

*Universidad de las Ciencias Informáticas*

*Facultad 8*



*Propuesta de sistemas Integrados*

*del Proyecto Infodrez*

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

**Autores:**

Yadira Arguelles Blanco

Lienys Abreu Hernández

**Tutor:**

Ing. Alison Muñoz Capote.

Ciudad de La Habana, 2009

“Año del 50 Aniversario del Triunfo de la Revolución”



### DECLARACIÓN DE AUTORÍA

Declaramos ser autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Yadira Arguelles Blanco**

\_\_\_\_\_

Firma del Autor

**Lienys Abreu Hernández**

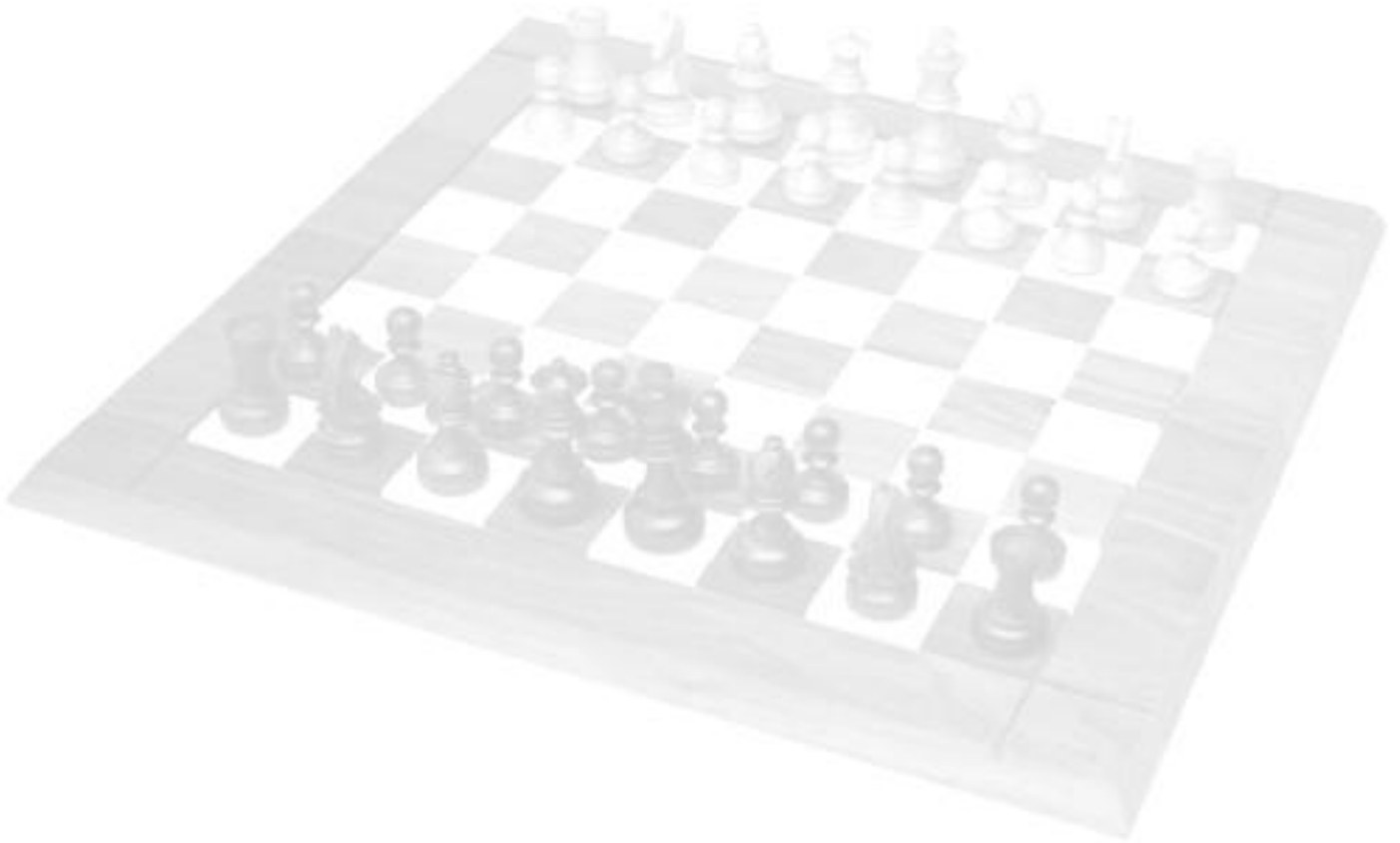
\_\_\_\_\_

Firma del Autor

**Ing. Alison Muñoz Capote**

\_\_\_\_\_

Firma del Tutor



*“Masificar el ajedrez, colocaría a este país con mucha más capacidad de pensar, más eficiente; es como saber una asignatura básica...”*

*Fidel Castro*



### *Agradecimientos:*

*A la Revolución y a la Universidad de Ciencias Informáticas por brindarnos la oportunidad de formarnos como profesionales y por poner a nuestra disposición los recursos necesarios para lograr nuestros sueños.*

*A nuestro tutor Alison por su compromiso ante esta importante tarea y apoyarnos en todo momento.*

*A todos los integrantes del proyecto Infodrez por su ayuda incondicional.*

*A todos los profesores que nos ayudaron para que este trabajo pudiera salir adelante con éxitos.*

*A las primas Yaimara, Yeleine y Rosalba por ayudarnos ante las dificultades que se presentaban.*

### *Yadira*

*A mis padres por todo lo que han pasado para que yo saliera adelante, por sus consejos, sacrificios, paciencia, tanta dedicación, su amor incondicional y apoyo mientras he estado lejos de casa, que me dieron las fuerzas para seguir adelante y no flaquear. A ellos les debo lo que soy y les estaré eternamente agradecida.*

*A mi familia por ofrecerme tanto amor y armonía desde pequeña, por lograr que me haya convertido en la mujer que soy hoy, por todos los momentos buenos y malos que nos han tocado vivir y que hemos estado juntos y sobre todo por su apoyo en esos tiempos difíciles que no pude estar con ellos y supieron darme las fuerzas para soportar el dolor y seguir adelante.*

*A Roxana por ser mi amiga de toda la vida y haberme dado la oportunidad de contar con ella siempre*

*A mi novio Alberto que me ha enseñado que el amor puede vencer a pesar de la distancia, por la dicha de tenerlo y ser lo mejor que me ha pasado, por ser mi consejero y amigo siempre que lo necesito.*



*A mis amigos del clan de Moa por darme la oportunidad de conocerlos y descubrir lo maravilloso que son todos, les agradezco por los momentos que hemos pasado juntos y por ser incondicionales.*

*A mis amigas inseparables Yádira, Lienys y Milenis por compartir conmigo en estos cinco años cada instante de dicha y tristeza, por ocupar un lugar tan especial en mi corazón.*

*A todos los demás amigos que he conocido durante estos años y que recordaré siempre.*

*Lienys*

*A mis padres, por el cariño y el amor que me han brindado toda la vida, por ser mis principales educadores, el motivo de inspiración de llevar todos estos sueños adelante y poder siempre contar con ellos. Por hacer hasta lo imposible porque durante estos 5 años lejos de casa no me faltara nada, ni su cariño aún estando lejos, y por confiar en mí hasta en los momentos más difíciles. También gracias a mi mamá por enseñarme que el esfuerzo siempre se premia y que todos tenemos derecho a equivocarnos y de aprender con los errores.*

*A mi hermano por constituir para mí un ejemplo a seguir y darme el aliento necesario para no retroceder ante las dificultades.*

*A mis abuelos, tíos y primos y en especial a mi tía Cristi, por ser mi segunda mamá, por apoyarme con su cariño incondicional y sus consejos, y por preocuparse por mí en cada momento durante toda mi carrera.*

*A Guille, mi novio, por ser lo mejor que me ha sucedido y la persona más especial que he conocido en estos años, por todo su amor, su preocupación, su paciencia ante mis cambios de ánimo y por soportar todas mis malcriadeces, por eso y mucho más...gracias.*



*A mis amigas Yadi A, Yady B, Milenis, y Yami, muchas gracias por todo el apoyo que me han brindado durante estos años compartiendo juntas, por ganarse todo mi cariño y respeto, por ser personas especiales llenas de valores dignos de admiración, y más que todo por saber ser amigas en los momentos difíciles. Gracias por aguantar mis locuras.*

*A todo el piquete de Moa, a las amistades que he conocido, y a los que vienen conmigo compartiendo desde el preuniversitario, a todos muchas gracias por estar siempre ahí.*



*Yadira*

*Dedico este trabajo a mis padres por alumbrarme el camino, por ser mis guías en todo momento, por confiar en mí y lograr a pesar de los obstáculos que haya terminado mis estudios y seguir superándome para ser alguien en la vida.*

*A mis hermanos por ser mi mayor inspiración y las personas que más adoro*

*A mis abuelitas Mirna y Vidalina que tanto las extraño y aunque ya no estén conmigo estoy segura que estarían muy orgullosas de su nietecita.*

*Lienys*

*A mis padres que son mi razón de ser y de existir va dedicado todo el empeño que he puesto en este trabajo, por ser personas maravillosas que me supieron guiar con su ejemplo, sus consejos y su amor.*

*A mi tía Cristina por el aliento y el optimismo que siempre me inculcó.*

*A alguien también muy especial para mí, que a pesar de no estar hoy físicamente a mi lado, nunca me ha abandonado porque en mi corazón ocupa un lugar muy importante, a mi abuelita Celia que le hubiese encantado ver a su nieta "trigueña" graduada.*



## Resumen

Infodrez ha jugado un papel muy importante en el desarrollo de la informatización del ajedrez. Ha desarrollado productos en las principales áreas de este deporte que satisfacen las necesidades de los especialistas de la cátedra Remberto A. Fernández González de la Universidad de las Ciencias Informáticas. Estos productos presentan algunas deficiencias, aunque se considera que la mayor de ellas es el hecho de estar separados y se dificulta la consulta de toda la información simultáneamente. En el presente trabajo se le da solución a esta problemática, con el diseño de la primera versión de la plataforma Infodrez. Esta propuesta abarca principalmente funcionalidades presentes en los módulos de arbitraje, estadísticas y visor de torneo.

En el desarrollo de ésta propuesta se utiliza una de las metodologías tradicionales más usadas actualmente en el mundo, el Proceso Unificado de Desarrollo (RUP) unido al lenguaje de modelado UML. Durante el levantamiento de requisitos se definen algunas técnicas para la obtención y especificación de los requerimientos, continuando con su validación a través de prototipos presentados al usuario antes de entrar a modelar el sistema. Durante el análisis y el diseño se aplican los patrones de asignación de responsabilidades GRASP así como el estilo arquitectónico N-Capas. En la modelación de los artefactos generados durante el paso por estos flujos de trabajo se utiliza la herramienta Visual Paradigm propuesta por el proyecto.

Se contó con la asesoría de especialistas de Ingeniería del Software así como de ajedrez para la validación de los resultados obtenidos.





## INDICE

<b>RESUMEN.....</b>	<b>VI</b>
<b>CAPÍTULO 1 .....</b>	<b>5</b>
1.1 INTRODUCCIÓN.....	5
1.2 DESARROLLO .....	5
1.2.1 Metodologías de desarrollo.....	5
1.2.2 Lenguajes de modelado.....	15
1.2.3 Herramientas CASE.....	17
1.2.4 Técnicas para la recopilación de requisitos.....	20
1.2.5 Lenguaje del lado del cliente.....	23
1.2.6 Leguaje del lado del servidor .....	25
1.2.7 Patrones .....	26
1.3 CONCLUSIONES.....	28
<b>CAPÍTULO 2 .....</b>	<b>29</b>
2.1 INTRODUCCIÓN.....	29
2.2 DESARROLLO .....	29
2.2.1 Fundamentación de la situación problemática .....	29
2.2.2 Propuesta del sistema.....	31
2.2.3 Modelo de negocio.....	31
2.2.5 Requerimientos.....	36
<b>CAPÍTULO 3 .....</b>	<b>68</b>
3.1 INTRODUCCIÓN.....	68
3.2 DESARROLLO .....	68
3.2.1 Arquitectura y Estilos Arquitectónicos .....	68
3.2.2 Análisis del Sistema.....	69
3.2.3 Diseño del Sistema .....	83
3.3 CONCLUSIONES.....	103



<b>CONCLUSIONES GENERALES .....</b>	<b>101</b>
<b>RECOMENDACIONES:.....</b>	<b>102</b>
<b>REFERENCIA BIBLIOGRÁFICA .....</b>	<b>103</b>
<b>BIBLIOGRAFÍA .....</b>	<b>104</b>
<b>ANEXOS .....</b>	<b>107</b>
<b>ANEXO 1: APLICACIÓN DE LAS TÉCNICAS DE CAPTURA DE REQUISITOS .....</b>	<b>107</b>
<b>ANEXO 2: MODELO DE DOMINIO DEL SISTEMA INFODREZ .....</b>	<b>109</b>
<b>ANEXO 3: VALIDACIÓN DE REQUISITOS .....</b>	<b>110</b>
<b>ANEXO 4: DESCRIPCIONES DE LOS CUS .....</b>	<b>137</b>
<b>ANEXO 5: DIAGRAMA DE CLASES DEL ANÁLISIS.....</b>	<b>203</b>
<b>ANEXO 6: DIAGRAMAS DE COLABORACIÓN DEL ANÁLISIS.....</b>	<b>219</b>
<b>ANEXO 7: DIAGRAMAS DE CLASES DEL DISEÑO .....</b>	<b>250</b>
<b>ANEXO 8: DESCRIPCIONES DE LAS CLASES DE DISEÑO WEB.....</b>	<b>264</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>306</b>



# Introducción

El ajedrez es un juego de ingenio que simboliza la guerra, practicado en el mundo por infinidad de personas de todas las edades, no solo considerado un juego sino un arte, una ciencia y un deporte mental. Es jugado entre dos oponentes, en el que no interviene para nada el azar, sino el ingenio de los competidores.

La informática ha significado para el ajedrez el comienzo de una nueva era, los avances tecnológicos se han venido perfeccionando y hoy en día, ajedrecistas de distintos niveles, los usan para mejorar su juego, plantear y solucionar problemas, y probar nuevas aperturas antes de intentar jugarlas en un evento ajedrecístico. Incluso, a través de Internet, se pueden jugar partidas, acceder a bancos de datos con colecciones de las mismas y obtener variada información referente a dichas partidas como aperturas, posiciones, jugadores y ganadores.

En la actualidad existen herramientas que se encargan de mostrar o gestionar información referente a las diversas áreas relacionados con el ajedrez; arbitraje, estadística, torneos y juegos en tiempo real. En términos de arbitraje se pueden mencionar el Swiss Perfect y el Swiss Manager; ambos para gestionar torneos en sistemas "suizos" y "Round Robin", la diferencia entre ellos es que el primero no está pensado para torneos por equipos, pero tiene una opción de MultiGame (varias partidas) que generalmente es suficiente para gestionar un torneo por equipos, y el segundo gestiona torneos tanto individuales como por equipos, además de que en torneos por sistema suizo, puede realizar tanto emparejamientos automáticos como manuales. El emparejamiento automático permite distintas variedades de emparejamiento suizo.

En el ámbito referente a torneos se han publicado sitios en Internet como el [www.ajedrezmadrid.com](http://www.ajedrezmadrid.com) donde se brindan datos de los torneos más importantes de ajedrez en el mundo: resultados, clasificaciones, partidas, noticias, opinión, consultas on-line de Elo, artículos de opinión, información sobre el material de juego, tablero y piezas etc. Otro ejemplo lo constituye el [www.Chessresult.com](http://www.Chessresult.com), Web dedicada a la publicación de los resultados de todo tipo de evento de ajedrez que se disputan a nivel mundial.



Por otro lado surgen programas que tienen como objetivo ofrecer reportes estadísticos, brindando la posibilidad de aumentar el rendimiento de los jugadores y obtener información útil para el estudio de sus adversarios. Entre estas herramientas podemos destacar el caso del ChessBase que se ha convertido en una de las más utilizadas, entre sus ventajas muestra la posibilidad de poder analizar partidas, tanto las que realiza cualquier persona como las partidas míticas que han disputado las mejores estrellas del ajedrez. El programa permite acceder a datos de los mejores ajedrecistas de la historia. Dispone de un libro de aperturas, algo que es realmente útil ya que con él se podrá ensayar todas las variantes de las salidas favoritas del participante. En resumen es un programa con bastantes funciones, pero no llega al nivel de otros como Fritz. Este último es adecuado tanto para principiantes como para jugadores de club o profesionales: todos pueden sacar el mejor partido del programa porque Fritz se adapta automáticamente a las necesidades y a la fuerza de juego de cada jugador, posibilitando consultar los datos de los mismos aunque no estén presentes en el servidor en ese momento.

Después de haber realizado una breve descripción sobre las funcionalidades de las herramientas más utilizadas en cada una de las áreas vinculadas al ajedrez a nivel internacional, observamos que todas abarcan de manera muy amplia las funcionalidades necesarias para resolver problemáticas disímiles en cada una de las mismas.

### **La informática y el ajedrez en Cuba**

Cuba se caracteriza por ser una potencia deportiva y el ajedrez no es la excepción, es un deporte muy seguido por la afición cubana y lo demuestran los resultados alcanzados en Olimpiadas y de forma individual por algunos de sus jugadores. La informática también ha tenido su impacto positivo en el desarrollo de este deporte, pero no ha tenido total influencia en cada una de las áreas descritas en los párrafos anteriores, debido a algunos factores como son que los software son privativos y además de que los servicios los brindan aplicaciones publicadas en Internet y los usuarios en nuestro país no tienen casi o ningún acceso a ella desde la red nacional. Motivado por esta problemática, la UCI, a través del proyecto Infodrez, se trazó como meta la informatización del ajedrez cubano, a partir de la creación de aplicaciones



que se utilizarían en las diversas áreas de este deporte y enfocada a las problemáticas reales del mismo en nuestro país. Las propuestas de productos identificados en el curso 2007-2008 fueron:

- ❖ Confección y desarrollo del módulo Gestor de Estadísticas del proyecto Infodrez.
- ❖ Desarrollo de la versión 2.0 del módulo Juego Online de Infodrez.
- ❖ Concepción y desarrollo del módulo Arbitraje del Proyecto Infodrez.
- ❖ Concepción del módulo de ajedrez por correspondencia de Infodrez
- ❖ Gestionar el Componente del Rating Elo.

Después de haber efectuado revisiones en cada una de las propuestas de productos del proyecto Infodrez, se identificaron las siguientes irregularidades en cada uno de los módulos:

- ❖ Réplica de código.
- ❖ Réplica de reportes.
- ❖ Información no centralizada.
- ❖ Información no estandarizada.

Teniendo en cuenta estas deficiencias, la inexistencia de un sistema que integre la gestión de las principales áreas de ajedrez, y en vista a que el proyecto desea entrar en competencia en el mercado internacional, se hace necesario crear un producto que cubra las expectativas de los usuarios y sus funcionalidades tengan la eficiencia de los más utilizados en la actualidad. El presente trabajo abordará las áreas referentes a arbitraje, estadísticas e información sobre torneos de ajedrez.

Enunciada la situación problemática, puede definirse el **problema científico** a partir de la siguiente interrogante: ¿Cómo gestionar los principales módulos del proyecto Infodrez?

El **objeto de estudio** fijado para la presente investigación se enmarca en el proceso de análisis y diseño de software de gestión y tomando como **campo de acción** la modelación de los módulos del proyecto Infodrez.

Como **objetivo general** la realización del análisis y diseño de un sistema integrado con los módulos del proyecto Infodrez.

Se plantean como **objetivos específicos**:



1. Analizar el estado de cada uno de los módulos del proyecto.
2. Identificar deficiencias que presentan cada uno de los sistemas por separado.
3. Identificar ventajas que debe tener el sistema propuesto sobre los sistemas existentes.
4. Elaborar una propuesta de sistema de la primera versión de la plataforma de Infodrez con los módulos unificados.

Para dar cumplimiento a los objetivos se plantean un grupo de **tareas de investigación**:

1. Analizar los modelos propuestos por las tesis anteriores.
2. Analizar los sistemas similares existentes.
3. Entrevistar a especialistas de los sistemas existentes en el proyecto.
4. Refinar los requisitos de cada uno de los sistemas existentes en el proyecto.
5. Elaborar el modelo propuesto.

Como **idea a defender** se parte de que es posible la modelación de un sistema integrado para crear una herramienta que permita la gestión de los módulos del proyecto Infodrez.



# Capítulo 1

## 1.1 Introducción

El sistema que se desea proponer no tiene antecedentes en cuanto a sistemas similares existentes. En la fundamentación de este capítulo se realiza una descripción de los aspectos que se tienen en cuenta para modelar el sistema, se efectúa un análisis de las metodologías de desarrollo del software, lenguajes de modelado, herramientas CASE, técnicas de captura de requisitos más utilizadas a nivel mundial además de lenguajes del lado del cliente y del servidor, realizando una selección de las más adecuadas para el desarrollo de la propuesta que se pretende modelar.

## 1.2 Desarrollo

### 1.2.1 Metodologías de desarrollo

Hoy en día el desarrollo del software requiere mayor esfuerzo debido a que los clientes piden productos más grandes y con mayor calidad, por lo que se hace necesario el uso de una metodología de desarrollo que guíe este proceso y garantice la satisfacción del cliente. No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc.).

Metodología de desarrollo de software en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información [1]. Es definido por Mario Piattini Velthuis como un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevo software (Piattini, 1996). Piattini expresa que "una metodología representa el camino para desarrollar software de una manera sistemática".

Las metodologías se pueden enmarcar en dos grupos: tradicionales (pesadas) y ágiles (ligeras). Las metodologías tradicionales se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y



notaciones que se usarán. Las metodologías ágiles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad.

Las metodologías de desarrollos más conocidas en la actualidad son: Programación Extrema (XP), Proceso Unificado de Rational (RUP) y Scrum.

### 1.2.1.1 Metodología XP

La Programación Extrema, o Extreme Programming (XP según siglas en inglés), es la más utilizada de las metodologías ágiles de desarrollo de software. Surge como posible solución a los problemas derivados del cambio en los requerimientos; esta metodología ofrece la posibilidad de cambiar los requisitos en cualquier momento de la vida de un proyecto, ya que es adaptable a estos cambios. Se centra en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. [2]

XP es un proceso muy orientado a la implementación, en el que se genera poca documentación y en que la funcionalidad exacta del sistema final no se define nunca formal y contractualmente. Es por eso que este método es más aplicable para desarrollos internos. (Palmero Sánchez y otros, 2007)

El ciclo de vida ideal de XP consta de seis fases [2]:

- ❖ Exploración.
- ❖ Planificación de la Entrega (*Release*).
- ❖ Iteraciones.
- ❖ Producción.
- ❖ Mantenimiento.
- ❖ Muerte del Proyecto.





### Proceso XP

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste de manera general en los siguientes pasos [3]:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

### Las Historias de Usuario en XP

Las historias de usuario son utilizadas por la metodología XP como una técnica para especificar los requisitos del usuario, tanto requisitos no funcionales como funcionales. Se trata de tarjetas en las cuales el usuario describe las características que el sistema debe poseer, de forma muy breve.

Las características fundamentales del método son:

- ❖ Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- ❖ Pruebas unitarias continuas: frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión: Se aconseja escribir el código de la prueba antes de la codificación.
- ❖ Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera



el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.

- ❖ Frecuente integración del equipo de programación en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- ❖ Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario.

### Roles XP

La metodología XP de acuerdo con la propuesta original de Kent Beck [2] propone los siguientes roles:

- ❖ Programador.
- ❖ Cliente.
- ❖ Encargado de pruebas (*Tester*).
- ❖ Encargado de seguimiento (*Tracker*).
- ❖ Entrenador (*Coach*).
- ❖ Consultor.
- ❖ Gestor (*Big boss*).

### 1.2.1.2 Metodología Scrum

Scrum es un proceso de desarrollo de software iterativo e incremental utilizado comúnmente en entornos basado en la metodología ágil de desarrollo de software. Está especialmente indicada para proyectos con un rápido cambio de requisitos.

Las principales características de la metodología Scrum:

- ❖ El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*.
- ❖ Las reuniones a lo largo proyecto.



La magnitud de cada sprint o iteración es definida por el equipo, generalmente comprende un período entre 15 y 30 días. En este período el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene del *Product Backlog*, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del *Product Backlog* que forman parte del sprint se determinan durante la reunión de *Sprint Planning*. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. Durante esta reunión, el *Product Owner* identifica los elementos del *Product Backlog* que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint [4]. Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.

### Roles en Scrum

En Scrum se definen varios roles:

#### ❖ Product Owner

El Product Owner representa la voz del cliente. Se asegura de que el equipo Scrum trabaja de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.

#### ❖ ScrumMaster (o Facilitador)

El Scrum es facilitado por un ScrumMaster, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El *ScrumMaster* no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El ScrumMaster se asegura de que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan.

#### ❖ Equipo

El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 5 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (diseñador, desarrollador, etc.).



- ❖ Usuarios

Se refiere a las personas para las cuales va construido el software.

- ❖ Stakeholders (Clientes, Proveedores)

Personas que hacen posible el proyecto y para los cuales el proyecto producirá el beneficio acordado que lo justifica. Sólo participan directamente durante las revisiones del sprint.

- ❖ Managers

Personas que establecen el ambiente para el desarrollo del producto.

### 1.2.1.3 Metodología RUP

El Proceso Unificado de Rational (*Rational Unified Process* en inglés, conocido como RUP), es el resultado de varios años de desarrollo y uso práctico, unifica los mejores elementos de las metodologías anteriores. Es un proceso de desarrollo de software que unido al Lenguaje Unificado de Modelado UML, constituye una de las metodologías más utilizadas para el análisis, implementación y documentación de sistemas orientados a objetos.

RUP se divide en cuatro fases:

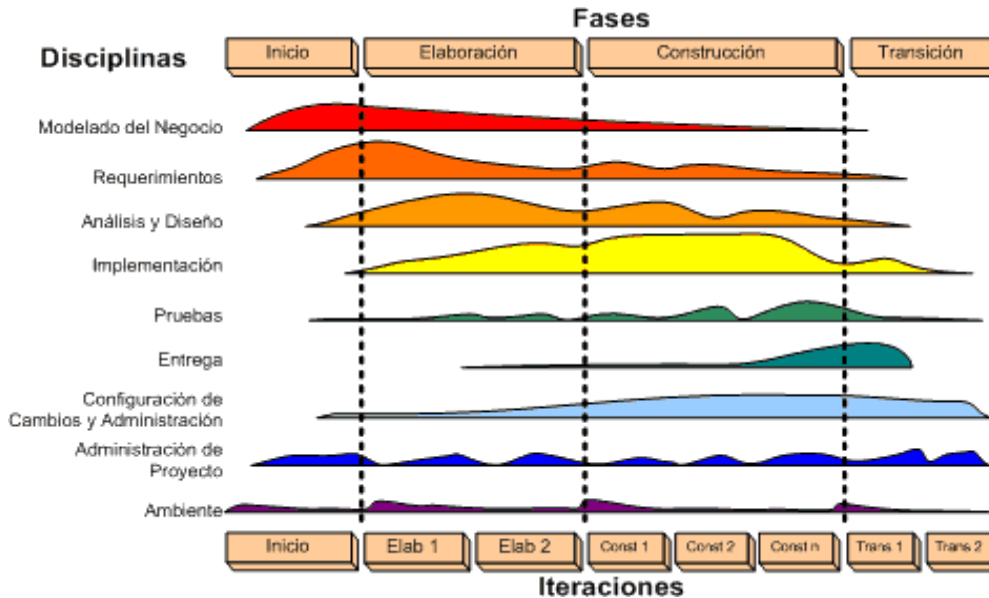
- ❖ Fase de Inicio: se define la idea, la visión del producto, como se enmarca en el negocio y el alcance del proyecto.
- ❖ Fase de Elaboración: se planifican las actividades necesarias y los recursos requeridos, especificando las características y el diseño de la arquitectura.
- ❖ Fase de Construcción: se construye el producto, la arquitectura y los planes, hasta que el producto está listo para ser enviado a la comunidad de usuarios.
- ❖ Fase de Transición: se realiza la transición del producto a los usuarios, lo cual incluye: manufactura, envío, entrenamiento, soporte y mantenimiento del producto, hasta que el cliente esté satisfecho.

El proceso propuesto por RUP posee dos dimensiones: la primera, representa el aspecto dinámico del proceso, y está expresado en términos de ciclos, fases, iteraciones e hitos, la segunda, representa el



aspecto estático, que se describe en términos de componentes, actividades, flujos de trabajo, artefactos, y actores como se muestra en la figura

**Figura 1: RUP en dos dimensiones**



La realización de las cuatro fases de RUP produce una generación del producto. Cada fase tiene una o más iteraciones de todos los flujos y finaliza en un hito. Al finalizar cada fase, en estos hitos ha de cumplirse que:

- ❖ Fase de Inicio: Objetivos (Visión).
- ❖ Fase de Elaboración: Arquitectura.
- ❖ Fase de Construcción: Capacidad Operacional Inicial.
- ❖ Fase de Transición: Release del Producto.

RUP cuenta con 9 flujos de trabajo, los 6 primeros de ingeniería y los 3 últimos de apoyo:

- ❖ Modelamiento del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.



- ❖ **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- ❖ **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- ❖ **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- ❖ **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- ❖ **Instalación:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- ❖ **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- ❖ **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- ❖ **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

RUP en su modelación define como sus principales elementos:

- ❖ **Trabajadores (“quién”):** Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- ❖ **Actividades (“cómo”):** Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.



- ❖ Artefactos (“qué”): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- ❖ Flujo de actividades (“Cuándo”): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

El ciclo de vida de RUP se caracteriza por:

- ❖ Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- ❖ Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.
- ❖ Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

Algunos artefactos generados por RUP.

- ❖ Plan de Desarrollo del Software.
- ❖ Modelo del Negocio.
- ❖ Glosario.
- ❖ Modelo de Casos de Uso.
- ❖ Visión.
- ❖ Especificación de los requisitos de software.
- ❖ Prototipos de Interfaces de Usuario.
- ❖ Modelo de Análisis y Diseño.



Esta metodología propone una serie de roles a ser desarrollados por los integrantes del proyecto, algunos de los más importantes son:

- ❖ Jefe de Proyecto.
- ❖ Analista del sistema.
- ❖ Arquitecto de software.
- ❖ Especificador de casos de uso.
- ❖ Desarrollador o programador.
- ❖ Diseñador de Interfaz de Usuario.
- ❖ Diseñador de Bases de Datos.
- ❖ Diseñador de pruebas.

RUP pretende implementar las mejores prácticas actuales en ingeniería de software:

- ❖ Desarrollo iterativo del software.
- ❖ Administración de requerimientos.
- ❖ Uso de arquitecturas basadas en componentes.
- ❖ Modelado visual del software.
- ❖ Verificación de la calidad del software.
- ❖ Control de cambios.

En resumen, las metas de RUP según Kruchten son:

- ❖ Asegurar la producción de un software de alta calidad que reúna las necesidades de los usuarios finales dentro de un plan y un presupuesto predecible.
- ❖ Proveer un enfoque disciplinado para asignar tareas y responsabilidades dentro del desarrollo del sistema.
- ❖ Proveer un camino metódico, sistemático para desarrollar, diseñar y validar una arquitectura.





- ❖ Reducir en gran medida el riesgo que representa la construcción de sistemas complejos, porque evoluciona de forma incremental partiendo de sistemas más pequeños en los que ya se tiene confianza.

### **1.2.1.4 Selección de la metodología de desarrollo a utilizar.**

Después de haber realizado un análisis de algunas de las metodologías existentes se ha decidido aplicar a la propuesta de sistema la metodología RUP (Proceso Unificado de Rational), ya que es la metodología que mejor se adapta al proceso, como metodología tradicional realiza una fuerte planificación durante todo el proceso de desarrollo, realizando una intensa etapa de análisis y diseño antes de la construcción del sistema. Además de que en la universidad se encuentran grandes volúmenes de información y experiencia con respecto a la misma.

### **1.2.2 Lenguajes de modelado**

Un modelo es una descripción de (parte de) un sistema, descrito en un lenguaje bien definido. Un lenguaje bien definido es un lenguaje con una sintaxis y semántica precisa, y que puede ser interpretado y manipulado por un ordenador.

Los modelos proporcionan un mayor nivel de abstracción, permitiendo trabajar con sistemas mayores y más complejos, y facilitando el proceso de codificación e implementación del sistema de forma distribuida y en distintas plataformas.

#### **1.2.2.1 Lenguaje PML**

Process Modeling Language, denominado PML es un lenguaje de modelado orientado a objetos diseñado para describir el comportamiento del sistema físico mediante estructuras de representación modulares (clases de modelado). Las clases PML representan conceptos físicos que son familiares al modelador. El conocimiento físico declarado por las clases se utiliza para analizar los modelos



estructurados, obteniéndose de manera automatizada la representación matemática de las dinámicas de interés (Ramos González, 2003).

PML es capaz de automatizar el proceso de modelado asegurando la reusabilidad de modelos que pueden ser predefinidos de manera independiente al contexto físico donde serán reutilizados.

### 1.2.2.2 Lenguaje UML

El desarrollo de las metodologías de análisis y diseño orientados a objetos trajo consigo la existencia de diversas notaciones y propuestas, entre las que cabe destacar la metodología de Booch [5], la Object Modeling Technique (OMT) de Rumbaugh [6] y OOSE de Ivar Jacobson [7]. Ante el problema de la proliferación de notaciones distintas, los autores más relevantes unieron esfuerzos para producir una notación unificada, y el resultado de esto fue el Lenguaje Unificado de Modelado o Unified Modeling Language, UML [8]. Entre los lenguajes de modelado que define OMG (Object Management Group) el más conocido y usado es el UML.

Este no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. Los diagramas se realizan con la finalidad de presentar diversas perspectivas de un sistema, a los que se les denominan modelos, estos describen lo que supuestamente hará un sistema, pero no dicen como implementarlo.

UML posibilita el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Se ha convertido en un estándar de facto en la industria, ya que permite visualizar, especificar, construir y documentar los elementos que forman un sistema de software orientado a objetos (Jacobson, y otros, 2000). A continuación se definen cada una de las operaciones que permite UML.

- ❖ Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.



- ❖ Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- ❖ Construir: A partir de los modelos especificados se puede construir los sistemas diseñados.
- ❖ Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado, sirviendo para su futura revisión.

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Está formado por tres clases de bloques de construcción. El primero de ellos son los elementos, es decir, las abstracciones de cosas reales o ficticias (como objetos y acciones); luego se encuentran las Relaciones, esto es la relación entre los elementos (la cual puede ser de dependencia, asociación, generalización, etc.); finalmente, se encuentran los Diagramas, que son colecciones de elementos con sus relaciones.

### 1.2.2.3 Selección del lenguaje de modelado

Se decide utilizar como lenguaje de modelado el UML, debido a que este es uno de los más usados en la modelación de los artefactos que se van generando en el proceso de desarrollo del software. Al ser creado por los autores de la metodología RUP, este lenguaje da soporte a la misma, por lo que cubre todas las necesidades de esta metodología formando un buen complemento a la hora de desarrollar un producto.

### 1.2.3 Herramientas CASE

Las herramientas CASE (**C**omputer **A**ided **S**oftware **E**ngineering, Ingeniería de Software Asistida por Ordenador) sirven de ayuda a lo largo de todo el ciclo de vida de desarrollo del software permitiendo realizar el diseño del proyecto, la compilación automática, documentación o detección de errores entre otras funcionalidades.

Estas herramientas aumentan la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Cuentan también con una credibilidad y exactitud que



tienen un reconocimiento universal, siendo usadas por cualquier desarrollador y / o programador que busca un resultado óptimo y eficiente, pero sobre todo, que busca esa minuciosidad necesaria de los procesos y entre los procesos.

Algunos ejemplos de aplicaciones CASE son: Visual Paradigm for UML, Rational ClearCASE, Rational Rose.

### 1.2.3.1 Visual Paradigm for UML

Visual Paradigm es una herramienta CASE que utiliza el UML como lenguaje de modelado, soportando hasta la fecha la versión 2.1. Cubre el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, realizar ingeniería tanto directa como inversa y genera la documentación del proyecto automáticamente en varios formatos como PDF, HTML y MS Word.

Es una herramienta colaborativa; soporta múltiples usuarios trabajando sobre el mismo proyecto. Proporciona abundantes tutoriales, demostraciones interactivas y proyectos de UML. Combina el modelado UML, código de ingeniería avanzada y una excelente interoperabilidad en una única plataforma de desarrollo.

VP-UML (Visual Paradigm for UML) ayuda a construir aplicaciones en menor tiempo, de una mejor forma y más económica. Se debe destacar su robustez, usabilidad y portabilidad.

Características de Visual Paradigm:

- ❖ Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento Modelado colaborativo con CVS y Subversión.
- ❖ Interoperabilidad con modelos UML2 (meta-modelos UML 2.x para plataforma Eclipse) a través de XML.
- ❖ Generación de código - Modelo a código, diagrama a código.
- ❖ Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.



- ❖ Diagramas EJB - Visualización de sistemas EJB.
- ❖ Generación de código y despliegue de EJB's - Generación de beans para el desarrollo y despliegue de aplicaciones.
- ❖ Diagramas de flujo de datos.
- ❖ Soporte ORM - Generación de objetos Java desde la base de datos.
- ❖ Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- ❖ Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- ❖ Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- ❖ Importación y exportación de ficheros XML.
- ❖ Soporta múltiples plataformas.

### 1.2.3.2 Rational Rose

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML la cual soporta de forma completa la versión 1.1 de este lenguaje estando enfocada hacia el uso de la metodología RUP. Permite especificar, analizar, diseñar el sistema antes de codificarlo.

Propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. Utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Permite que haya varias personas trabajando al mismo tiempo, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.



Puede generar código en distintos lenguajes de programación a partir de un diseño en UML. Proporciona mecanismos para realizar la denominada ingeniería inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

### **1.2.3.3 Selección de la herramienta CASE a utilizar**

Visual Paradigm 6.0 ha sido la herramienta seleccionada debido a que se pueden realizar las modelaciones del sistema que se propondrá de manera cómoda y es soportada por la distribución Devian del sistema operativo Linux, que es la plataforma que soporta el ambiente de desarrollo del proyecto Infodrez.

### **1.2.4 Técnicas para la recopilación de requisitos.**

La obtención de requisitos es el proceso mediante el cual los interesados en un sistema de software descubren, revelan, articulan y entienden sus requisitos (Raghavan *et al.*, 1994). En muchos casos, se requiere tiempo para llegar a especificar claramente lo que el interesado espera de la aplicación de software, por lo que se hace necesario por parte de los analistas el empleo de técnicas que permitan establecer una buena comunicación con los interesados del producto y así lograr la satisfacción del cliente.

Entre las técnicas de recopilación de requisitos más utilizadas se encuentran las mencionadas a continuación:

#### **1.2.4.1 Entrevistas**

Es la más tradicional de las técnicas de obtención y consiste en reuniones analista-interesado en las cuales se suceden preguntas y respuestas para extraer el dominio de la aplicación (Goguen y Linde, 1993). En Pressman (2005) se presentan conjuntos de preguntas que se pueden utilizar en el desarrollo de esta técnica, que tiene una alta participación del analista y se realiza en conjunto con otras técnicas.



### **1.2.4.2 Técnicas para facilitar las especificaciones de una aplicación (TFEA)**

Estas son una variación de las entrevistas buscando identificar el problema, proponer elementos de solución, negociar diferentes enfoques y especificar un conjunto preliminar de requisitos de la solución (Pressman, 2005). A pesar de ir un paso más allá de las entrevistas convencionales, precisan aún de una alta participación del analista.

### **1.2.4.3 Despliegue de la función de calidad (DFC)**

Esta técnica incluye entrevistas y documentación de la organización con las cuales se construye la tabla de opinión del interesado. Esta tabla se analiza con diagramas, matrices y métodos de evaluación para extraer los requisitos esperados e intentar obtener requisitos innovadores (Pressman, 2005). También esta técnica requiere una alta interacción con el analista.

### **1.2.4.4 Tormenta de ideas (brainstorming)**

Es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios. Puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de captura, cuando los requisitos son todavía muy difusos (Raghavan et al., 1994). Sin embargo, también se requiere participación intensiva del analista.

### **1.2.4.5 Juego de roles**

En su forma más simple, consiste en que el desarrollador, el analista y cada uno de los miembros del equipo de desarrollo del software toman el lugar del interesado y ejecutan la actividad de trabajo que éste desempeña. Ellos experimentan las inexactitudes y problemas ligados con el sistema que se está especificando. Se busca suministrarle al analista una perspectiva nueva del problema que le permita la obtención de los requisitos del sistema por construir (Raghavan *et al.*, 1994). Esta técnica también presenta alta participación de los involucrados.

### **1.2.4.6 Introspección**

Esta técnica recomienda que el analista se ponga en el lugar del interesado y trate de imaginar cómo desearía éste la aplicación de software. Basado en estas suposiciones, el analista entrega



recomendaciones al interesado sobre la funcionalidad que debería tener dicha aplicación (Goguen y Linde, 1993). El problema radica en que un analista no es un tipo normal de interesado, pues posee un conocimiento técnico más elevado; por ello, es posible que entre las recomendaciones haya cosas que el interesado aún no necesita o que incluso no sabe que necesitará en un futuro. En este caso, el discurso se refiere más a la solución que al dominio del problema.

### **1.2.4.7 Casos de uso o escenarios**

Son descripciones que incluyen actores, eventos, operaciones y objetivos de esas operaciones, generalmente ligados con el funcionamiento de una solución informática (Leffingwell y Widrig, 1999; Raghavan *et al.*, 1994; Pressman, 2005; Sommerville 2001). Los escenarios como técnica de obtención poseen dos limitaciones: exigen una alta participación del interesado en su elaboración y necesitan que él realice una concepción completa de la solución informática, que sólo sería posible al final del proceso de obtención de requisitos.

### **1.2.4.8 Talleres**

Los requisitos tienen a menudo implicaciones cruzadas desconocidas para las personas implicadas individuales y que a menudo no se descubren en las entrevistas o quedan incompletamente definidas durante la misma. Estas implicaciones cruzadas pueden descubrirse realizando en un ambiente controlado, talleres facilitados por un analista del negocio, en donde las personas implicadas participan en discusiones para descubrir requisitos, analizan sus detalles y las implicaciones cruzadas. A menudo es útil la selección de un secretario dedicado a la documentación de la discusión, liberando al analista del negocio para centrarse en el proceso de la definición de los requisitos y para dirigir la discusión.

### **1.2.4.9 Cuestionario**

Es una técnica para recopilar información por medio de preguntas escritas, que se responden en ausencia del analista. No es muy confiable, ya que las respuestas obtenidas no pueden ser fácilmente verificadas o pueden estar "elaboradas por motivos propios o de grupo".

Se recomienda su uso:

- ❖ Cuando las personas que posean la información se encuentren dispersas geográficamente.
- ❖ Cuando se requiere obtener información al mismo tiempo de numerosas personas.





- ❖ Cuando la información necesaria está conformada por tablas o datos que pueden ser verificados por otra vía.

### 1.2.4.10 Selección de las técnicas de captura de requisitos

Se decide utilizar para la captura de los requisitos de la propuesta de sistema la tormenta de ideas y las técnicas para facilitar las especificaciones de una aplicación (TFEA). [Ver Anexo # 1](#)

### 1.2.5 Lenguaje del lado del cliente

Los lenguajes del lado del cliente son aquellos que pueden ser directamente "digeridos" por el navegador es decir, es el navegador el que soporta la carga de procesamiento y son totalmente independientes del servidor.

En la realización de la propuesta de sistema integrado del proyecto Infodrez se utiliza el JavaScript y el HTML como lenguajes del lado del cliente.

#### 1.2.5.1 HTML

HTML, siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). Estas constituyen la filosofía de este lenguaje, por medio de ellas se pueden controlar los elementos tipográficos del texto: tipo, color y tamaño de las fuentes, el estilo, así como también la inclusión de tablas, listas, formularios, la inserción de fotos, sonidos, fondos, los enlaces mencionados anteriormente. Las etiquetas se pueden modificar por medio de sus atributos, éstos son del tipo atributo="valor" y se colocan detrás del nombre de la etiqueta



El nombre de la etiqueta y sus atributos se colocan entre los símbolos `<` y `>` y normalmente se usan dos, una de inicio y otra final, para conseguir el efecto deseado.

Etiquetas fundamentales que se encuentran en HTML:

- ❖ `<html>`: define el inicio del documento HTML, le indica al navegador que lo que viene a continuación debe ser interpretado como código HTML.
- ❖ `<head>`: define la cabecera del documento HTML, ésta cabecera suele contener información sobre el documento que no se muestra directamente al usuario.
- ❖ `<title>`: define el título de la página. Por lo general, el título aparece en la barra de título encima de la ventana.
- ❖ `<link>`: para vincular el sitio a hojas de estilo o iconos.
- ❖ `<style>`: para colocar el estilo interno de la página, ya sea usando CSS, JavaScript u otros lenguajes similares. No es necesario colocarlo si se va a vincular a un archivo externo usando la etiqueta `<link>`.
- ❖ `<body>`: define el contenido principal o cuerpo del documento. Esta es la parte del documento html que se muestra en el navegador; dentro de esta etiqueta pueden definirse propiedades comunes a toda la página, como color de fondo y márgenes.

### 1.2.5.2 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas Web dinámicas. Sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Personas que no tengan experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica.

Los programas JavaScript van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos, entre otras.



JavaScript es un lenguaje de programación interpretado, es decir, no requiere de compilación. Orientado a objetos, utilizado principalmente en páginas Web y con sintaxis semejante a la del lenguaje Java y el lenguaje C. Compatible con todos los navegadores modernos, por lo que es el lenguaje de programación del lado del cliente más utilizado.

Las normas básicas que definen la sintaxis de JavaScript son las siguientes:

- ❖ No se tienen en cuenta los espacios en blanco y las nuevas líneas.
- ❖ Se distinguen las mayúsculas y minúsculas.
- ❖ No se define el tipo de las variables.
- ❖ No es necesario terminar cada sentencia con el carácter de punto y coma.
- ❖ Se pueden incluir comentarios.

### 1.2.6 Leguaje del lado del servidor

Los lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Éstos son independientes del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo.

#### 1.2.6.1 PHP

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas. Se muestra como código impregnado dentro de una página HTML. El modo de operación del PHP es el siguiente:

- ❖ El Navegador realiza una petición al servidor (se escribe la URL).
- ❖ Después el servidor ejecuta el código PHP solicitado y retorna el código HTML generado al Navegador.
- ❖ Por último el Navegador muestra la respuesta del servidor.



Las principales características de PHP son: su rapidez; su facilidad de aprendizaje; su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos; y el hecho de que se distribuye de forma gratuita bajo una licencia abierta.

### Ventajas

- ❖ Es un lenguaje multiplataforma.
- ❖ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- ❖ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- ❖ Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ❖ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ❖ Permite las técnicas de Programación Orientada a Objetos.
- ❖ Biblioteca nativa de funciones sumamente amplia e incluida.
- ❖ No requiere definición de tipos de variables.

### 1.2.7 Patrones

“Cada patrón describe un problema que ocurre una y otra vez en nuestro 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 siquiera dos veces de la misma forma”. (Christopher Alexander, 1977)

Patrones GRASP (General Responsibility Assignment Software Patterns)

Estos patrones describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Las mismas están relacionadas con las obligaciones de un objeto en cuanto a su comportamiento. Estas responsabilidades pueden ser en cuanto a dos aspectos:



- ❖ Conocer: conocer los datos privados encapsulados, conocer los objetos relacionados, conocer las cosas que puede derivar o calcular.
- ❖ Hacer: hacer algo él mismo; como crear un objeto o hacer un cálculo, iniciar una acción en otros objetos, controlar y coordinar actividades en otros objetos.

Se describen a continuación cada uno de los patrones analizados.

### 1.2.7.1 Experto

Asigna una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

### 1.2.7.2 Creador

Asigna a la clase B la responsabilidad de crear una instancia de clase A en uno de los siguientes casos:

- ❖ B agrega los objetos A.
- ❖ B contiene los objetos A.
- ❖ B registra las instancias de los objetos A.
- ❖ B utiliza específicamente los objetos A.
- ❖ B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado (así que B es un Experto respecto a la creación de A).
- ❖ B es un creador de los objetos A.

Si existe más de una opción, prefiera la clase B que agregue o contenga la clase A.

### 1.2.7.3 Bajo acoplamiento

Asigna una responsabilidad para mantener bajo acoplamiento.

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras.



### 1.2.7.4 Alta cohesión

Asigna una responsabilidad de modo que la cohesión siga siendo alta.

En la perspectiva del diseño orientado a objetos, la cohesión (0, más exactamente, la cohesión funcional) es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

### 1.2.7.5 Controlador

Asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones:

- ❖ El "sistema" global (controlador de fachada).
- ❖ La empresa u organización global (controlador de fachada).
- ❖ Algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas).
- ❖ Un manejador artificial de todos los eventos del sistema de un caso de uso, generalmente denominados "Manejador<NombreCasodeUso>"(controlador de casos de uso).

## 1.3 Conclusiones

En este capítulo se realizó un estudio del arte con el objetivo de analizar las herramientas que van a permitir la modelación del sistema propuesto, analizando las funcionalidades de cada una de ellas. Como se pretende que el mismo cubra las expectativas de los usuarios, se utilizará RUP como metodología de desarrollo, unido al lenguaje de modelado UML, y como herramienta CASE el Visual Paradigm. Se definen como técnicas de captura de requisitos a utilizar la tormenta de ideas y las técnicas para facilitar las especificaciones de una aplicación (TFEA). Además de la utilización del JavaScript y HTML como lenguajes del lado del cliente y PHP del lado del servidor.



# Capítulo 2

## 2.1 Introducción

En el presente capítulo se abordarán temas referentes al sistema, profundizando en sus características y en la situación problemática existente.

La metodología de desarrollo seleccionada para realizar la propuesta de sistema integrado de los módulos del proyecto Infodrez plantea una serie de flujos de trabajos en los cuáles se generan artefactos que constituyen la entrada al próximo flujo. Este capítulo se centra en la fase de Inicio, realizando actividades comprendidas en el “Modelo de Negocio” y “Requerimientos”, creando una serie de modelos que facilitan la comunicación, debido a la utilización del lenguaje estándar UML, y ayudan encontrar la solución a un nivel determinado de abstracción. Se utiliza para la construcción de estos modelos el Visual Paradigm, producto a las ventajas que proporciona.

Se realiza un modelo de dominio como alternativa al modelo de negocio. Se identifican los requisitos funcionales y no funcionales del sistema y se utilizan técnicas para la validación de los funcionales, persiguiendo detectar posibles errores o inconsistencias, evitar la obtención de resultados inesperados y la pérdida de tiempo al trabajar sobre la base de un mal levantamiento de requisitos.

## 2.2 Desarrollo

### 2.2.1 Fundamentación de la situación problemática

La Cátedra Honorífica de Ajedrez “Remberto Fernández” perteneciente a la Universidad de las Ciencias Informáticas unido al proyecto Infodrez juegan un papel fundamental en la informatización del ajedrez cubano. Actualmente este proyecto tiene concebido herramientas que gestionan información relacionada con las principales áreas de este deporte: arbitraje, estadística, torneos, y juegos en tiempo real.



El área de arbitraje permite informatizar la labor de un árbitro en un evento, a partir de la creación de torneos, el control de las rondas que lo componen además de los reportes que genera que ofrecen información de cada jugador del torneo. La estadística posibilita a jugadores y entrenadores hacer un estudio del rendimiento en un período de tiempo determinado, además de obtener las características de sus partidas. Mientras que el visor de torneos facilita la gestión de la información relacionada con los torneos de ajedrez, para que toda la información se encuentre disponible en un mismo lugar y pueda ser consultada de forma eficiente y rápida.

Según lo expuesto anteriormente en algunos de los módulos se gestiona u ofrece la misma información, lo que provoca una serie de problemas que traen consigo serias consecuencias para el cliente, además de no ser una solución informática factible. Estos problemas son:

### **Réplica de código y réplica de reportes**

Visor de torneos y arbitraje trabajan con información referente a los torneos, en común tienen código sobre los reportes que generan los resultados de cada una de las rondas, la variación de Elo y rating performance de cada jugador durante el evento. Los reportes que se generan en cada uno de estos módulos no son estándares entre sí, aunque hacen referencia al mismo tipo de información.

Ambos sistemas trabajaban la gestión del torneo presentando las siguientes funcionalidades:

- ❖ Creación de un torneo (características, definición de grupo y asignación de jugadores).
- ❖ Generación del emparejamiento de cada ronda según sistema de juego
- ❖ Sistema de clasificación del torneo

### **Información no centralizada**

Los módulos de visor de torneos, arbitraje y estadística, trabajan en principio con la información que se genera a partir del desempeño de un jugador en un torneo. Existe inconsistencia entre esta información debido a la no centralización de la misma. Los principales problemas identificados han sido:

- ❖ Nombres de jugadores duplicados y escritos de formas diferentes
- ❖ Nombres de eventos duplicados y escritos de formas diferentes
- ❖ No correspondencia entre partidas jugadas en un torneo





### **Información no estandarizada**

Las informaciones y reportes que se generan en visor de torneos, arbitraje y estadística son similares sin embargo la manera de mostrarse a los usuarios son diversas.

### **2.2.2 Propuesta del sistema**

Se pretende diseñar una plataforma integrada usando software libre que abarque todos los ámbitos de la cultura ajedrecística, configurable según las necesidades de los usuarios y reutilizable para una amplia gama de manifestaciones deportivo-recreativas de similar carácter.

Con el diseño de esta plataforma se proporcionará una aplicación que podrá ser utilizada para brindar características de cualquier jugador, ver torneos que estén registrados o efectuándose en tiempo real en la misma. El sistema debe satisfacer los requisitos funcionales y las necesidades de los usuarios a los cuales va enfocado el producto.

### **2.2.3 Modelo de negocio**

El primero de los flujos que propone la metodología RUP es el Modelo de Negocio, donde se requiere identificar el problema para luego ofrecer una solución, lo que hace necesario lograr una efectiva comunicación entre usuarios e integrantes del equipo de desarrollo.

#### **2.2.3.1 Modelo de dominio**

Luego de un análisis de los módulos que se pretenden integrar se pone en evidencia que no todos tienen los procesos del negocio bien definidos y en vista a buscar uniformidad en la modelación general se decide utilizar un modelo de dominio, siendo este un subconjunto del modelo de negocio. El modelo de dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema; se realiza un modelo del entorno del software, y no del software. Se concibe como un diccionario visual de abstracciones que será utilizado en fases posteriores y cuya función principal es ayudar a comprender el problema a tratar.



A continuación una breve explicación de las clases conceptuales que definen el dominio del problema para un mejor entendimiento del diagrama de clases del dominio:

- ❖ Director del torneo: Máximo responsable de un torneo de ajedrez. Controla que estén todas las condiciones creadas para la realización del torneo.
- ❖ Federación de Ajedrez: Es la organización rectora de todos los eventos relacionados con el ajedrez a la cual se le entregan todos los reportes oficiales de los diferentes torneos para que sean remitidos a la FIDE.
- ❖ Organizador: Es la persona encargada de organizar todos los torneos ajedrecísticos.
- ❖ Jugador: Son los protagonistas en una partida de ajedrez en un torneo.
- ❖ Árbitro: Es la persona encargada guiar la ejecución de un torneo de ajedrez y velar por el cumplimiento de las leyes ajedrecísticas dentro del mismo.
- ❖ Ronda: Momento (día) en el que se enfrentan los jugadores en un torneo. El evento está compuesto por varias rondas.
- ❖ Sistema de clasificación: Sistema mediante el cual, a partir de los resultados de los jugadores durante el torneo, permite realizar una competencia entre los mismos a partir de los parámetros de clasificación, ya sea clasificación general del evento, clasificación por puntuación o clasificación por rendimiento.
- ❖ Número de sorteo: Número escogido por el jugador para determinar sus rivales en cada ronda según el sistema del juego definido.
- ❖ Tablas de emparejamiento: Tabla que registra orden de los jugadores con sus oponentes, en que ronda se enfrentarán y con qué color de piezas juegan. Muestra resultados que se van obteniendo en cada ronda.
- ❖ Comité de Apelaciones: Se define durante el Congresillo Técnico, formado por un grupo de árbitros que toman decisiones sobre reclamaciones que se realizan durante el torneo.
- ❖ Cuadro Sinóptico: Cuadro donde se refleja el resultado del enfrentamiento entre cada uno de los jugadores en cada una de las rondas durante todo el torneo, puntos acumulados por cada jugador, cual fue el promedio de Elo final de cada uno, la categoría del torneo y el por ciento de combatividad.



## Capítulo 2: Propuesta del sistema

---

- ❖ **Congresillo Técnico:** Reunión que se realiza antes de comenzar el torneo, se definen las bases técnicas del torneo con la aceptación de todos los jugadores.
- ❖ **Bases Técnicas:** En estas se define el sistema de desempate, los premios a otorgar, el comité de apelaciones, entre otros aspectos técnicos.
- ❖ **Rating Elo:** Fuerza de juego, de cada jugador la cual se representa por un valor numérico.
- ❖ **Grupo de torneo:** Grupos de jugadores en los que se divide el torneo.
- ❖ **Invitado:** Persona que solicita información en un torneo de ajedrez.
- ❖ **Partida:** partida de Ajedrez
- ❖ **Torneo:** evento donde participan dos o más jugadores
- ❖ **Reporte:** Contiene información solicitada por una persona sobre jugadores, rondas, partidas e imágenes del torneo.
- ❖ **Bibliografía del Jugador:** Toda la información sobre un jugador determinado.
- ❖ **Reporte Rondas:** Información sobre las rondas del torneo.
- ❖ **Galería de imágenes:** reporte a través de imágenes de los jugadores y de todo lo que acontece en el torneo.
- ❖ **Reporte del Jugador:** Información sobre las partidas y los resultados de un jugador.

A continuación se muestra el modelo de dominio de cada uno de los módulos de forma separada. Para ver el modelo de dominio de los módulos integrados consultar [Anexo # 2](#)



Figura 2: Modelo de dominio del módulo Arbitraje

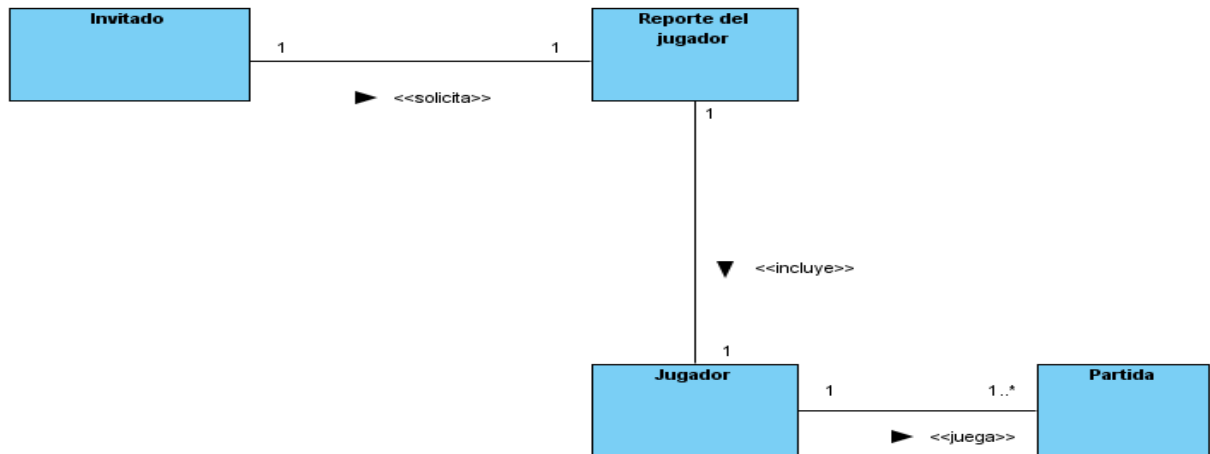
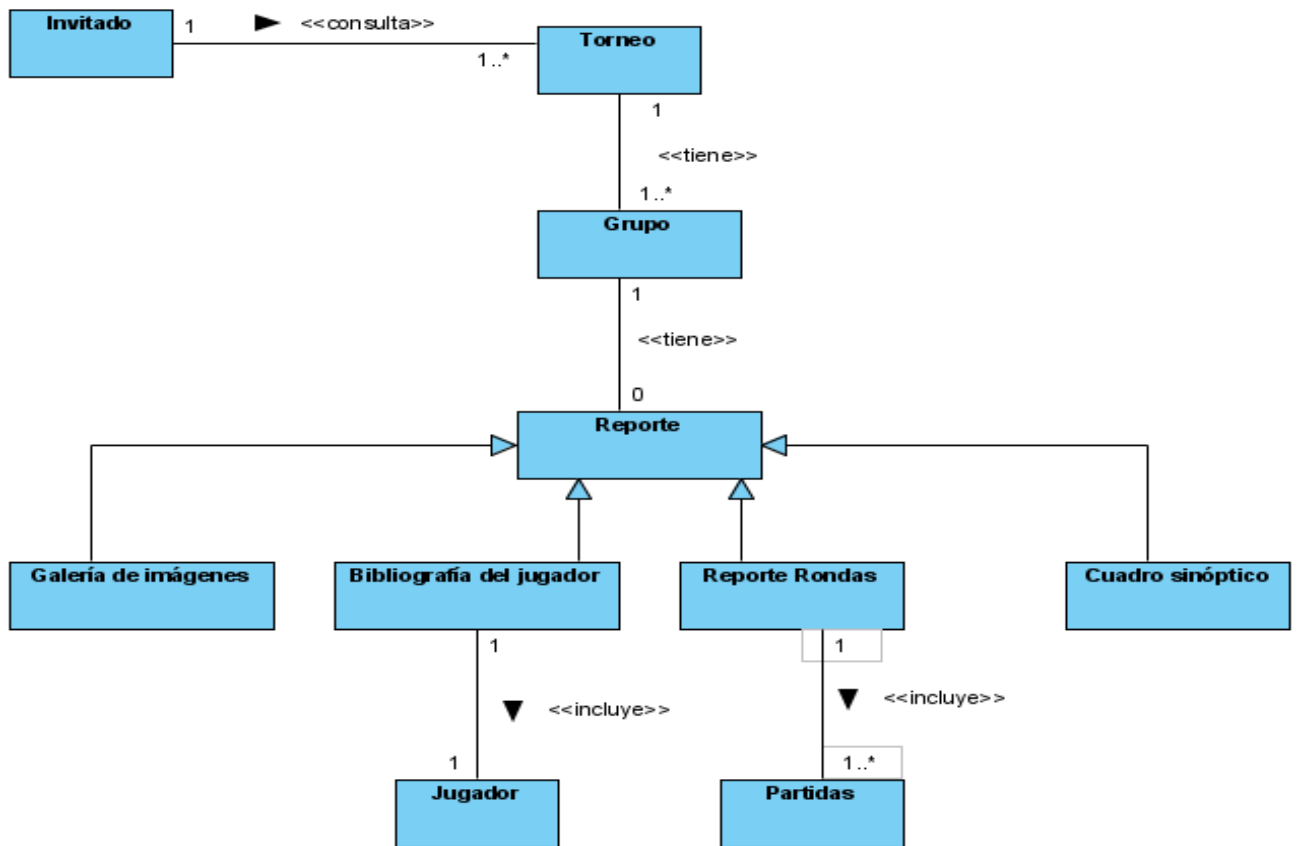


Figura 3: Modelo de dominio del módulo Visor de Torneos







### 2.2.3.2 Reglas del Negocio

Para un mejor entendimiento del negocio seguidamente se establecen las reglas del mismo:

- ❖ Todos los nomencladores deben tener un nombre real asociado (distinto de vacío)
- ❖ Los títulos de árbitros FIDE que existen son árbitro internacional (AI) y árbitro FIDE (AF). Las federaciones pueden establecer (en su marco) las que ellas deseen, pero no serán reconocidos por la FIDE a no ser que sean los establecidos por ella.
- ❖ Los títulos de jugadores, reconocidos por la FIDE son GM, GMF, MI, MIF, MF, MFF. Las federaciones pueden establecer (en su marco) las que ellas deseen, pero no serán reconocidos por la FIDE a no ser que sean los establecidos por ella.
- ❖ Para los ritmos de juegos existen 3 categorías: 1- Blitz (partidas de menos de 25 minutos), Rápidos (partidas entre 25 minutos y 1 hora) y FIDE (Mas de 1 hora).
- ❖ Para el desempate entre jugadores o equipos en un torneo, se escogerán los sistemas de desempate escogidos antes de comenzar el evento, por el orden establecido también en el mismo.
- ❖ En los torneos por equipos, cada tablero puede solo ascender en el orden, además, puede hacerlo las n posiciones que se definan la reglamentación del evento siempre y cuando se hayan puesto a descansar, n jugadores de tableros superior en su equipo.
- ❖ Durante el sorteo, a cada jugador se le asigna un único número, el cual tendrá el mismo valor durante el evento.
- ❖ Un árbitro gestiona solo partidas oficiales, o sea, que se hayan desarrollado en un torneo y que se tenga como constancia la planilla de la misma por ambos jugadores.

### 2.2.5 Requerimientos

En el flujo de trabajo de Requerimientos se define que es lo que debe hacer el sistema, siendo necesaria la captura de requisitos para identificar las funcionalidades requeridas y las restricciones impuestas. La IEEE (Institute of Electrical and Electronics Engineers), define requerimiento como:



- ❖ Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo
- ❖ Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

Características que deben tener los requerimientos:

- ❖ Especificados por escrito.
- ❖ Posibles de probar o verificar.
- ❖ Descritos como una característica del sistema a entregar.
- ❖ Lo más abstracto y conciso posible. Para evitar malas interpretaciones.

Los requisitos se pueden clasificar en 2 categorías:

- ❖ Requisitos funcionales.
- ❖ Requisitos no funcionales.

### **2.2.5.1 Requisitos funcionales**

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. En el negocio existen actividades que serán posibles automatizaciones, estas no constituyen exactamente requisitos funcionales, pero son el inicio para identificar qué debe hacer el sistema.

Funcionalidades que debe cumplir el sistema:

#### **RF 1 Autenticar usuario**

#### **RF 2 Gestionar usuario**

##### **2.1 Adicionar Usuario**

##### **2.2 Ver Usuario**



2.3 Modificar Usuario

2.4 Eliminar Usuario

### **RF 3 Gestionar Árbitro**

3.1 Adicionar Árbitro

3.2 Ver Árbitro

3.3 Modificar Árbitro

3.4 Eliminar Árbitro

### **RF 4 Gestionar Ciudad**

4.1 Adicionar Ciudad

4.2 Modificar Ciudad

4.3 Eliminar Ciudad

### **RF 5 Gestionar Federación**

5.1 Adicionar Federación

5.2 Ver Federación

5.3 Modificar Federación

5.4 Eliminar Federación

### **RF 6 Gestionar Título del Árbitro**

6.1 Adicionar Título del Árbitro

6.2 Modificar Título del Árbitro

6.3 Eliminar Título del Árbitro





### **RF 7 Gestionar Título del Jugador**

7.1 Adicionar Título del Jugador

7.2 Modificar Título del Jugador

7.3 Eliminar Título del Jugador

### **RF 8 Adicionar Ritmo de Juego**

### **RF 9 Mostrar Ritmo de Juego**

### **RF 10 Mostrar Sistemas de desempates**

### **RF 11 Gestionar Jugador**

11.1 Adicionar Jugador

11.2 Ver Jugador

11.3 Modificar Jugador

11.4 Eliminar Jugador

### **RF 12 Gestionar Equipo**

12.1 Adicionar Equipo

12.2 Ver Equipo

12.3 Modificar Equipo

12.4 Eliminar Equipo

### **RF 13 Gestionar Torneo**

13.1 Adicionar Torneo

13.2 Ver Torneo



13.3 Modificar Torneo

13.4 Eliminar Torneo

### **RF 14 Gestionar jugadores del torneo**

14.1 Adicionar jugadores a un torneo

14.2 Eliminar jugadores de un torneo

### **RF 15 Buscar jugador**

15.1 Listar jugadores por criterios de búsqueda

### **RF 16 Definir sistemas de desempate**

### **RF 17 Elaborar Bases técnicas**

### **RF 18 Realizar Sorteo**

### **RF 19 Realizar emparejamiento**

### **RF 20 Gestionar Ronda**

20.1 Adicionar Ronda

20.2 Modificar Ronda

### **RF 21 Gestionar partida**

21.1 Registrar Partida

21.2 Ver Partida

21.3 Modificar Partida

21.4 Eliminar Partida

### **RF 22 Cálculo del Rating Elo**



**RF 23 Mostrar Cuadro Sinóptico**

**RF 24 Mostrar emparejamiento y resultados**

**RF 25 Mostrar Clasificación General**

**RF 26 Gestionar Grupo**

26.1 Adicionar Grupo

26.2 Ver Grupo

26.3 Modificar Grupo

26.4 Eliminar Grupo

**RF 27 Gestionar equipos de un grupo**

27.1 Adicionar equipos a un grupo

27.2 Eliminar equipos de un grupo

**RF 28 Buscar equipo**

28.1 Listar equipos por criterios de búsqueda

**RF 29 Gestionar jugadores de un equipo**

29.1 Adicionar jugadores a un equipo

29.2 Eliminar jugadores de un equipo

**RF 30 Modificar tableros**

**RF 31 Definir Alineación**

**RF 32 Mostrar clasificación individual por tableros**

**RF 33 Consultar jugador**



**33.1** Listar jugadores

**33.2** Ver detalles de los jugadores

### **RF 34 Consultar partida**

**34.1** Listar partidas

**34.2** Ver detalles de las partidas

### **RF 35 Gestionar Imagen**

**35.1** Adicionar Imagen

**35.2** Eliminar Imagen

### **RF 36 Consultar información del torneo**

**36.1** Listar Torneos

**36.2** Ver detalles de los torneos

### **RF 37 Importar PGN**

#### **2.2.5.2 Técnicas de validación de requerimientos**

El proceso de obtención de requerimientos es un aspecto importante para el desarrollo de las etapas posteriores, un error en los flujos de trabajos iniciales puede traer como consecuencias un sistema que no cubra las expectativas de los usuarios.

Una vez definidos los requisitos, éstos necesitan ser validados con el objetivo de comprobar que definen realmente el sistema que el cliente desea o el usuario necesita. Se debe asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de los requisitos son correctos. Existen



técnicas para la validación de estos requerimientos, las aplicadas a la propuesta de sistema integrado de los módulos del proyecto Infodrez son:

❖ **Reviews o Walk-throughs**

Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante.

❖ **Prototipos**

Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario (Olsina, 1999).

- Un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y entender mejor el problema y su solución.
- Un prototipo puede revelar errores u omisiones en los requerimientos propuestos, favorece la comunicación entre clientes y desarrolladores, da una primera visión del producto.
- Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

Para ver resultado de la aplicación de esta técnica consultar [Anexo # 3](#)

### **2.2.5.3 Requisitos no funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Características que hacen al producto usable, atractivo, rápido, confiable, entre otras. Usualmente están vinculados a requerimientos funcionales; una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.



Requisitos no funciones que debe tener el producto:

❖ **Requerimientos de Software**

Para la instalación del servidor:

- Servidor Apache versión 2 o superior.
- PHP 4 o superior con módulo LDAP para autenticar usuarios de Directorio Activo.
- Servidor de bases de datos MySQL versión 4 o superior o Postgre versión 8 o superior.
- Sistemas Operativos GNU/Linux o Microsoft Windows.

Para la interpretación por clientes (Navegadores Web):

- Internet Explorer v6 o superior (recomendado).
- Mozilla Firefox 2.\* o superior (recomendado).

❖ **Requerimientos de Hardware**

- Mínimo Requerido: RAM 256 Mb, Disco duro 3 Gb.
- Recomendado: RAM 512 Mb, Disco duro 10Gb.
- Conexión de red por MODEM o fibra óptica.

❖ **Restricciones en el diseño y la implementación**

El sistema se recomienda que funcione sobre una plataforma de sistema operativo GNU/Linux a modo texto, basado en el modelo cliente-servidor. Utilizar además para su desarrollo herramientas como el Quanta, Bluefish, ZendStudio, phpmyadmin y Visual Paradigm.

Portabilidad: es un sistema multiplataforma.

❖ **Requerimientos de apariencia o interfaz externa**

Será fácil de entender, de comprender y de ser usado, con una interfaz amigable. Rápida en cuanto al tiempo de respuesta a los usuarios y a la velocidad de generación de páginas y de gráficos.

❖ **Requerimientos de Seguridad**

Verificar autenticidad y niveles de privilegio del usuario antes de hacer cualquier operación



dentro del sistema. Comprobar que las funcionalidades sean mostradas en correspondencia con los niveles de privilegios. Protección contra operaciones no autorizadas que en algún momento puedan afectar la integridad de los datos. La autenticación se realiza contra un LDAP que solo permite el acceso de usuarios del dominio.

Confidencialidad: solo se permitirá conexiones mediante usuarios del directorio activo.

### ❖ Requerimientos de Usabilidad

Interfaz Web fácil de usar para personas que no son especialistas en informática, permitiendo un mejor uso e interacción con los usuarios, y mayor facilidad de intercambio de información.

### ❖ Requerimientos de Soporte

Legales: este módulo está adjunto a la plataforma Infodrez que está basada en la licencia GNU/GPL.

Confiabilidad: realiza control de errores y recuperación ante fallos.

#### 2.2.5.4 Actores del sistema

Un actor del sistema está definido como el individuo, grupo de individuos, sistema automatizado o máquina que interactúa con el mismo intercambiando información. Los trabajadores del negocio que tienen actividades a automatizar son candidatos a actores del sistema. Si algún actor del negocio continúa interactuando con el sistema, entonces también será actor del sistema

Actores del sistema con su correspondiente descripción:

**Tabla 1: Descripción de los actores del sistema**

Actor del sistema	Descripción
Usuario	Es un actor genérico del cual heredan Árbitro y Administrador.



Árbitro	Encargado de velar que se cumplan las leyes del ajedrez y realizar todo el proceso de arbitraje en un torneo.
Administrador	Gestiona toda la información manejada en el sistema.
Invitado	Persona que consulta la información de un torneo o un jugador registrado en el sistema.

### 2.2.5.5 Diagrama de casos de uso del sistema

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

El diagrama de casos de uso del sistema describe la relación entre actores y casos de usos. Es un artefacto que modela las funciones deseadas para el sistema y su entorno.







### 2.2.5.6 Descripción de los casos de uso del sistema

La descripción de los casos de uso del sistema describe las acciones que se realizan durante la interacción entre el actor y el sistema, o sea describe el flujo de actividades que realiza un actor al hacer uso del sistema y las respuestas que emite el mismo.

A continuación las descripciones de los casos de uso críticos del sistema. Para ver la descripción de los restantes consultar [Anexo # 4](#)

**Tabla 1: Descripción del CUS Gestionar Árbitro**

<b>Caso de Uso:</b>	Gestionar Árbitro
<b>Actores:</b>	Administrador
<b>Propósito:</b>	Permite al administrador gestionar toda la información referente a los árbitros: adicionar, ver, modificar o eliminar árbitros en el sistema.
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador dentro de las opciones de administración va a tener la posibilidad de adicionar, modificar o eliminar árbitros.
<b>Precondiciones:</b>	
<b>Poscondiciones:</b>	-Árbitro adicionado a la Base de Datos.  - Árbitro eliminado de la Base de Datos.  -Datos del Árbitro actualizados.
<b>Referencias</b>	RF 3, RF 3.1, RF 3.2, RF 3.3, RF 3.4
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	



Acción del Actor	Respuesta del Sistema
1-El administrador dentro de la página principal escoge la opción de Gestionar Árbitro del Menú de Administración	1.1- El sistema brinda las opciones de:  -Adicionar Árbitro  -Ver Árbitro  -Modificar Árbitro.  -Eliminar Árbitro.
2 -El administrador selecciona una de las opciones.	2.1- Si selecciona “Adicionar Árbitro” ver sección <b>Adicionar Árbitro</b>  - Si selecciona “Ver Árbitro” ver sección <b>Ver Árbitro</b>  -Si selecciona “Modificar Árbitro” ver sección <b>Modificar Árbitro</b> .  -Si selecciona “Eliminar Árbitro” ver sección <b>Eliminar Árbitro</b> .
<b>Sección: “Adicionar Árbitro”</b>	
Acción del Actor	Respuesta del Sistema
1-El administrador selecciona la opción de Adicionar Árbitro	1.1-El sistema muestra el formulario con los siguientes campos a llenar:  -Nombre  -Apellidos  -Rol



## Capítulo 2: Propuesta del sistema

	-Título
2- El administrador llena lo campos necesarios pulsa el botón Aceptar	2.1-El sistema verifica que los datos introducidos son correctos.
	2.2- El sistema verifica que el árbitro no esté registrado, en caso positivo lo registra en la base de datos y muestra un mensaje evidenciando el registro del nuevo árbitro, concluyendo con esto el caso de uso del sistema.
<b>Flujo Alternativo de Eventos</b>	
<b>Sección: "Adicionar Árbitro"</b>	
	2.1-Si los datos introducidos no son correctos el sistema muestra un mensaje informando que los datos no son válidos. Se procede al flujo normal de eventos a partir del paso 2.
	2.2- Si el árbitro está registrado el sistema muestra un mensaje informando la situación.
<b>Sección: "Ver Árbitro"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1-El administrador selecciona el árbitro	1.1-El sistema muestra los datos del árbitro, termina así el caso de uso
<b>Sección: "Modificar Árbitro"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>



## Capítulo 2: Propuesta del sistema

1- El administrador selecciona la opción de Modificar Árbitro	1.1-El sistema muestra un listado de todos los árbitros registrados
2- El administrador selecciona el árbitro que desea y pulsa el botón Aceptar.	2.1-El sistema muestra un formulario con los datos del árbitro que se desea modificar.
3- El administrador modifica los datos necesarios y pulsa el botón Aceptar.	3.1- El sistema verifica que los datos modificados son válidos, en caso positivo estos se actualizan en la base de datos, concluyendo así el caso de uso del sistema.
<b>Flujo Alternativo de Eventos</b>	
<b>Sección: "Modificar Árbitro"</b>	
	3.1-Si los datos no son correctos, éste muestra un mensaje indicando en que parte del formulario fue donde estuvo el error. Se procede al flujo normal de eventos desde la acción 3.
<b>Sección: "Eliminar Árbitro "</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1-El administrador selecciona la opción de Eliminar Árbitro	1.1- El sistema muestra un listado con los árbitros registrados
2- El administrador selecciona el árbitro que desea eliminar del sistema y pulsa el botón Aceptar	2.1- El sistema elimina el árbitro de la base de datos y muestra un mensaje de que éste ha sido eliminado, terminando así el caso de uso.



**Tabla 2: Descripción del CU Gestionar Jugador**

<b>Caso de Uso:</b>	Gestionar Jugador
<b>Actores:</b>	Árbitro
<b>Propósito:</b>	Permite al árbitro gestionar toda la información referente a los jugadores: adicionar, ver, modificar y eliminar un jugador.
<b>Resumen:</b>	El caso de uso se inicia cuando el árbitro dentro del Menú de Arbitraje tiene la opción de gestionar la información de los jugadores.
<b>Precondiciones:</b>	
<b>Poscondiciones:</b>	-Jugador adicionado a la Base de Datos. -Jugador eliminado de la Base de Datos. -Datos del Jugador actualizados.
<b>Referencias</b>	RF 11, RF 11.1, RF 11.2, RF 11.3, RF 11.4
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El árbitro selecciona dentro de la página principal la opción Gestionar Jugador del Menú Torneo.	1.1- El sistema brinda las opciones de: - Adicionar Jugador -Ver Jugador -Modificar Jugador



	-Eliminar Jugador
2 -El árbitro selecciona una de las opciones.	2.1- Si selecciona “Adicionar Jugador” ver sección <b>Adicionar Jugador</b>  - Si selecciona “Ver Jugador” ver sección <b>Ver Jugador</b>  -Si selecciona “Modificar Jugador” ver sección <b>Modificar Jugador</b>  -Si selecciona “Eliminar Jugador” ver sección <b>Eliminar Jugador</b>
<b>Sección: “Adicionar Jugador”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El árbitro selecciona la opción de Adicionar Jugador	1.1- El sistema muestra el formulario con todos las opciones para que el árbitro entre todos los datos que se necesitan para adicionar un jugador al sistema.  -Nombre  -Apellido  -Sexo  -Titulo  -Código FIDE  -Federación



	-Fecha de nacimiento  -Elo  -K
2- El árbitro introduce los datos necesarios para adicionar un nuevo jugador y pulsa el botón Aceptar.	2.1-El sistema verifica que los datos insertados son correctos.
	2.2-El sistema verifica que el jugador no se encuentre registrado a la base de datos, en caso afirmativo lo adiciona a la base de datos y muestra un mensaje informando que el registro se realizó correctamente, concluyendo con esto el caso de uso del sistema.
<b>Flujo Alternativo de Eventos</b>	
<b>Sección: "Adicionar Jugador"</b>	
	2.1-Si los datos introducidos son incorrectos el sistema muestra un mensaje indicando en que elemento del formulario fue donde estuvo el error. Se procede al flujo normal de eventos desde el paso 2.
	2.2-Si el jugador ya existe el sistema emite un mensaje informando la existencia del mismo.
<b>Sección: "Ver Jugador"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>





1-El árbitro selecciona el jugador a consultar datos	1.1-El sistema muestra los datos del jugador seleccionado, termina así el caso de uso
<b>Sección: “Modificar Jugador ”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El árbitro selecciona el jugador que desea modificar y marca la opción de Modificar Jugador	1.1- El sistema verifica que el jugador no pertenezca a un torneo que no haya finalizado, en caso positivo muestra los datos del jugador a ser modificados
2- El árbitro modifica los datos y pulsa el botón Aceptar	2.1-El sistema verifica que los datos modificados son válidos, en caso positivo estos se actualizan en la base de datos, concluyendo así el caso de uso del sistema.
<b>Flujo Alternativo de Eventos</b>	
<b>Sección: “Modificar Jugador ”</b>	
	1.1-Si el jugador pertenece a un torneo que no haya finalizado se muestra un mensaje informando sobre la situación
	2.1- Si los datos que se modifican son incorrectos el sistema mostrará un mensaje señalando el campo donde los datos no son válidos. Se procede al flujo normal de eventos a partir del paso 3.
<b>Sección: “Eliminar Jugador ”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>



1- El árbitro selecciona el jugador que desea eliminar y marca la opción de Eliminar Jugador	1.1- El sistema verifica que el jugador no pertenezca a un torneo que no haya finalizado, en caso positivo muestra un mensaje de confirmación para verificar que realmente se quiere eliminar el jugador.
2- El árbitro pulsa el Aceptar del mensaje de confirmación.	2.1- El sistema elimina el jugador de la base de datos y muestra un mensaje de que ha sido eliminado, terminando así el caso de uso.
<b>Flujo Alternativo de Eventos</b>	
<b>Sección: "Eliminar Jugador"</b>	
	1.1- Si el jugador pertenece a un torneo que no haya finalizado se muestra un mensaje informando sobre la situación
2-El árbitro pulsa el Cancelar del mensaje de confirmación.	

**Tabla 4: Descripción del CU Gestionar Equipo**

<b>Caso de Uso:</b>	Gestionar Equipo
<b>Actores:</b>	Árbitro
<b>Propósito:</b>	Brinda la posibilidad a los árbitros de gestionar la información relacionada con los equipos: adicionar, ver, modificar o eliminar equipos.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario selecciona la opción Equipo del Menú



	Arbitraje, el sistema brinda la posibilidad de gestionar los equipos.
<b>Precondiciones:</b>	
<b>Poscondiciones:</b>	-Nuevo equipo del torneo registrado en la Base de Datos  -Equipo del torneo eliminado de la Base de Datos  -Actualización de los datos de un equipo del torneo
<b>Referencias</b>	RF 12, RF 12.1, RF 12.2, RF 12.3
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1-1-El árbitro selecciona dentro de la página principal la opción Gestionar Equipo del Menú Torneo.	1.1- El sistema brinda las opciones de:  -Adicionar Equipo  -Ver Equipo  - Modificar Equipo  -Eliminar Equipo
2 -El árbitro selecciona una de las opciones.	2.1- Si selecciona “Adicionar Equipo” ver sección <b>Adicionar Equipo</b>  - Si selecciona “Ver Equipo” ver sección <b>Ver Equipo</b>  -Si selecciona “Modificar Equipo” ver sección <b>Modificar Equipo</b>  - Si selecciona “Eliminar Equipo” ver sección <b>Eliminar</b>



	Equipo
<b>Sección: “Adicionar Equipo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El árbitro selecciona la opción de Adicionar Equipo	1.1- El sistema muestra un formulario con los datos a ser llenados:  - Nombre  -Federación  -Logo
2- El árbitro introduce los datos para llenar los campos correspondientes, y pulsa el botón Aceptar.	2.1-El sistema verifica que los datos introducidos por el árbitro son válidos.
	2.2- El sistema verifica que el quipo no se encuentre registrado, en caso positivo se insertan los mismos en la base de datos, concluyendo así el caso de uso.
<b>Flujo Alternativo de Eventos</b>	
<b>Sección: “Adicionar Equipo”</b>	
	2.1- Si los datos que se introducen son incorrectos el sistema mostrará un mensaje señalando la presencia de un error al adicionar un equipo. Se procede al flujo normal de eventos a partir de la acción 2.
	2.2-Si el equipo ya existe el sistema emite un mensaje informando su existencia.



<b>Sección: “Ver Equipo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1-El árbitro selecciona el equipo	1.1-El sistema muestra los datos del equipo seleccionado
<b>Sección: “Modificar Equipo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario selecciona el equipo que desea modificar y marca la opción de Modificar Equipo	1.1-El sistema verifica que el equipo no pertenezca a un torneo que no haya finalizado, en caso positivo muestra todos los datos del equipo de torneo para que sean modificados.
2- El usuario modifica los datos deseados y pulsa el botón Aceptar.	2.1- El sistema verifica que los datos modificados son válidos, en caso positivo estos se actualizan en la base de datos, concluyendo así el caso de uso del sistema.
<b>Flujo Alternativo de Eventos</b>	
<b>Sección: “Modificar Equipo”</b>	
	1.2-Si el equipo pertenece a un torneo que no finalizó muestra un mensaje informando sobre la situación.
	2.1- Si los datos que se introducen son incorrectos el sistema mostrará un mensaje señalando el campo donde los datos no son válidos. Se procede al flujo normal de eventos a partir del paso 2.
<b>Sección: “Eliminar Equipo”</b>	



Acción del Actor	Respuesta del Sistema
1- El usuario selecciona el equipo que desea eliminar y marca la opción Eliminar Equipo	1.1-El sistema verifica que el equipo no pertenezca a un torneo que no haya finalizado, en caso positivo muestra un mensaje de confirmación para verificar que se quiere eliminar el equipo.
2- El usuario pulsa el botón Aceptar del mensaje de confirmación.	2.1 -El sistema lo elimina de la base de datos y muestra un mensaje informando que el equipo ha sido eliminado, terminando así el caso de uso.
<b>Flujo Alternativo de Eventos</b>	
<b>Sección: "Eliminar Equipo "</b>	
	1.2-Si el equipo pertenece a un torneo que no finalizó muestra un mensaje informando sobre la situación.
2- El usuario pulsa el botón Cancelar del mensaje de confirmación.	

**Tabla 5: Descripción del CU Gestionar Torneo**

<b>Caso de Uso:</b>	Gestionar Torneo
<b>Actores:</b>	Árbitro
<b>Propósito:</b>	Brinda la posibilidad a los árbitros de gestionar la información relacionada con los torneos: adicionar, ver, modificar, y eliminar torneos.
<b>Resumen:</b>	El caso de uso inicia cuando el árbitro selecciona la opción Torneo del Menú Arbitraje donde el sistema brinda la posibilidad de gestionar la información relacionada con los diferentes torneos.



<b>Precondiciones:</b>	
<b>Poscondiciones:</b>	<ul style="list-style-type: none"><li>- Torneo registrado en la Base de Datos.</li><li>- Torneo actualizado.</li><li>- Torneo eliminado de la Base de Datos.</li></ul>
<b>Referencias</b>	RF 13, RF 13.1, RF 13.2, RF 13.3, RF 13.4
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El árbitro selecciona dentro de la página principal la opción Gestionar Torneo del Menú Torneo.	1.1- El sistema brinda las opciones de: <ul style="list-style-type: none"><li>- Adicionar Torneo</li><li>-Ver Torneo</li><li>-Modificar Torneo</li><li>-Eliminar Torneo</li></ul>
2 -El árbitro selecciona una de las opciones.	2.1- Si selecciona “Adicionar Torneo” ver sección <b>Adicionar Torneo</b> <ul style="list-style-type: none"><li>- Si selecciona “Ver Torneo” ver sección <b>Ver Torneo</b></li><li>-Si selecciona “Modificar Torneo” ver sección <b>Modificar Torneo</b></li><li>-Si selecciona “Eliminar Torneo” ver sección</li></ul>



	<b>Eliminar Torneo</b>
<b>Sección: “Adicionar Torneo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El árbitro selecciona la opción de Adicionar Torneo	1.1- El sistema muestra un formulario con los datos a ser llenados:  -Nombre  -Fecha de inicio  -Fecha final  -Cantidad de grupos  -Árbitro Principal
2- El Árbitro introduce los datos para llenar los campos correspondientes y pulsa el botón Aceptar	2.1- El sistema verifica que los datos introducidos por el árbitro son válidos.
	2.2-El sistema verifica que el torneo no está registrado, en caso positivo se insertan los mismos en la base de datos y muestra un mensaje informando que el registro se realizó correctamente concluyendo así el caso de uso
<b>Flujo Alternativo de Eventos</b>	
<b>Sección: “Adicionar Torneo”</b>	
	2.1- Si los datos que se introducen son incorrectos el sistema mostrará un mensaje señalando el campo donde los datos no son válidos. Se procede al flujo normal de





	eventos a partir de la acción 2.
	2.2- Si el torneo ya está registrado el sistema muestra un mensaje informando la situación
<b>Sección: “Ver Torneo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1-El árbitro selecciona el torneo	1.1-El sistema muestra los datos del torneo, finalizando así el caso de uso.
<b>Sección: “Modificar Torneo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El árbitro selecciona el torneo que desea modificar y marca la opción de Modificar Torneo	1.1-El sistema verifica que el torneo no hay finalizado, en caso positivo muestra un formulario con los datos del torneo seleccionado
2- El árbitro modifica los datos y pulsa el botón Aceptar	2.1- El sistema verifica que los datos modificados son válidos, en caso positivo estos se actualizan en la base de datos, termina así el caso de uso.
<b>Flujo Alternativo de Eventos</b>	
<b>Sección: “Modificar Torneo”</b>	
	1.1-Si el torneo ya finalizó el sistema muestra el mensaje: “No se puede eliminar el torneo”.
	2.1- Si los datos que se introducen son incorrectos el sistema mostrará un mensaje señalando el campo donde los datos no son válidos. Se procede al flujo normal de



	eventos a partir del paso 2.
<b>Sección: “Eliminar Torneo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El árbitro selecciona el torneo que desea eliminar y marca la opción de Eliminar Torneo	1.1-El sistema verifica que el torneo no hay finalizado, en caso positivo muestra un mensaje de confirmación para verificar que realmente se quiere eliminar el torneo.
2- El árbitro pulsa el botón Aceptar del mensaje de confirmación.	2.1- El sistema elimina el torneo de la base de datos y muestra un mensaje de que ha sido eliminado, termina así el caso de uso.
<b>Flujo Alternativo de Eventos</b>	
<b>Sección: “Eliminar Torneo”</b>	
	1.2-Si el torneo ya finalizó el sistema muestra el mensaje: “No se puede eliminar el torneo”.
2-El árbitro pulsa el botón Cancelar del mensaje de confirmación.	

**Tabla 6: Descripción del CU Realizar Sorteo**

<b>Caso de Uso:</b>	Realizar Sorteo
<b>Actores:</b>	Árbitro
<b>Propósito:</b>	El propósito de éste caso de uso es realizar sorteo conociendo inicialmente la lista de jugadores o de equipos que participarán en el torneo.



<b>Resumen:</b>	El caso de uso comienza cuando el árbitro escoge la opción Sorteo del Menú Torneo y brinda la posibilidad de sortear lo mismo los torneos desarrollados en Round Robin individual como por equipos.
<b>Precondiciones:</b>	Debe estar creada la lista de jugadores o de equipos participantes en el torneo.
<b>Poscondiciones:</b>	Cada jugador o equipo tiene un número aleatorio asignado.
<b>Referencias:</b>	RF 18
<b>Prioridad:</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Sección: “Realizar Sorteo para un torneo Round Robin individual”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El árbitro selecciona dentro de la página principal la opción Sorteo del Menú Torneo.	1.1- El sistema muestra todos los jugadores participantes en el torneo, brindando la posibilidad de que el árbitro pueda realizar el sorteo.
2- El árbitro sortea los jugadores por orden de Rating Elo y Título.	2.1-El sistema asigna de forma aleatoria un número a cada jugador, mostrando dicho número al árbitro y brindando la posibilidad de deshacer el sorteo si es necesario
<b>Sección: “Realizar Sorteo para un torneo Round Robin por equipos ”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El árbitro selecciona la opción Sorteo del Menú Torneo.	1.1- El sistema muestra los equipos participantes en el torneo, brindando la posibilidad de que el árbitro pueda realizar el sorteo.
2- El árbitro sortea los equipos por el	2.1-El sistema asigna de forma aleatoria un número a cada



Elo promedio del equipo.	equipo, mostrando dicho número al árbitro y brindando la posibilidad de deshacer el sorteo si es necesario terminando así el caso de uso.
--------------------------	---

**Tabla 7: Descripción del CU Realizar emparejamiento**

<b>Caso de Uso:</b>	Realizar emparejamiento
<b>Actores:</b>	Árbitro
<b>Propósito:</b>	Mostrar el pareo de los torneos que se desarrollan tanto los Round Robin individuales como por equipos.
<b>Resumen:</b>	El caso de uso inicia cuando el árbitro selecciona la opción Pareo del menú Torneo, luego se muestra el emparejamiento de todas las rondas de un grupo de torneo.
<b>Precondiciones:</b>	Ya debe existir un torneo creado y debe haberse realizado el sorteo de los jugadores o de los equipos.
<b>Poscondiciones:</b>	Se muestra la tabla de pareo de un grupo del torneo que contiene los enfrentamientos por ronda.
<b>Referencias</b>	RF 19
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Sección: “Realizar emparejamiento para un torneo Round Robin individual”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El árbitro dentro de la página principal	1.1-El sistema verifica que estén registrados todos los



## Capítulo 2: Propuesta del sistema

pulsa el botón Pareo del Menú Torneo.	jugadores que van a participar en el torneo.
	1.2-El sistema verifica que todos los jugadores participantes tengan un número de sorteo asignado.
	1.3 - El sistema aplica el algoritmo del pareo, es decir guiado por el número de sorteo de cada jugador para el torneo busca el nombre de cada uno y construye un arreglo con todas las partidas organizadas por rondas.
	1.4-El sistema muestra todas las partidas organizadas por ronda con sus resultados en caso de que ya los tenga.
<b>Flujo Alternativo de Eventos</b>	
	1.1-Si no están todos los jugadores registrados en el torneo el sistema muestra un mensaje informando la situación.
	1.2- Si existe algún participante en el torneo que no cuenta aún con un número de sorteo el sistema muestra un mensaje informando que se debe realizar primero el sorteo para luego realizar el pareo.
<b>Sección: “Realizar emparejamiento para un torneo Round Robin por equipos ”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El árbitro pulsa el botón Pareo del Menú Torneo.	1.1-El sistema verifica que estén registrados todos los equipos que van a participar en el torneo.
	1.2-El sistema verifica que todos los equipos participantes tengan un número de sorteo asignado.
	1.3-El sistema ejecuta el algoritmo encargado de realizar



	pareo.
	1.4 - El sistema guiado por el número de sorteo de cada equipo para el torneo busca el nombre de cada uno de ellos y construye un arreglo con los enfrentamientos entre los equipos en cada una de las rondas
	1.5-El sistema muestra todos los enfrentamientos por ronda con sus resultados en caso de que ya los tenga.
<b>Flujo Alternativo de Eventos</b>	
	1.1-Si no están todos los equipos registrados en el torneo el sistema muestra un mensaje informando la situación.
	1.2- Si existe algún equipo en el torneo que no cuenta aún con un número de sorteo el sistema muestra un mensaje informando que se debe realizar primero el sorteo para luego realizar el pareo.

**Tabla 8: Descripción del CU Consultar información del torneo**

<b>Caso de Uso:</b>	Consultar información del torneo
<b>Actores:</b>	Invitado
<b>Propósito:</b>	Permite al invitado ver los datos de un torneo.
<b>Resumen:</b>	El caso de uso se inicia cuando el invitado visita la plataforma y dentro del Menú Consultar torneo le brinda la posibilidad de consultar información de



	un torneo.
<b>Precondiciones:</b>	El sistema debe estar disponible.
<b>Poscondiciones:</b>	Información del torneo mostrado al Invitado.
<b>Referencias</b>	RF 36
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1-El invitado dentro de la página principal accede a la opción Consultar Torneo.	1.1-El sistema muestra los siguientes criterios de búsqueda:  -Nombre  -Fecha de inicio  -Árbitro Principal
2-El invitado selecciona el torneo del que desea ver información	2.1-El sistema accede a la Sección "Ver torneo" del caso de uso Gestionar Torneo



### 2.3 Conclusiones

En éste capítulo se crea una propuesta de las características funcionales que se proponen para el sistema. Se realiza el modelo de dominio, como alternativa al modelo de negocio a partir cada uno de los módulos para su posterior integración. Se validan los requisitos funcionales con la aplicación de las técnicas planteadas anteriormente, aplicándose solamente a los requisitos de los módulos no implementados. Luego de este proceso se identifican los actores y casos de usos del sistema, construyéndose el diagrama de casos de uso correspondiente y la descripción de cada uno de los casos de uso.





# Capítulo 3

### 3.1 Introducción

El desarrollo de este capítulo se enmarca en las actividades que propone RUP para la fase de elaboración fundamentalmente el flujo de trabajo de Análisis y Diseño, modelándose una serie de artefactos que contribuyen a la implementación de la propuesta de sistema integrado de los módulos del proyecto Infodrez. Se realizan los diagramas de clases del análisis y del diseño, así como los diagramas de colaboración para el análisis.

Durante el diseño se aplican los patrones GRASP, para la asignación de responsabilidades a las clases.

### 3.2 Desarrollo

#### 3.2.1 Arquitectura y Estilos Arquitectónicos

La arquitectura de software está formada por la estructura de los elementos de un programa o sistema, sus interrelaciones y los principios y reglas que gobierna su diseño y evolución a lo largo del tiempo. También es definida por David Garlan y Mary Shaw como un nivel de diseño que hace foco en aspectos "más allá de los algoritmos y estructuras de datos de la computación; el diseño y especificación de la estructura global del sistema es un nuevo tipo de problema".

Para desarrollar una arquitectura potente se utilizan estilos arquitectónicos, éstos indican los tipos de componentes y conectores involucrados, así como patrones y restricciones de interconexión o composición entre ellos.

En el desarrollo de la propuesta de sistema integrado de los módulos del proyecto Infodrez se utiliza el estilo N Capas.



### 3.2.1.4 Estilo N Capas

El Estilo N Capas se utiliza para descomponer un sistema en capas (Presentación, Lógica de Negocio y Acceso a Datos), posibilitando el manejo de su complejidad. Cada una de éstas debe ocuparse de un nivel específico del problema, y debe tener poca cohesión con las demás. Este estilo arquitectónico presenta algunas ventajas como la reutilización de las capas, además de que el cambio en una de ellas no afectaría en gran medida las demás capas.

### 3.2.2 Análisis del Sistema

Durante el análisis se analizan los requisitos que se describen en la captura de requisitos, refinándolos y estructurándolos, es decir, el análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales.

Se generan artefactos como las clases del análisis y las realizaciones de los casos de uso del análisis.

#### 3.2.2.1 Clases del análisis

Representan la abstracción a un concepto, con operaciones y atributos poco detallados. Se pueden representar mediante 3 estereotipos básicos estándares: clase interfaz, clase control, y clase entidad.

A continuación se muestran los diagramas de clases del análisis de los casos de uso críticos del sistema. Para ver las restantes consultar [Anexo #5](#)



Figura 6: CU Gestionar Árbitro

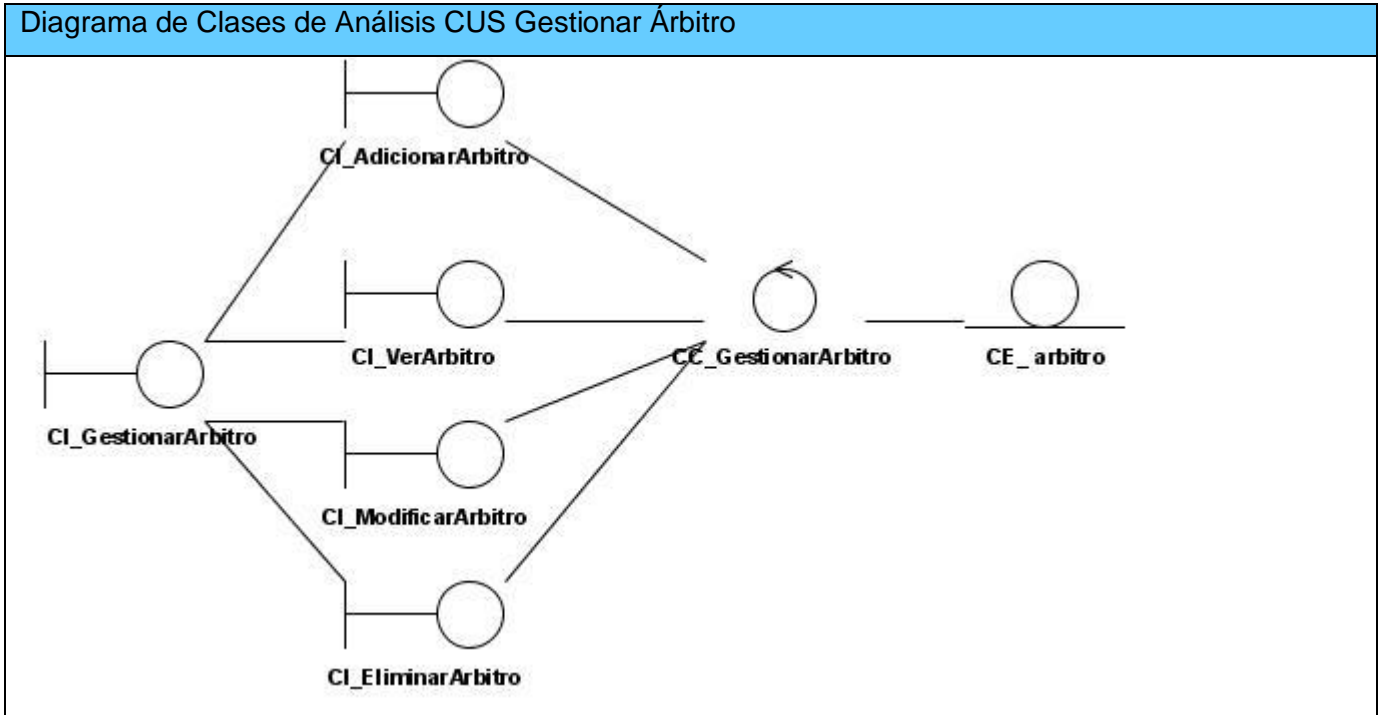




Figura 7: CU Gestionar Jugador

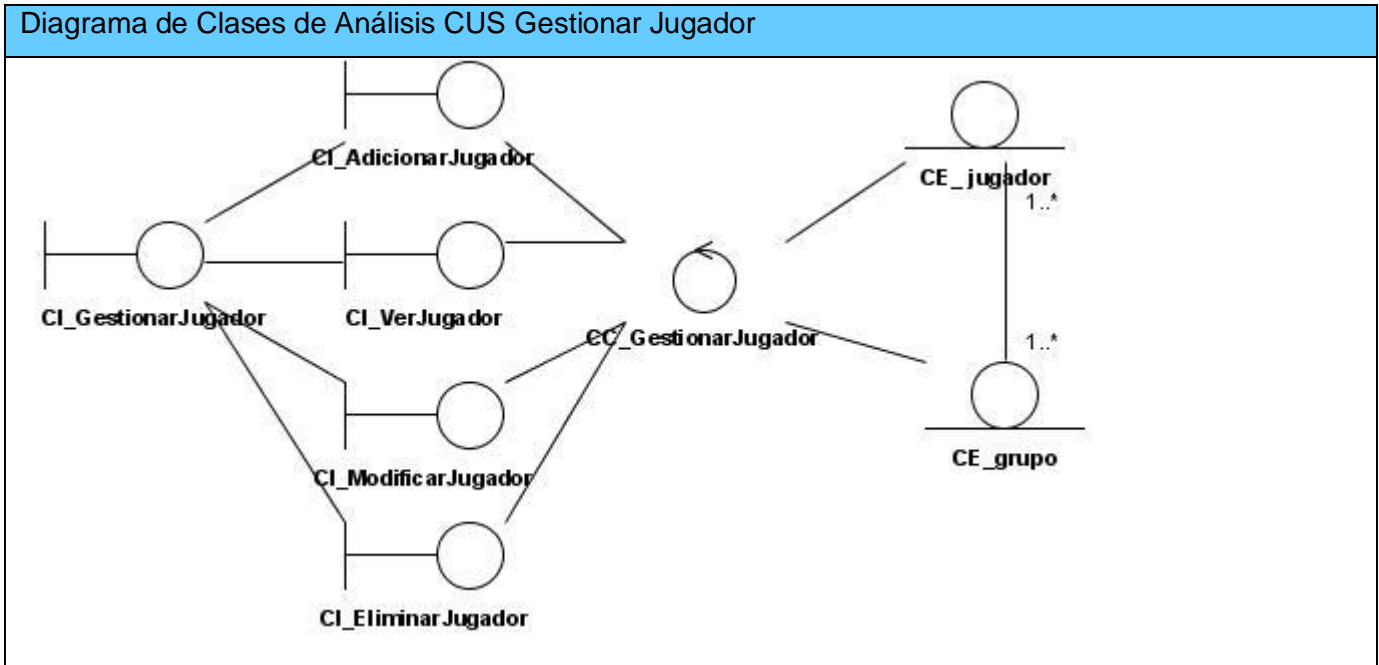
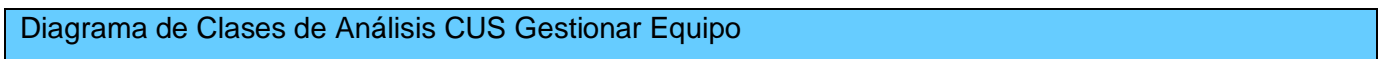


Figura 8: CU Gestionar Equipo



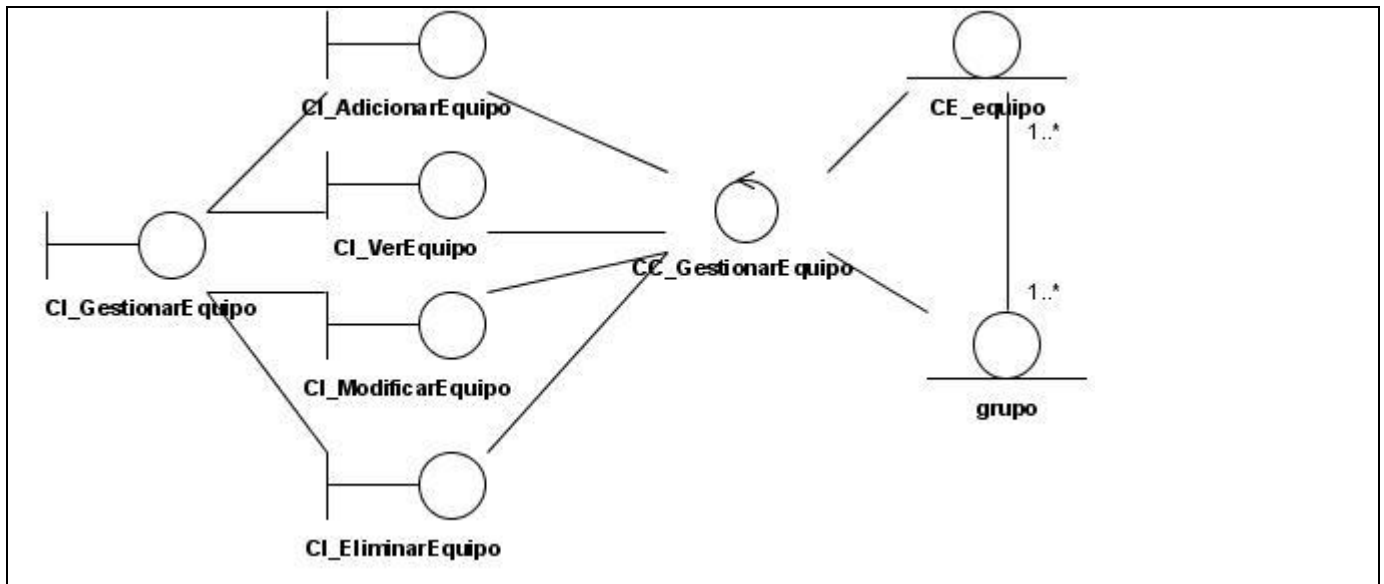


Figura 9: CU Gestionar Torneo

Diagrama de Clases de Análisis CUS Gestionar Torneo

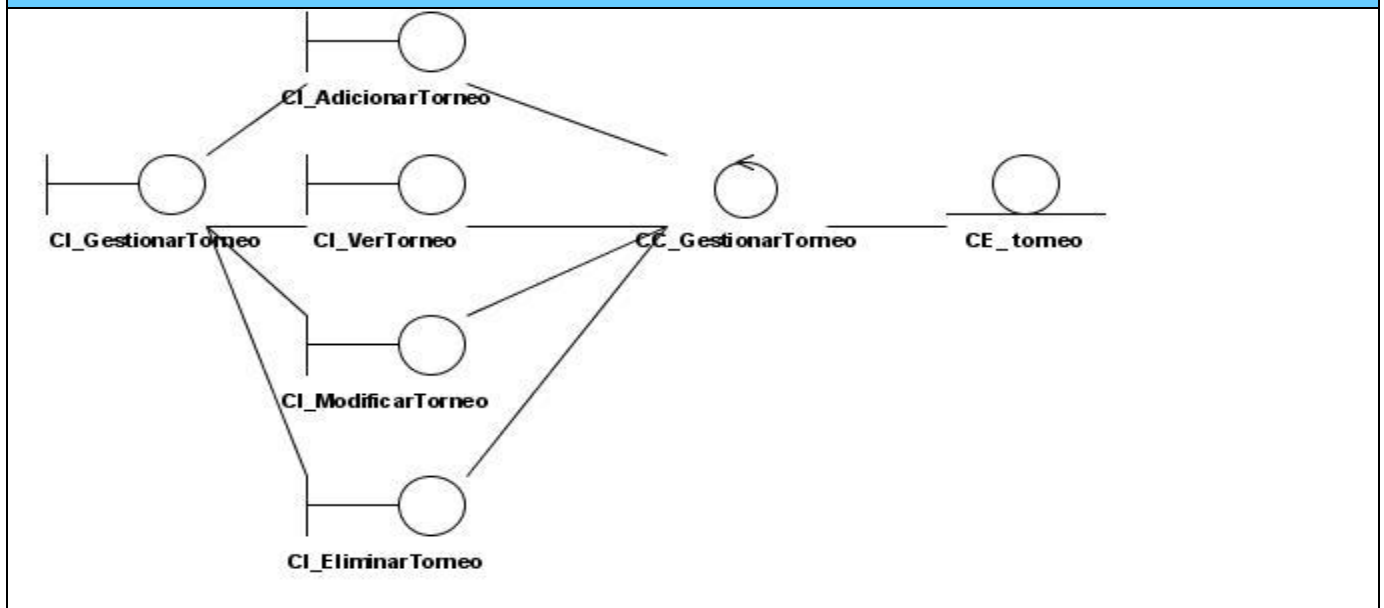




Figura 10: CU Realizar Sorteo

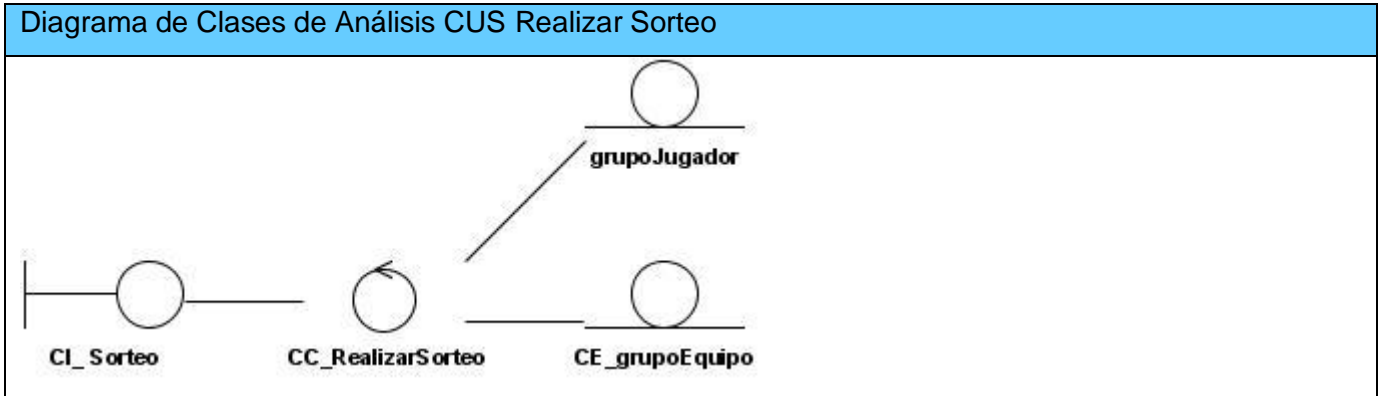


Figura 11: CU Realizar emparejamiento

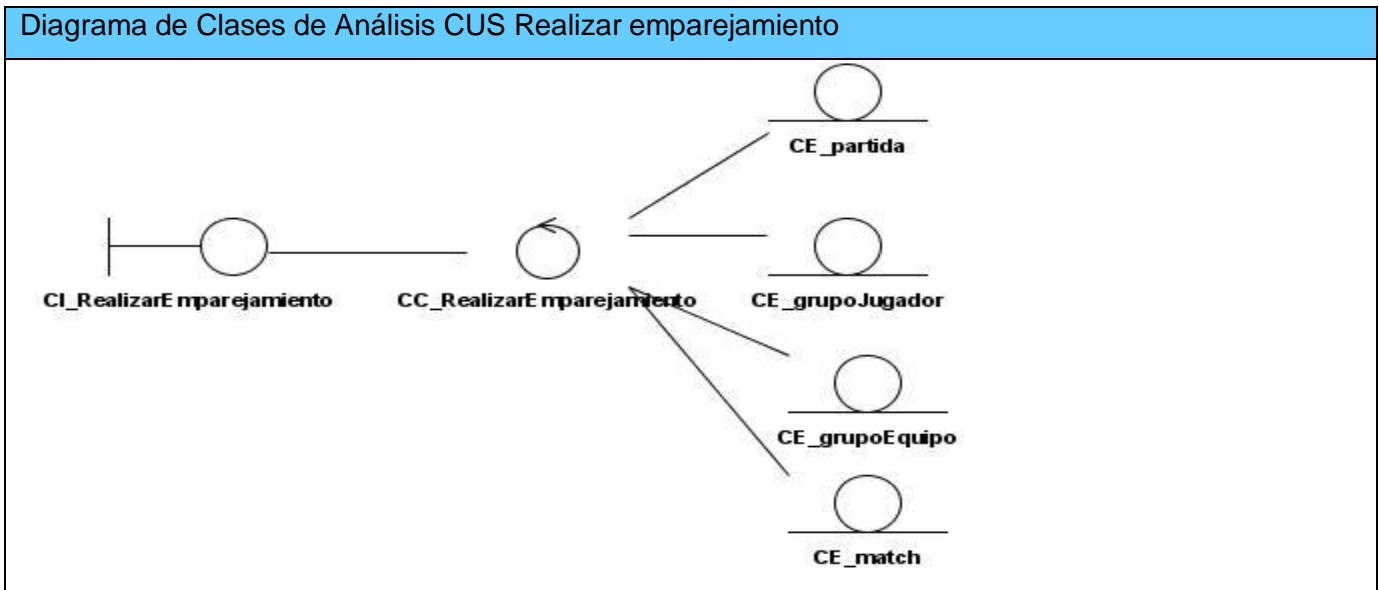
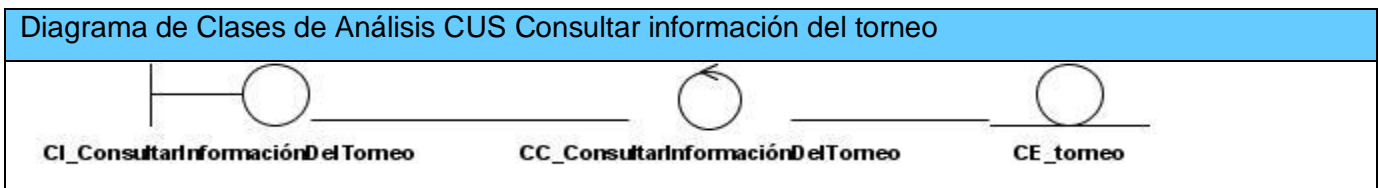


Figura 12: CU Consultar información del torneo





### 3.2.2.2 Realización de los casos de uso del análisis

La realización de los casos de uso del análisis describe como se lleva a cabo y se ejecuta un caso de uso en término de las clases del análisis y de sus objetos en interacción. Se realiza a través de dos tipos de diagramas: diagrama de secuencia y de colaboración.

En la propuesta de sistema integrado de los módulos del proyecto Infodrez se utiliza para describir los casos de uso del análisis los diagramas de colaboración. Estos muestran una interacción organizada en torno a los objetos que efectúan operaciones. Modela los objetos y los enlaces existentes entre ellos que se necesitan para implementar una operación de nivel más elevado.

A continuación se representan los diagramas de colaboración correspondientes a cada escenario de los casos de uso críticos del sistema. Para ver los restantes consultar [Anexo # 6](#)

**Figura 13: CU Gestionar Árbitro**

#### Diagrama de Colaboración del Análisis Sección “Adicionar Árbitro”

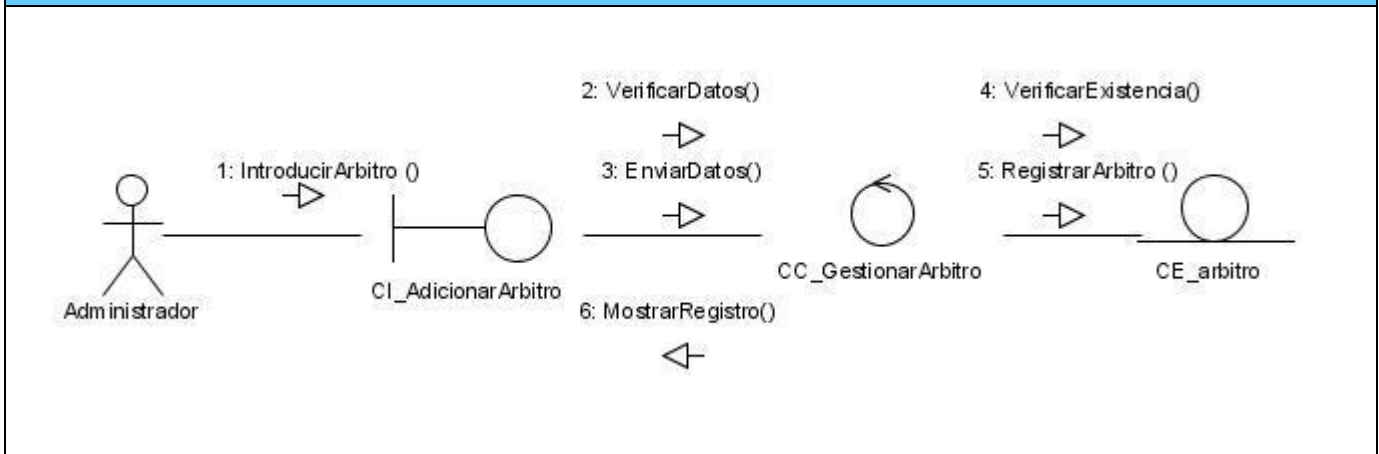




Diagrama de Colaboración del Análisis Sección “Ver Árbitro”

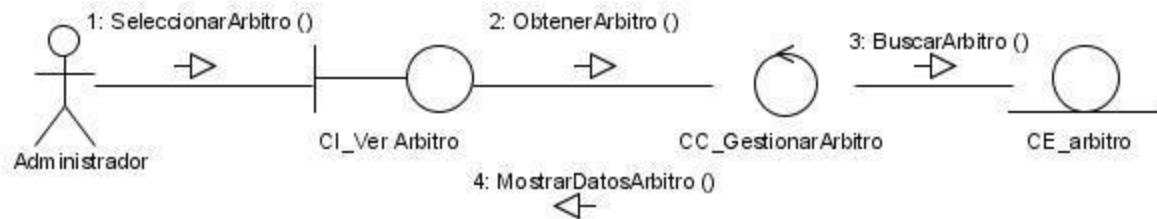


Diagrama de Colaboración del Análisis Sección “Modificar Árbitro”

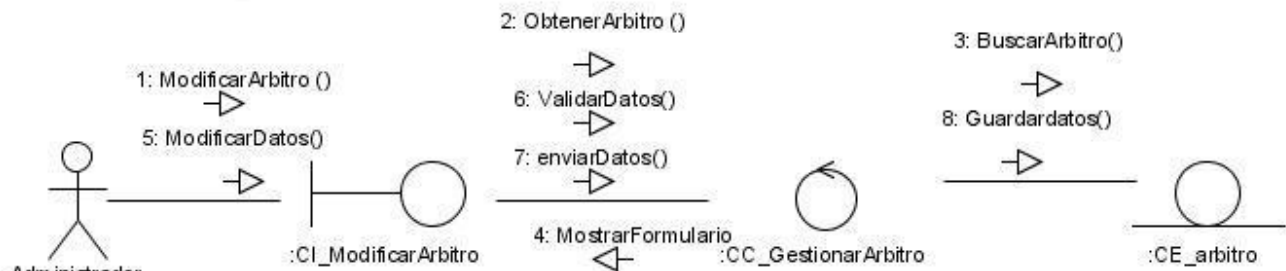


Diagrama de Colaboración del Análisis Sección “Eliminar Árbitro”

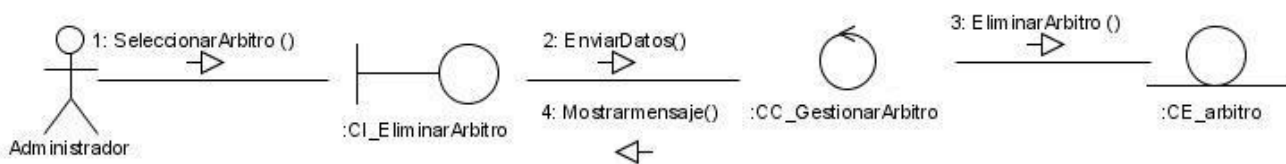






Figura 14: CU Gestionar Jugador

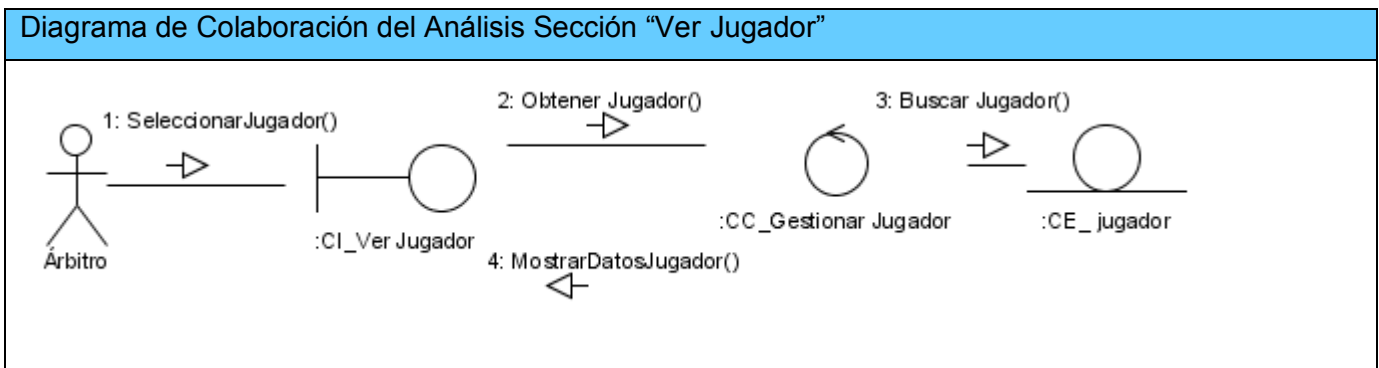
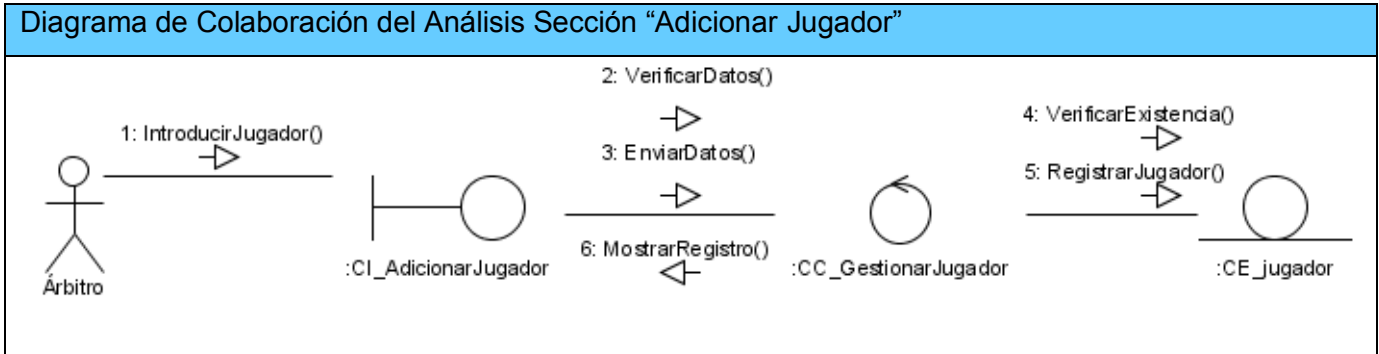




Diagrama de Colaboración del Análisis Sección "Modificar Jugador"

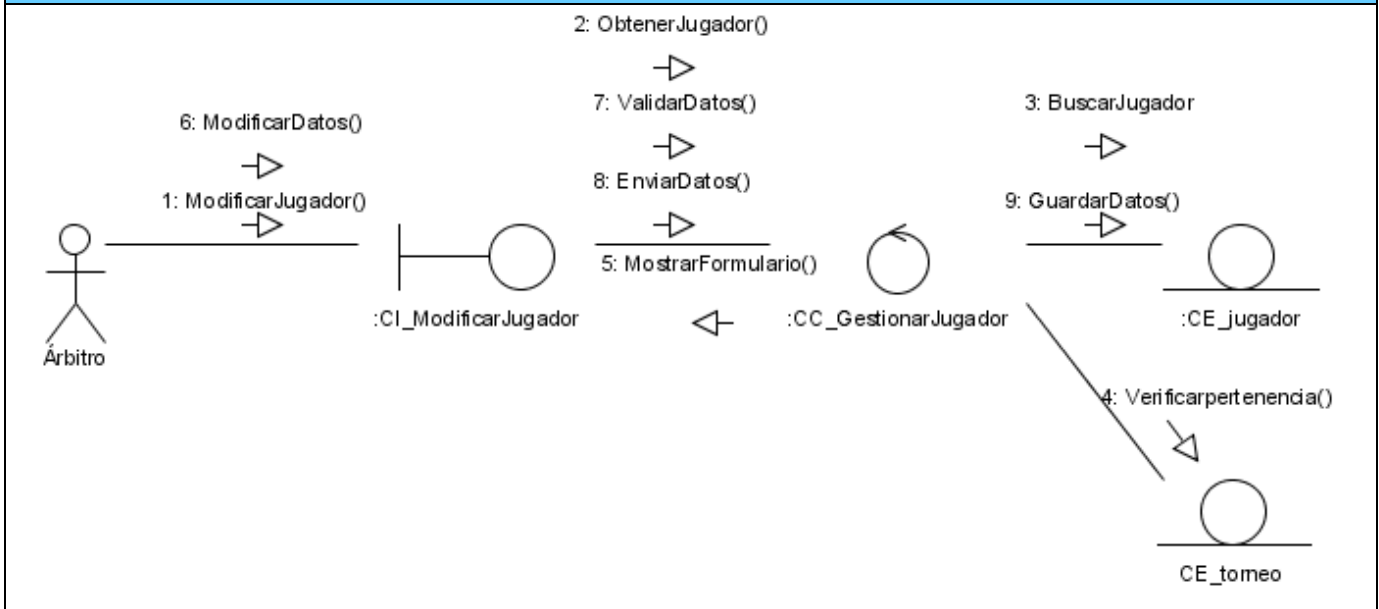


Diagrama de Colaboración del Análisis Sección "Eliminar Jugador"

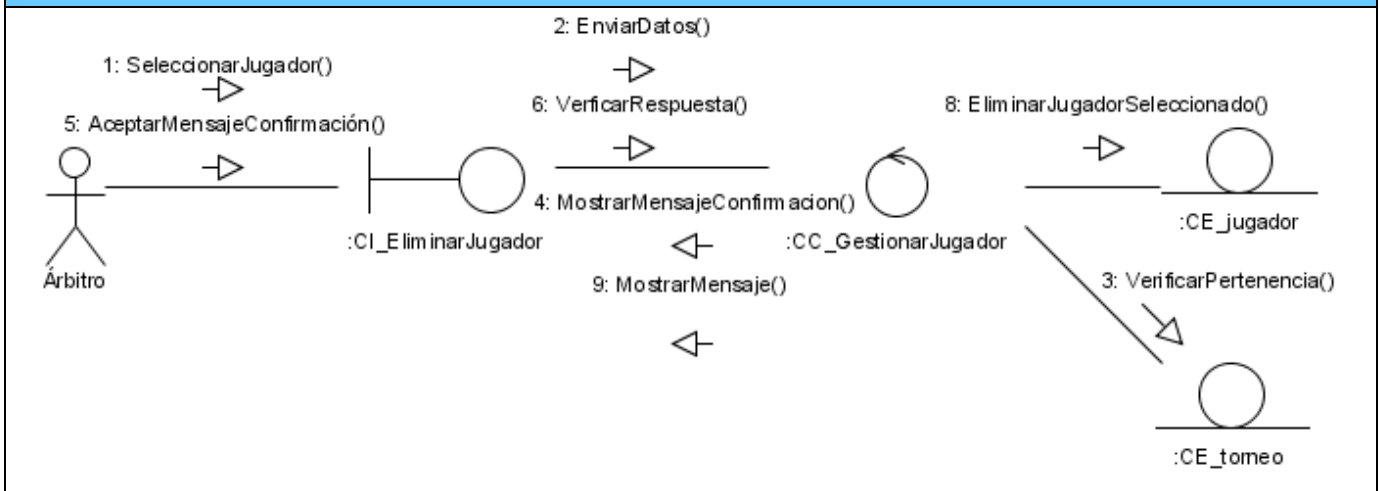




Figura 15: CU Gestionar Equipo

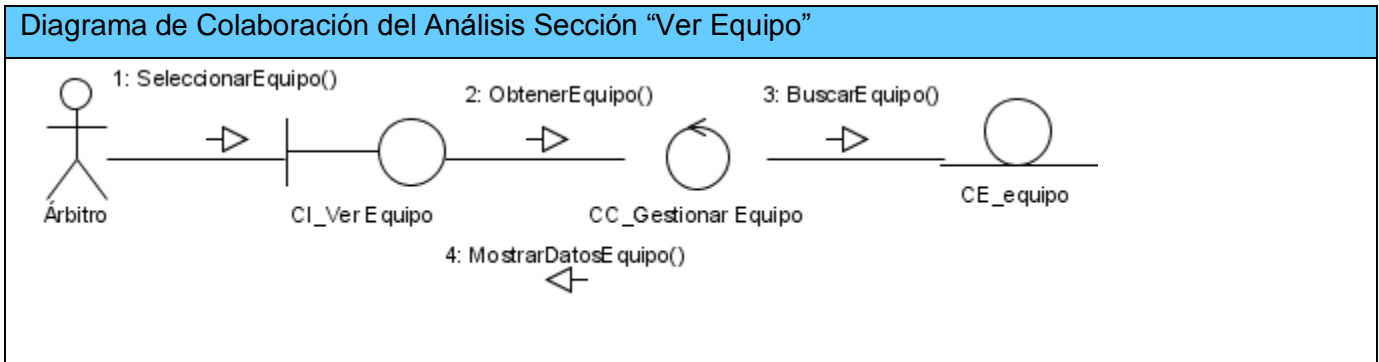
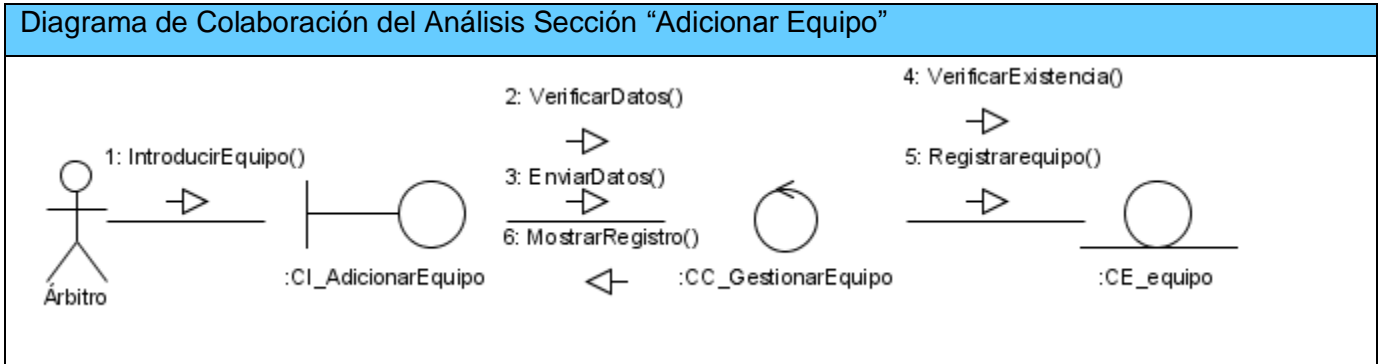




Diagrama de Colaboración del Análisis Sección “Modificar Equipo”

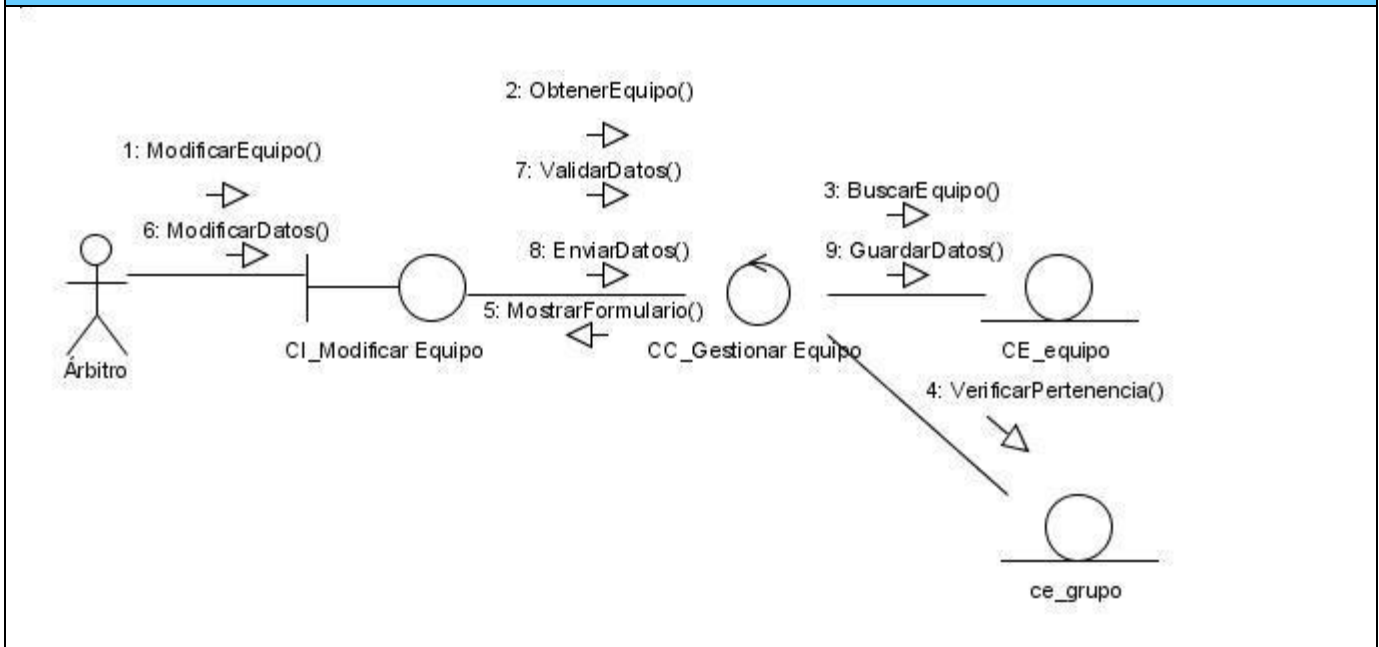


Diagrama de Colaboración del Análisis Sección “Eliminar Equipo”

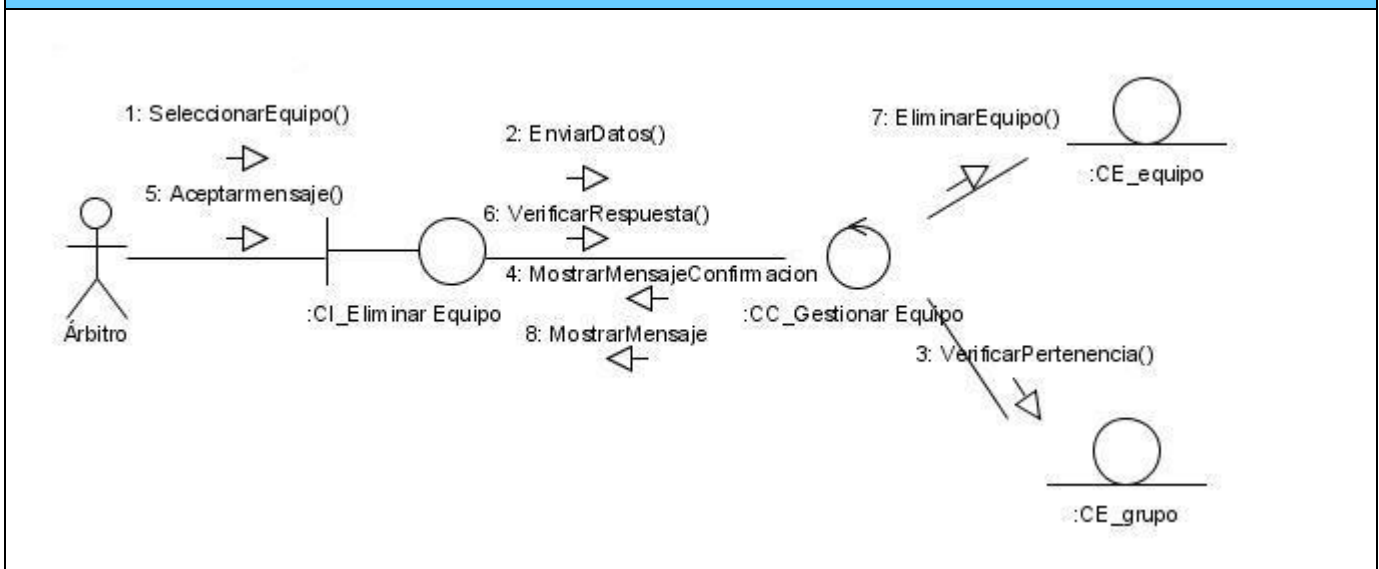




Figura 16: CU Gestionar Torneo

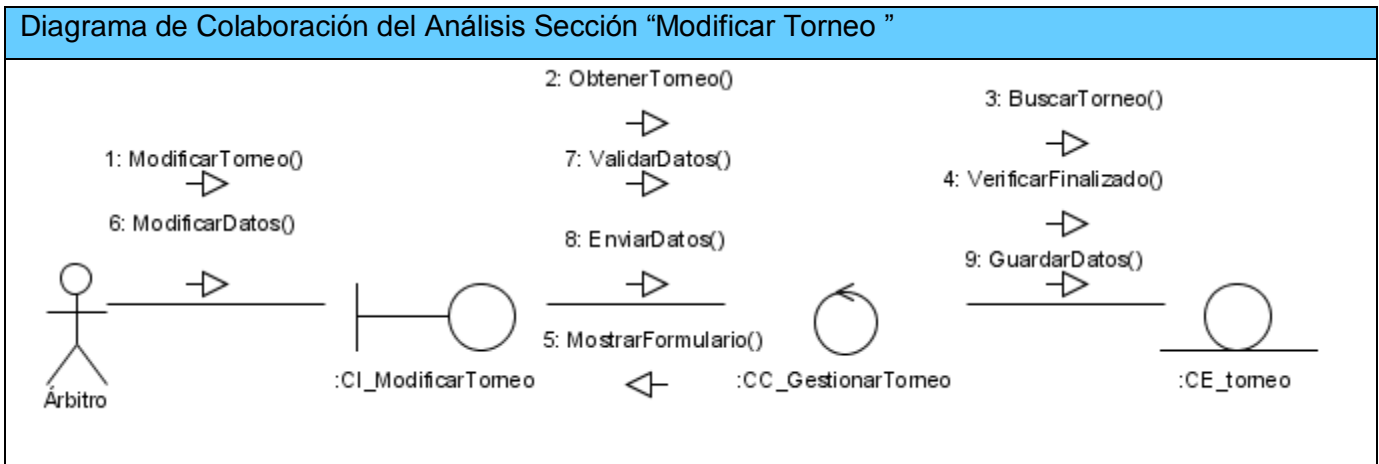
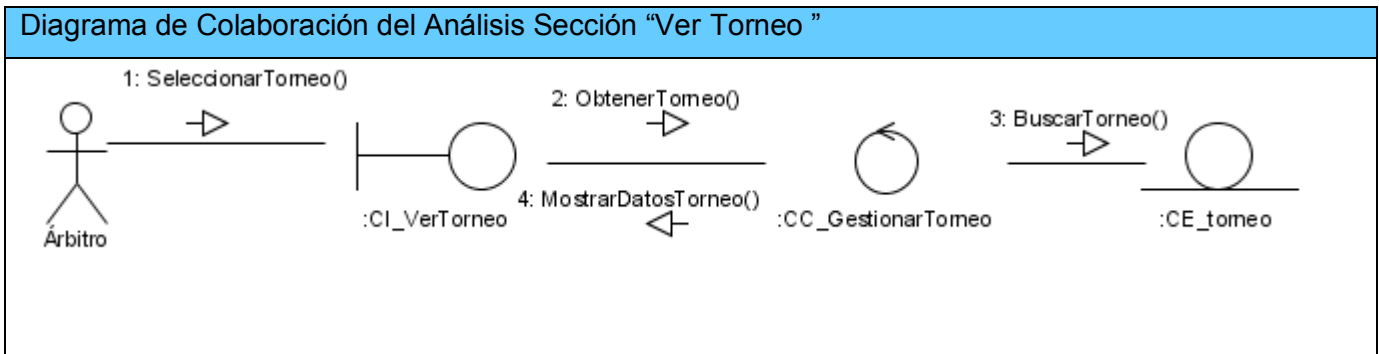
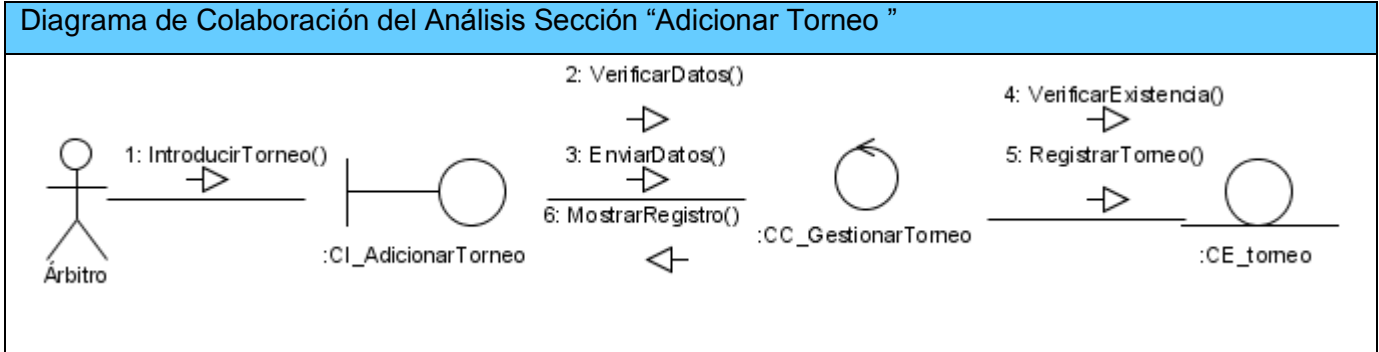




Diagrama de Colaboración del Análisis Sección “Eliminar Torneo”

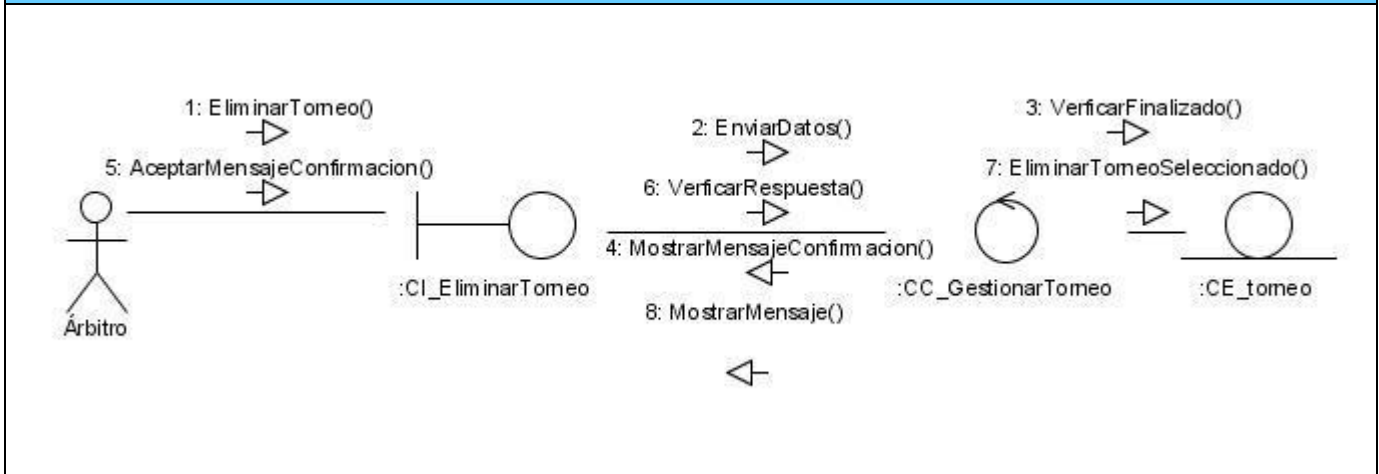


Figura 17: CU Realizar Sorteo

Diagrama de Colaboración del Análisis Sección: “Realizar Sorteo para un torneo RRI”

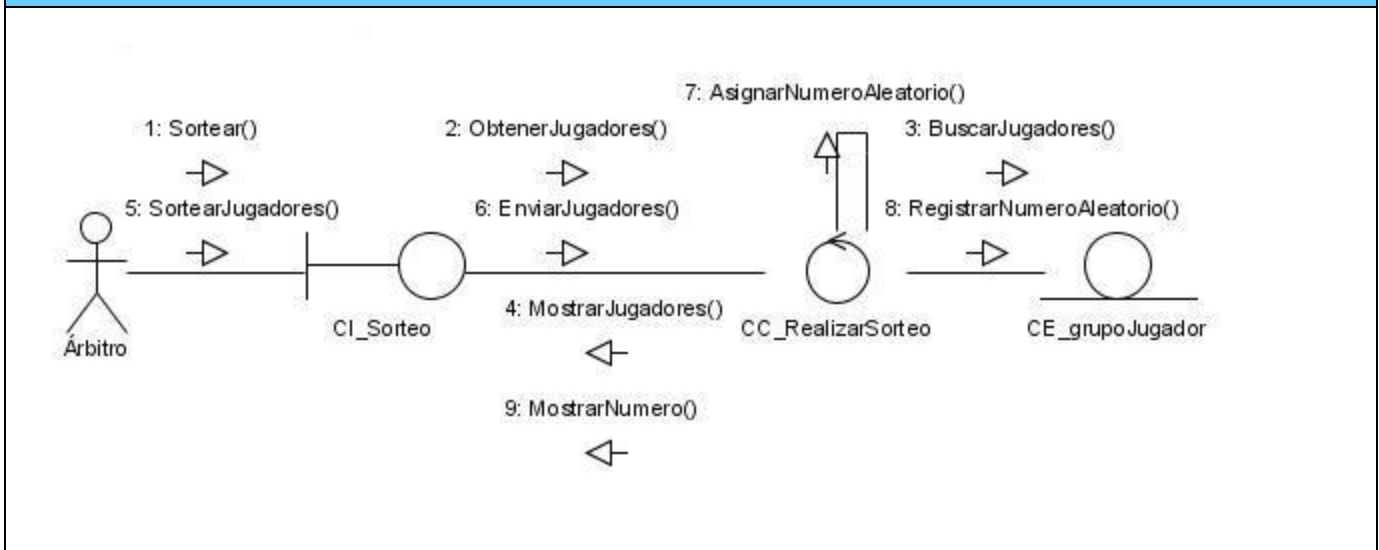




Diagrama de Colaboración del Análisis Sección: "Realizar Sorteo para un torneo RRE"

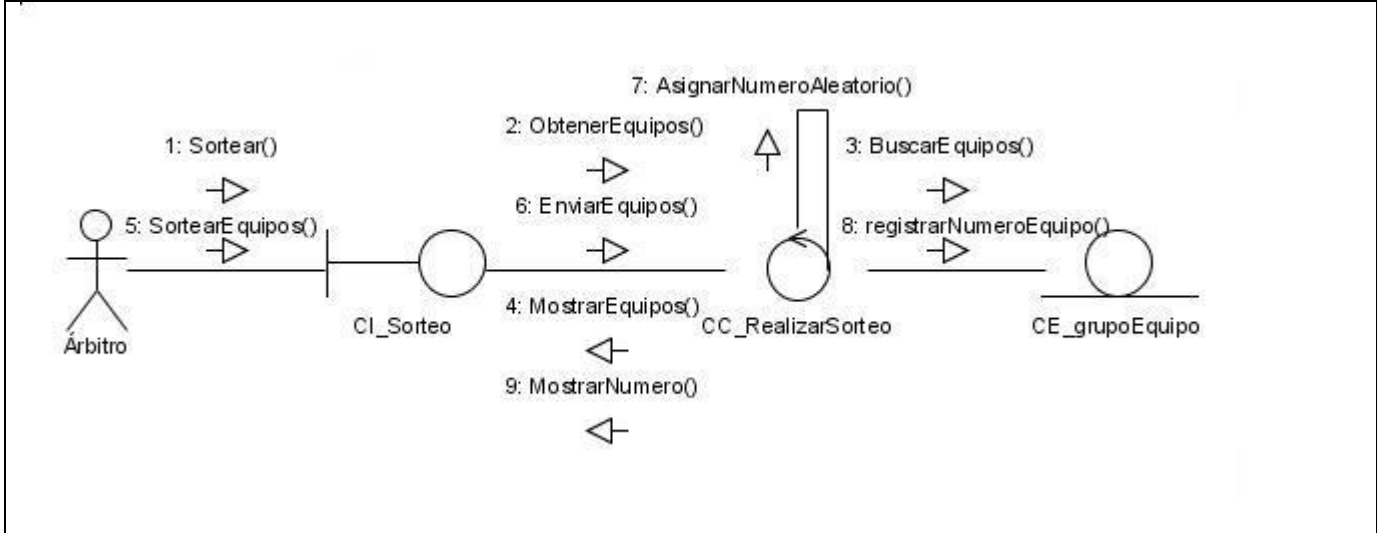


Figura 18: CU Realizar emparejamiento

Diagrama de Colaboración del Análisis Sección: "Realizar emparejamiento para un torneo RRI"

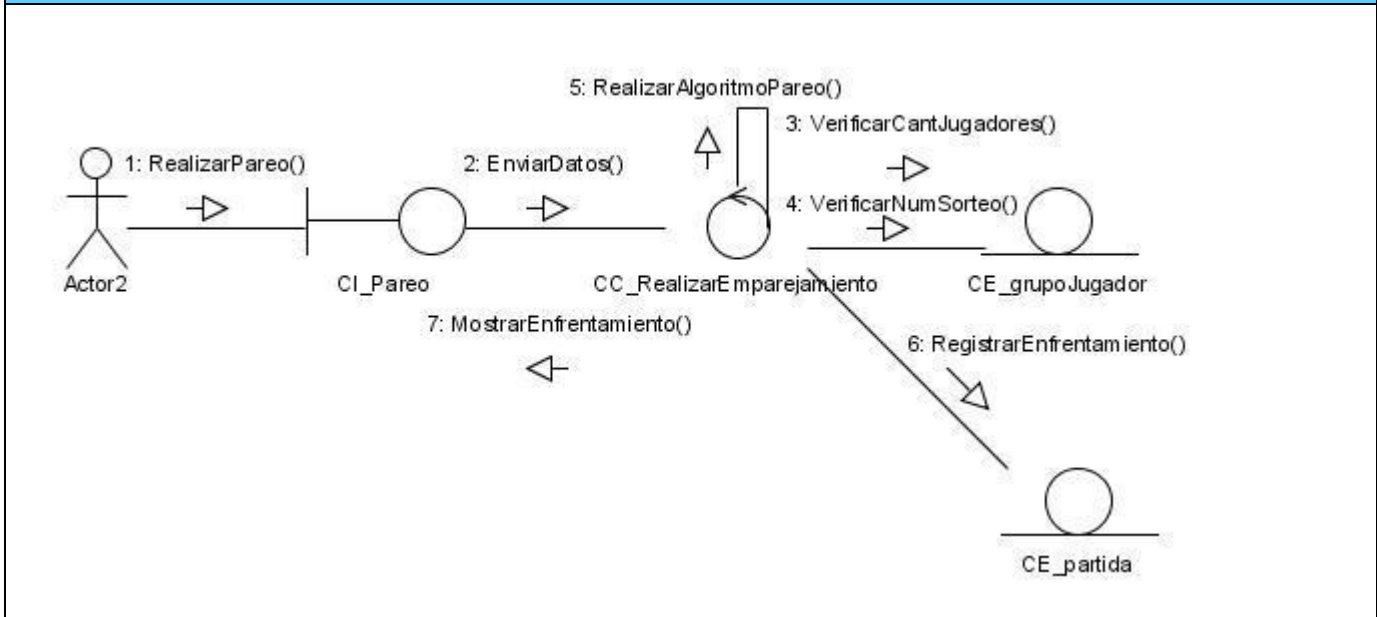




Diagrama de Colaboración del Análisis Sección: "Realizar emparejamiento para un torneo RRE"

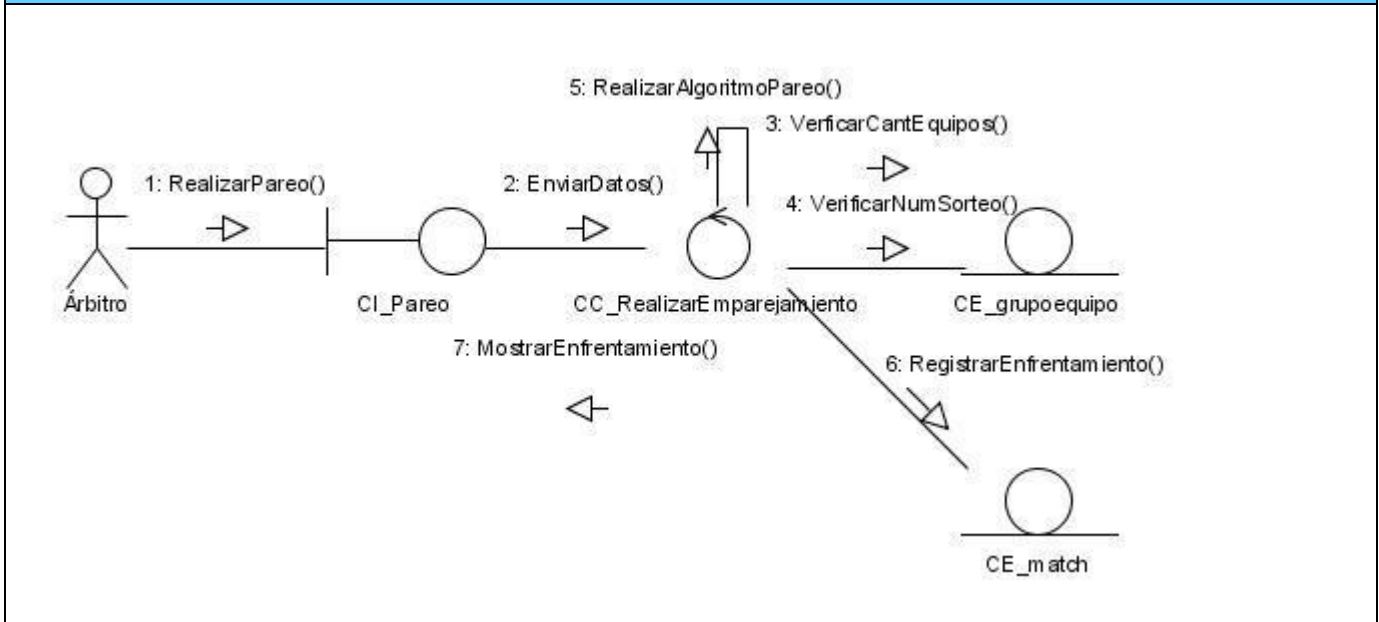
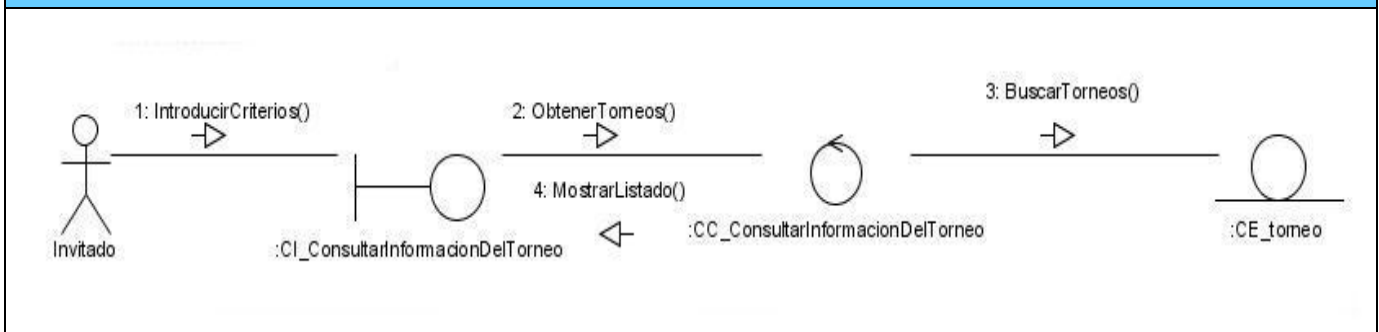


Figura 19: CU Consultar información del torneo

Diagrama de Colaboración del Análisis Consultar información del torneo



### 3.2.3 Diseño del Sistema

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva como cumple el sistema sus objetivos. Se tienen en cuenta las restricciones impuestas principalmente con respecto al lenguaje de programación. Debe ser suficiente para que el sistema pueda ser





implementado sin ambigüedades, constituyendo la entrada fundamental al flujo de Implementación. En el diseño las clases definidas en el análisis se detallan y se añaden nuevas clases para manejar áreas técnicas como base de datos, interfaz del usuario, comunicación, dispositivos, entre otras.

### 3.2.3.1 Clases del Diseño

Una clase de diseño es una construcción similar a la implementación del sistema, es especificada en el mismo lenguaje de programación a utilizar, y los métodos tienen correspondencia directa con el correspondiente método en la implementación de las clases.

Las clases del diseño varían en relación con la aplicación que se va a realizar. En la propuesta de sistema integrado de los módulos del proyecto Infodrez se realiza el diagrama de clases del diseño utilizando los estereotipos Web que propone UML: “Server Page”, “Client Page” y “Form”, empleados para el código servidor, código cliente y formularios respectivamente.

A continuación se representan los diagramas de clases del diseño correspondientes a los casos de uso críticos del sistema. Para consultar los restantes ver [Anexo #7](#)



Figura 20: Diagrama de Clases del Diseño del CU Gestionar Árbitro

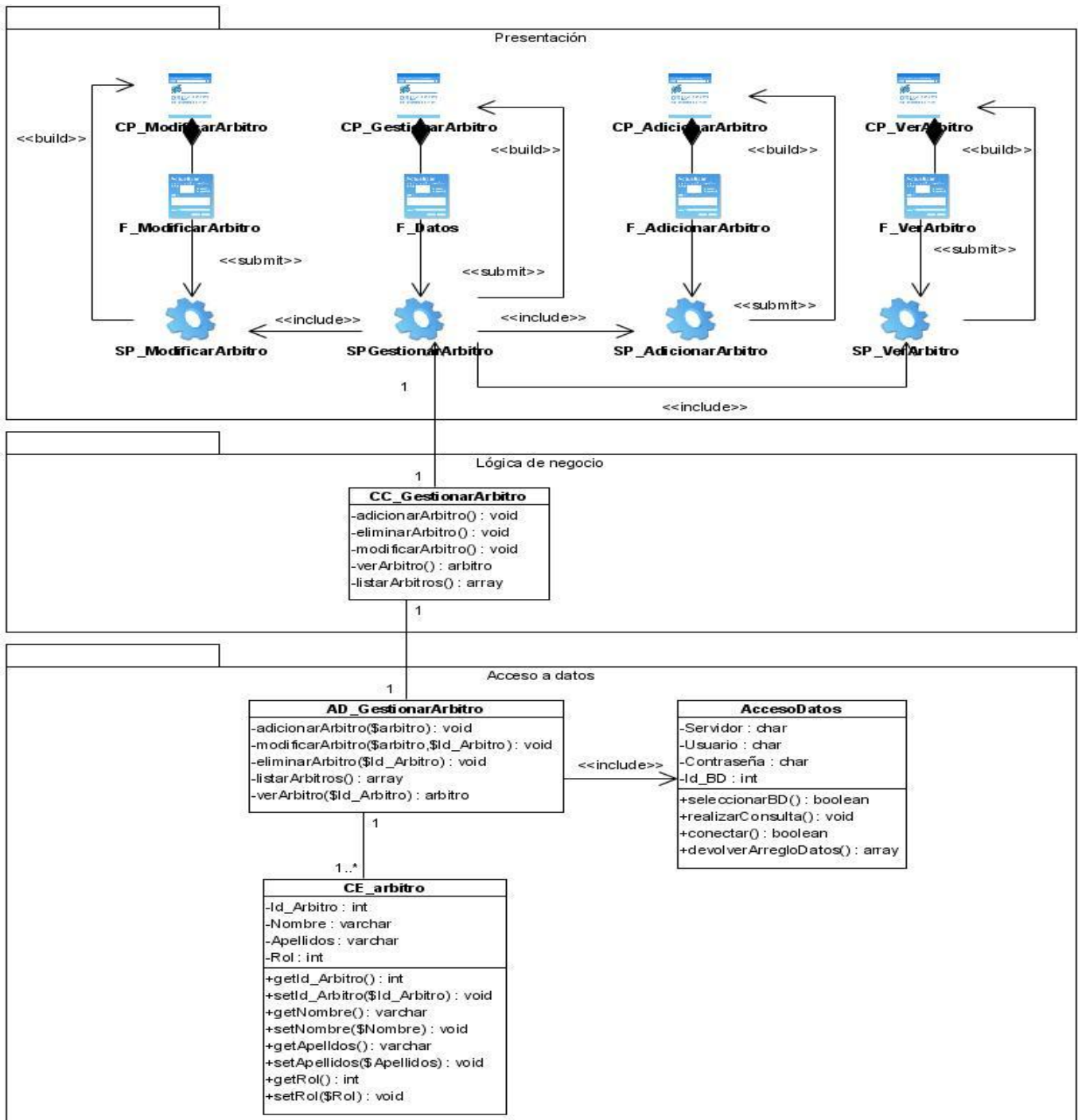




Figura 21: Diagrama de Clases del Diseño del CU Gestionar Jugador

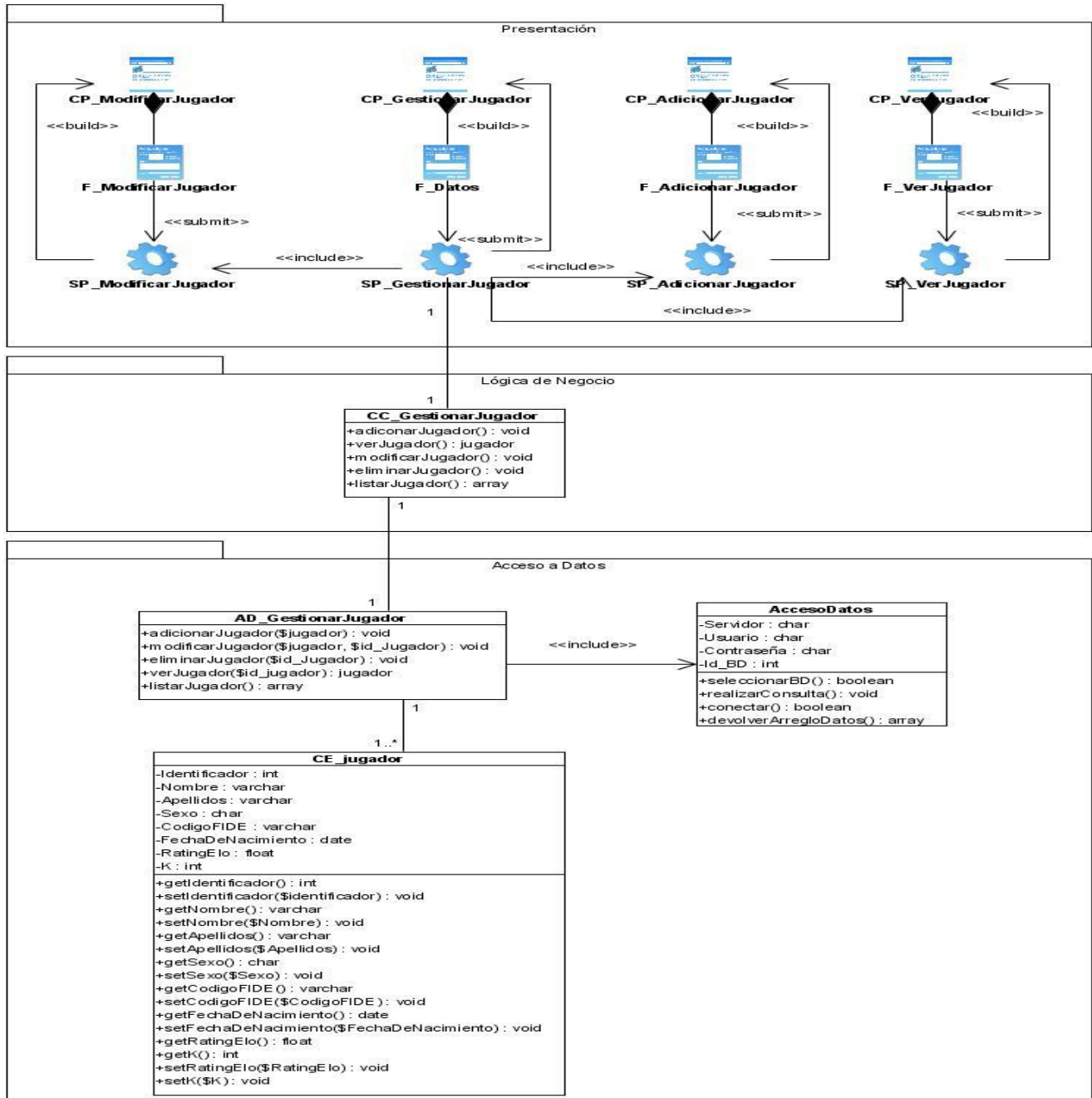




Figura 22: Diagrama de Clases del Diseño del CU Gestionar Equipo

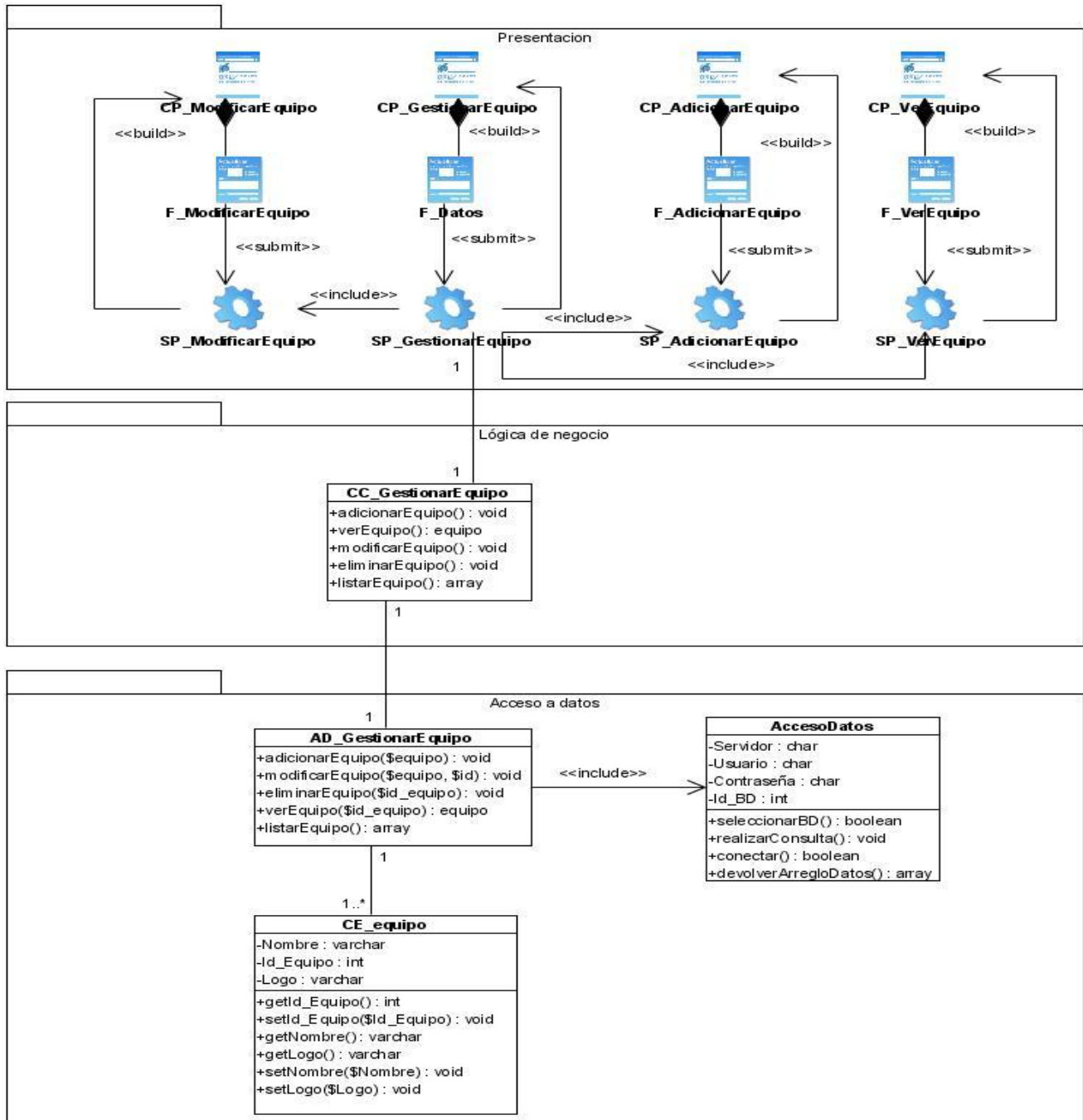




Figura 23: Diagrama de Clases del Diseño del CU Gestionar Torneo

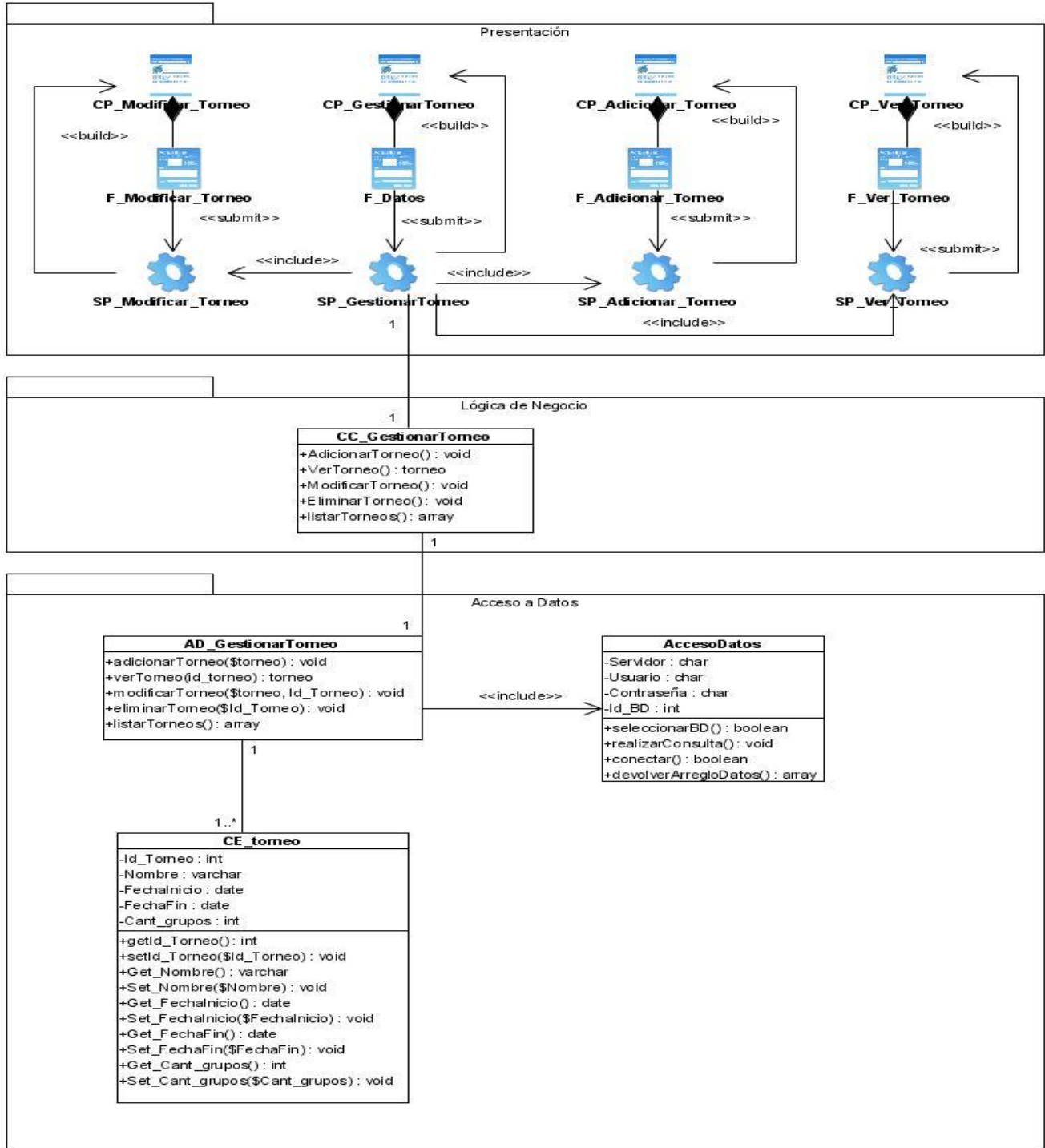




Figura 24: Diagrama de Clases del Diseño del CU Realizar Sorteo

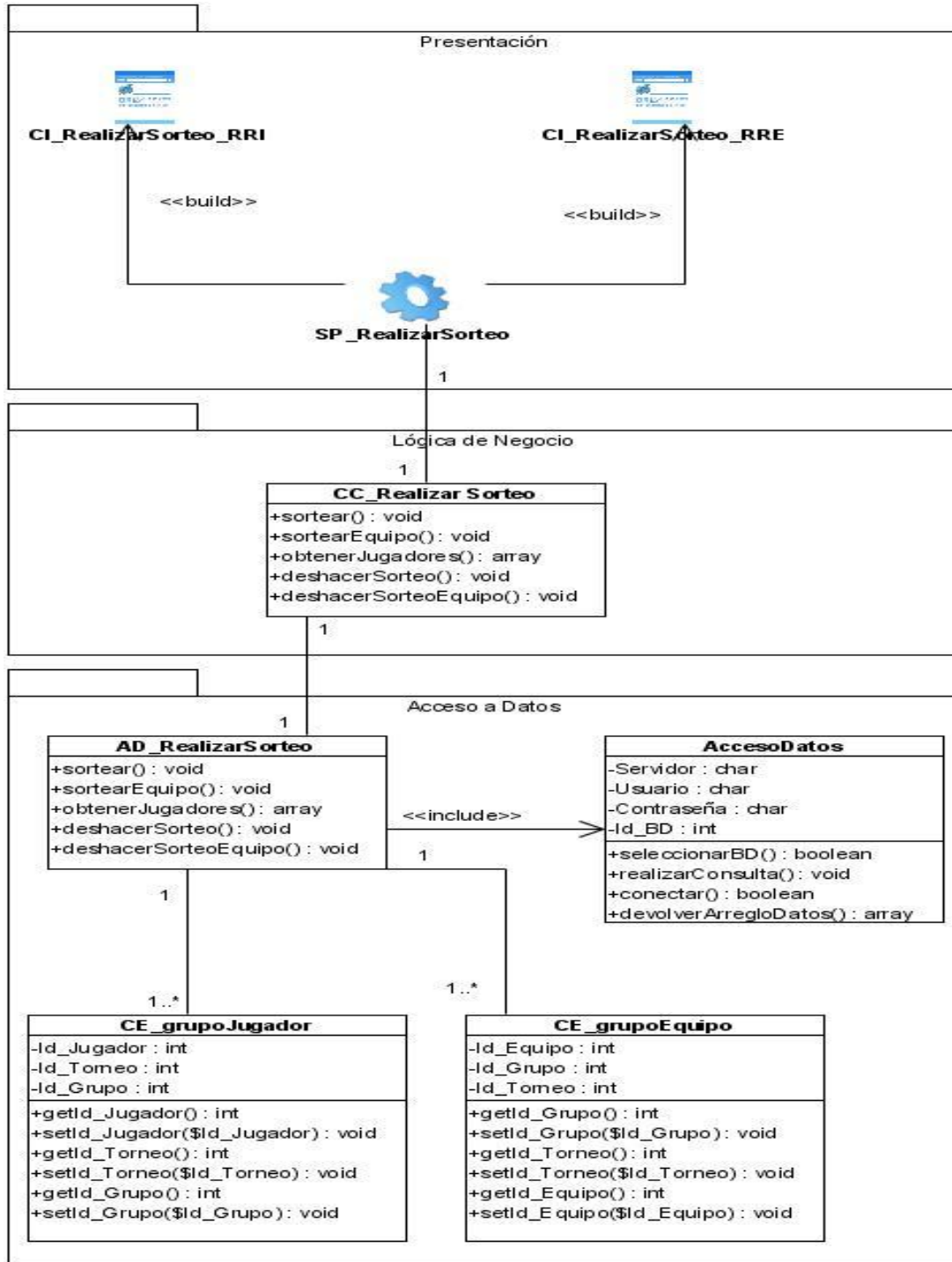




Figura 25: Diagrama de Clases del Diseño del CU Realizar emparejamiento

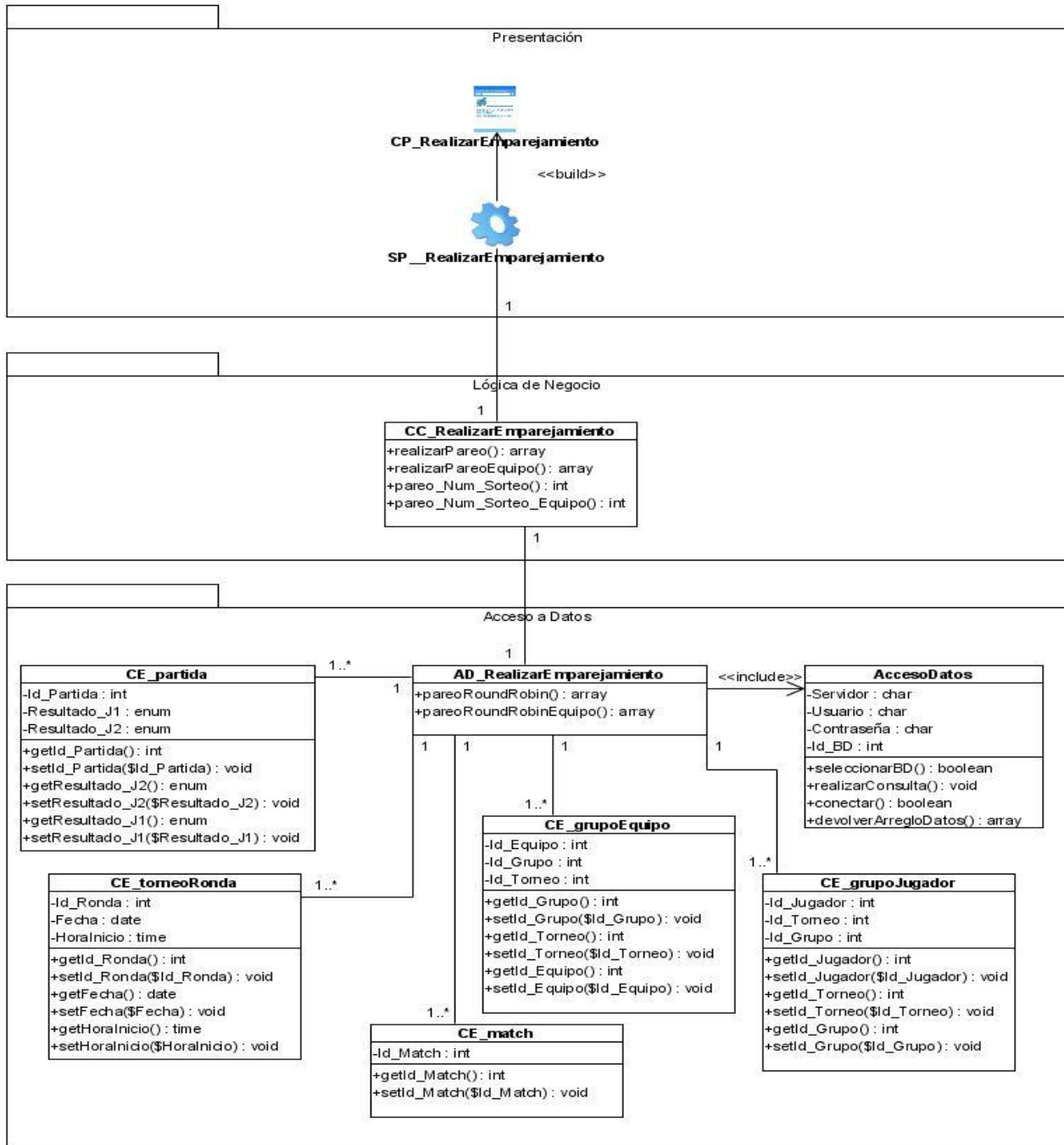
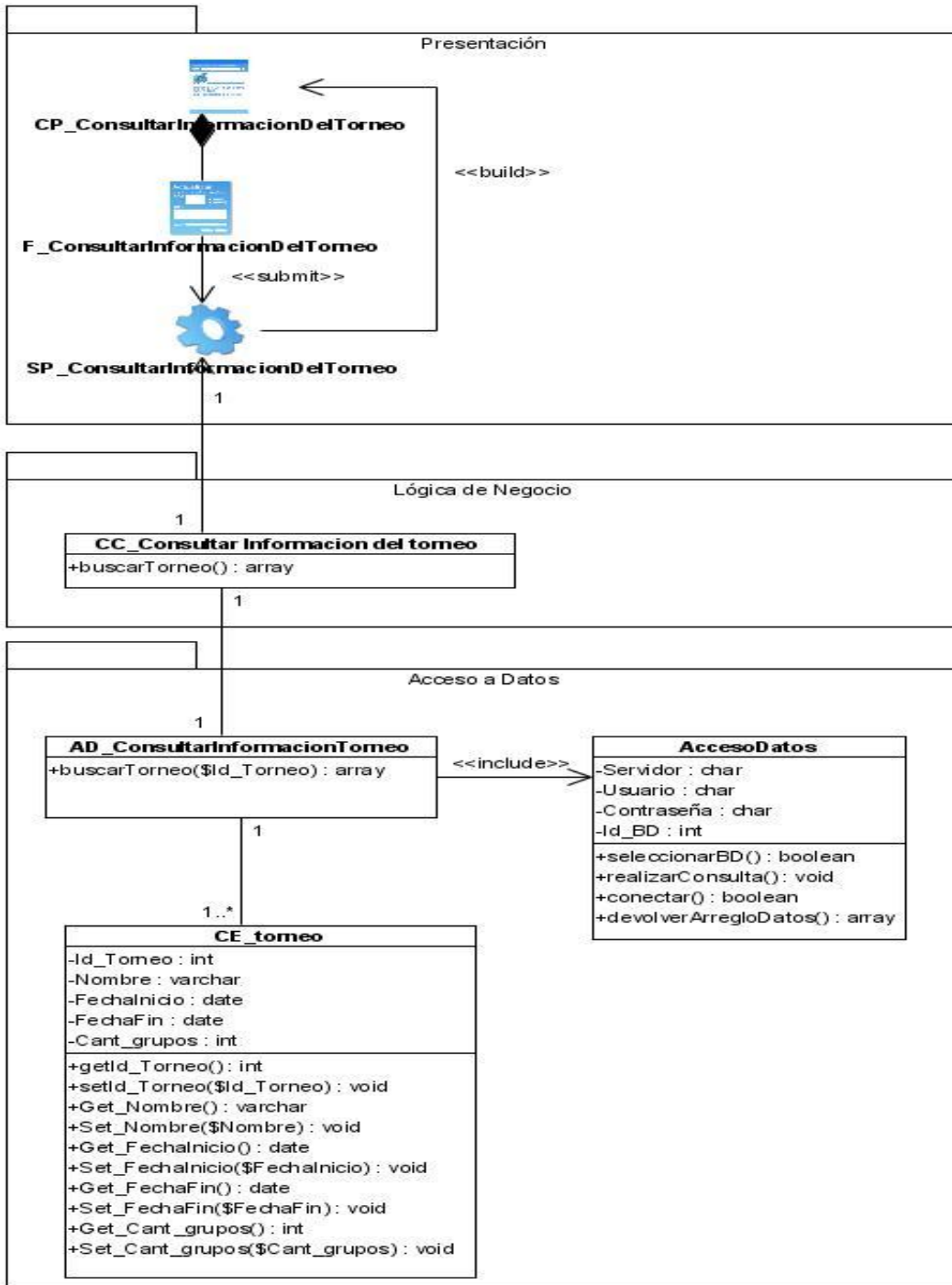




Figura 26: Diagrama de Clases del Diseño del CU Consultar Información del torneo







### Descripción Textual de las clases del diseño Web

La descripción textual de las clases que se modelan en el diagrama de clases del diseño permite entender las responsabilidades de las mismas dentro del sistema, ya que en la descripción se especifican claramente cuáles son sus funcionalidades además de la información que maneja cada una de ellas. Seguidamente se exponen las descripciones de las clases que responden a algunos de los casos de uso críticos, el resto se detalla en el [Anexo 8](#)

**Tabla 9: Descripción de la clase CE\_arbitro**

<b>Nombre</b>	CE_arbitro	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_Arbitro	int	
Nombre	varchar	
Apellidos	varchar	
Rol	int	
<b>Responsabilidad</b>		
	getId_Arbitro()	
	setId_Arbitro(\$Id_Arbitro)	
	getNombre()	
	setNombre(\$Nombre)	



	getRol()
	setRol(\$Rol)
	getApellidos()
	setApellidos(\$Apellidos)
<b>Descripción</b>	Clase entidad que representa la tabla Arbitro en la base de datos

**Tabla 10: Descripción de la clase CE\_jugador**

<b>Nombre</b>	CE_jugador	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_Jugador	int	
Nombre	varchar	
Apellidos	varchar	
Sexo	char	
CodigoFIDE	varchar	
FechaDeNacimiento	date	
RatingElo	float	
K	int	
<b>Responsabilidad</b>		



	getId_Jugador()
	setId_Jugador(\$Id_Jugador)
	getNombre()
	setNombre(\$Nombre)
	getApellidos()
	setApellidos(\$Apellidos)
	getSexo()
	setSexo(\$Sexo)
	getCodigoFIDE()
	setCodigoFIDE(\$CodigoFIDE)
	getFechaDeNacimiento()
	setFechaDeNacimiento(\$FechaDeNacimiento)
	getRatingElo()
	setRatingElo(\$RatingElo)
	getK()
	setK(\$K)
<b>Descripción</b>	Clase entidad que representa la tabla Jugador en la base de datos



Tabla 11: Descripción de la clase CE\_torneo

<b>Nombre</b>	CE_torneo	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
nombre	varchar	
Id_Torneo	int	
fechaInicio	date	
fechaFin	date	
<b>Responsabilidad</b>		
	getNombre()	
	setNombre(\$Nombre)	
	getId_Torneo()	
	setId_Torneo(\$Id_Torneo)	
	getFechaInicio()	
	setFechaInicio(FechaInicio)	
	getFechaFin()	
	setFechaFin(\$FechaFin)	
<b>Descripción</b>	Clase entidad que representa la tabla torneo en la base de datos	



Tabla 12: Descripción de la clase CE\_equipo

<b>Nombre</b>	CE_equipo
<b>Tipo de Clase</b>	Entidad
<b>Atributo</b>	<b>Tipo</b>
Id_Equipo	int
Nombre	varchar
Federacion	varchar
Logo	varchar
<b>Responsabilidad</b>	
	getId_Equipo()
	setId_Equipo(\$Id_Equipo)
	getNombre()
	setNombre(\$Nombre)
	getFederacion()
	setFederacion(\$Federacion)
	getLogo()
	setLogo(\$Logo)
<b>Descripción</b>	Clase entidad que representa la tabla Equipo en la base de datos



**Tabla 13: Descripción de la clase AD\_GestionarTorneo**

<b>Nombre</b>	AD_GestionarTorneo	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarTorneo(\$torneo)	
	modificarTorneo(\$torneo,\$Id_Torneo)	
	eliminarTorneos(\$Id_Torneo)	
	verTorneo(\$Id_Torneo)	
	listarTorneos()	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Torneo.	

**Tabla 14: Descripción de la clase AD\_GestionarJugador**

<b>Nombre</b>	AD_GestionarJugador	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	



Responsabilidad	
	adicionarJugador (\$jugador)
	modificarJugador(\$jugador, \$id_Jugador)
	eliminarJugador (\$id_Jugador)
	verJugador(\$id_jugador)
	listarJugadores()
Descripción	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Jugador.

Tabla 15: Descripción de la clase AD\_GestionarEquipo

Nombre	AD_GestionarEquipo	
Tipo de Clase	AD	
Atributo	Tipo	
Responsabilidad		
	adicionarEquipo (\$equipo)	
	modificarEquipo (\$equipo,\$Id_Equipo)	
	eliminarEquipo (\$Id_Equipo)	
	listarEquipos()	



	verEquipo (\$Id_Equipo)
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Equipo.

Tabla 16: Descripción de la clase AD\_GestionarArbitro

<b>Nombre</b>	AD_GestionarArbitro	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarArbitro(\$arbitro)	
	modificarArbitro(\$arbitro,\$Id_Arbitro)	
	eliminarArbitro(\$Id_Arbitro)	
	listarArbitros()	
	verArbitro(\$Id_Arbitro)	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Arbitro.	





Tabla 17: Descripción de la clase CC\_GestionarTorneo

<b>Nombre</b>	CC_GestionarTorneo	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarTorneo ()	
	modificarTorneo ()	
	eliminarTorneo ()	
	listarTorneos ()	
	verTorneo()	
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a los torneos	

Tabla 18: Descripción de la clase CC\_GestionarJugador

<b>Nombre</b>	CC_GestionarJugador	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		



	adicionarJugador ()
	verJugador()
	modificarJugador()
	eliminarJugador ()
	listarJugadores()
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a los jugadores

**Tabla 19: Descripción de la clase CC\_GestionarEquipo**

<b>Nombre</b>	CC_GestionarEquipo	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarEquipo ()	
	modificarEquipo ()	
	eliminarEquipo ()	
	listarEquipo ()	
	verEquipo()	
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a los equipos	



**Tabla 20: Descripción de la clase CC\_GestionarArbitro**

<b>Nombre</b>	CC_GestionarArbitro	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarArbitro ()	
	modificarArbitro()	
	verArbitro()	
	listar Arbitros()	
	eliminarArbitro()	
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a los árbitros.	

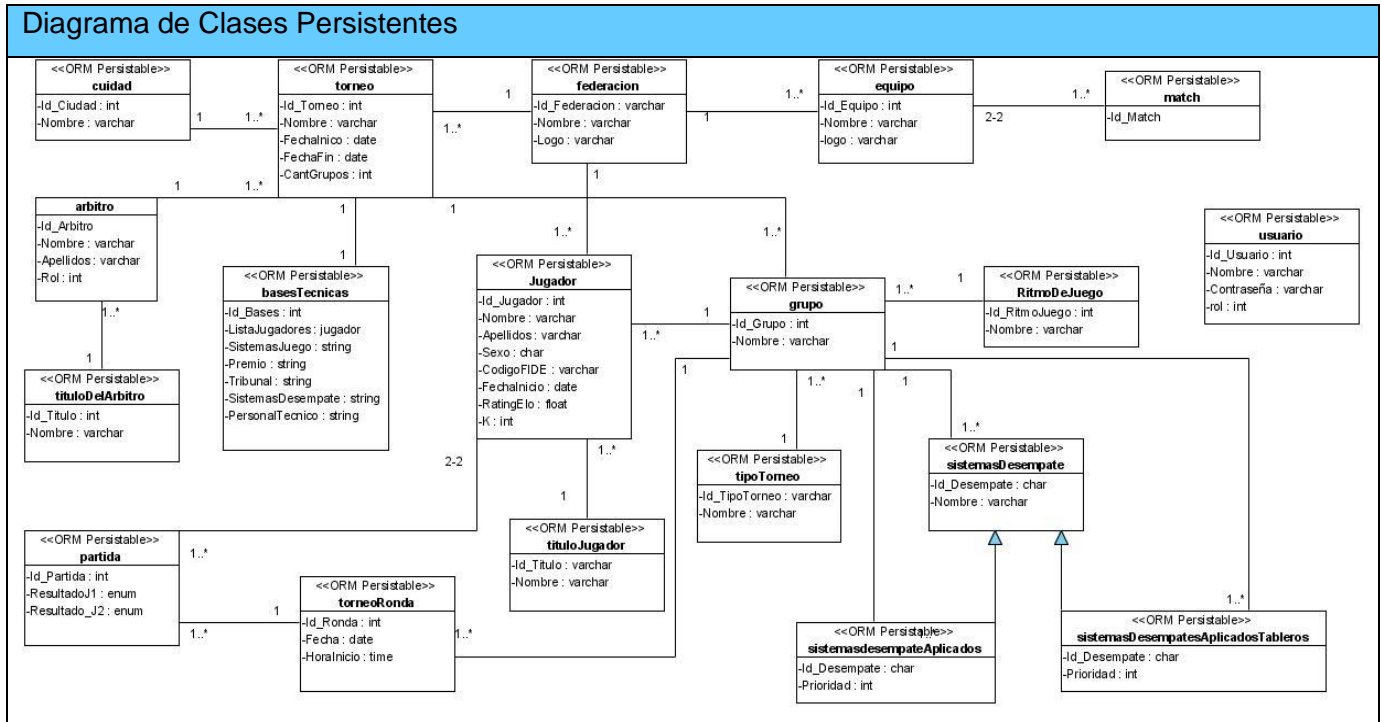
### Diagrama de clases persistentes

Las clases persistentes son capaces de guardar su estado en un medio permanente, debido a que sus objetos pueden mantener su valor en el espacio y el tiempo. Deben ser almacenados en algún repositorio como una base de datos.

A continuación se muestra el diagrama de clases persistentes que representa la interacción que existe entre las mismas.



Figura 27: Diagrama de Clases Persistentes



### 3.3 Conclusiones

En el presente capítulo fueron expuestos los distintos elementos que muestran la manera en que se encuentra construido el sistema, en términos de clases del análisis, del diseño y los diagramas de interacción, dando la posibilidad de tener una mejor comprensión de la lógica del mismo. Se utilizan los patrones GRASP en el diseño para la asignación de responsabilidades a las clases, logrando de ellas una mayor organización y entendimiento. Se realiza el modelo de clases persistentes de la base de que contiene la información física que se utilizó para la construcción de la aplicación



### Conclusiones Generales

Como resultado de este trabajo se obtuvo la propuesta de sistema integrado de los módulos (arbitraje, visor de torneos y estadística) del proyecto Infodrez en su primera versión. Se realizó un modelo de dominio para describir los objetos más importantes en el entorno del software, además de describirse las reglas del negocio presentes. Se capturan los requisitos del sistema y se validan cada uno de ellos a través de diversas técnicas. Se realizó el diseño del sistema dando cumplimiento con esto al objetivo principal del trabajo. Se cumplen cada una de las tareas definidas al comenzar el desarrollo de este trabajo, arribando a las siguientes conclusiones:

La metodología de desarrollo utilizada unida al lenguaje de modelado UML y a la herramienta Visual Paradigm ayudó a guiar el proceso de diseño de la propuesta de sistema de los módulos del proyecto Infodrez, además de permitir que el trabajo se desarrollara con mayor eficiencia.

Las técnicas de captura de requisitos brindaron la posibilidad de definir de forma clara las funcionalidades del sistema que sería diseñado.

Los prototipos facilitaron el entendimiento con el cliente y constituyeron una manera de validar las funcionalidades del sistema.

La realización del análisis ayudó a entender el sistema antes de ser diseñado.

El diseño realizado permitirá que los desarrolladores de la propuesta de sistema se sientan guiados hacia una implementación sin ambigüedades.

Durante el desarrollo del trabajo todos los artefactos generados permitieron un mejor entendimiento entre el cliente y el equipo de trabajo, dando cumplimiento así al objetivo propuesto.



### **Recomendaciones:**

Para una próxima versión del trabajo se recomienda tener en cuenta las siguientes funcionalidades:

- ❖ Adicionar una funcionalidad que permita al administrador ir registrando los movimientos de una partida y brindar la opción de mostrarla al usuario a medida que se va realizando el registro.
- ❖ Mostrar en un grafico el comportamiento de cada jugador durante un torneo.
- ❖ Ampliar los campos de búsqueda y la información resultante de la misma tanto para los casos de uso Consultar jugador y Consultar Partida
- ❖ Integrar la ingeniería del juego online a la propuesta de sistema



### Referencia Bibliográfica

[1] Department of Health Human Services.USA, CENTERS for MEDICARE & MEDICAID SERVICES. [Online] Marzo 27, 2008. [Cited: ENERO 30, 2009.]

Disponible en :[http://es.wikipedia.org/wiki/Metodolog%C3%ADa\\_de\\_desarrollo\\_de\\_software](http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software).

[2] Beck, Kent. *“Extreme Programming Explained. Embrace Change”*. 1999.

[3] Jeffries, R., Anderson, A., Hendrickson, C. *“Extreme Programming Installed”*. 2001.

[4] Schwaber, Ken. *Agile Project Management with Scrum*. s.l. : Microsoft Press, 2004. ISBN/ 0-7356-1993-X.

[5] G, Booch. *Object-Oriented Analysis and Design with Applications, 2nd edition*. s.l. : Addison-Wesley Object Technology Series, February 1994. ISBN/0805353402. *Análisis y diseño orientado a objetos con aplicaciones, segunda edición*. s.l. : Addison-Wesley / Díaz de Santos, 1996. ISBN/0-201-60122-2.

[6] Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorenson W. *Object-Oriented Modelling and Design*. s.l. : Prentice Hall, enero de 1991. ISBN/0136298419..

[7] Jacobson I., Booch G ., Rumbaugh J. *The Unified Software Development Process*. s.l. : Addison-Wesley, enero de 1999. ISBN/ 0201571692.

Disponible en : [http://kuainasi.ciens.ucv.ve/ideas07/documentos/articulos\\_ideas/Articulo08.pdf](http://kuainasi.ciens.ucv.ve/ideas07/documentos/articulos_ideas/Articulo08.pdf).

[8] José L. Pastrana, Ernesto Pimentel, Miguel Katrib. [www.kuainasi.ciens.ucv.ve](http://www.kuainasi.ciens.ucv.ve). [www.kuainasi.ciens.ucv.ve](http://www.kuainasi.ciens.ucv.ve). [Online] [Cited: febrero 2, 2009.]



### Bibliografía

1- [http://es.wikipedia.org/wiki/Programacion\\_Extrema](http://es.wikipedia.org/wiki/Programacion_Extrema)

Fecha de consulta: 30/01/09 Hora 3:35 PM

2 - <http://teleformacion.uci.cu/mod/resource/view.php?id=11402>

Fecha de consulta 30/01/2009 Hora 4:00 PM

3-[http://es.wikipedia.org/wiki/Scrum#cite\\_note-schwaber-3](http://es.wikipedia.org/wiki/Scrum#cite_note-schwaber-3)

Fecha de consulta: 1/02/2009 Hora: 9:00 PM

4- <http://www.willydev.net/descargas/masyxp.pdf>

Fecha de consulta: 2/02/2009 Hora 9:54 PM

5 -<http://www.lcc.uma.es/~av/Publicaciones/04/UMLProfiles-Novatica04.pdf>

Fecha de consulta: 2/02/2009 Hora: 9:30 PM

6-<http://es.wikipedia.org/wiki/CASE>

Fecha de consulta: 2/02/2009 Hora 9:45 PM

7-<http://www.tdx.cat/TDX-0611104-154319>

Fecha de consulta: 2/02/2009 Hora 10:05 PM

8- Larman, C; "UML y patrones" Tomo I Capítulo 4, Páginas 185-217.

9- <http://teleformacion.uci.cu/mod/resource/view.php?id=11553>

Fecha de consulta 20/02/2009 Hora 2:30 PM

10- Rumbaugh, J.; Jacobson, I. y Booch, G.; "El Lenguaje Unificado de Modelado".





11-JACOBSON, Ivar; RUMBAUGH, James; BOOCH, Grady, "El proceso unificado de desarrollo".2000.

12- <http://teleformacion.uci.cu/mod/resource/view.php?id=13151>

Fecha de consulta 12/03/2009 Hora 11:20 AM

13- <http://es.wikipedia.org/wiki/DBMS>

Fecha de consulta: 18/03/2009 Hora: 5:00PM

14- <http://platea.pntic.mec.es/~abercian/guiahtml/comienzo.htm>

Fecha de consulta: 5/04/2009 Hora: 10:35 AM

15- <http://www-app.etsit.upm.es/~alvaro/manual/manual.html#1>

Fecha de consulta: 5/04/2009 Hora 11:00 AM

16- [http://es.wikipedia.org/wiki/C%C3%B3digo\\_HTML](http://es.wikipedia.org/wiki/C%C3%B3digo_HTML)

Fecha de consulta: 5/04/2009 Hora: 11:20 AM

17-<http://es.wikipedia.org/wiki/Javascript>

Fecha de consulta: 5/04/2009 Hora: 11:30 AM

18- Pérez Eguíluz, Javier, "Introducción a JavaScript".2009.

Disponible en <http://www.librosweb.es/javascript>

Fecha de consulta: 5/04/2009 Hora: 11:36 AM

19-[http://es.wikipedia.org/wiki/.php#PHP\\_4](http://es.wikipedia.org/wiki/.php#PHP_4)

Fecha de consulta: 6/04/2009 Hora: 3:00 PM

20-<http://www.cgsite.cl/php1.htm>



Fecha de consulta: 6/04/2009 Hora 3:15 AM

21-<http://es.wikipedia.org/wiki/PostgreSQL>

Fecha de consulta: 6/04/2009 Hora 3:26 AM

22-<http://es.wikipedia.org/wiki/MySQL>

Fecha de consulta: 6/04/2009 Hora 4:00PM

23-<http://www.mysqla.com.ar/temarios/descripcion.php?cod=2&punto=1>

Fecha de consulta: 6/04/2009 Hora: 4:15 PM

24-[http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html)

Fecha de consulta: 6/04/2009 Hora: 4:25 PM

25-[http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador)

Fecha de consulta: 20/04/2009 Hora: 10:40 AM



## Anexos

### Anexo 1: Aplicación de las técnicas de captura de requisitos

Técnica	Tormenta de Ideas		
Fecha:	10/02/2009	Hora:	9:00 PM
Lugar:	Cátedra de Ajedrez "Remberto Fernández"		
Participantes:	Líder del proyecto, analista principal, jefe de los módulos arbitraje, estadística y visor de torneos, analistas de los módulos mencionados y especialista funcional.		
Resumen:	La reunión comenzó a la hora señalada, donde los analistas de cada uno de los módulos exponen de forma libre los requisitos presentes en su módulo. Los jefes de módulo hacen sugerencias a nuevos requerimientos que deberían estar en la unificación y el líder del proyecto hace alusión a aspectos a tener en cuenta a la hora de diseñar la propuesta de sistema integrado.		

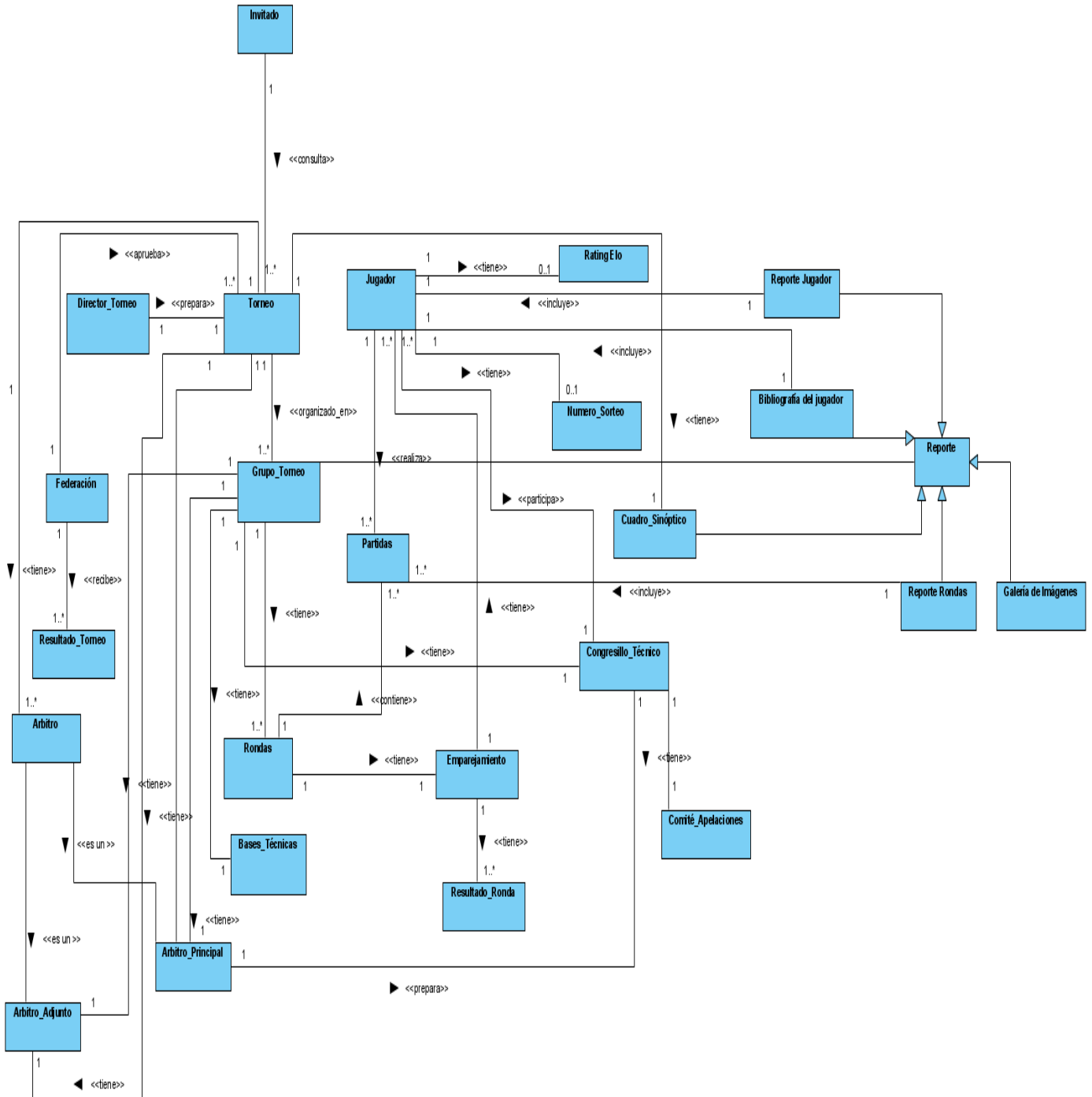
Técnica	Técnicas para facilitar las especificaciones de una aplicación (TFEA)		
Fecha:	15/02/2009	Hora:	10:00 AM
Lugar:	Aula del docente 5		
Participantes:	Líder del proyecto, analista principal, jefe de los módulos arbitraje, estadística y visor de torneos, analistas de los módulos mencionados y especialista		



	funcional.
Resumen:	La reunión comenzó con una exposición por parte de las autoras del presente trabajo de diploma, donde exponen su propuesta de requisitos y el Modelo de Casos de Uso del sistema. Cada una de las personas presentes expresa su opinión acerca de lo expuesto, realizando sugerencias para mejorar la propuesta. Se finaliza la actividad con la aprobación de la propuesta de requisitos y el Modelo de Casos de Uso del sistema de la plataforma Infodrez en su primera versión.



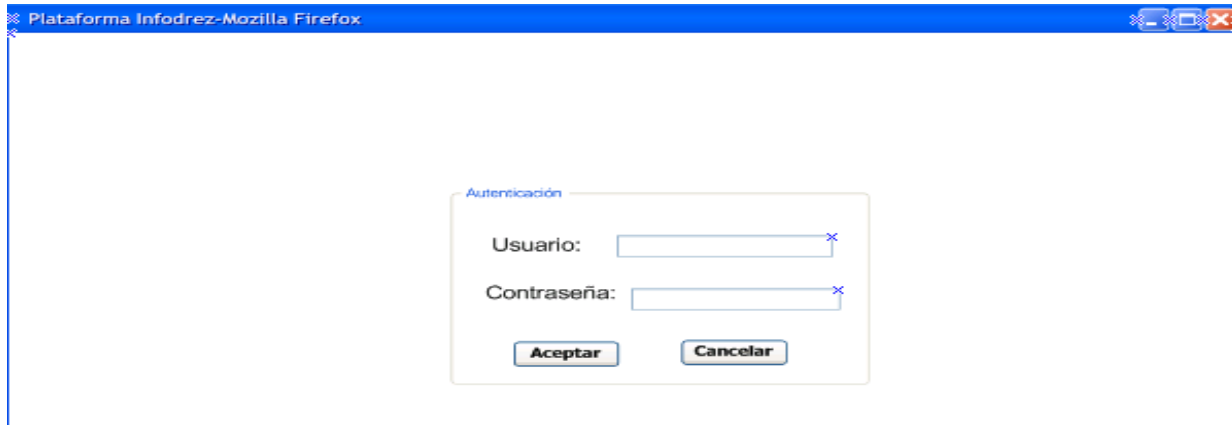
### Anexo 2: Modelo de dominio del sistema Infodrez





### Anexo 3: Validación de requisitos

Prototipo: Autenticar Usuario



Prototipo: Gestionar usuario



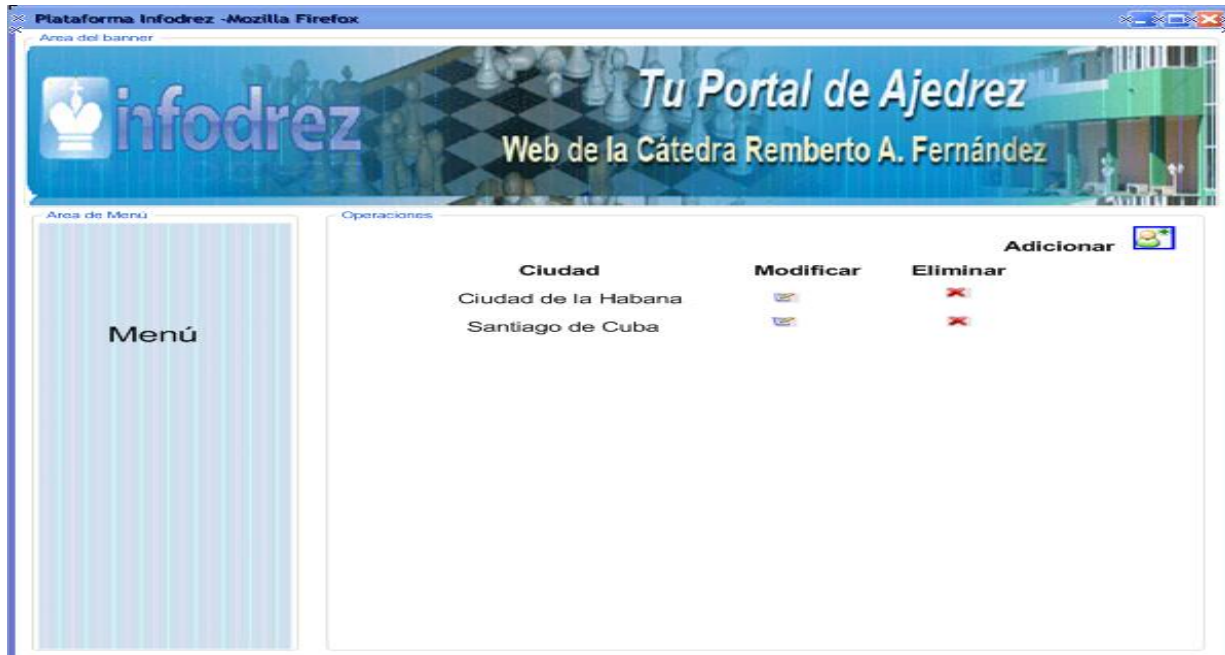


Prototipo: Gestionar Árbitro





Prototipo: Gestionar Ciudad







Prototipo: Gestionar Federación





Prototipo: Gestionar Título del Árbitro



Prototipo: Gestionar Título del Jugador





Prototipo: Adicionar Ritmo de Juego

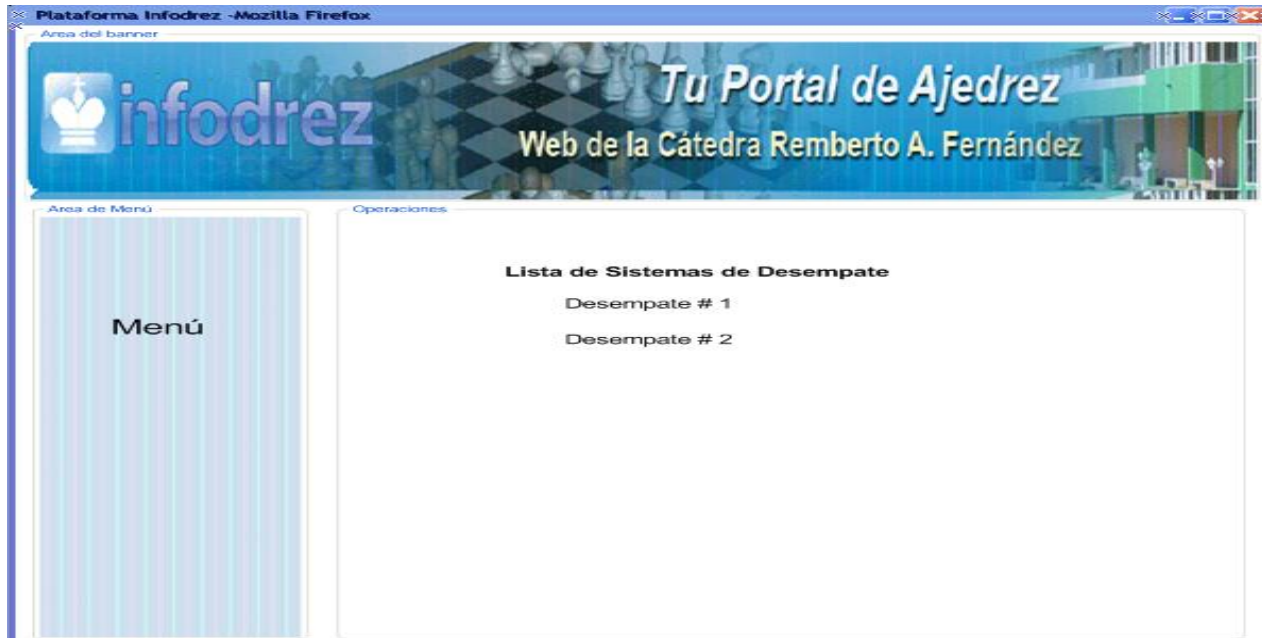


Prototipo: Mostrar Ritmo de Juego





Prototipo: Mostrar Sistemas de Desempate





Prototipo: Gestionar Jugador





Prototipo: Gestionar Equipo





Prototipo: Gestionar Torneo





Prototipo: Gestionar jugadores del torneo



Prototipo: Buscar Jugador







Prototipo: Definir sistemas de desempate





Prototipo: Elaborar Bases Técnicas





Prototipo: Realizar Sorteo





Prototipo: Realizar emparejamiento





Prototipo: Gestionar Ronda





Prototipo: Gestionar partida





Prototipo: Cálculo del Rating Elo

Nombre	Equipo	Rating ELO Inicial	Constante K	Puntos Alcanzados	Puntos Validos ELO	Expectancia	Rating Performance	Rating ELO Final	Variacion ELO
Henry	UH	2267	15	1	1	1.34	2170	2262	-5
Lelys	SCU	2510	15	2	2	2.48	2410	2503	-7

Prototipo: Mostrar Cuadro Sinóptico

Rk	Equipos	1	2	3	4	TB1	TB2
1	SCU	■	1%	1	3	5.5	3
2	UH	1%	■	1%	2	5	4
3	UCI	2	1%	■	1	4.5	3
4	Cujae	0	1	2	■	3	2

TB1: Puntos Totales  
TB2: Match Point



Prototipo: Mostrar emparejamiento y resultados

Area del banner

**infodrez** Tu Portal de Ajedrez  
Web de la Cátedra Remberto A. Fernández

Area de Menú

Operaciones

Ronda 1 Ronda 2 Ronda 3

**Ronda 1: 1985-01-10 a las 00:00:00**

Bo.	1	Cujae	Tab	2-1	4	UCI	Tab
1.1	MF	Lopez Vinas, Loes A	1	1-0		Hernandez, Ramon	1
1.2		Leonard, Pedro	2	0-1			2
1.3	WGM		3	1-0		Catala, Nette	3
Bo.	2	UH	Tab	1½-1½	3	SCU	Tab
2.1		Hernandez, Ramon	1	1-0	GM	Bruzon, Lazaro	1
2.2		Cruz De La Osa, Reider	2	½-½	GM	Gonzalez, Yuri	2
2.3	MF	Soto, Henry	3	0-1	GM	Martinez, Lelys	3





Prototipo: Mostrar Clasificación General

Plataforma Infodrez -Mozilla Firefox

Area del banner

**infodrez** Tu Portal de Ajedrez  
Web de la Cátedra Remberto A. Fernández

Area de Menú

Operaciones

### Clasificación General

RK	Nombre	Federacion	Enfrentamientos	Tb1	Tb2
1	SCU	CUB	3	5,5	3
2	UH	CUB	3	5	4
3	UCI	CUB	3	4,5	3
4	Cujae	CUB	3	3	2

Tb1: Puntos Totales  
Tb2: Match Point



Prototipo: Gestionar Grupo



Prototipo: Gestionar equipos de un grupo





Prototipo: Buscar equipo



Prototipo: Modificar tableros





Prototipo: Definir Alineación



Prototipo: Mostrar clasificación individual por tableros





Prototipo: Consultar Jugador

Plataforma Infodrez -Mozilla Firefox

Area del banner

**infodrez** Tu Portal de Ajedrez  
Web de la Cátedra Remberto A. Fernández

Area de Menú

Operaciones

**Consultar jugador**

Nombre del Jugador

Título

Federación

Elo

Resultados de la búsqueda:

Jugador 1

Jugador 2

Menú



Prototipo: Consultar Partida





Prototipo: Consultar información del torneo

Plataforma Infodrez -Mozilla Firefox

Area del banner

Area de Menú

Operaciones

**Consultar información del Torneo**

Nombre del Torneo

Fecha de inicio

Árbitro Principal

Resultados de la búsqueda:

Capablanca in Memoriam



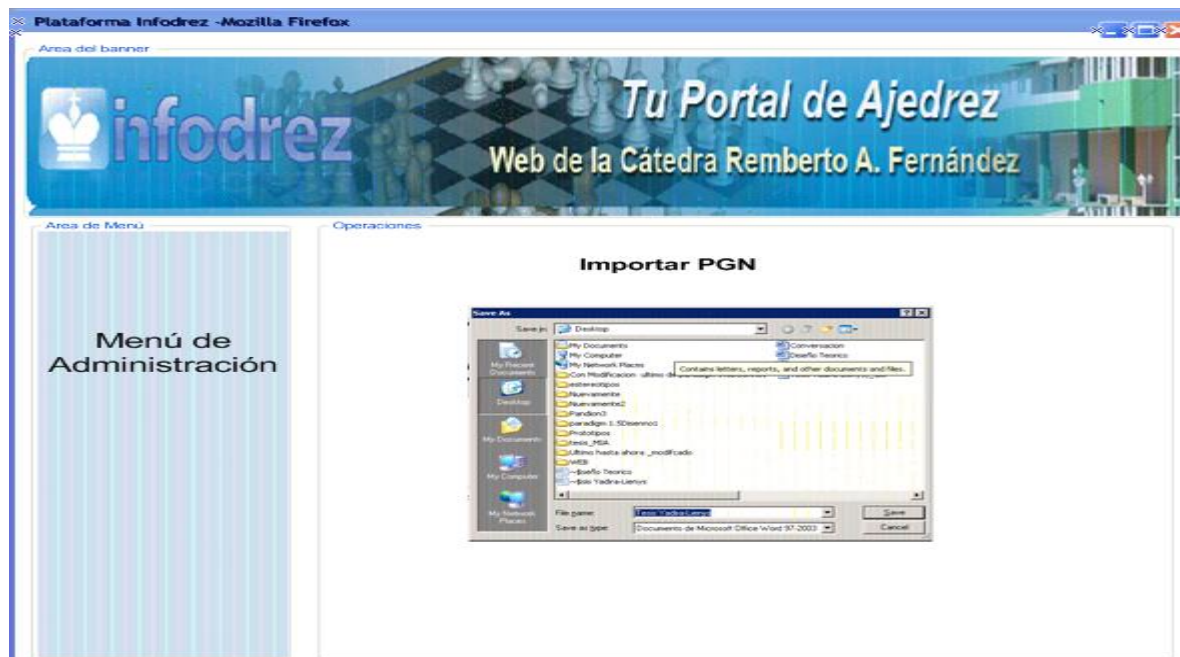
Prototipo: Gestionar Imagen







Prototipo: Importar PGN



Anexo 4: Descripciones de los CUS

Tabla 21: Descripción del CUS Autenticar Usuario

Caso de Uso:	Autenticar usuario
Actores:	Usuario
Resumen:	El CU se inicia cuando el usuario introduce sus datos en el sistema para autenticarse; y en este, según los datos introducidos le son asignados al usuario sus privilegios y en caso de que no sean correctos se le niega el acceso mostrando un mensaje de error.



Precondiciones:	
Poscondiciones:	La autenticación se realiza de forma correcta.
Referencias	RF 1
Prioridad	Secundario
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario entra sus datos de acceso al sistema	1.1- El sistema verifica que los datos de acceso introducidos son correctos.
	1.2- El sistema verifica que el usuario está registrado, el sistema le asigna sus privilegios según su rol y le permite entrar al sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.1 -Si los datos no están correctos el sistema muestra un mensaje denegando el acceso.
	1.2- Si el usuario no está registrado muestra un mensaje informando la situación

Tabla 22: Descripción del CU Gestionar Usuario

Caso de Uso:	Gestionar Usuario
--------------	-------------------



Actores:	Administrador
Resumen:	El Administrador del sistema tendrá la posibilidad de insertar, ver, modificar o eliminar la información referente a los usuarios.
Precondiciones:	El administrador debe estar previamente autenticado en el sistema.
Poscondiciones	
Referencias	RF2, RF2.1, RF2.3, RF2.3, RF 2.4
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El Administrador dentro de la página principal escoge la opción Gestionar Usuario del menú Administración	1.1-El sistema brinda las opciones: -Adicionar usuario -Ver Usuario -Modificar Usuario -Eliminar Usuario
2 El Administrador selecciona una de las opciones	2.1-Si selecciona “Adicionar Usuario ” ver sección Adicionar Usuario -Si selecciona “Ver Usuario ” ver sección Ver Usuario -Si selecciona “Modificar Usuario ” ver sección Modificar Usuario - Si selecciona “Eliminar Usuario ” ver sección Eliminar



	Usuario
Flujo Normal de Eventos	
Sección "Adicionar usuario"	
Acción del Actor	Respuesta del Sistema
1-El administrador escoge la opción Adicionar Usuario	2.1-El sistema muestra los campos a llenar para adicionar un nuevo usuario, los mismos son:  -Usuario  -Nombre  -Correo  -Rol  -Contraseña
2-El Administrador del sistema llena los campos mostrados por el sistema y pulsa el botón "Aceptar".	2.1- El sistema verifica que los datos estén correctos
	2.2-El sistema verifica si el usuario no está registrado en el sistema, en caso positivo adiciona el usuario al sistema y muestra un mensaje informando que el registro se realizó correctamente, terminando así el caso de uso
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1-El sistema muestra un mensaje señalando donde estuvo el error, se procede al flujo normal



	de eventos a partir de paso 2.
	2.2-Si el usuario ya está registrado el sistema muestra un mensaje informando la situación
Flujo Normal de Eventos	
Sección "Ver Usuario"	
Acción del Actor	Respuesta del Sistema
1-El administrador selecciona el usuario a consultar información	1.1-El sistema muestra los datos del usuario , termina así el caso de uso
Flujo Normal de Eventos	
Sección "Modificar Usuario"	
Acción del Actor	Respuesta del Sistema
1-El administrador escoge la opción Modificar Usuario	1.1 El sistema muestra un listado de todos los usuarios
2 El Administrador escoge el usuario que desea modificar	2.1 El sistema muestra un formulario con los datos del usuario a modificar
3-El administrador modifica los datos y pulsa el botón Aceptar	3.1-El sistema verifica que los datos están correctos, en caso positivo actualiza la base de datos y muestra un mensaje informando que los datos han sido modificados correctamente
Flujos Alternos	
Acción del Actor	Respuesta del Sistema



	2.1-Si los datos modificados son incorrectos el sistema muestra un mensaje informando sobre los campos que contienen errores.
Flujo Normal de Eventos	
Sección "Eliminar Usuario"	
Acción del Actor	Respuesta del Sistema
1- El administrador escoge la opción Eliminar Usuario	1.1-El sistema muestra un listado de todos los usuarios registrados
2-El Administrador escoge el usuario que desea eliminar y pulsa el botón Aceptar	2.1-El sistema elimina el usuario y muestra un mensaje informando que el usuario se ha eliminado.

Tabla 23: Descripción del CU Gestionar Federación

Caso de Uso:	Gestionar Federación
Actores:	Árbitro
Propósito:	Permite al árbitro gestionar toda la información referente a la federación: adicionar, ver, modificar y eliminar.
Resumen:	El caso de uso se inicia cuando el árbitro dentro del Menú de Arbitraje tiene la opción de gestionar la información de la federación.
Precondiciones:	
Poscondiciones:	- Federación registrada en la Base de Datos.



	-Datos de la federación actualizados.  - Federación eliminada de la Base de Datos.
Referencias	RF 5, RF 5.1, RF 5.2, RF 5.3, 5.4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona dentro de la página principal la opción Gestionar Federación del Menú Arbitraje.	1.1- El sistema brinda las opciones de:  - Adicionar Federación  -Ver Federación  -Modificar Federación  -Eliminar Federación
2 -El árbitro selecciona una de las opciones.	2.1- Si selecciona “Adicionar Federación” ver sección Adicionar Federación  - Si selecciona “Ver Federación” ver sección Ver Federación  -Si selecciona “Modificar Federación” ver sección Modificar Federación  -Si selecciona “Eliminar Federación” ver sección Eliminar Federación
Sección: “Adicionar Federación”	



Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona la opción de Adicionar Federación	1.1- El sistema muestra el formulario para que el árbitro llene los campos :  -Nombre  -Identificador  -Logo
2- El árbitro llena los datos y pulsa el botón Aceptar	2.1-El sistema verifica que los datos estén correctos.
	2.2-El sistema verifica que la federación no esté registrada y en caso positivo inserta los datos en la base de datos y muestra un mensaje informando que el registro se realizó correctamente Termina así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Adicionar Federación"	
	2.2- Si la federación ya está registrada el sistema muestra un mensaje informando la situación
Sección: "Ver Federación"	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona la federación a consultar información	1.1- El sistema muestra los datos de la federación, termina así el caso de uso





Sección: "Modificar Federación "	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la federación que desea modificar y marca la opción de Modificar Federación	1.1-El sistema muestra todos los datos de la federación para que sean modificados:  -Nombre  - Identificador  -Logo
2- El árbitro modifica los datos y pulsa el botón Aceptar	2.1- El sistema verifica que la a la federación a modificar no pertenezca ningún torneo que se esté efectuando, en caso positivo
	2.2- El sistema verifica que los datos modificados son válidos, en caso positivo estos se actualizan en la base de datos, concluyendo así el caso de uso del sistema.
Flujo Alternativo de Eventos	
Sección: "Modificar Federación "	
	2.1- Si la federación pertenece algún torneo que se esté efectuando en ese momento el sistema muestra el mensaje:  "No se puede eliminar la federación hasta no terminar el torneo"
	2.2- Si los datos que se introducen son incorrectos el sistema mostrará un mensaje



	señalando el campo donde los datos no son válidos. Se procede al flujo normal de eventos a partir del paso 2.
Sección: "Eliminar Federación "	
Acción del Actor	Respuesta del Sistema
1- El árbitro escoge la federación que desea eliminar y marca la opción Eliminar federación	1.1- El sistema verifica que la a la federación a eliminar no pertenezca ningún torneo que se esté efectuando, en caso positivo muestra un mensaje de confirmación para verificar que realmente se quiere eliminar la federación.
2- El árbitro pulsa el botón Aceptar del mensaje de confirmación.	2.1- El sistema elimina la federación de la base de datos y muestra un mensaje de que ha sido eliminada, termina así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Eliminar Federación "	
	1.1- Si la federación pertenece algún torneo que se esté efectuando en ese momento el sistema muestra el mensaje:  "No se puede eliminar la federación hasta no terminar el torneo"
2-El árbitro pulsa el botón Cancelar del mensaje de confirmación.	



Tabla 24: Descripción del CU Buscar jugador

Caso de Uso:	Buscar Jugador
Actores:	CU Gestionar jugadores de un equipo, CU Gestionar jugadores de un torneo.
Propósito:	Permite al árbitro realizar una búsqueda de jugadores.
Resumen:	Puede iniciarse cuando se ejecuta el caso de uso Gestionar jugadores de un equipo o CU Gestionar jugadores de un torneo.
Precondiciones:	Deben existir jugadores registrados en la base de datos.
Poscondiciones:	Lista de jugadores
Referencias	RF 15
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1-El sistema muestra un formulario especificando criterios para la búsqueda de él o los jugadores:  -Nombre  -Apellido  -Título  -Fecha de nacimiento



	-Rating Elo  -Código FIDE
2-El árbitro llena los campos necesarios para realizar la búsqueda y pulsa el botón Aceptar	2.2- El sistema muestra un listado de él o los jugadores registrados que respondan a los criterios de búsqueda, finalizando así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Buscar Jugador"	
	2.2- No existe ningún jugador que cumpla con los criterios especificados y el sistema muestra el mensaje :  "No hay resultado para esta búsqueda". Si el árbitro desea seguir buscando vuelve al paso 2.

Tabla 25: Descripción del CUS Definir Sistemas de Desempate

Caso de Uso:	Definir Sistemas de Desempate
Actor:	Árbitro
Propósito:	Permite al árbitro definir los Sistemas de Desempate a utilizar en cada Grupo de un Torneo.
Resumen:	El caso de uso se inicia cuando el árbitro selecciona dentro del Menú Gestionar Grupo, la opción Desempates, y el sistema brinda la posibilidad de definir que Sistemas de Desempate se utilizarán en cada uno de los Grupos del Torneo.
Precondiciones:	Debe estar creado al menos un grupo del torneo.



Poscondiciones:	Definición de los Sistemas de Desempate.
Referencias	RF 16
Prioridad	Crítico
Flujo Normal de Eventos	
Sección: "Definir Sistemas de Desempate para un Torneo Round Robin individual "	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona dentro de la página principal la opción Desempates del Menú Torneo	1.1- El sistema muestra todos los Sistemas de Desempates para un torneo Round Robin Individual brindando la posibilidad de establecer la prioridad que va a tener cada uno.
2- El árbitro selecciona los Sistemas de Desempate a utilizar, establece la prioridad de cada uno y pulsa el botón Aceptar.	2.1- El sistema guarda los sistemas de desempate seleccionados, junto a su prioridad, terminando así el caso de uso.
Sección: "Definir Sistemas de Desempate para un Torneo Round Robin por Equipos"	
1-El árbitro selecciona la opción Desempates del Menú Gestionar Grupo	1.1- El sistema muestra todos los Sistemas de Desempate para un torneo Round Robin por Equipos, brindando la posibilidad de establecer la prioridad que va a tener cada uno.
2- El árbitro selecciona los Sistemas de Desempate a utilizar, establece la prioridad de cada uno y pulsa el botón Aceptar.	2.1- El sistema guarda los sistemas de desempate seleccionados, junto a su prioridad, terminando así el caso de uso.



Tabla 26: Descripción del CU Gestionar Ronda

Caso de Uso:	Gestionar Ronda
Actores:	Árbitro
Propósito:	Permite al árbitro gestionar toda la información referente a las rondas: adicionar y modificar rondas.
Resumen:	El caso de uso se inicia cuando el árbitro selecciona dentro del Menú Torneo Gestionar Ronda y el sistema le brinda la posibilidad de adicionar o modificar rondas de un torneo.
Precondiciones:	El torneo debe estar creado.
Poscondiciones:	-Rondas adicionadas a la Base de Datos. -Rondas actualizadas.
Referencias	RF 20, RF 20.1, RF 20.2
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona dentro de la página principal la opción Gestionar Ronda del Menú Torneo	1.1- El sistema brinda las opciones de:  - Adicionar Ronda  -Modificar Ronda
2 -El árbitro selecciona una de las opciones	2.1- Si selecciona "Adicionar Ronda" ver sección Adicionar Ronda



	-Si selecciona "Modificar Ronda" ver sección Modificar Ronda
Sección: "Adicionar Ronda"	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona la opción de Adicionar Ronda	1.1-El sistema verifica que no estén todas las rondas adicionadas a ese torneo y en caso positivo muestra un formulario con los datos de la ronda a llenar:  -Fecha  -Hora
2- El árbitro llena los campos y pulsa el botón Aceptar	2.1- El sistema verifica que los datos son correctos y en caso afirmativo inserta los datos en la base de datos y muestra un mensaje informando que el registro se realizó correctamente, concluyendo con esto el caso de uso del sistema.
Flujo Alternativo de Eventos	
Sección: "Adicionar Ronda"	
	1.1 -Si ya están todas las rondas adicionadas muestra un mensaje diciendo que ya todas las rondas están adicionadas.
	2.1- Si los datos introducidos por el árbitro son incorrectos el sistema muestra un mensaje indicando que los resultados no son válidos. Se procede al flujo normal de eventos desde el paso 2



Sección: "Modificar Ronda"	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona la opción de Modificar Ronda	1.1-El sistema brinda la posibilidad de escoger la ronda que se desea modificar.
2- El árbitro selecciona la ronda a modificar	2.1- El sistema muestra los datos de la ronda a ser modificados.
3- El árbitro modifica los datos de la ronda y pulsa el botón Aceptar	3.1- El sistema verifica que los datos modificados son válidos, en caso positivo se actualizan en la base de datos, concluyendo así el caso de uso del sistema.
Flujo Alternativo de Eventos	
Sección: "Modificar Ronda"	
	3.1- Si los datos que se introducen son incorrectos el sistema mostrará un mensaje señalando que los datos no son válidos. Se procede al flujo normal de eventos a partir del paso 3.

Tabla 27: Descripción del CU Gestionar Partida

Caso de Uso:	Gestionar Partida
Actores:	Árbitro
Propósito:	Permite al árbitro gestionar toda la información referente a las partidas: adicionar, modificar y eliminar las partidas.





Resumen:	El caso de uso se inicia cuando el árbitro selecciona dentro del Menú Torneos Gestionar Partidas: registrar, modificar o eliminar cada una de las partidas.
Precondiciones:	El torneo y las rondas deben estar creados, además de haberse realizado el sorteo
Poscondiciones:	-Partidas registradas en la Base de Datos -Partidas actualizadas en la base de datos -Partidas eliminadas de la base de datos
Referencias	RF 21, RF 21.1, RF 21.2, RF 21.3
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona dentro de la página principal la opción Gestionar Partida del Menú Torneo.	1.1- El sistema brinda las opciones de: - Registrar Partida -Ver Partida -Modificar Partida -Eliminar Partida
2 -El árbitro selecciona una de las opciones.	2.1- Si selecciona "Registrar Partida" ver sección Registrar Partida -Si selecciona "Ver Partida" ver sección Ver Partida -Si selecciona "Modificar Partida" ver sección Modificar



	Partida  --Si selecciona "Eliminar Partida" ver sección Eliminar Partida
Sección: "Registrar Partida"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Registrar Partida	1.1-El sistema verifica que la partida no esté registrada  1.2- El sistema verifica que ya se haya realizado el pareo y en caso positivo muestra un listado con todas las partidas que se han realizado.
2-El árbitro inserta las partidas y pulsa el botón Aceptar	2.1- El sistema verifica que los datos son correctos y en caso positivo los inserta en la base de datos .Termina así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Registrar Partida"	
	1.1- Si ya está registrada la partida el sistema muestra un mensaje diciendo que ya la partida está registrada.  1.2- Si no se ha realizado el pareo muestra un mensaje informando que debe realizar el pareo primero.
	2.1- Si los datos no son correctos el sistema muestra un mensaje de que los resultados no son correctos y que debe volver a entrar los datos. Vuelve al paso 2
Sección: "Ver Partida"	



Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la partida	1.1-El sistema muestra los datos de la partida seleccionada
Sección: "Modificar Partida"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Modificar Partida	1.1-El sistema brinda la opción de escoger la ronda en la que va a modificar la partida
2 - El árbitro selecciona la ronda.	2.1-El sistema muestra las partidas pertenecientes a esa ronda para que se puedan modificar
3- El árbitro realiza las modificaciones y pulsa el botón Aceptar	3.1- El sistema verifica que los datos estén correctos, en caso positivo registra los cambios en la base de datos finalizando así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Modificar Partida"	
	3.1- Si los datos que se introducen son incorrectos el sistema mostrará un mensaje señalando los datos que no son válidos. Se procede al flujo normal de eventos a partir del paso 3.
Sección: "Eliminar Partida"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Eliminar Partida	1.1-El sistema brinda la opción de escoger la ronda en la que va a eliminar la partida



2 - El árbitro selecciona la ronda.	2.1- El sistema muestra las partidas pertenecientes a esa ronda.
3-El sistema escoge la partida a eliminar y pulsa el botón Aceptar	3.1- El sistema verifica que la partida no pertenezca a un torneo que no haya finalizado , en caso positivo muestra un mensaje de confirmación
4-El árbitro acepta el mensaje de confirmación	4.1-El sistema elimina la partida concluyendo así el caso de uso
Flujo Alternativo de Eventos	
Sección: "Eliminar Partida"	
	3.1- Si el torneo no ha finalizado muestra un mensaje de que no se puede eliminar la partida hasta que el torneo no haya finalizado.
4- El árbitro presiona el botón Cancelar del mensaje de confirmación.	

Tabla 28: Descripción del CU Gestionar Grupo

Caso de Uso:	Gestionar Grupo
Actores:	Árbitro
Propósito:	Brinda la posibilidad a los árbitros de gestionar la información relacionada con los grupos que componen un torneo: adicionar, ver, modificar o eliminar un grupo.
Resumen:	El caso de uso inicia cuando el árbitro selecciona la opción Grupo del Menú Torneo y el sistema brinda la posibilidad de gestionar grupos del torneo.



Precondiciones:	El torneo debe estar creado.
Poscondiciones:	-Nuevo grupo de torneo registrado en la Base de Datos - Actualización de los datos de un grupo de torneo -Grupo de torneo eliminado de la Base de Datos
Referencias	RF 26, RF 26.1, RF 26.2, RF 26.3, RF 26.4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-1-El árbitro selecciona dentro de la página principal la opción Gestionar Grupo del Menú Torneo.	1.1- El sistema brinda las opciones de: -Adicionar Grupo. -Ver Grupo - Modificar Grupo. -Eliminar Grupo.
2 -El árbitro selecciona una de las opciones.	2.1- Si selecciona "Adicionar Grupo" ver sección Adicionar Grupo - Si selecciona "Ver Grupo" ver sección Ver Grupo -Si selecciona "Modificar Grupo" ver sección Modificar Grupo - Si selecciona "Eliminar Grupo" ver sección Eliminar Grupo



Sección: "Adicionar Grupo"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Adicionar Grupo	1.1-El sistema verifica que no estén creados todos los grupos especificados para ese torneo, en caso positivo el sistema muestra un formulario con los datos a ser llenados:  - Número del Grupo  -Nombre del Grupo  -Tipo de Torneo  -Ritmo de Juego
2- El árbitro introduce los datos para llenar los campos correspondientes y pulsa el botón Aceptar	2.1-El sistema verifica que el grupo no se encuentre adicionado.
	2.2-El sistema verifica que los datos introducidos por el árbitro son válidos, en caso positivo se insertan los mismos en la base de datos, concluyendo así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Adicionar Grupo"	
	1.1-Si ya están creados todos los grupos de ese torneo el sistema muestra el mensaje:  "Ya todos los grupos de este torneo se encuentran registrados en el sistema"
	2.1-Si el grupo ya existe el sistema emite un mensaje



	informando la existencia del mismo.
	2.2- Si los datos que se introducen son incorrectos el sistema mostrará un mensaje señalando el campo donde los datos no son válidos. Se procede al flujo normal de eventos a partir de la acción 2.
Sección: "Ver Grupo"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona el grupo	1.1-El sistema muestra los datos del grupo seleccionado
Sección: "Modificar Grupo"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción Modificar Grupo	1.1- El sistema verifica que el grupo no pertenezca a un torneo que no haya finalizado, en caso positivo muestra todos los datos del grupo de torneo para que sean modificados.
2- El árbitro modifica los datos deseados y pulsa el botón Aceptar	2.1 -El sistema verifica que los datos modificados son válidos, en caso positivo estos se actualizan en la base de datos, concluyendo así el caso de uso del sistema.
Flujo Alternativo de Eventos	
Sección: "Modificar Grupo"	
	1.1-Si el grupo pertenece a un torneo que no haya finalizado se muestra un mensaje informando la situación
	2.1- Si los datos que se introducen son incorrectos el



	sistema mostrará un mensaje señalando el campo donde los datos no son válidos. Se procede al flujo normal de eventos a partir del paso 2.
Sección: "Eliminar Grupo"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Eliminar Grupo	1.1- El sistema verifica que el grupo no pertenezca a un torneo que no haya finalizado, en caso positivo el sistema muestra un mensaje de confirmación para eliminar el grupo.
2-El árbitro pulsa el botón Aceptar del mensaje de confirmación.	2.1- El sistema lo elimina de la base de datos y muestra un mensaje de que el grupo ha sido eliminado , terminando así el caso de uso
Flujo Alternativo de Eventos	
Sección: "Eliminar Grupo"	
	1.1- Si el grupo pertenece a un torneo que no ha finalizado el sistema muestra un mensaje informando la situación
2- El árbitro pulsa el botón Cancelar del mensaje de confirmación.	

Tabla 29: Descripción del CU Gestionar jugadores de un equipo

Caso de Uso:	Gestionar jugadores de un equipo
Actores:	Árbitro





Propósito:	Permite al árbitro gestionar toda la información referente a los jugadores que van a participar en el torneo por equipos, posibilitando la opción de adicionar o eliminar jugadores.
Resumen:	El caso de uso se inicia cuando el árbitro selecciona dentro del Menú Torneo Gestionar jugadores y el sistema le brinda la posibilidad de ir adicionando o eliminando jugadores a los equipos que forman el mismo.
Precondiciones:	El torneo debe estar creado.
Poscondiciones:	-Jugadores adicionados al equipo.  - Jugadores eliminados del equipo.
Referencias	RF 29, RF 29.1, RF 29.2
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona dentro de la página principal la opción Gestionar jugadores de un equipo del Menú Torneo	1.1- El sistema brinda las opciones de:  - Adicionar jugadores a un equipo  - Eliminar jugadores de un equipo
2 -El usuario selecciona una de las opciones	2.1- Si selecciona “Adicionar jugadores ” ver sección Adicionar jugadores a un equipo  -Si selecciona “Eliminar jugadores” ver sección Eliminar jugadores de un equipo
Sección: “Adicionar jugadores a un equipo”	



Acción del Actor	Respuesta del Sistema
1- El usuario selecciona la opción de Adicionar jugadores	1.1-El sistema permite seleccionar el equipo al que se desea adicionar jugadores
2- El usuario selecciona el equipo deseado	2.1-El sistema muestra un listado con los jugadores pertenecientes al equipo con el número del tablero correspondiente.  2.2-Puede ejecutarse el caso de uso “Buscar Jugador”
3- El usuario selecciona él o los jugadores deseados y pulsa el botón Aceptar	3.1- El sistema verifica que el jugador no esté asignado ya a ese equipo, o a otro equipo de ese mismo grupo de torneo, en caso positivo muestra el listado actualizado de los jugadores del equipo finalizando así el caso de uso.
Flujo Alternativo de Eventos	
Sección: “Adicionar jugadores a un equipo”	
	3.1- Si el jugador ya está asignado a algún equipo de ese mismo grupo, el sistema muestra un mensaje para informar que el jugador ya está asignado.
Sección: “Eliminar jugadores de un equipo”	
Acción del Actor	Respuesta del Sistema
1- El usuario selecciona la opción de Eliminar Jugadores	1.1-El sistema posibilita seleccionar el equipo del que se desea eliminar jugadores
2- El usuario selecciona el equipo	2.1-El sistema muestra un listado con los jugadores pertenecientes al equipo con el número del tablero correspondiente, además del grupo y el torneo al que



	pertenece el equipo
3- El usuario selecciona el jugador a eliminar	3.1- El sistema verifica que los jugadores no pertenezcan a un torneo que no haya finalizado, en caso positivo muestra un mensaje de confirmación.
4-EL usuario pulsa el botón Aceptar del mensaje de confirmación.	4.1- El sistema verifica si en ese equipo no se ha realizado el sorteo, en caso positivo el sistema muestra un mensaje de confirmación, terminando así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Eliminar jugadores de un equipo "	
4- EL usuario pulsa el botón Cancelar del mensaje de confirmación.	4.1- Si ya se ha realizado el sorteo, entonces el sistema muestra un mensaje informando que no se pueden eliminar estos jugadores hasta no deshacer el sorteo.

Tabla 30: Descripción del CU Consultar Partida

Caso de Uso:	Consultar Partida
Actores:	Invitado
Propósito:	Permite al invitado ver los datos de una partida.
Resumen:	El caso de uso se inicia cuando el invitado visita la plataforma y dentro del Menú Consultar Partida le brinda la posibilidad de consultar información de alguna partida.



Precondiciones:	El sistema debe estar disponible.
Poscondiciones:	Información de la partida mostrada al Invitado.
Referencias	RF 34
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El invitado dentro de la página principal accede a la opción Consultar Partida.	1.1-El sistema muestra los siguientes criterios de búsqueda :  -Nombre del Torneo  -Fecha del Torneo  - Nombre Jugador
2-El invitado introduce los criterios de búsqueda y pulsa el botón Aceptar.	2.1-El sistema muestra un listado de las partidas que corresponden al criterio especificado.
4- El Invitado selecciona la partida de la que desea consultar información.	4.1-El sistema accede a la Sección “Ver partida” del CU Gestionar Partida.

Tabla 31: Descripción del CU Importar PGN

Caso de Uso:	Importar PGN
--------------	--------------



Actores:	Administrador
Propósito:	Permite al administrador importar archivos PGN
Resumen:	El caso de uso se inicia cuando el administrador dentro de las opciones de administración va a tener la posibilidad de importar archivos PGN
Precondiciones:	
Poscondiciones:	Archivo PGN almacenado en la base de datos
Referencias	37
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El administrador dentro de la página principal accede a la opción Importar PGN	1.1-El sistema muestra una ventana para buscar en el directorio de ficheros el archivo PGN a importar
2-El administrador selecciona el fichero	2.1-El sistema guarda la información interpretada (en forma de torneo) del fichero en la base de datos y muestra un mensaje de que la operación se ha realizado con éxito.

Tabla 32: Descripción del CU Gestionar Ciudad

Caso de Uso:	Gestionar Ciudad
Actores:	Árbitro



Propósito:	Permite al árbitro gestionar toda la información referente a la ciudad: adicionar, modificar y eliminar.
Resumen:	El caso de uso se inicia cuando el usuario dentro del Menú de Arbitraje tiene la opción de gestionar la información de la ciudad.
Precondiciones:	
Poscondiciones:	-Ciudad adicionada a la Base de Datos. -Datos de la ciudad actualizados. - Ciudad eliminada de la Base de Datos.
Referencias	RF 4 RF 4.1, RF 4.2, RF 4.3
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona dentro de la página principal la opción Gestionar Ciudad del Menú Arbitraje.	1.1- El sistema brinda las opciones de: - Adicionar Ciudad -Modificar Ciudad -Eliminar Ciudad
2 -El árbitro selecciona una de las opciones.	2.1- Si selecciona “Adicionar Ciudad” ver sección Adicionar Ciudad -Si selecciona “Modificar Ciudad” ver sección Modificar Ciudad



	-Si selecciona "Eliminar Ciudad" ver sección Eliminar Ciudad
Sección: "Adicionar Ciudad"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Adicionar Ciudad	1.1- El sistema muestra el formulario con la opción para que el árbitro entre el nombre de la ciudad
2-El árbitro introduce el nombre y pulsa el botón Aceptar	2.1- El sistema verifica que la ciudad no esté registrada, en caso positivo inserta la ciudad en la base de datos y muestra un mensaje informando que el registro se realizó correctamente, terminando así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Adicionar Ciudad"	
	2.1- Si la ciudad ya está registrada el sistema muestra un mensaje informando la situación
Sección: "Modificar Ciudad"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona al ciudad que desea modificar y marca la opción de Modificar Ciudad	1.1- El sistema muestra los datos de la ciudad a ser modificada
2- El árbitro modifica los datos y pulsa el botón Aceptar	2.1- El sistema verifica que la ciudad que se desea eliminar no tiene algún torneo efectuándose, en caso positivo se actualizan los cambios en la base



	de datos, terminando así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Modificar Ciudad"	
	2.1- Si la ciudad tiene algún torneo efectuándose el sistema muestra un mensaje informando la situación
Sección: "Eliminar Ciudad"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la ciudad que desea eliminar y marca la opción de Eliminar Ciudad	1.1- El sistema verifica que la ciudad que se desea eliminar no tiene algún torneo efectuándose, en caso positivo muestra un mensaje de confirmación para verificar que realmente se quiere eliminar la ciudad.
2- El árbitro pulsa el botón Aceptar del mensaje de confirmación.	2.1- El sistema elimina la ciudad de la base de datos y muestra un mensaje de que ha sido eliminada, termina así el caso de uso
Flujo Alternativo de Eventos	
Sección: "Eliminar Ciudad"	
	1.1- Si la ciudad tiene algún torneo efectuándose el sistema muestra el mensaje:  "No se puede eliminar la ciudad hasta no terminar el torneo"





2-El árbitro pulsa el botón Cancelar del mensaje de confirmación.	
---	--

Tabla 33: Descripción del CU Gestionar Título del Árbitro

Caso de Uso:	Gestionar Título del Árbitro
Actores:	Árbitro
Propósito:	Permite al árbitro gestionar toda la información referente a los títulos que puede tener un árbitro: adicionar, modificar y eliminar título del árbitro.
Resumen:	El caso de uso se inicia cuando el árbitro dentro del Menú de Arbitraje tiene la opción de gestionar la información del título del árbitro.
Precondiciones:	
Poscondiciones:	<ul style="list-style-type: none"><li>- Título del árbitro registrado en la Base de Datos.</li><li>- Título de árbitro actualizado.</li><li>- Título de árbitro eliminado de la Base de Datos.</li></ul>
Referencias	RF 6, RF 6.1, RF6.2, RF 6.3
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona dentro de la página principal la opción Gestionar	1.1- El sistema brinda las opciones de: <ul style="list-style-type: none"><li>- Adicionar Título del Árbitro</li></ul>



Título del Árbitro del Menú Arbitraje.	-Modificar Título del Árbitro  -Eliminar Título del Árbitro
2 -El árbitro selecciona una de las opciones.	2.1- Si selecciona “Adicionar Título del Árbitro” ver sección Adicionar Título del Árbitro  -Si selecciona “Modificar Título del Árbitro” ver sección Modificar Título del Árbitro  -Si selecciona “Eliminar Título del árbitro” ver sección Eliminar Título del Árbitro
Sección: “Adicionar Título del Árbitro”	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Adicionar Título del Árbitro	1.1- El sistema muestra el formulario con la opción para que el árbitro entre el título del árbitro
2- El árbitro introduce el título del árbitro y pulsa el botón Aceptar	2.1-El sistema verifica si el título del árbitro es correcto.
	2.2-El sistema verifica si el título del árbitro ya está registrado, en caso positivo lo inserta en la base de datos y muestra un mensaje informando que el registro se realizó correctamente. Termina así el caso de uso.
Flujo Alternativo de Eventos	
Sección: “Adicionar Título de Arbitro”	
	2.1- Si los datos introducidos no son correctos el sistema muestra un mensaje informando que los datos no son correctos. Se procede al flujo normal de eventos a partir



	del paso 2.
	2.2- Si el título del árbitro ya está registrado el sistema muestra un mensaje informando la situación
Sección: "Modificar Título del Árbitro "	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona el Título de Árbitro que desea modificar y marca la opción de Modificar Título del Árbitro	1.1-El sistema muestra el título del Árbitro que se va a modificar.
2- El árbitro modifica el título y pulsa el botón Aceptar	2.1- El sistema verifica que los datos modificados son válidos, en caso positivo estos se actualizan en la base de datos, termina así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Modificar Título del Árbitro"	
	2.1- Si los datos que se introducen son incorrectos el sistema mostrará un mensaje señalando el campo donde los datos no son válidos. Se procede al flujo normal de eventos a partir del paso 2.
Sección: "Eliminar Título del Árbitro"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona el Título de Arbitro que desea eliminar y marca la opción de Eliminar Título del Árbitro	1.1-El sistema muestra un mensaje de confirmación para verificar que realmente se quiere eliminar el título del árbitro.



2- El árbitro pulsa el botón Aceptar del mensaje de confirmación.	2.1- El sistema elimina el título del árbitro de la base de datos y muestra un mensaje de que ha sido eliminado, termina así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Eliminar Título del Árbitro"	
2-El árbitro pulsa el botón Cancelar del mensaje de confirmación.	

Tabla 34: Descripción del CU Gestionar Título del Jugador

Caso de Uso:	Gestionar Título del Jugador
Actores:	Árbitro
Propósito:	Permite al árbitro gestionar toda la información referente a los títulos que puede tener un jugador de un torneo: adicionar, modificar y eliminar títulos de jugador.
Resumen:	El caso de uso se inicia cuando el árbitro dentro del Menú de Arbitraje tiene la opción de gestionar la información del título del jugador.
Precondiciones:	
Poscondiciones:	- Título del jugador registrado en la Base de Datos. - Título del jugador actualizado. - Título del jugador eliminado de la Base de Datos.
Referencias	RF 7, RF 7.1, RF 7.2, RF 7.3



Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona dentro de la página principal la opción Gestionar Título del Jugador del Menú Arbitraje.	1.1- El sistema brinda las opciones de:  - Adicionar Título del Jugador  -Modificar Título del Jugador  -Eliminar Título del Jugador
2 -El árbitro selecciona una de las opciones.	2.1- Si selecciona “Adicionar Título del Jugador” ver sección Adicionar Título del Jugador  -Si selecciona “Modificar Título del Jugador” ver sección Modificar Título del Jugador  -Si selecciona “Eliminar Título del Jugador” ver sección Eliminar Título del Jugador
Sección: “Adicionar Título del Jugador”	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Adicionar Título del Jugador	1.1- El sistema muestra el formulario con la opción para que el árbitro entre el título del jugador
2- El árbitro introduce el título del jugador y pulsa el botón Aceptar	2.1-El sistema verifica que el título del jugador esté correcto.
	2.2-El sistema verifica que el título del jugador no esté registrado y en caso positivo lo inserta en la base de datos y muestra un mensaje informando



	que el registro se realizó correctamente, terminando así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Adicionar Título de Jugador"	
	2.1- Si los datos introducidos no son correctos el sistema muestra un mensaje informando que los datos no son válidos. Se procede al flujo normal de eventos a partir del paso 2.
	2.2- Si el título del jugador ya está registrado el sistema muestra un mensaje informando la situación
Sección: "Modificar Título del Jugador "	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona el título del jugador que desea modificar y marca la opción de Modificar Título del Jugador	1.1- El sistema muestra el título del jugador a modificar.
2- El árbitro modifica los datos y pulsa el botón Aceptar	2.1- El sistema verifica que los datos modificados son válidos, en caso positivo estos se actualizan en la base de datos, termina así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Modificar Título del Jugador"	
	2.1- Si los datos que se introducen son incorrectos el sistema mostrará un mensaje señalando el



	campo donde los datos no son válidos. Se procede al flujo normal de eventos a partir del paso 2.
Sección: "Eliminar Título del Jugador"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona el título del jugador que desea eliminar y marca la opción de Eliminar Título del Jugador	1.1- El sistema muestra un mensaje de confirmación para verificar que realmente se quiere eliminar el título del jugador.
2- El árbitro pulsa el botón Aceptar del mensaje de confirmación.	2.1- El sistema elimina el título del jugador de la base de datos y muestra un mensaje de que ha sido eliminado, termina así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Eliminar Título del Jugador"	
2-El árbitro pulsa el botón Cancelar del mensaje de confirmación.	

Tabla 35: Descripción del CU Adicionar Ritmo de Juego

Caso de Uso:	Adicionar Ritmo de Juego
Actores:	Árbitro
Propósito:	Permite al árbitro adicionar el ritmo de juego del torneo.
Resumen:	El caso de uso se inicia cuando el árbitro dentro del Menú de Arbitraje tiene la opción de adicionar el ritmo de juego.



Precondiciones:	
Poscondiciones:	- Ritmo de juego registrado en la Base de Datos.
Referencias	RF 8
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona dentro de la página principal la opción de Adicionar Ritmo de Juego	1.1- El sistema brinda la posibilidad de adicionar el ritmo de Juego.
2- El árbitro adiciona el ritmo de juego y pulsa el botón Aceptar	2.1- El sistema verifica que los datos introducidos sean correctos.
	2.2-El sistema verifica que el ritmo de juego no está registrado, en caso positivo guarda en la base de datos el ritmo de juego y muestra un mensaje informando que el registro se realizó correctamente, terminando así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Adicionar Ritmo de Juego"	
	2.1- Si los datos introducidos no son correctos el sistema muestra un mensaje informando que los datos no son válidos. Se procede al flujo normal de eventos a partir del paso 2.





	2.2- Si el ritmo de juego ya está registrado el sistema muestra un mensaje informando la situación
--	--

Tabla 36: Descripción del CU Mostrar Sistemas de Desempate

Caso de Uso:	Mostrar Sistemas de Desempate
Actor:	Árbitro
Propósito:	Permite mostrar los Sistemas de Desempates registrados en el Sistema.
Resumen:	El caso de uso se inicia al acceder a la opción Mostrar Desempate, dentro del Menú Arbitraje se muestra una lista de los sistemas de desempate, especificando si se aplican a Jugadores o Equipos.
Precondiciones:	
Poscondiciones:	Lista de los sistemas de desempate registrados.
Referencias	RF 10
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona dentro de la página principal la opción Mostrar Desempate, del menú Arbitraje	1.1- El sistema brinda un listado de todos los sistemas de desempates a utilizar.

Tabla 37: Descripción del CU Gestionar jugadores de un torneo



Caso de Uso:	Gestionar jugadores de un torneo
Actores:	Árbitro
Propósito:	Permite al árbitro gestionar toda la información referente a los jugadores que van a participar en el torneo de tipo individual posibilitando la opción de adicionar o eliminar jugadores.
Resumen:	El caso de uso se inicia cuando el árbitro selecciona dentro del Menú Torneo Gestionar jugadores y el sistema le brinda la posibilidad de ir adicionando o eliminando jugadores al torneo.
Precondiciones:	El torneo debe estar creado.
Poscondiciones:	-Jugadores adicionados al torneo. - Jugadores eliminados del torneo.
Referencias	RF 14, RF 14.1, RF 14.2
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona dentro de la página principal la opción Gestionar jugadores de un Torneo del Menú Torneo	1.1- El sistema brinda las opciones de: - Adicionar jugadores a un torneo - Eliminar jugadores de un torneo
2 -El árbitro selecciona una de las opciones	2.1- Si selecciona "Adicionar jugadores" ver sección Adicionar jugadores a un torneo



	-Si selecciona “Eliminar jugadores” ver sección Eliminar jugadores de un torneo
Sección: “Adicionar jugadores a un torneo ”	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona la opción de Adicionar jugadores	1.1-El sistema muestra un listado con los jugadores que ya están registrados en ese torneo unido al número del tablero correspondiente.
	1.2-Puede ejecutarse el caso de uso “Buscar Jugador”
2- El árbitro selecciona él o los jugadores deseados y pulsa el botón Aceptar	2.1- El sistema verifica que el jugador no esté asignado a ese torneo, en caso positivo muestra el listado actualizado de los jugadores del torneo finalizando así el caso de uso.
Flujo Alternativo de Eventos	
Sección: “Adicionar jugadores a un torneo ”	
	2.1- Si el jugador ya está asignado a ese torneo, el sistema muestra un mensaje para informar que el jugador ya se encuentra.
Sección: “Eliminar jugadores de un torneo ”	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Eliminar Jugadores	1.1- El sistema muestra un listado con los jugadores que ya están registrados en ese torneo unido al número del tablero correspondiente.



2- El árbitro selecciona el jugador a eliminar	2.1- El sistema verifica si en ese torneo no se ha realizado el sorteo, en caso positivo envía un mensaje de confirmación
3- EL árbitro pulsa el botón Aceptar del mensaje de confirmación.	3.1-El sistema elimina él o los jugadores seleccionados, terminando así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Eliminar jugadores de un torneo"	
	2.1- Si ya se ha realizado el sorteo, entonces el sistema muestra un mensaje informando que no se pueden eliminar estos jugadores hasta no deshacer el sorteo.
3- EL árbitro pulsa el botón Cancelar del mensaje de confirmación.	

Tabla 38: Descripción del CU Elaborar Bases Técnicas

Caso de Uso:	Elaborar Bases Técnicas
Actores:	Árbitro
Propósito:	Brindar la posibilidad al árbitro de elaborar un reporte con las Bases Técnicas de un torneo en formato .doc.
Resumen:	El caso de uso comienza cuando el Árbitro selecciona la opción Bases Técnicas del Menú Torneo. Se llena un formulario con las mismas, brindando la posibilidad de guardarlas como un documento, editarlo e imprimirlo.
Precondiciones:	El torneo debe haberse creado.



Poscondiciones	Se conforma el documento que contendrá las Bases Técnicas del torneo.
Referencias	RF 17
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona dentro de la página principal la opción Bases Técnicas del Menú Torneo.	1.1-El sistema muestra el formulario con un listado de los participantes y los siguientes datos a llenar:  -Reglas y Sistemas de Juegos  -Planificación del evento  -Casos de Desempates  -Premios  -Tribunal de Apelaciones  - Personal Técnico
2-El árbitro introduce los datos a llenar del formulario	2.1-El sistema verifica que estén todos los campos del formulario llenados y en caso positivo genera un documento con las Bases Técnicas del torneo, brindando la posibilidad de imprimir el mismo, terminando así el caso de uso.
Flujo Alterno de Eventos	
	2.1- Si no están todos los campos llenados el sistema muestra un mensaje informando:  "Faltan campos por llenar".



	Se procede al flujo normal de eventos a partir del paso 2
--	---

Tabla 39: Descripción del CU Cálculo del Rating Elo

Caso de Uso:	Cálculo del Rating Elo
Actores:	Árbitro
Propósito:	Brindar la posibilidad al árbitro de tener registrado información sobre cada uno de los parámetros para el cálculo del Elo de cada jugador participante en el torneo.
Resumen:	El caso de uso se inicia cuando el árbitro selecciona la opción Cálculo de Elo del menú Torneo mostrándose todos los jugadores con el valor actual de cada uno de los parámetros necesarios para calcular el Elo de cada jugador.
Precondiciones:	Deben estar registrados los jugadores en el torneo y existir al menos el resultado de alguna de las partidas.
Poscondiciones:	Actualización del Elo de cada jugador.
Referencias	RF 22
Prioridad	Crítico
Flujo Normal de Eventos	
Sección: "Cálculo del Rating Elo para un torneo Round Robin individual "	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona dentro de la página principal la opción Cálculo de Elo del Menú Torneo	1.1- El sistema verifica que el torneo tenga jugadores registrados.



	<p>1.2-El sistema verifica que exista al menos el resultado de una de las partidas jugadas, en caso positivo muestra un cuadro con la siguiente información:</p> <ul style="list-style-type: none"><li>-Lugares</li><li>-Nombre del jugador</li><li>- Rating Elo Inicial</li><li>- Constante K</li><li>- Puntos Alcanzados</li><li>- Puntos Válidos Elo</li><li>- Expectancia</li><li>- Rating Performance</li><li>- Rating Elo Final</li><li>- Variación Elo</li></ul>
Flujo Alternativo de Eventos	
	<p>1.1- Si no existen jugadores registrados en el torneo el sistema emite un mensaje informando que no existen jugadores en ese torneo.</p>
	<p>1.2-Si no existe el resultado de ninguna partida el sistema emite un mensaje informando que deben existir resultados de alguna partida.</p>
Sección: "Cálculo del Rating Elo para un torneo Round Robin por equipos "	



Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción Cálculo de Elo del Menú Torneo	1.1- El sistema verifica que el torneo tenga los equipos registrados con sus jugadores.
	1.2- El sistema verifica que exista al menos el resultado de una de las partidas jugadas, en caso positivo muestra un cuadro con la siguiente información:  -Nombre del jugador  -Equipo  - Rating Elo Inicial  - Constante K  - Puntos Alcanzados  - Puntos Válidos Elo  - Expectancia  - Rating Performance  - Rating Elo Final  - Variación Elo
Flujo Alternativo de Eventos	
	1.1- Si no existen jugadores registrados en el torneo el sistema emite un mensaje informando que no existen jugadores en ese torneo.
	1.2-Si no existe el resultado de ninguna partida el sistema





	emite un mensaje informando que deben existir resultados de alguna partida.
--	---

Tabla 40: Descripción del CU Mostrar Cuadro Sinóptico

Caso de Uso:	Mostrar Cuadro Sinóptico
Actores:	Árbitro
Propósito:	Brindar la posibilidad al árbitro de tener registrado en un cuadro sinóptico la información más relevante del torneo.
Resumen:	El caso de uso se inicia cuando el árbitro selecciona la opción Cuadro Sinóptico del menú Torneo mostrándose éste con toda la información obtenida hasta el momento.
Precondiciones:	Debe haber terminado al menos una de las rondas del torneo.
Poscondiciones:	Actualización del cuadro sinóptico según se van realizando las rondas.
Referencias	RF 23
Prioridad	Crítico
Flujo Normal de Eventos	
Sección: "Mostrar Cuadro sinóptico para un torneo Round Robin individual "	
Acción del Actor	Respuesta del Sistema
1- El árbitro dentro de la página principal pulsa el botón Cuadro Sinóptico del Menú Torneo.	1.1- El sistema verifica que ya se haya realizado el sorteo, en caso positivo permite seleccionar la ronda que desea de todas las que se hayan realizado hasta el momento, así



	como la del torneo una vez finalizado.
2-El árbitro escoge la opción que necesita	<p>2.1 - El sistema muestra el cuadro sinóptico de la opción escogida:</p> <ul style="list-style-type: none"><li>-Los resultados de las partidas de cada jugador con cada uno de sus adversarios organizados por el lugar obtenido</li><li>-Lugar</li><li>- Elo promedio</li><li>-Total de puntos del jugador</li><li>- Por ciento de combatividad</li><li>- Categoría del torneo</li><li>-Sistema de desempate</li></ul> <p>Finalizando así el caso de uso.</p>
Flujo Alternativo de Eventos	
	<p>1.1- Si no se ha realizado el sorteo el sistema muestra el mensaje:</p> <p>“Antes debe realizarse el sorteo”.</p>
Sección: “Mostrar Cuadro sinóptico para un torneo Round Robin por equipos ”	
Acción del Actor	Respuesta del Sistema
1- El árbitro pulsa el botón Cuadro Sinóptico del Menú Torneo.	1.1- El sistema verifica que ya estén definidos los sistemas de desempate, en caso positivo permite seleccionar la ronda que desea de todas las que se hayan realizado hasta el



	momento, así como la del torneo una vez finalizado.
2-El árbitro escoge la opción que necesita	2.1 - El sistema muestra el cuadro sinóptico de la opción escogida:  -Resultado de la partida entre los equipos enfrentados organizados por el lugar obtenido  -Lugar  - Ranking  -Sistemas de desempate utilizado.  Finalizando así el caso de uso.
Flujo Alternativo de Eventos	
	1.1- Si no se han definido los sistemas de desempate muestra el mensaje:  “Antes debe definir los sistemas de desempate”.

Tabla 41: Descripción del CU Mostrar emparejamiento y resultados

Caso de Uso:	Mostrar emparejamiento y resultados
Actores:	Árbitro
Propósito:	Mostrar el pareo de los torneos que se desarrollan unido a los resultados que se van obteniendo
Resumen:	El caso de uso inicia cuando el árbitro selecciona la opción Mostrar Emparejamientos del menú Torneo, luego se muestra el emparejamiento por



	tableros de todas las rondas de un grupo de torneo.
Precondiciones:	Ya debe existir un torneo creado y deben haberse obtenido resultado de alguna de las partidas
Poscondiciones:	Se muestra la tabla de pareo unido a los resultados que se van obteniendo.
Referencias	RF 24
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El árbitro dentro de la página principal pulsa el botón Mostrar Emparejamientos del Menú Torneo.	1.1-El sistema muestra todas las rondas del torneo.
2- El árbitro selecciona la ronda de la cual quiere ver el emparejamiento con los resultados.	2.1-El sistema muestra en esa ronda qué integrantes de un equipo se enfrentaron a los de otro, con los resultados individuales obtenidos hasta el momento y con el color de las piezas que se enfrentaron.

Tabla 42: Descripción del CU Mostrar Clasificación General

Caso de Uso:	Mostrar Clasificación General
Actores:	Árbitro
Propósito:	Se realiza con el objetivo de ver la clasificación general de un torneo.



Resumen:	El caso de uso inicia cuando el árbitro selecciona la opción Clasificación General del menú Torneo, brindando la posibilidad de ver como quedan los lugares de los equipos en un torneo, unido a otros datos como el sistema de desempate utilizado.
Precondiciones:	
Poscondiciones:	Se muestra una tabla de los equipos con los resultados finales del torneo.
Referencias	RF 25
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El árbitro dentro de la página principal pulsa el botón Clasificación General del Menú Torneo.	1.1-El sistema verifica que el torneo haya finalizado, en caso positivo muestra una tabla con la clasificación de los equipos organizados por el lugar obtenido, mostrando de cada uno de ellos los siguientes datos:  -Lugar  -RK  - Nombre  - Federación  - Enfrentamientos  -puntos totales obtenidos por el equipo  -Desempates utilizados con la puntuación obtenida



	(ordenados por prioridad)
Flujo Alternativo de Eventos	
	1.1-Si el torneo no ha culminado, el sistema muestra un mensaje informando que el torneo se está desarrollando.

Tabla 43: Descripción del CU Gestionar equipos de un grupo

Caso de Uso:	Gestionar equipos de un grupo
Actores:	Árbitro
Propósito:	Permite al árbitro gestionar toda la información referente a los equipos de un grupo de un torneo Round Robin por equipos: adicionar y eliminar equipos de un grupo.
Resumen:	El caso de uso se inicia cuando el usuario selecciona dentro del Menú Torneo Gestionar equipos y el sistema le brinda la posibilidad de agregar o eliminar equipos a un grupo del torneo.
Precondiciones:	El torneo y los equipos deben estar creados.
Poscondiciones:	-Equipos registrados en un grupo del torneo. - Equipos eliminados de un grupo del torneo.
Referencias	RF 27, RF 27.1, RF 27.2
Prioridad	Crítico
Flujo Normal de Eventos	



Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona dentro de la página principal la opción Gestionar Equipos de un Grupo del Menú Torneo	1.1- El sistema brinda las opciones de: <ul style="list-style-type: none"><li>- Adicionar equipos a un torneo</li><li>- Eliminar equipos de un torneo</li></ul>
2 -El árbitro selecciona una de las opciones	2.1- Si selecciona “Adicionar equipos” ver sección Adicionar equipos a un grupo  -Si selecciona “Eliminar equipos” ver sección Eliminar equipos de un grupo
Sección: “Adicionar equipos a un grupo”	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Adicionar Equipo	1.1-Puede ejecutarse el caso de uso Buscar Equipo.
2- El árbitro selecciona él o los equipos deseados y pulsa el botón Aceptar	2.1- El sistema verifica que el o los equipos ya no estén registrados en el grupo de torneo al que se quieren adicionar, en caso positivo muestra el listado actualizado de los equipos de ese grupo del torneo finalizando así el caso de uso.
Flujo Alternativo de Eventos	
Sección: “Adicionar equipos a un grupo”	
	2.1- Si el equipo ya está asignado al grupo que se desea adicionar, el sistema muestra un mensaje para informar que el equipo ya se encuentra en ese grupo de torneo.



Sección: "Eliminar equipos de un grupo"	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Eliminar Equipo	1.1- El sistema muestra un listado con los equipos de ese grupo del torneo, brindando la posibilidad de seleccionar los que se desean eliminar.
2- El árbitro selecciona él o los equipos que desea y pulsa el botón Eliminar	2.1-El sistema verifica que el equipo que se desea eliminar no pertenezca a un torneo que todavía no haya finalizado, en caso positivo el sistema muestra un mensaje de confirmación para eliminar el equipo.
3- El árbitro pulsa el botón Aceptar del mensaje de confirmación.	3.1- El sistema verifica que el sorteo no esté realizado, en caso positivo elimina el equipo de la base de datos y muestra un mensaje de que el equipo ha sido eliminado, terminando así el caso de uso.
Flujo Alternativo de Eventos	
Sección: "Eliminar equipos de un grupo"	
	2.1- Si el torneo al que pertenece el equipo no ha finalizado el sistema muestra el mensaje:  "No se puede eliminar el equipo hasta no terminar el torneo".
3-El árbitro pulsa el botón Cancelar del mensaje de confirmación.	3.1- Si ya se ha realizado el sorteo, entonces el sistema muestra un mensaje informando que no se pueden eliminar estos elementos hasta no deshacer el sorteo.

Tabla 44: Descripción del CU Modificar Tableros





Caso de Uso:	Modificar Tableros
Actores:	Árbitro
Propósito:	Permite al árbitro modificar los tableros asignados a los jugadores.
Resumen:	El caso de uso se inicia cuando el árbitro selecciona dentro del Menú Torneo Modificar Tableros y el sistema le brinda la posibilidad de ver en que tablero se enfrentarán cada uno de los jugadores permitiendo su modificación.
Precondiciones:	Debe estar creado el torneo con su respectivo pareo.
Poscondiciones:	-Nueva asignación de tableros a los jugadores.
Referencias	RF 30
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El árbitro selecciona dentro de la página principal la opción Modificar Tableros del Menú Torneo	1.1-El sistema brinda la posibilidad de escoger el equipo al que se le va a hacer la modificación.
2- El árbitro selecciona el equipo deseado	2.1- El sistema muestra el listado de los jugadores del equipo escogido, cada uno con el número de tablero.
3- El árbitro realiza el cambio de número del tablero y pulsa el botón Aceptar	3.1- El sistema sortea los demás tableros para que no se repita su número, guarda los cambios en la base de datos finalizando así el caso de uso.

Tabla 45: Descripción del CU Definir Alineación



Caso de Uso:	Definir Alineación
Actores:	Árbitro
Propósito:	Se realiza con el objetivo de definir el número del tablero con el que jugará cada integrante de un equipo.
Resumen:	El caso de uso inicia cuando el árbitro selecciona la opción Definir Alineación del menú Torneo, posibilitando al árbitro decidir los jugadores que se enfrentarán de cada uno de los equipos y con qué color de piezas lo harán.
Precondiciones:	Ya debe existir un torneo creado y el pareo debe haberse realizado.
Poscondiciones:	Se muestra la alineación entre los equipos en cada una de las rondas.
Referencias	RF 31
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El árbitro dentro de la página principal pulsa el botón Definir Alineación del Menú Torneo.	1.1-El sistema muestra cada una de las rondas del torneo.
2- El árbitro selecciona la ronda deseada para definir la alineación de los equipos	2.1-El sistema muestra en esa ronda los equipos que se enfrentarán.
3-El árbitro selecciona el enfrentamiento de los equipos a los cuáles se les desea definir la alineación.	3.1-El sistema muestra una tabla donde aparecen los integrantes de cada uno de los equipos.
4- El árbitro registra los jugadores que se	4.1-El sistema verifica que no existan jugadores



enfrentarán y con el color de las piezas que realizarán la partida.	repetidos, en caso positivo registra la alineación en la base de datos, finalizando así el caso de uso.
Flujo Alternativo de Eventos	
	4.1-Si existen jugadores repetidos el sistema muestra un mensaje informando que existen jugadores repetidos en un equipo.

Tabla 46: Descripción del CU Mostrar clasificación individual por tableros

Caso de Uso:	Mostrar clasificación individual por tableros
Actores:	Árbitro
Propósito:	Brindar la posibilidad al árbitro de tener registrado por tableros los jugadores que mejor resultado obtuvieron.
Resumen:	El caso de uso se inicia cuando el árbitro selecciona la opción Clasificación Individual del menú Torneo mostrándose los jugadores de mayor puntuación en cada tablero.
Precondiciones:	Debe haberse culminado alguna de las partidas en ese tablero.
Poscondiciones:	Clasificación Individual por tableros.
Referencias	RF 32
Prioridad	Crítico
Flujo Normal de Eventos	
Sección:	



Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona dentro de la página principal la opción Mostrar clasificación individual por tableros del Menú Torneo	1.2- El sistema muestra los tableros presentes en el torneo.
2- El árbitro selecciona el tablero del que desea ver los jugadores con mayor puntuación.	2.1- El sistema muestra una tabla con los siguientes datos del tablero seleccionado:  -RK  - Título  - Nombre del jugador  - Federación  - Juegos  -Resultado  - Puntos
Flujo Alternativo de Eventos	
	2.1-Si no se ha culminado ninguna partida en ese tablero los campos aparecen vacíos.

Tabla 47: Descripción del CU Consultar Jugador

Caso de Uso:	Consultar Jugador
--------------	-------------------



Actores:	Invitado
Propósito:	Permite al invitado ver los datos de un jugador.
Resumen:	El caso de uso se inicia cuando el invitado visita la plataforma y dentro del Menú Consultar Jugador le brinda la posibilidad de consultar información de algún jugador.
Precondiciones:	El sistema debe estar disponible.
Poscondiciones:	Información del jugador mostrada al Invitado.
Referencias	RF 33
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El Invitado dentro de la página principal accede a la opción Consultar Jugador.	1.1- El sistema muestra los siguientes criterios de búsqueda :  -Nombre  -Titulo  -Federación  -Elo
2- El Invitado introduce los criterios de búsqueda especificados y pulsa el botón Aceptar.	2.1-El sistema muestra un listado de los jugadores que responden al criterio de búsqueda introducido.
3- El Invitado selecciona el jugador del que desea	3.1-El sistema accede a la Sección "Ver Jugador"



información.	del CU Gestionar Jugador. Finalizando así el caso de uso.
--------------	---

Tabla 48: Descripción del CU Gestionar Imagen

Caso de Uso:	Gestionar Imagen
Actores:	Administrador
Propósito:	Permite al administrador gestionar las imágenes de un torneo, posibilitando la opción de adicionar o eliminar imágenes.
Resumen:	El caso de uso se inicia cuando el árbitro dentro del Menú Torneo, en Gestionar Imagen, el sistema le brinda la posibilidad de adicionar o eliminar imágenes del torneo.
Precondiciones:	El torneo debe estar creado.
Poscondiciones:	-Imágenes de la inauguración del torneo -Imágenes de la clausura del torneo -Imágenes de cada una de las rondas del torneo
Referencias	RF 35, RF 35.1, RF 35.2
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El árbitro dentro de la página principal selecciona la opción Gestionar imagen	1.1- El sistema brinda las opciones de:



del Menú Torneo	- Adicionar Imagen  - Eliminar Imagen
2 -El árbitro selecciona una de las opciones	2.1- Si selecciona “Adicionar Imagen ” ver sección Adicionar Imagen  -Si selecciona “Eliminar Imagen ” ver sección Eliminar Imagen
Sección: “Adicionar Imagen ”	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Adicionar Imagen	1.1-El sistema brinda la posibilidad de subir las imágenes deseadas.
2- El árbitro carga las imágenes y pulsa el botón Aceptar	2.1-El sistema verifica si la imagen ya está registrada
	2.2- El sistema verifica que la imagen cumpla con los requisitos estándares de tamaño y calidad, en caso positivo guarda la imagen en la base de datos. Termina así el caso de uso.
Flujo Alternativo de Eventos	
Sección: “Adicionar Imagen ”	
	2.1-Si la imagen ya está registrada en la base de datos el sistema muestra un mensaje informando la situación
	2.2- Si la imagen no cumple los parámetros muestra un mensaje informando que debe modificar la imagen para



	poder insertarla.
Sección: "Eliminar Imagen "	
Acción del Actor	Respuesta del Sistema
1- El árbitro selecciona la opción de Eliminar Imagen	1.1- El sistema muestra la galería de imágenes existentes del torneo
2- El árbitro selecciona la imagen que desea eliminar y pulsa el botón Eliminar	2.1-el sistema muestra un mensaje de confirmación
3- El árbitro pulsa el botón Aceptar del mensaje de confirmación.	3.1 El sistema actualiza la galería de imágenes.
Flujo Alternativo de Eventos	
Sección: "Eliminar Imagen "	
3- El árbitro pulsa el botón Cancelar del mensaje de confirmación.	





Tabla 49: Descripción del CU Mostrar Ritmo de Juego

Caso de Uso:	Mostrar Ritmo de Juego	
Actores:	Árbitro	
Propósito:	Permite al árbitro consultar información referente a los ritmos de juego.	
Resumen:	El caso de uso se inicia cuando el árbitro dentro del Menú de Arbitraje tiene la opción de consultar información de un ritmo de juego.	
Precondiciones:		
Poscondiciones:	Listado de los ritmos de juego registrados.	
Referencias	RF 9	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1- El árbitro selecciona dentro de la página principal la opción de Mostrar Ritmo de Juego	1.1- El sistema muestra un listado de todos los ritmos de juego registrados.	
2-El árbitro selecciona el ritmo de juego que desea consultar, terminando así el caso de uso.		



Tabla 50: Descripción del CU Buscar Equipo

Caso de Uso:	Buscar Equipo
Actores:	CU Gestionar equipos de un grupo
Propósito:	Permite al árbitro buscar un equipo.
Resumen:	Puede iniciarse cuando se ejecuta el caso de uso Gestionar equipos de un grupo.
Precondiciones:	El torneo debe estar creado.
Poscondiciones:	Listado de los equipos.
Referencias	RF 28
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1-El sistema muestra un formulario con los siguientes criterios de búsqueda:  - Nombre  - Federación
2-El árbitro busca el equipo que necesita según los criterios de búsqueda.	2.1- El sistema muestra un listado de los equipos según el criterio de búsqueda especificado, finalizando así el caso de uso.



Flujo Alternativo de Eventos	
	2.1- Si no existen equipos con los criterios de búsqueda especificados se muestra un mensaje informando que no hay resultados para esa búsqueda.

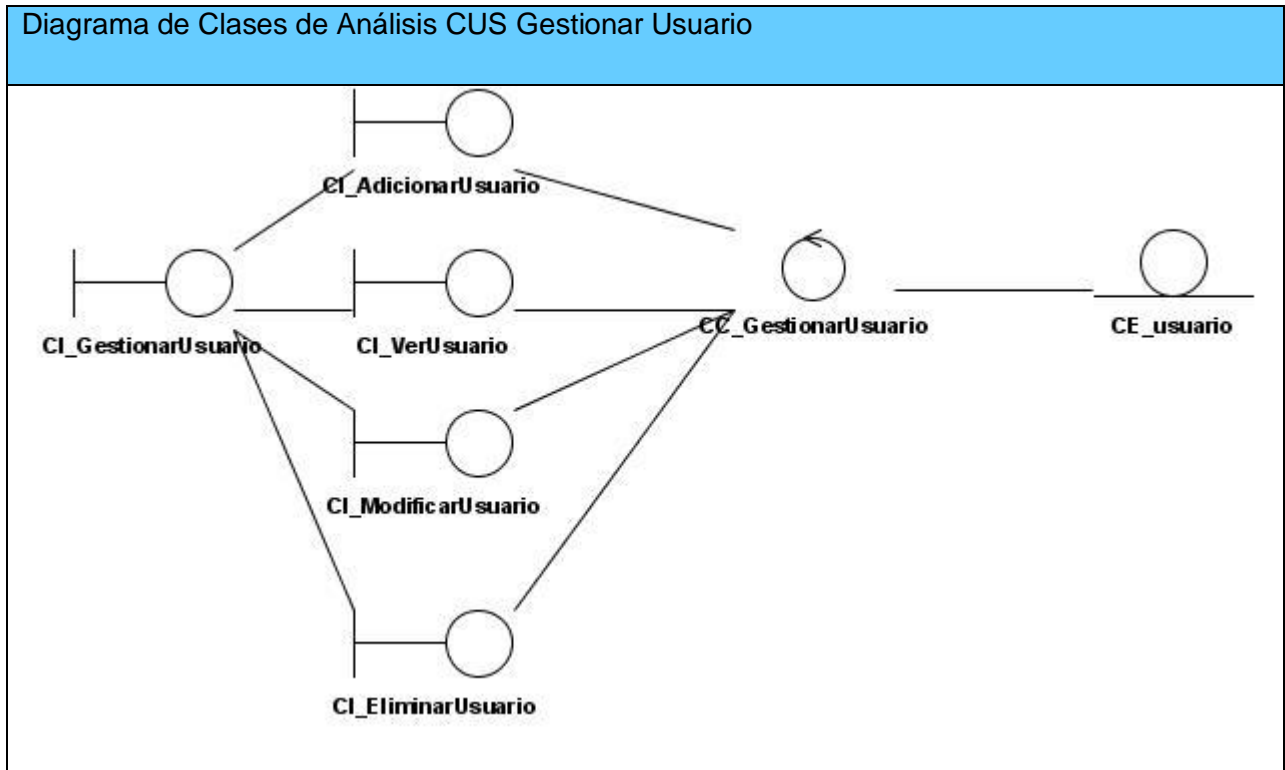
**Anexo 5: Diagrama de clases del Análisis**

**Figura 27:** CU Autenticar Usuario



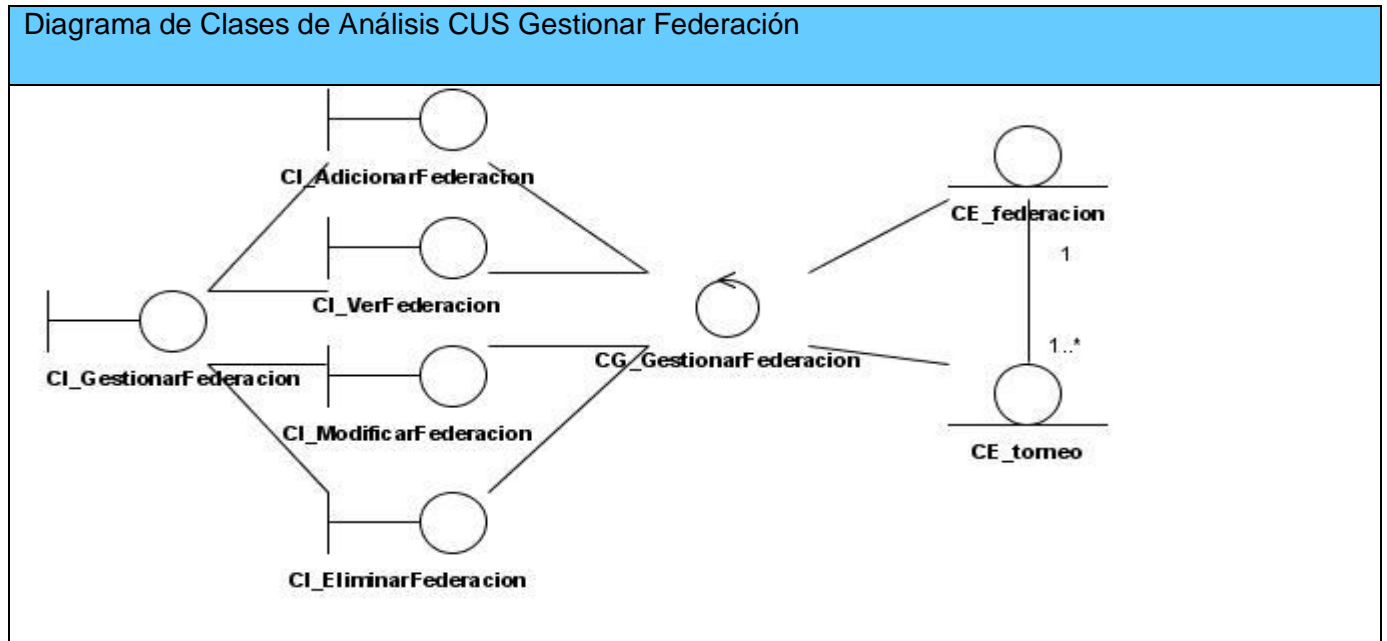


**Figura 28:** CU Gestionar Usuario

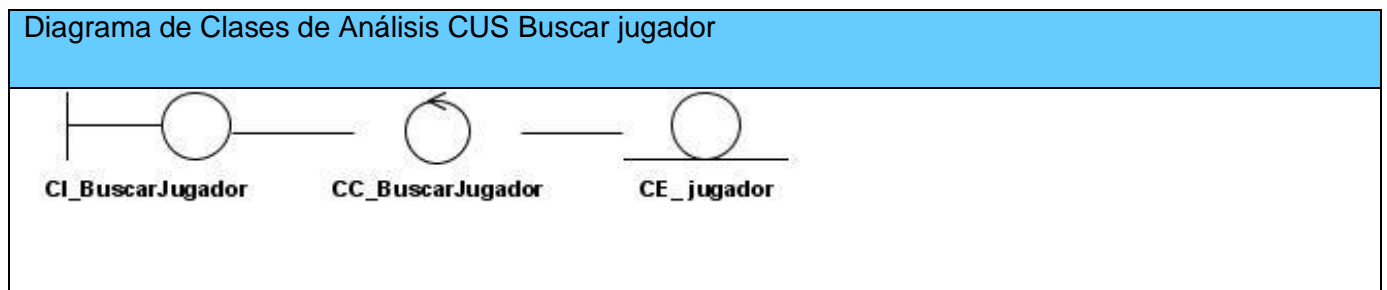




**Figura 29:** CU Gestionar Federación

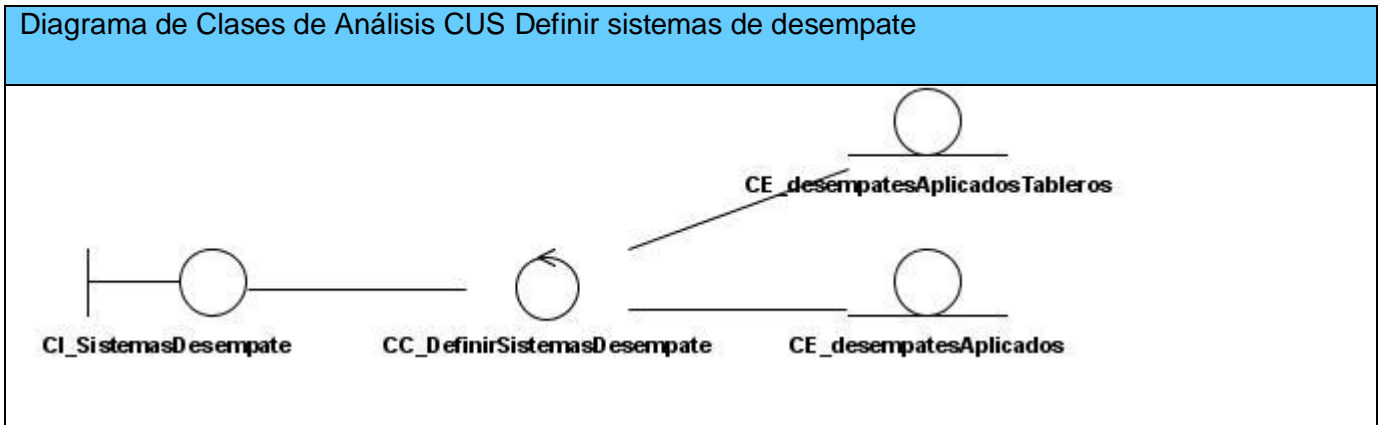


**Figura 30:** CU Buscar jugador

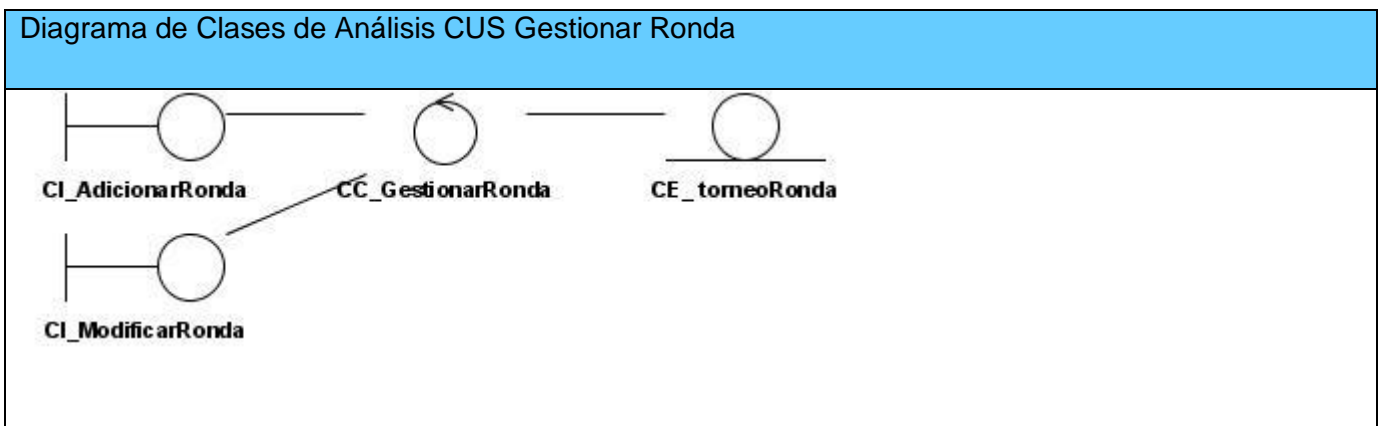




**Figura 31:** CU Definir sistemas de desempate



**Figura 32:** CU Gestionar Ronda





**Figura 33:** CU Gestionar Partida

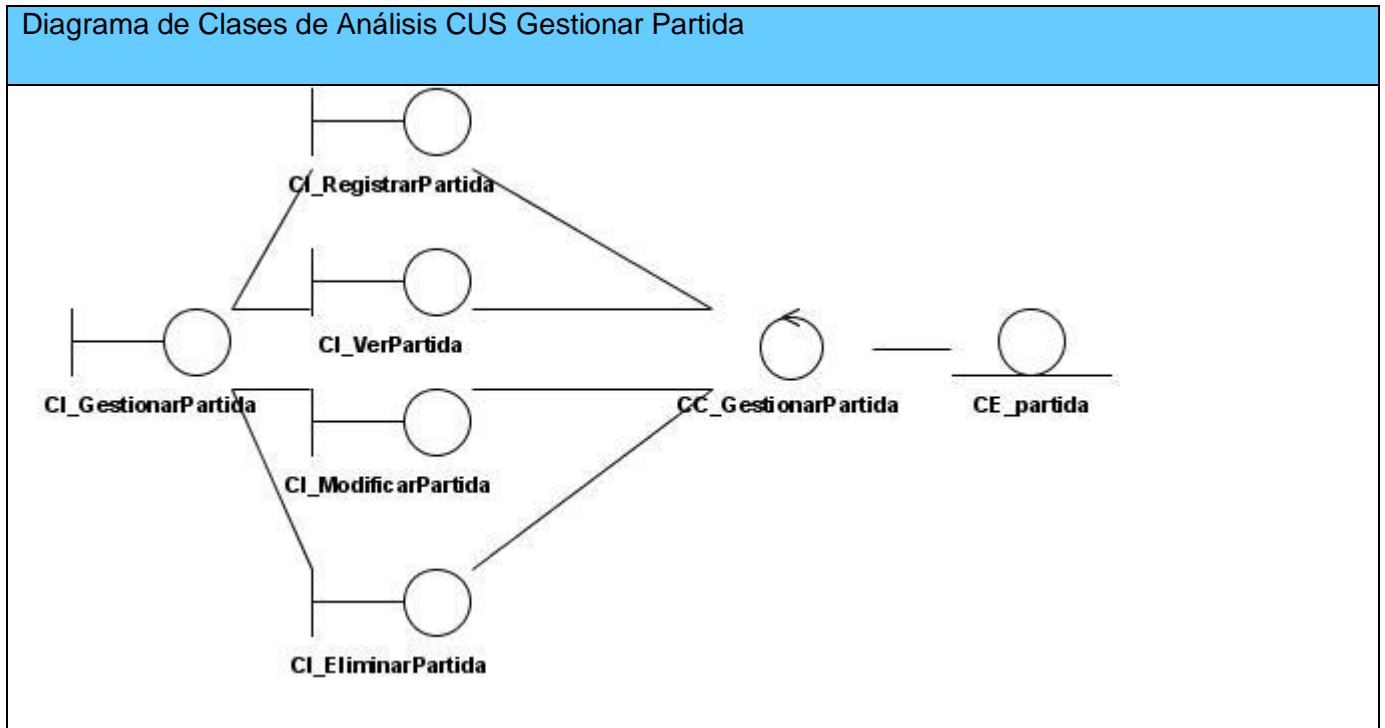




Figura 34: CU Gestionar Grupo

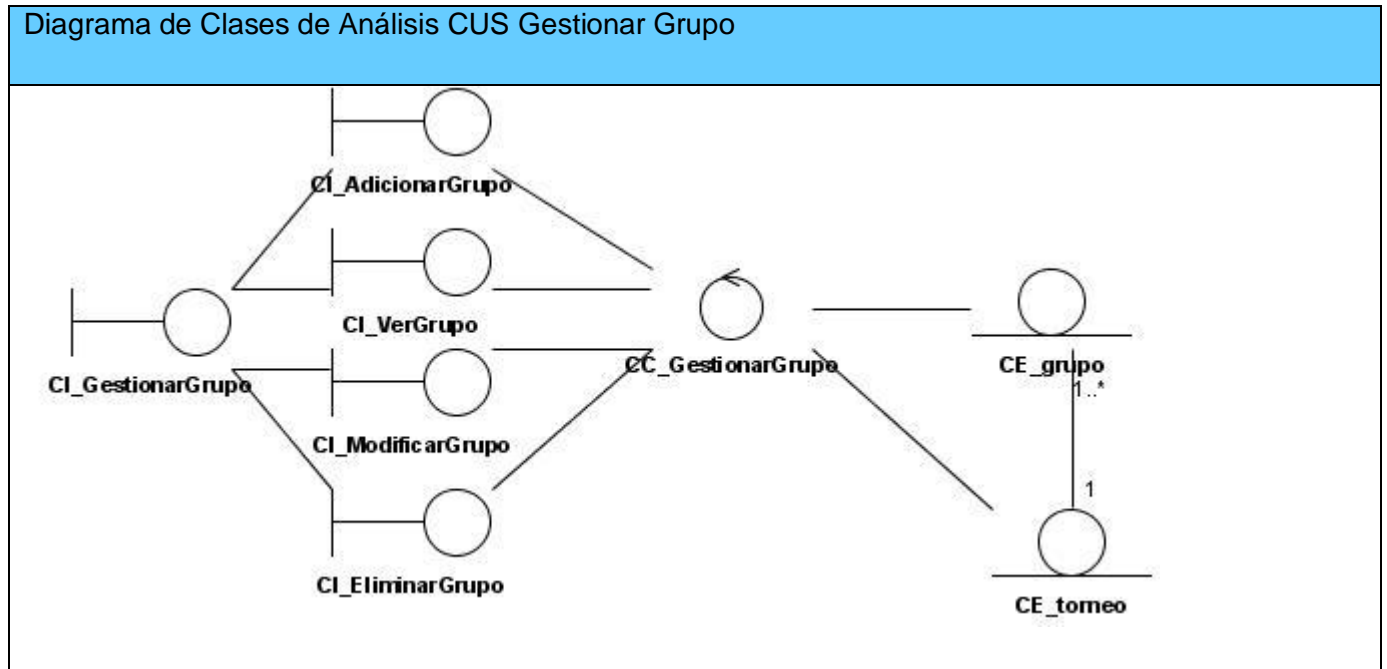
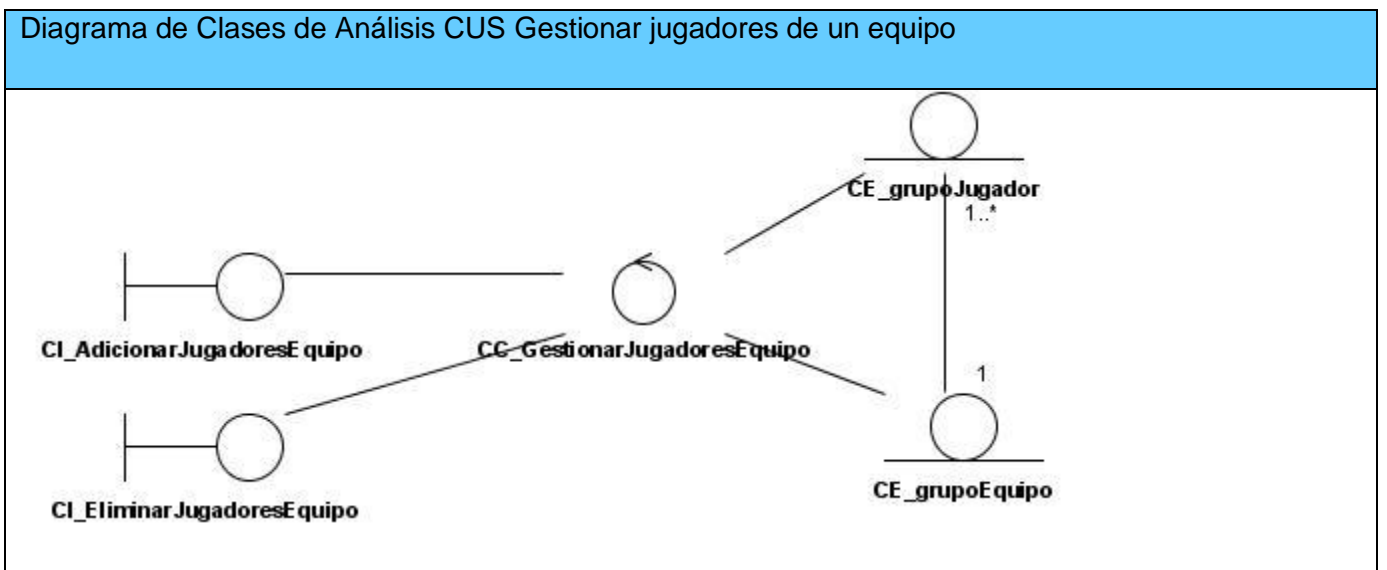


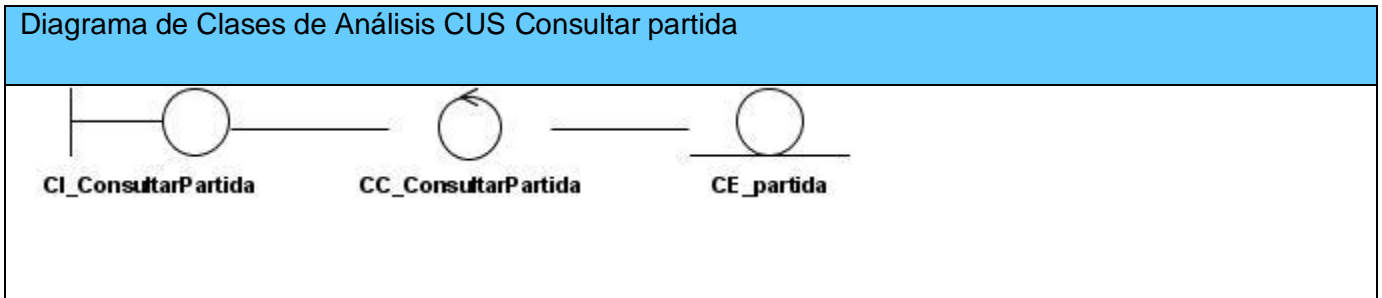
Figura 35: CU Gestionar jugadores de un equipo



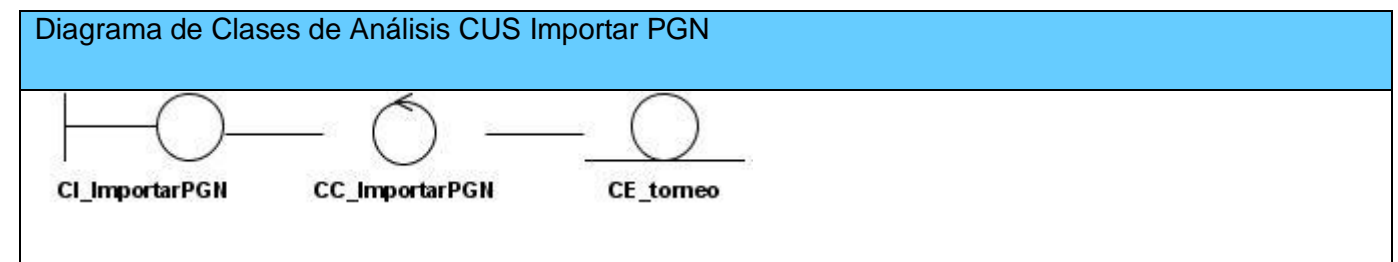




**Figura 36:** CU Consultar partida

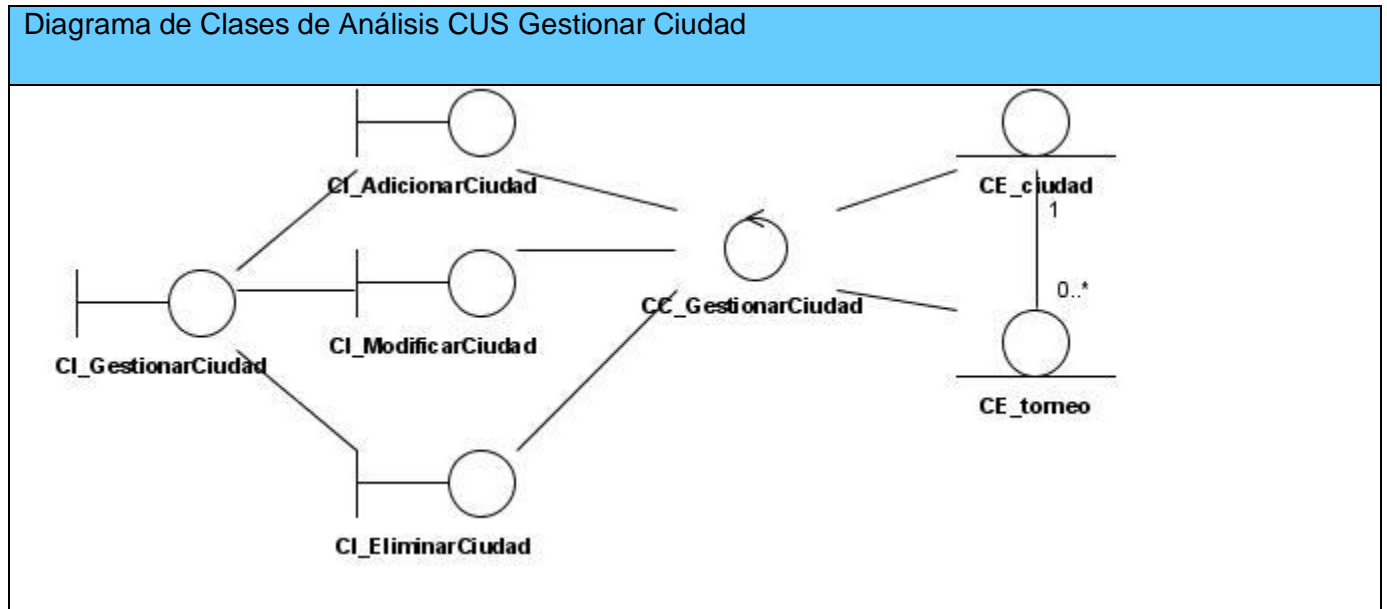


**Figura 37:** CU Importar PGN

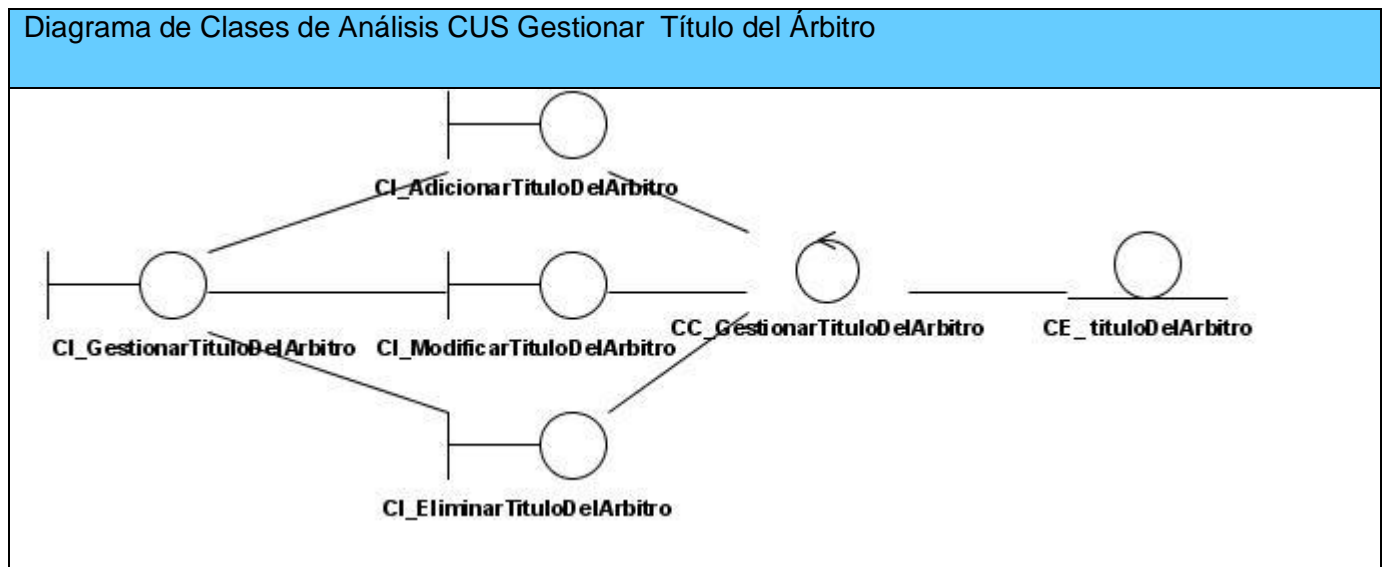




**Figura 38:** CU Gestionar Ciudad

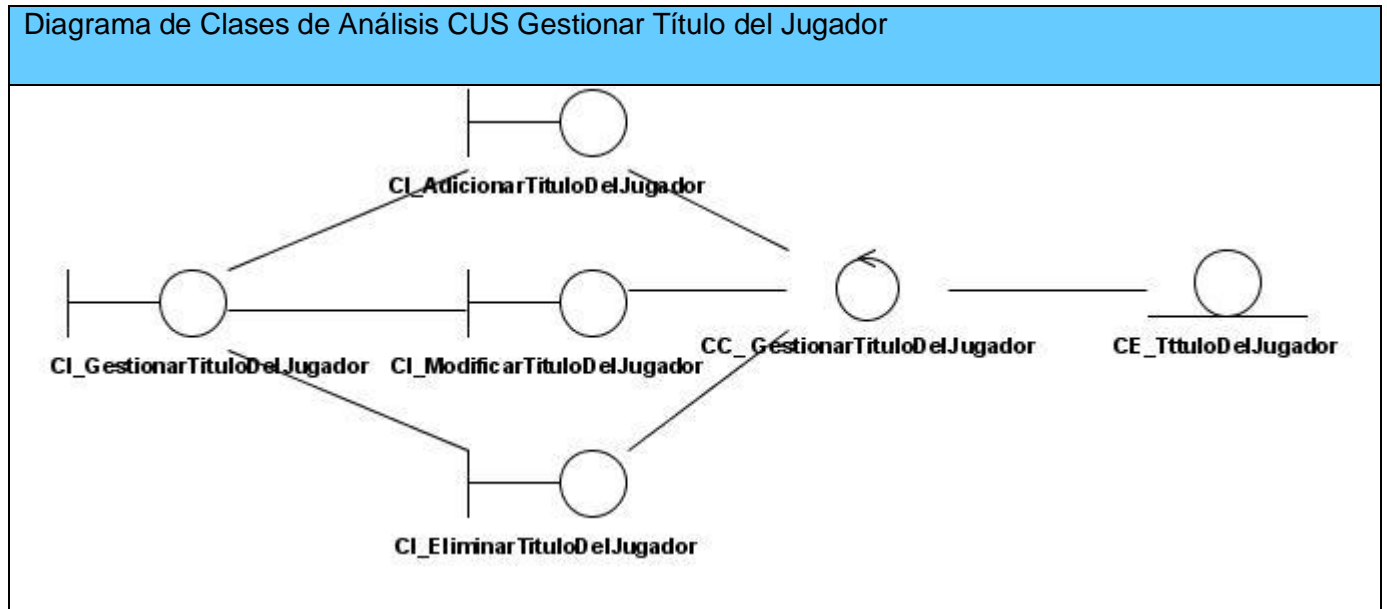


**Figura 39:** CU Gestionar Título del Árbitro

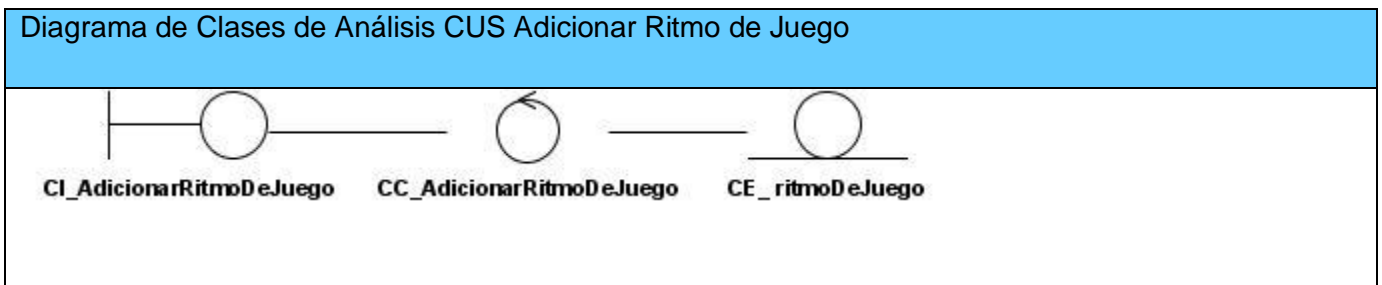




**Figura 40:** CU Gestionar Título del Jugador

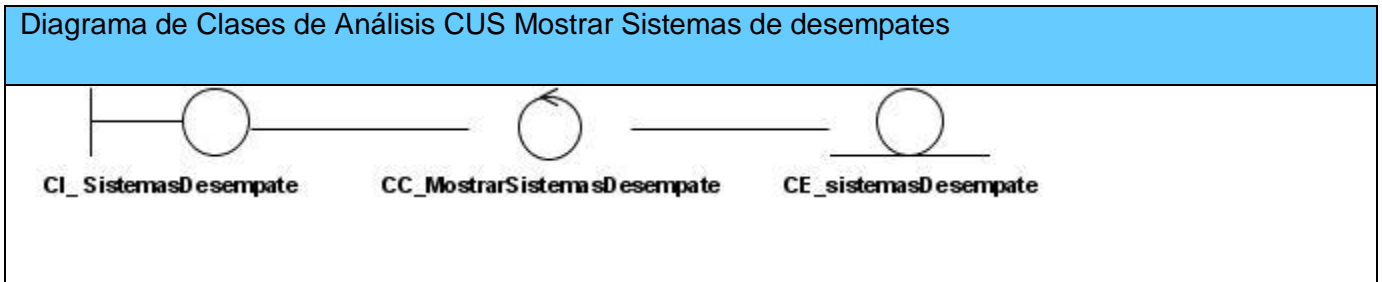


**Figura 41:** CU Adicionar Ritmo de Juego

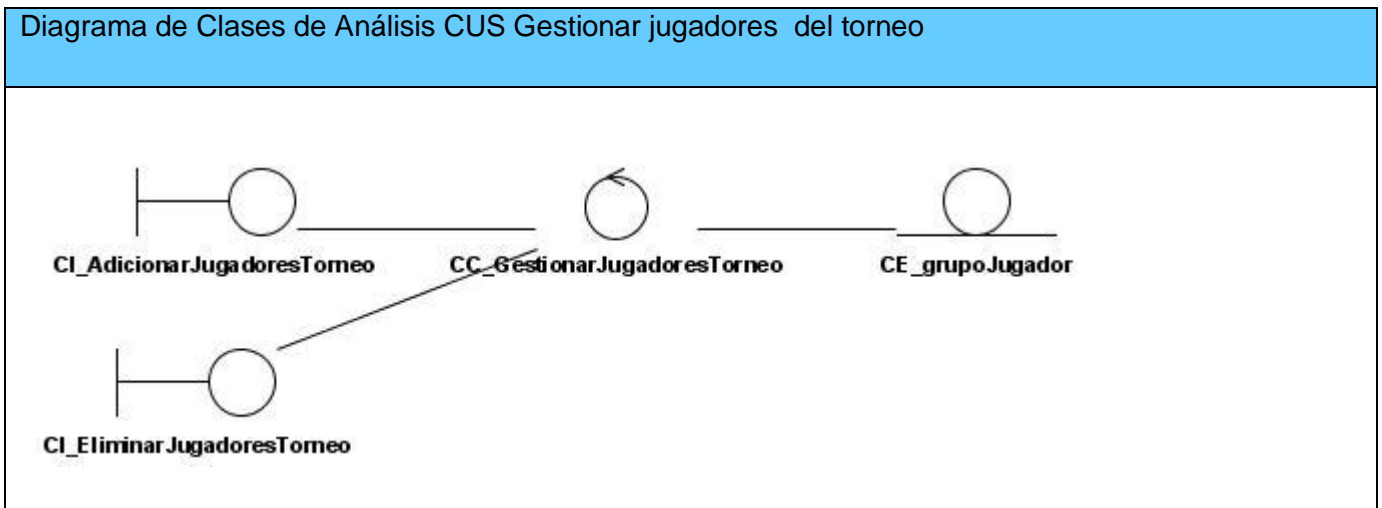




**Figura 42:** CU Mostrar Sistemas de desempates



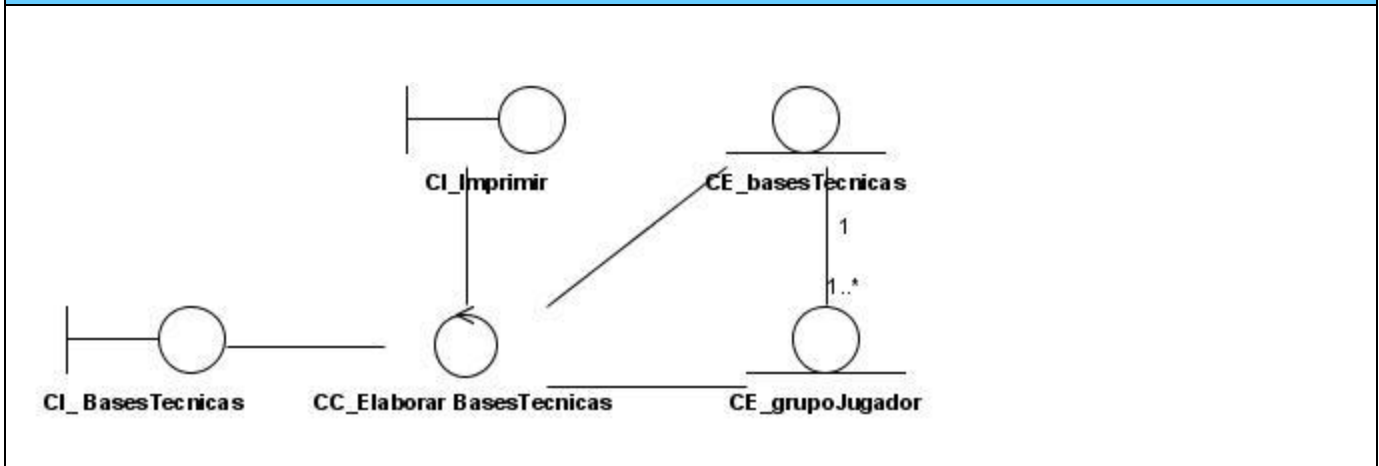
**Figura 43:** CU Gestionar jugadores del torneo





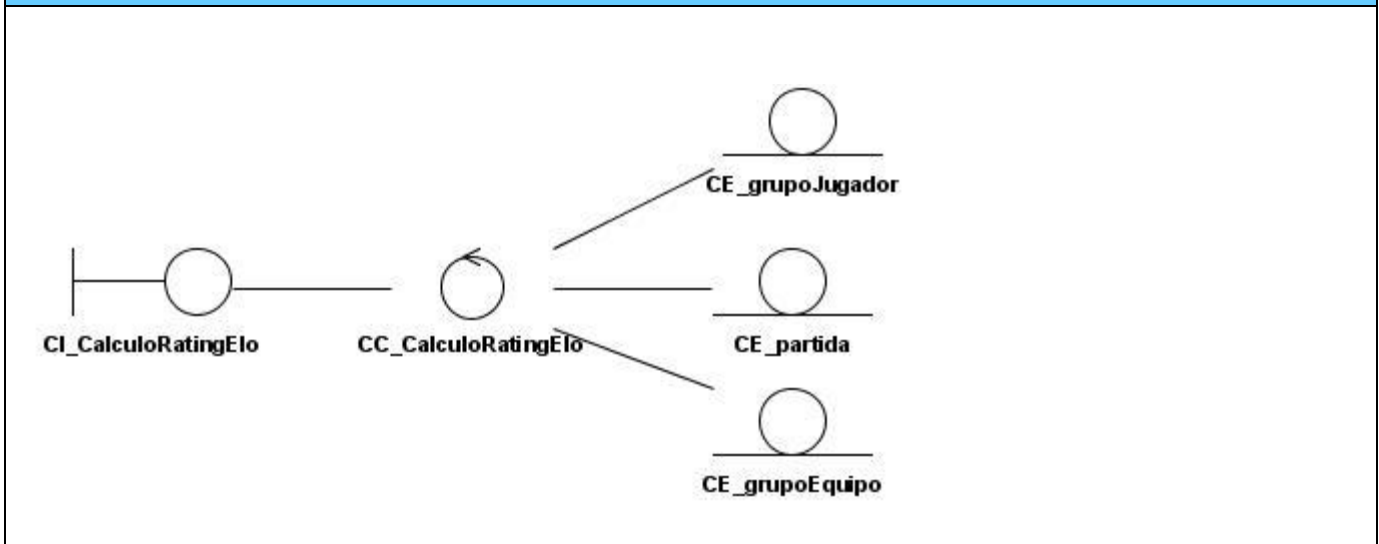
**Figura 44:** CU Elaborar Bases técnicas

Diagrama de Clases de Análisis CUS Elaborar Bases técnicas



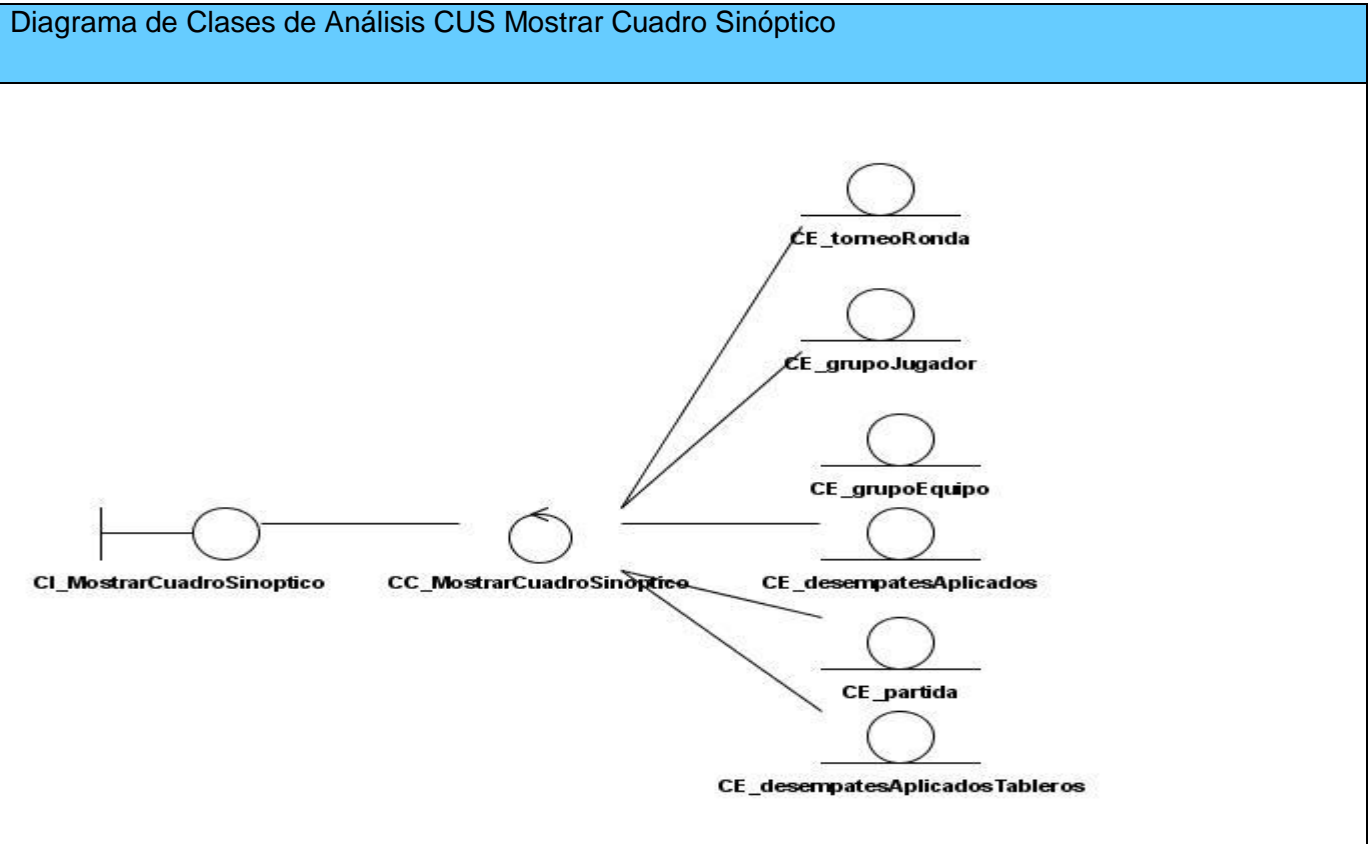
**Figura 45:** CU Cálculo del Rating Elo

Diagrama de Clases de Análisis CUS Cálculo del Rating Elo



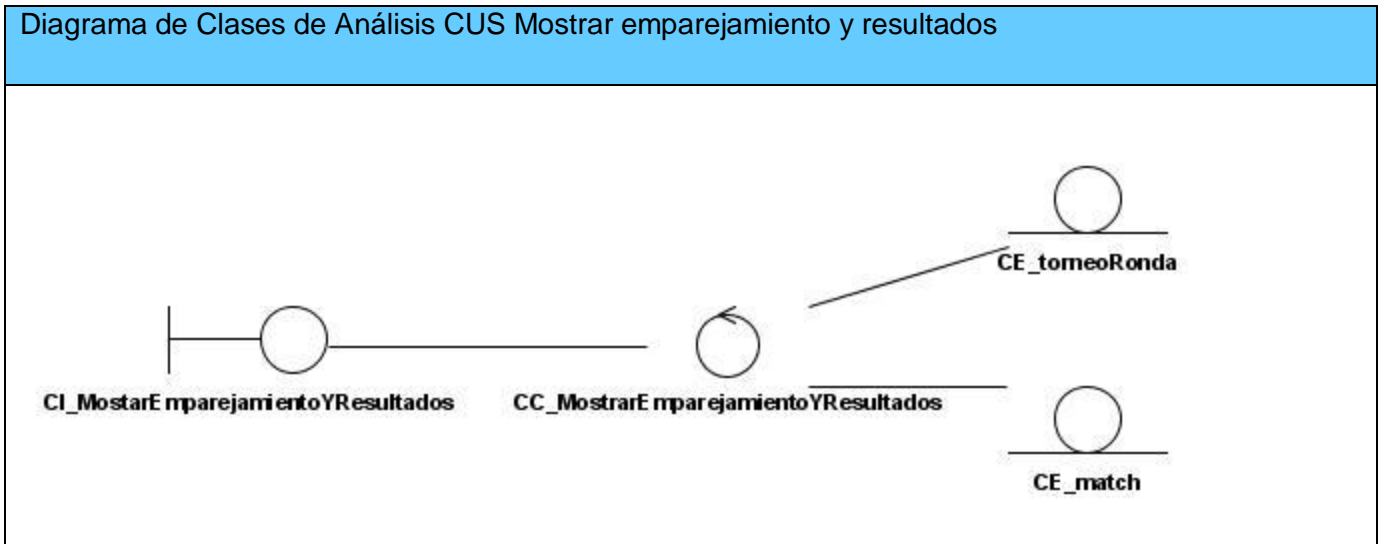


**Figura 46:** CU Mostrar Cuadro Sinóptico

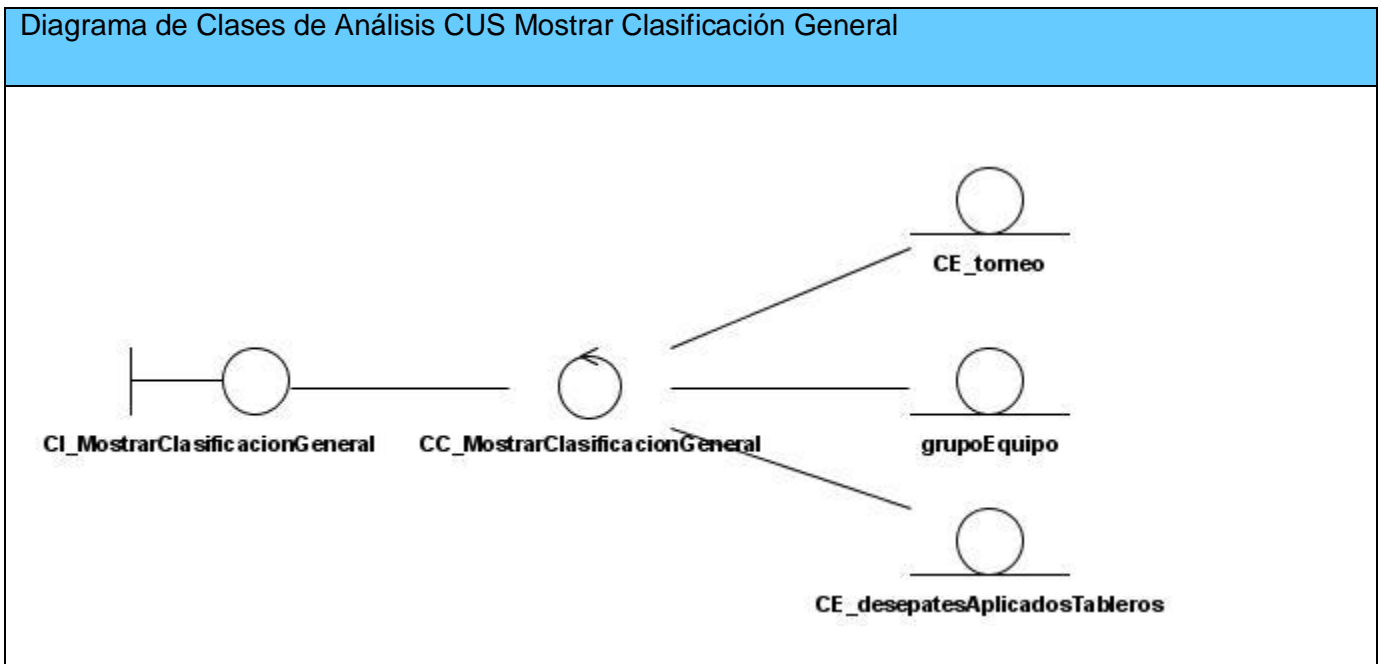




**Figura 47:** CU Mostrar emparejamiento y resultados

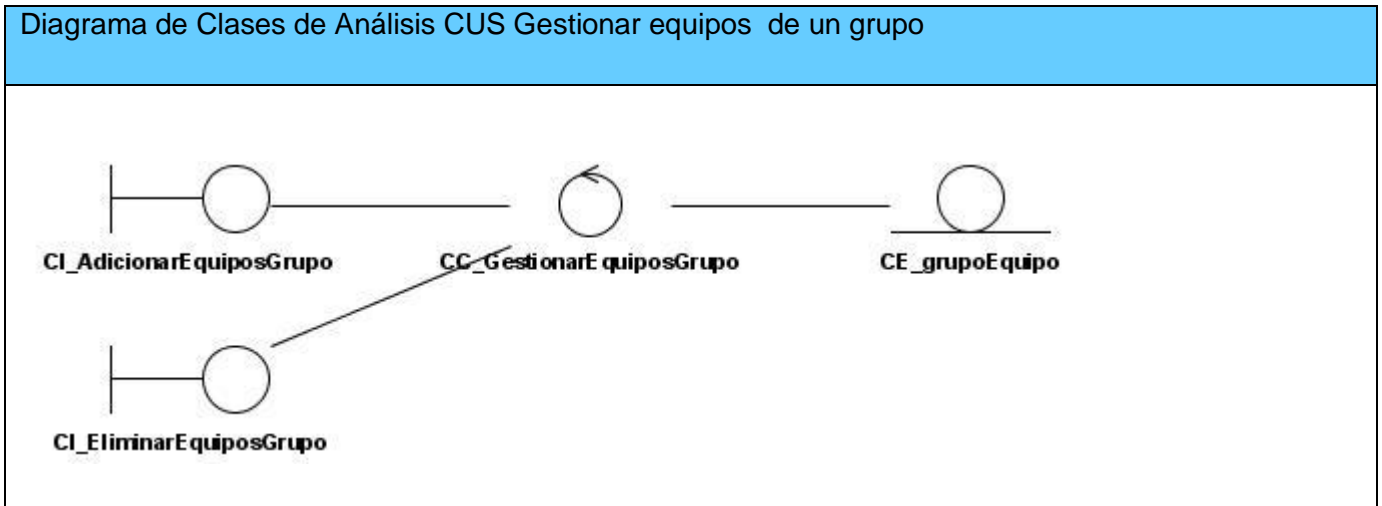


**Figura 48:** CU Mostrar Clasificación General

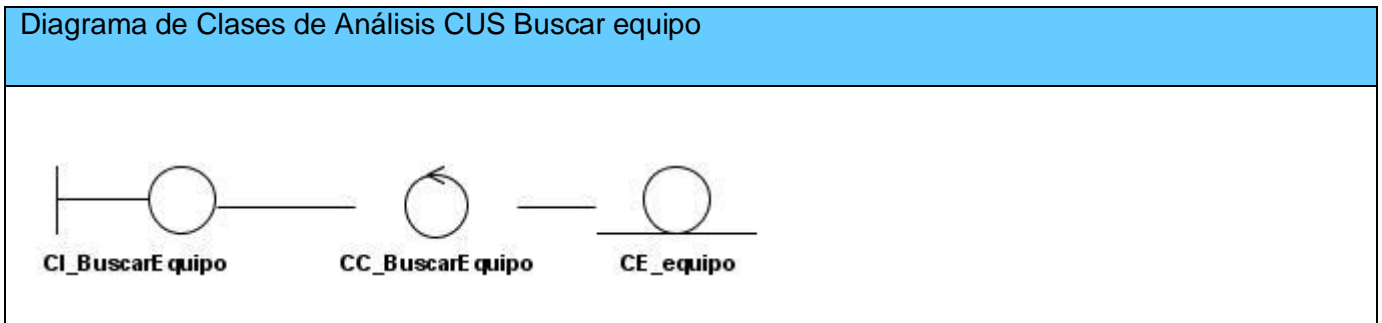




**Figura 49:** CU Gestionar equipos de un grupo



**Figura 50:** CU Buscar equipo

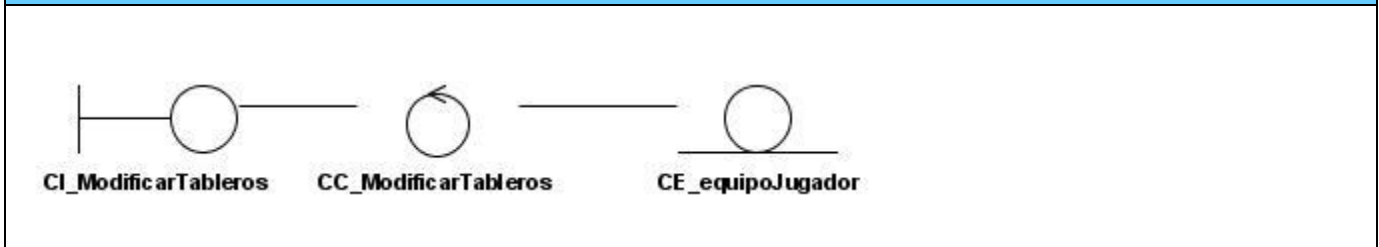






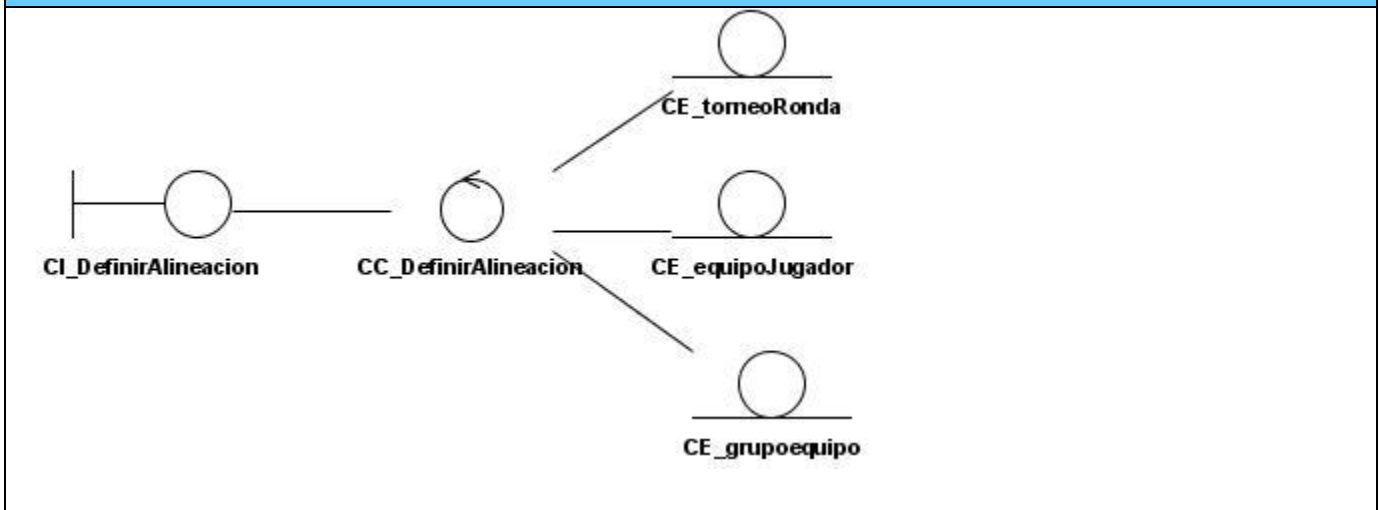
**Figura 51:** CU Modificar tableros

Diagrama de Clases de Análisis CUS Modificar tableros



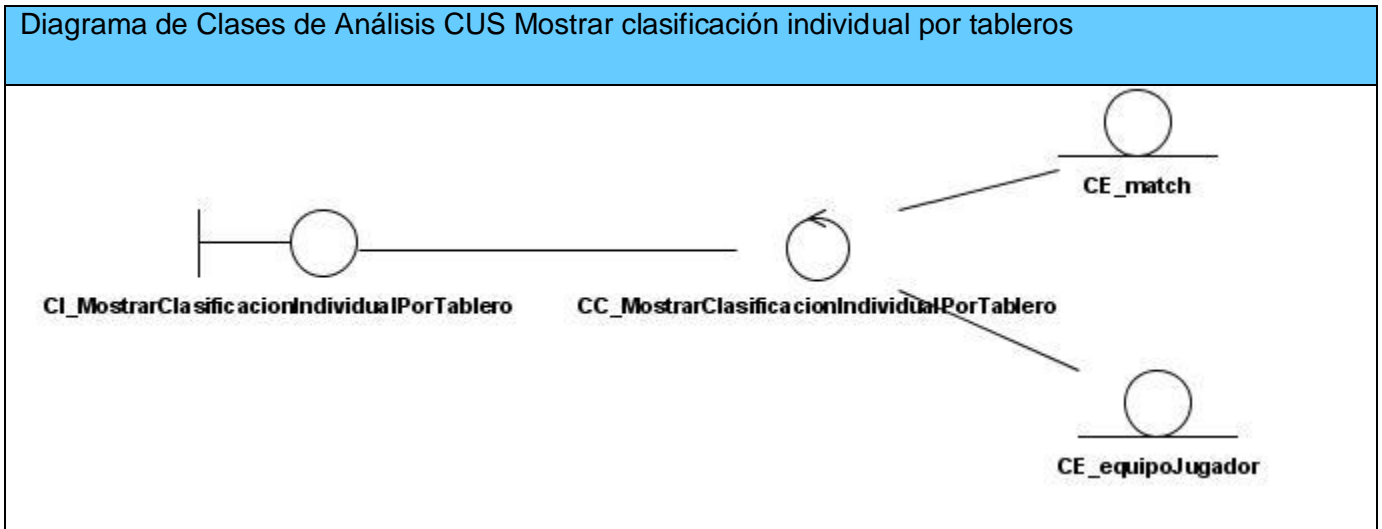
**Figura 52:** CU Definir Alineación

Diagrama de Clases de Análisis CUS Definir Alineación

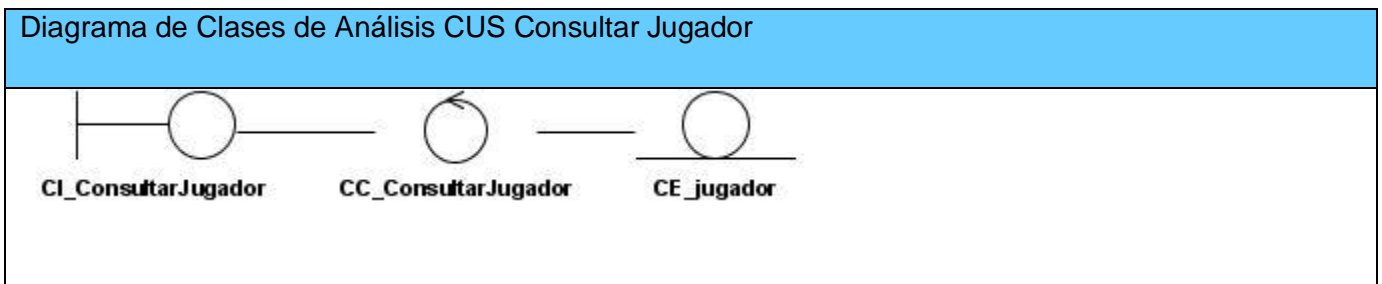




**Figura 53:** CU Mostrar clasificación individual por tableros

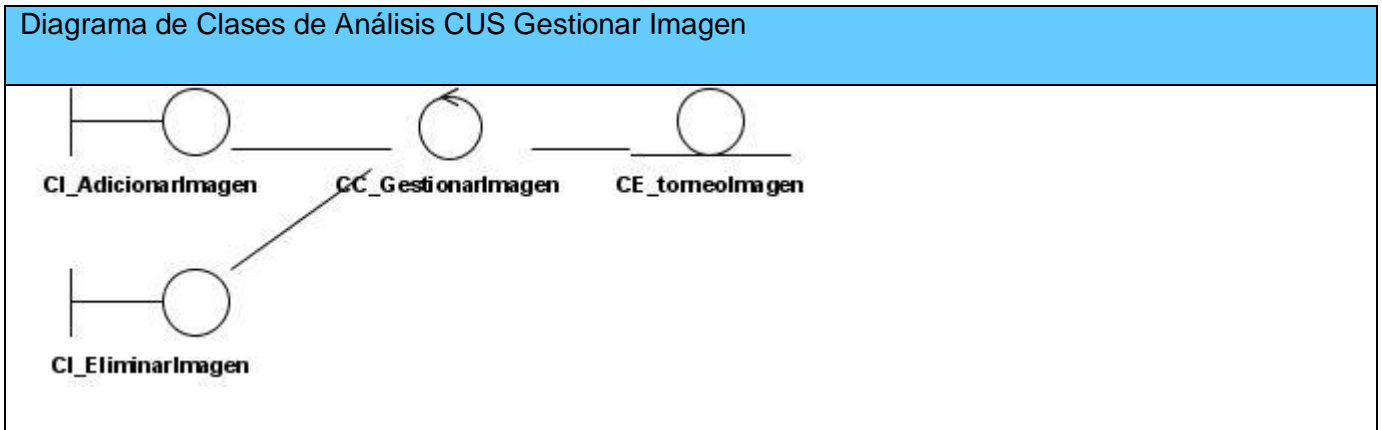


**Figura 54:** CU Consultar Jugador





**Figura 55:** CU Gestionar Imagen



**Anexo 6: Diagramas de colaboración del análisis**

**Figura 56:** Autenticar Usuario

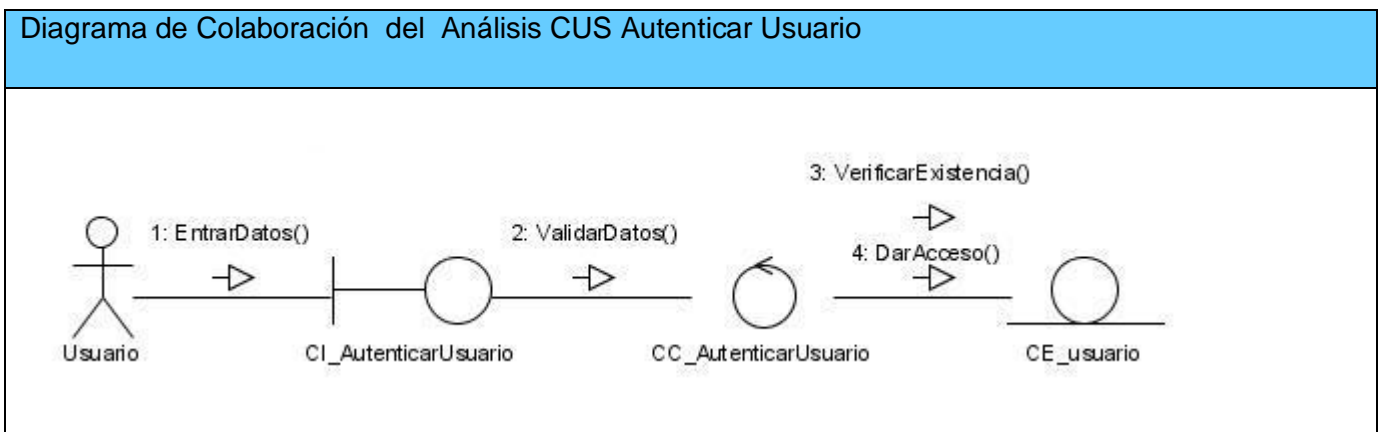




Figura 57: CU Gestionar Usuario

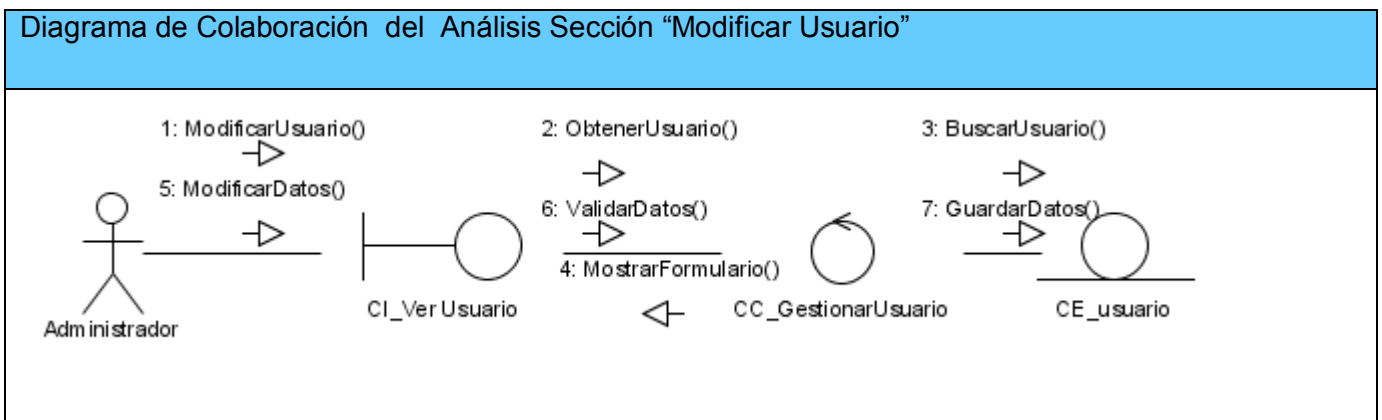
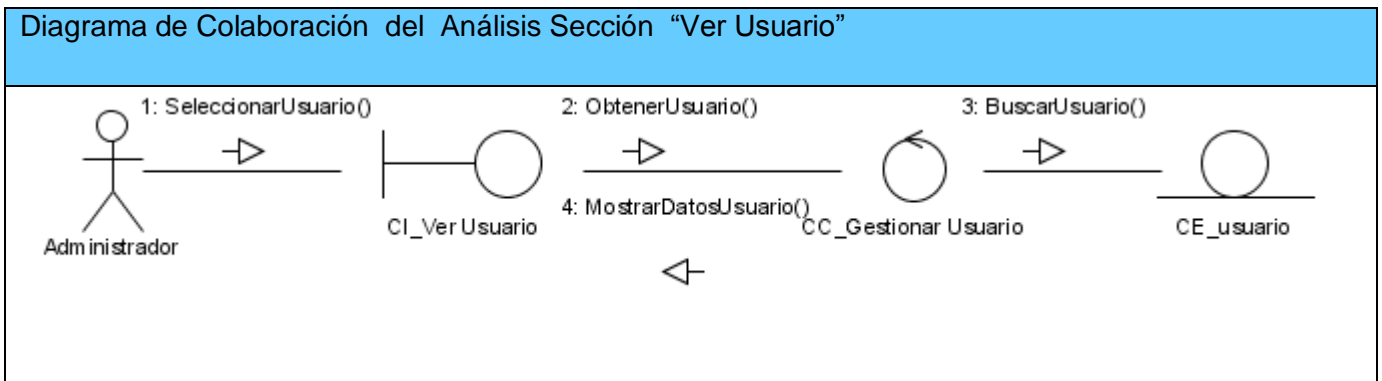
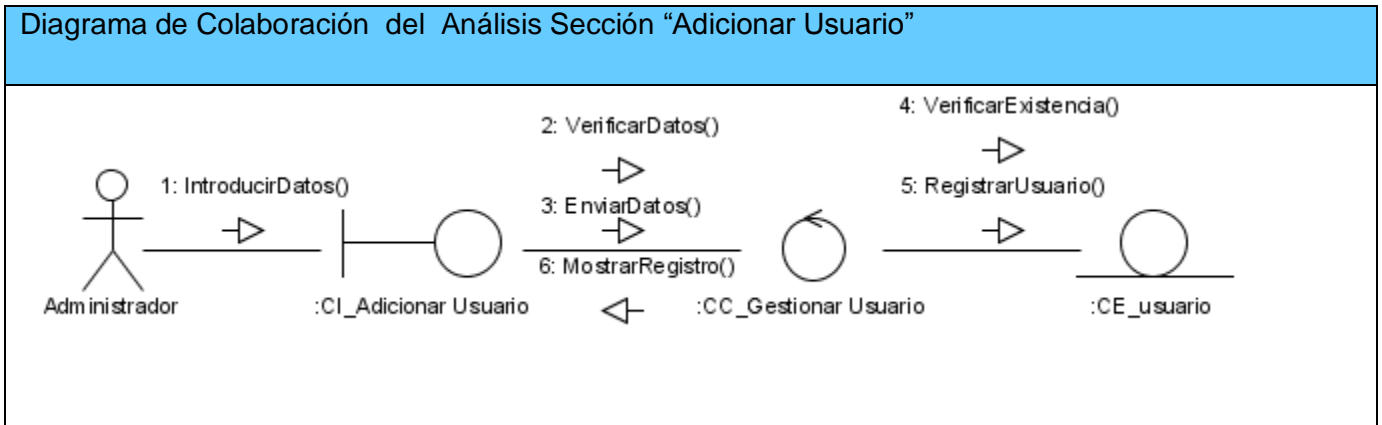




Diagrama de Colaboración del Análisis Sección “Eliminar Usuario”

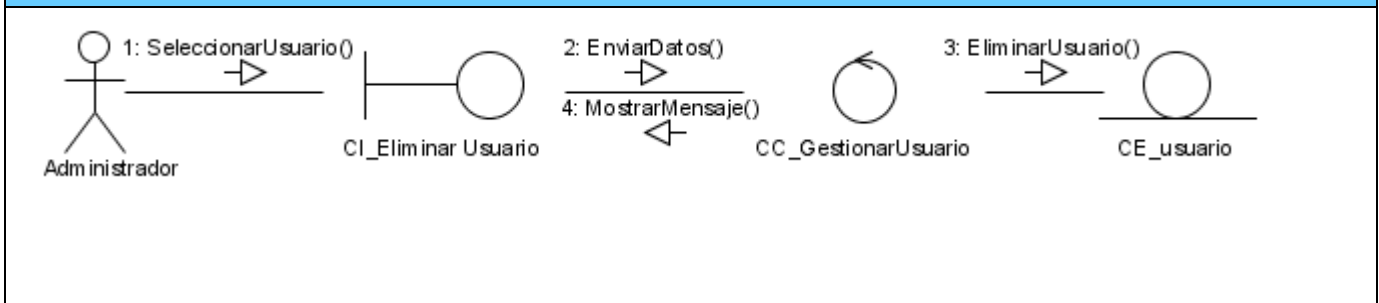


Figura 58: CU Gestionar Federación

Diagrama de Colaboración del Análisis Sección “Adicionar Federación”

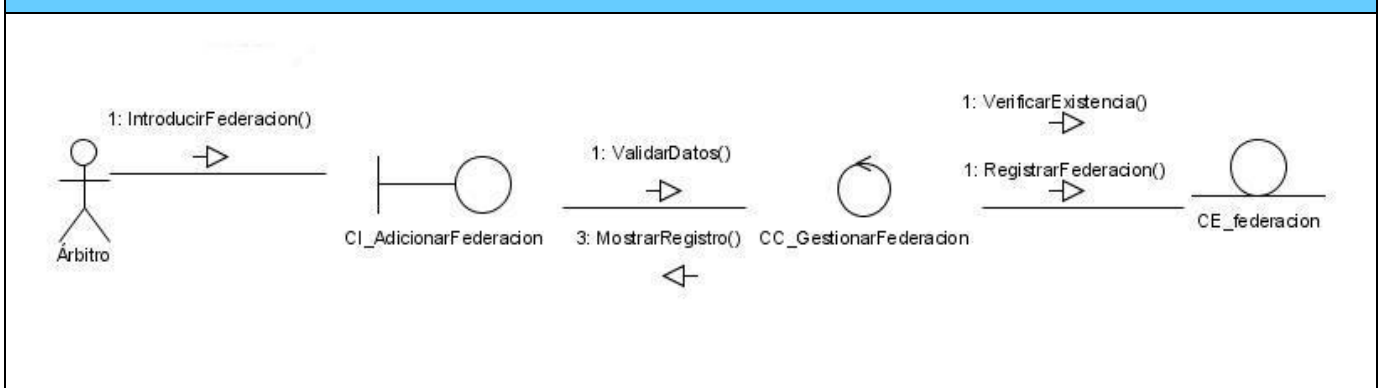




Diagrama de Colaboración del Análisis Sección “Ver Federación”

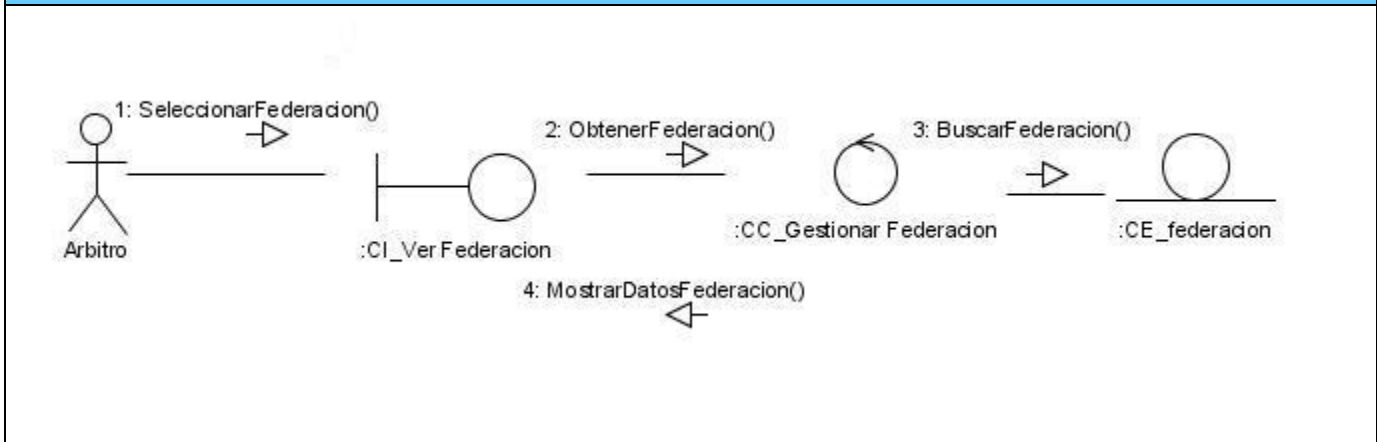


Diagrama de Colaboración del Análisis Sección “Modificar Federación”

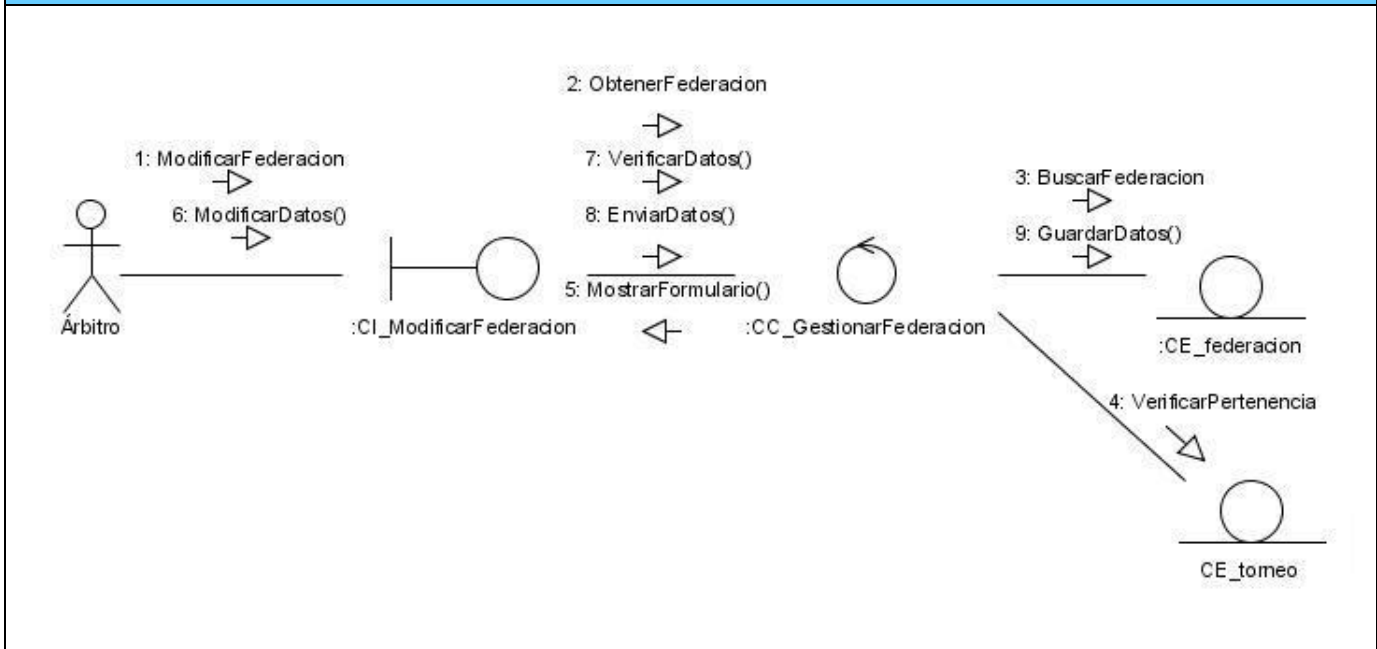




Diagrama de Colaboración del Análisis Sección “Eliminar Federación”

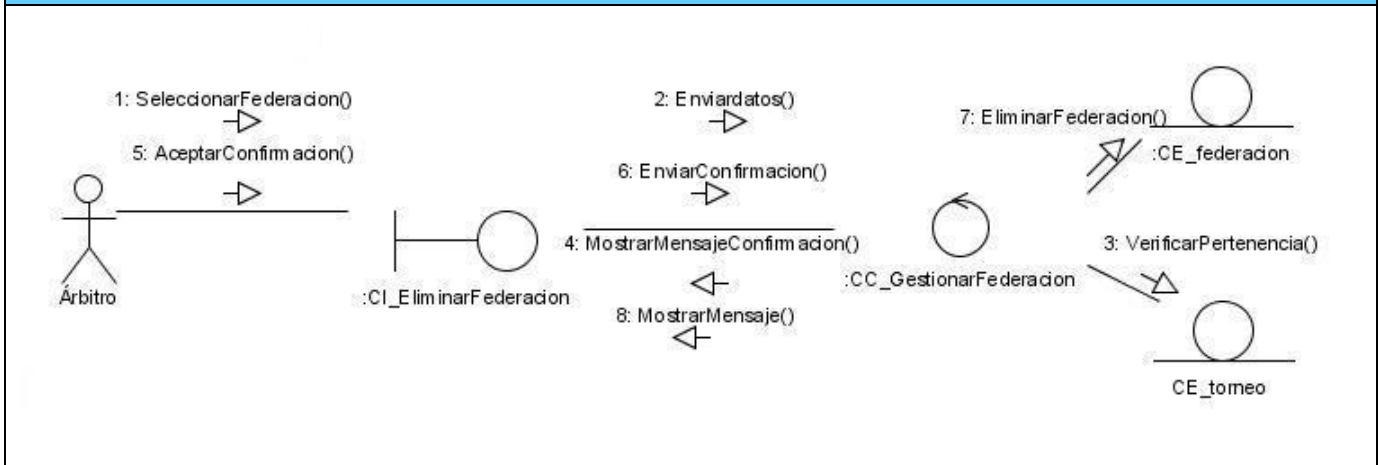
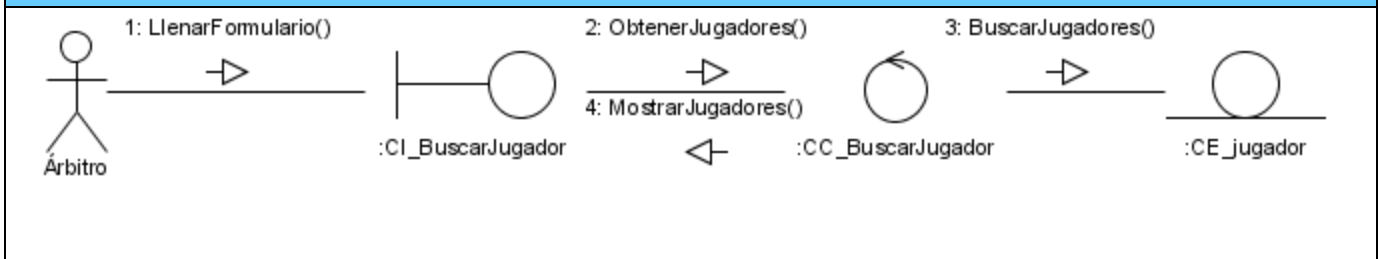


Figura 59: CU Buscar jugador

Diagrama de Colaboración del Análisis CUS Buscar jugador





**Figura 60:** CU Definir Sistemas de Desempate

Diagrama de Colaboración del Análisis Sección: “Definir Sistemas de Desempate para un torneo RRI”

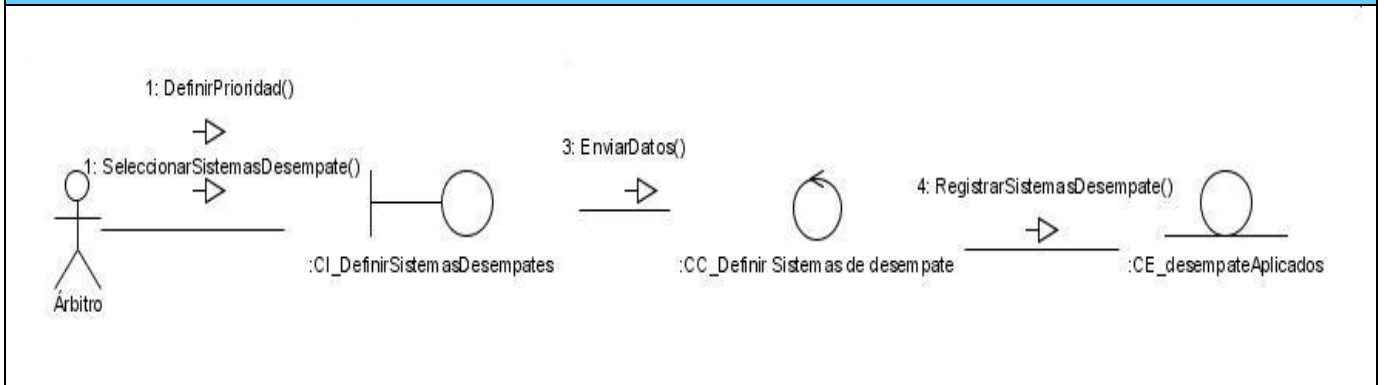
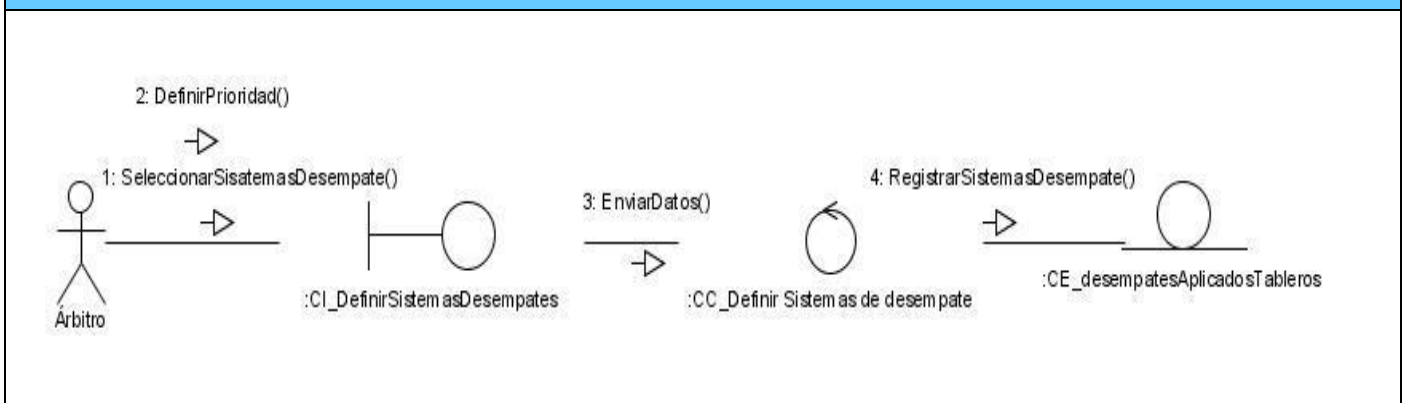


Diagrama de Colaboración del Análisis Sección: “Definir Sistemas de Desempate para un torneo RRE”







**Figura 61:** CU Gestionar Ronda

Diagrama de Colaboración del Análisis Sección: "Adicionar Ronda"

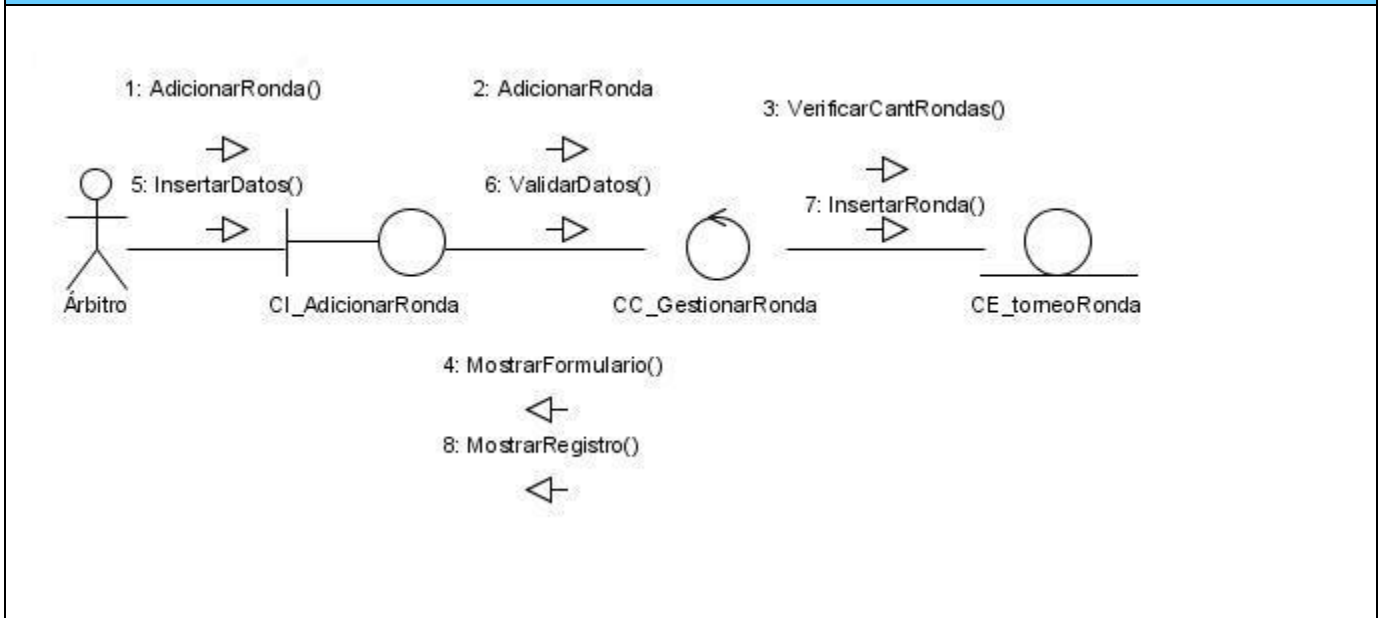
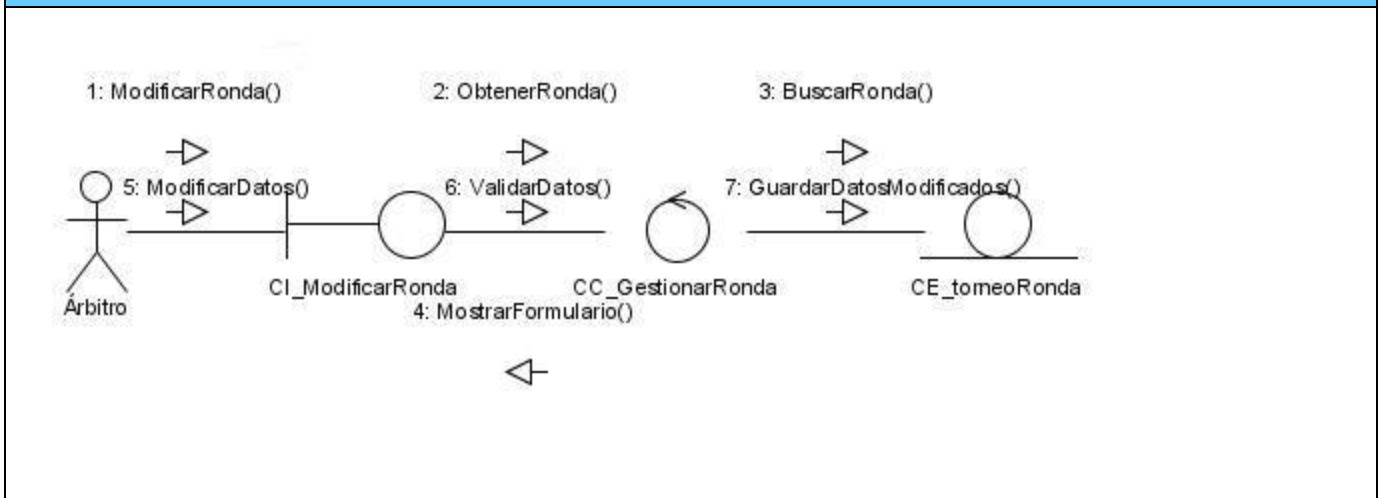


Diagrama de Colaboración del Análisis Sección: "Modificar Ronda"





**Figura 62:** CU Gestionar Partida

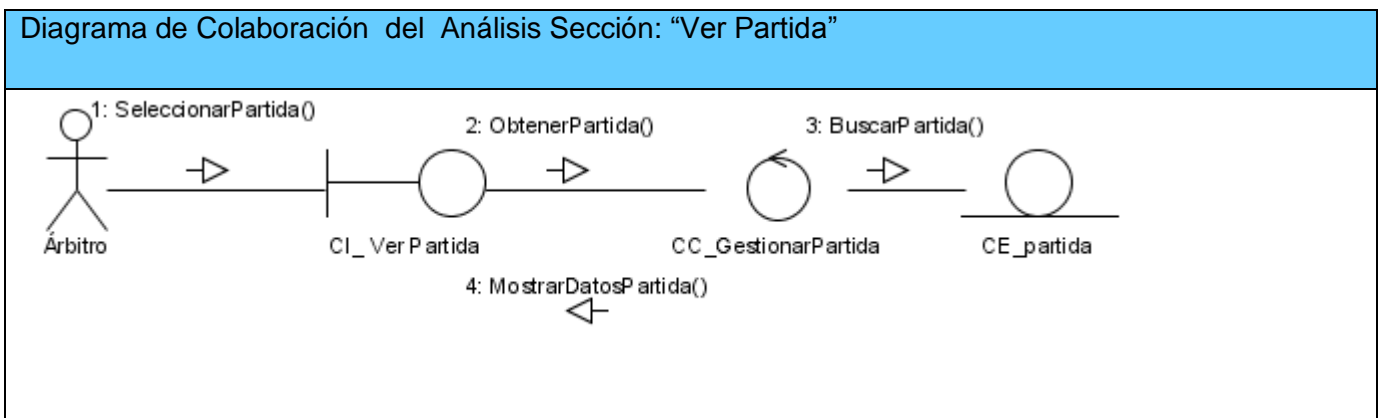
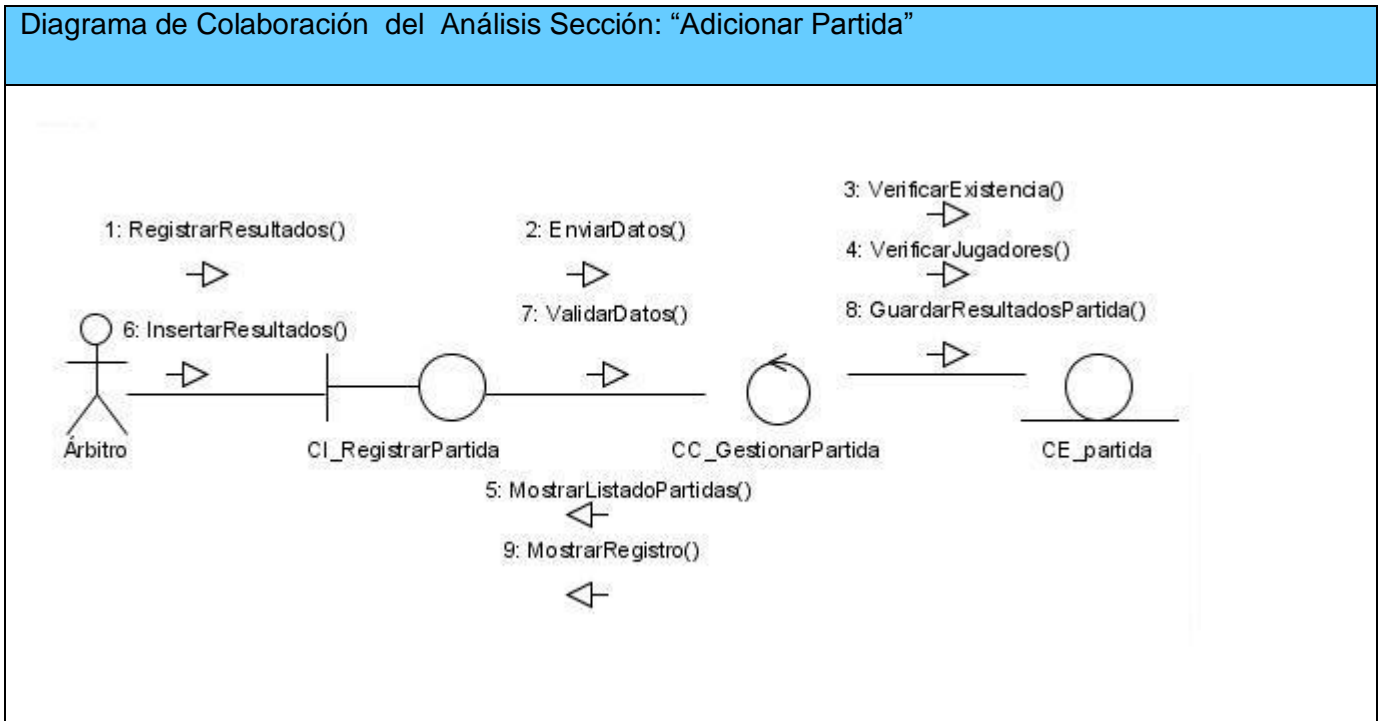




Diagrama de Colaboración del Análisis Sección: “Modificar Partida”

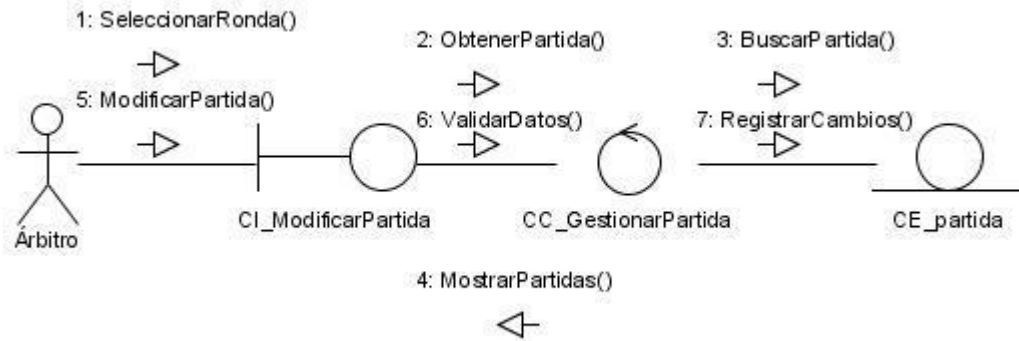


Diagrama de Colaboración del Análisis Sección: “Eliminar Partida”

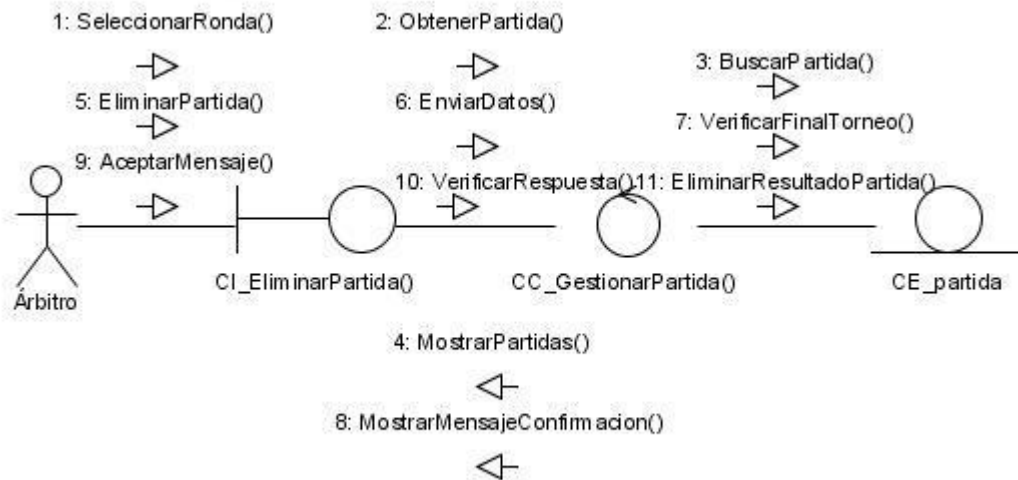




Figura 63: CU Gestionar Grupo

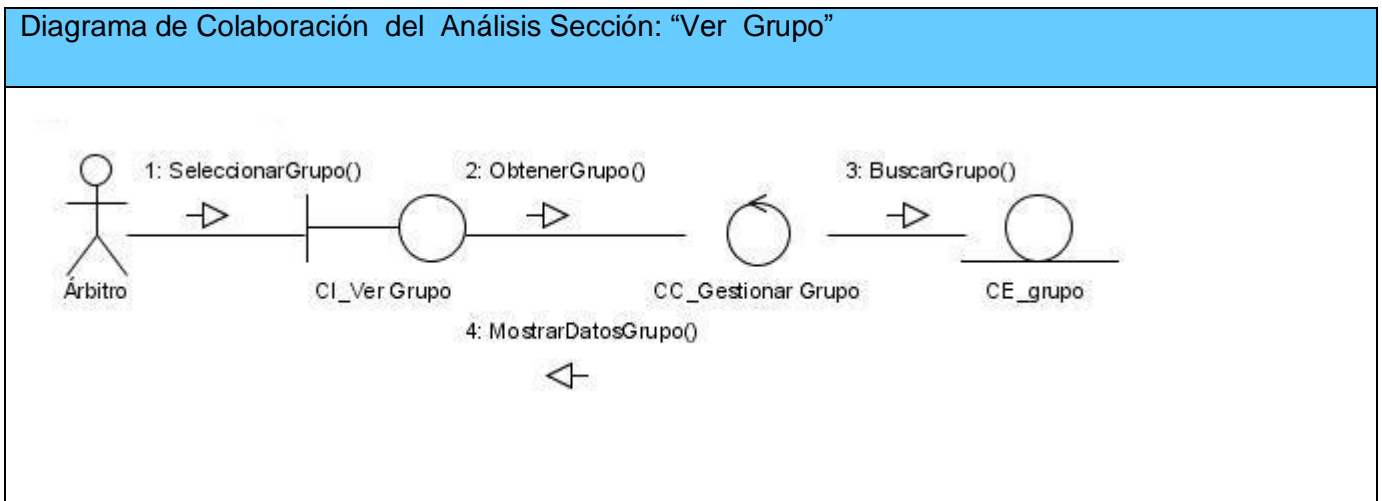
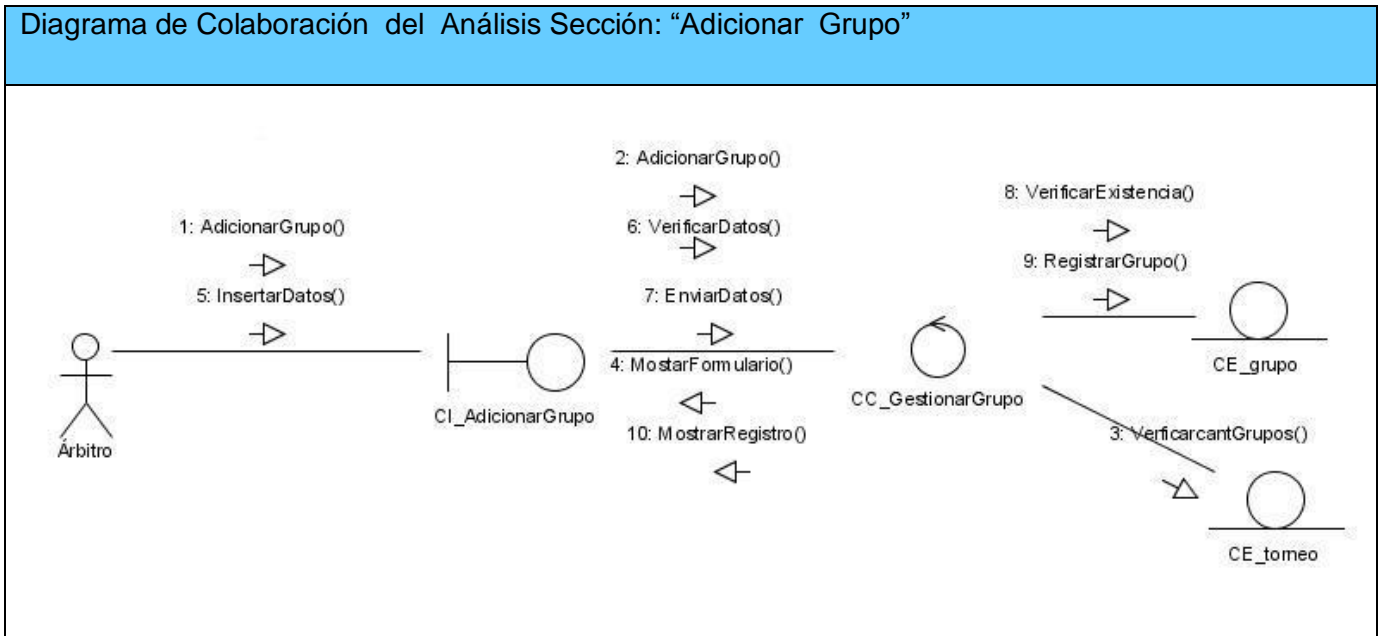




Diagrama de Colaboración del Análisis Sección: “Modificar Grupo”

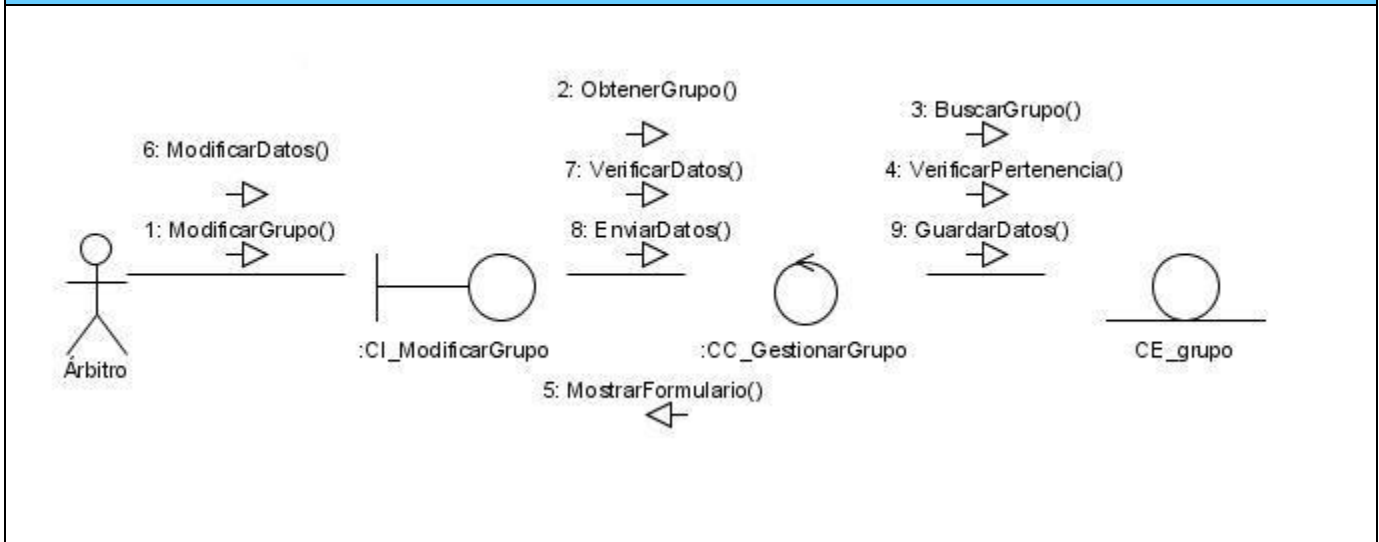
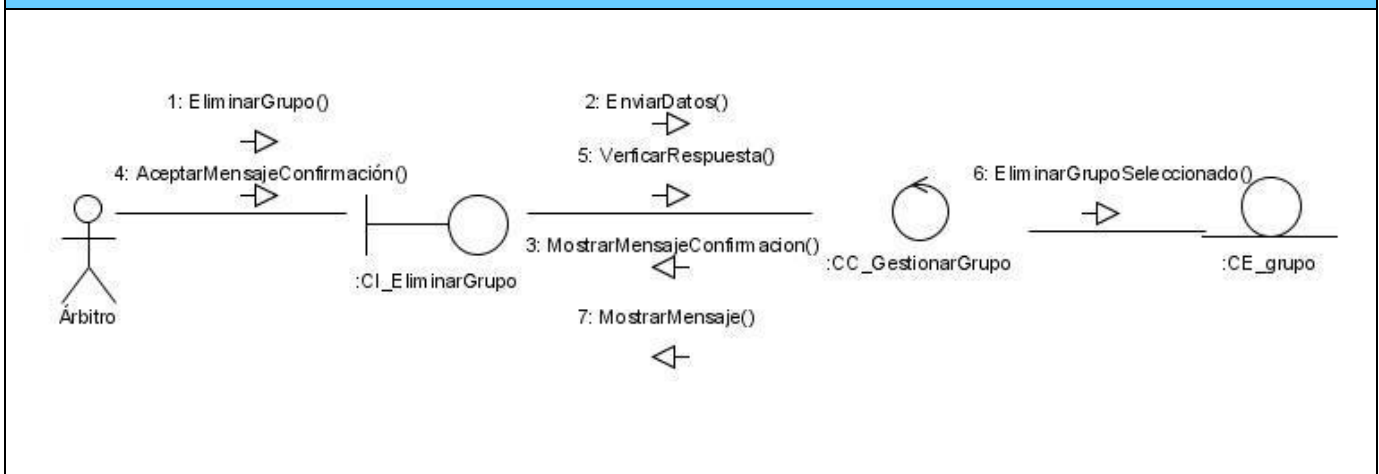


Diagrama de Colaboración del Análisis Sección: “Eliminar Grupo”





**Figura 64:** CU Gestionar jugadores de un equipo

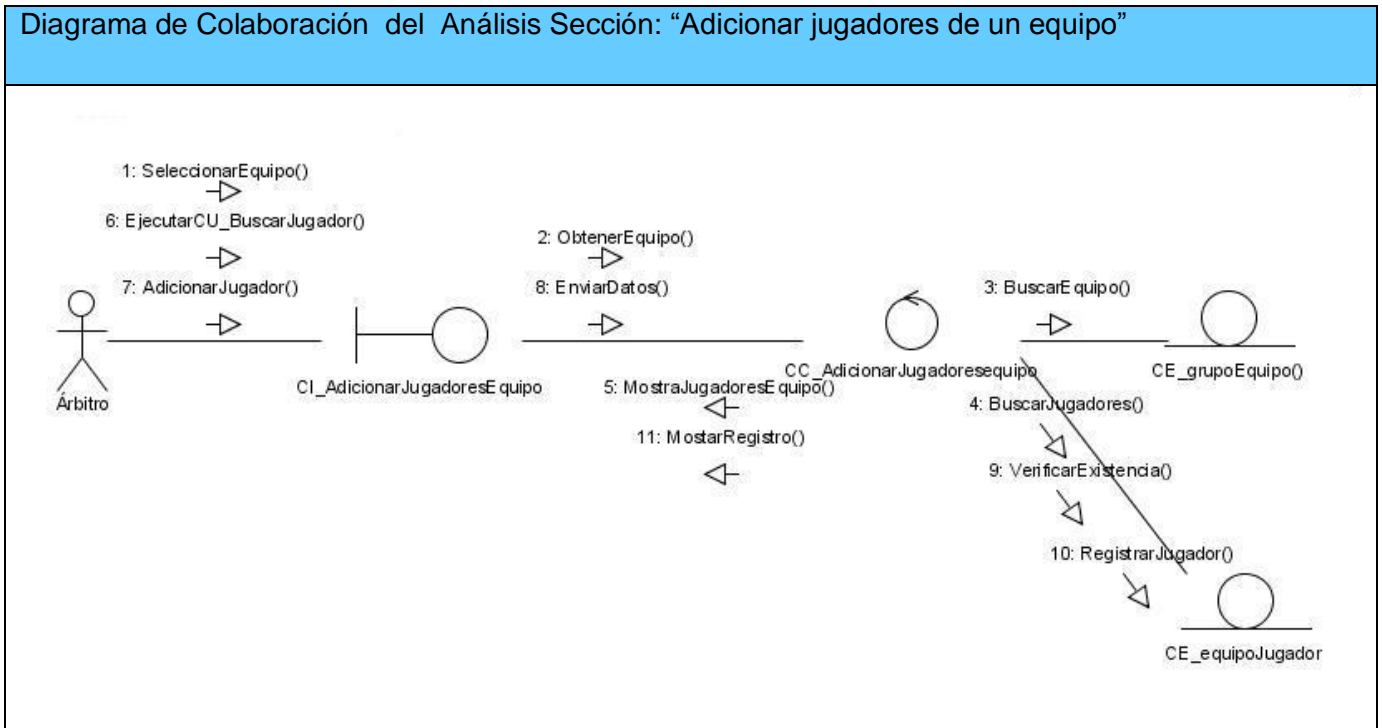




Diagrama de Colaboración del Análisis Sección: "Eliminar jugadores de un equipo"

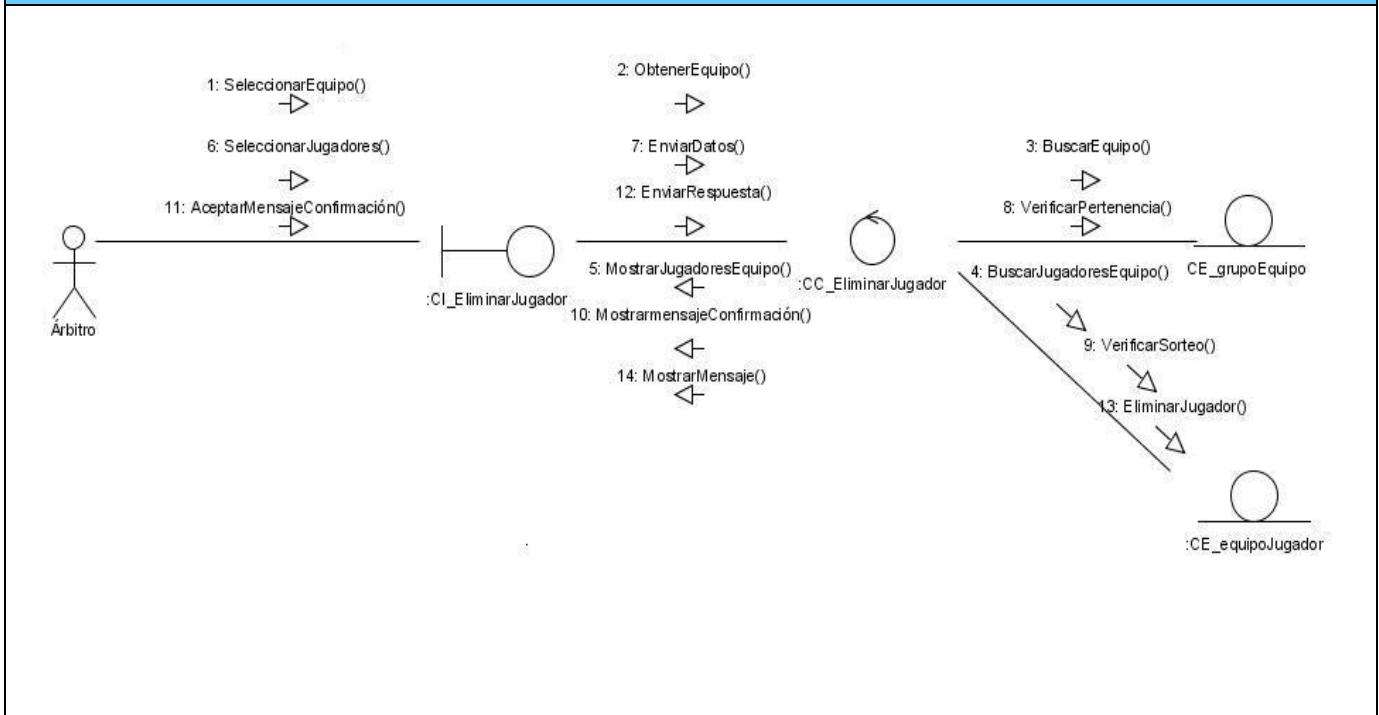
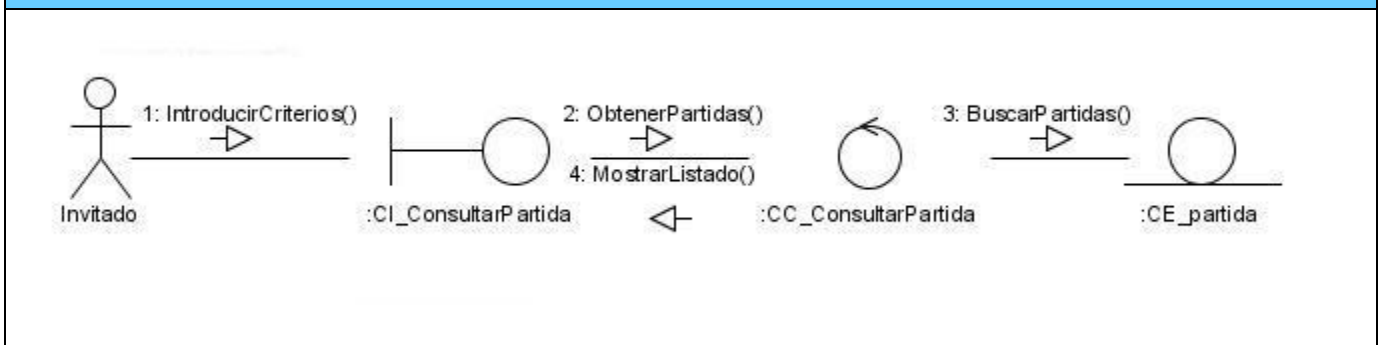


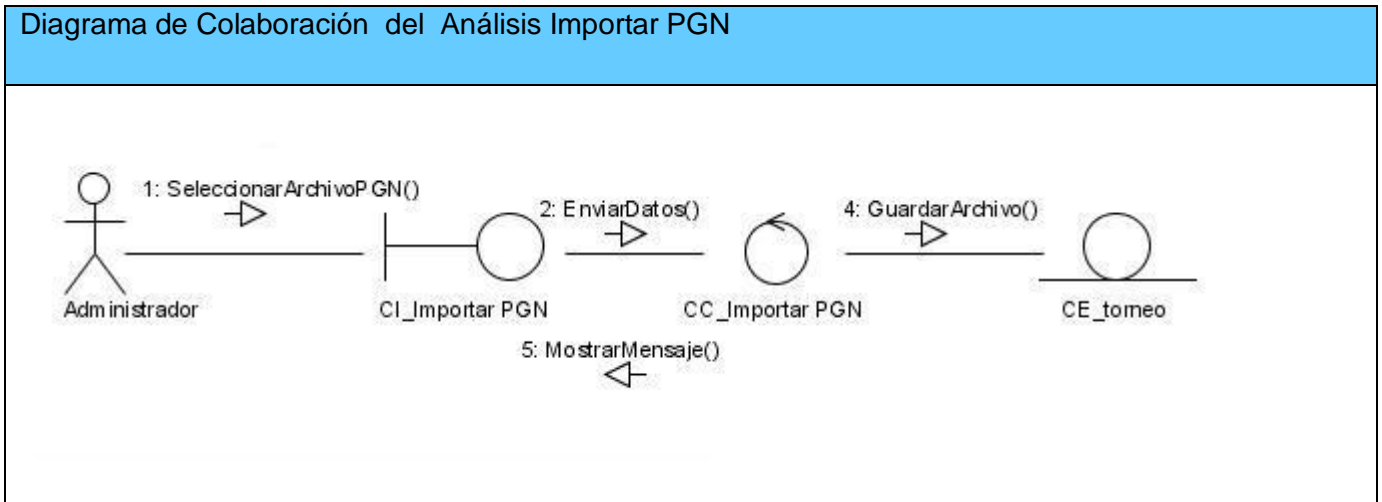
Figura 65: CU Consultar Partida

Diagrama de Colaboración del Análisis Consultar Partida





**Figura 66:** CU Importar PGN



**Figura 67:** CU Gestionar Ciudad

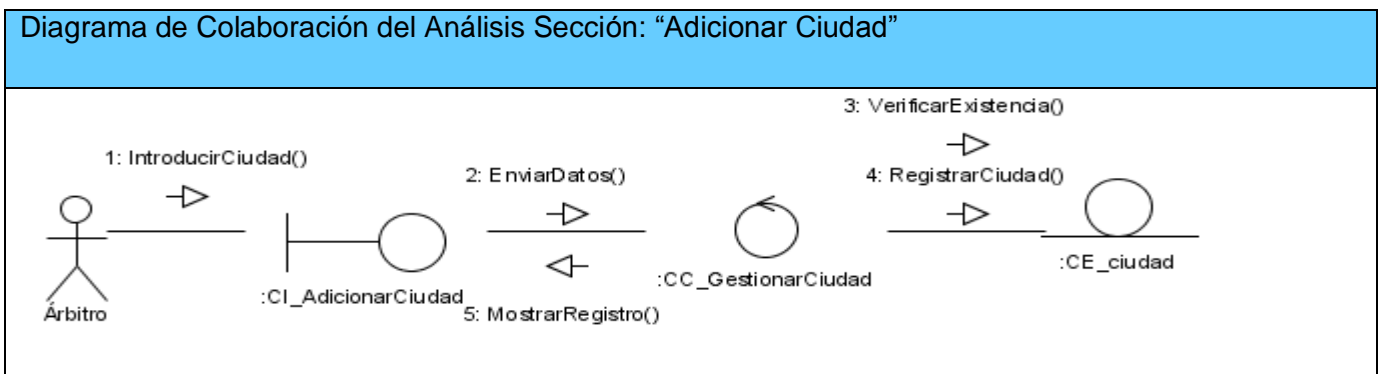






Diagrama de Colaboración del Análisis Sección: "Modificar Ciudad"

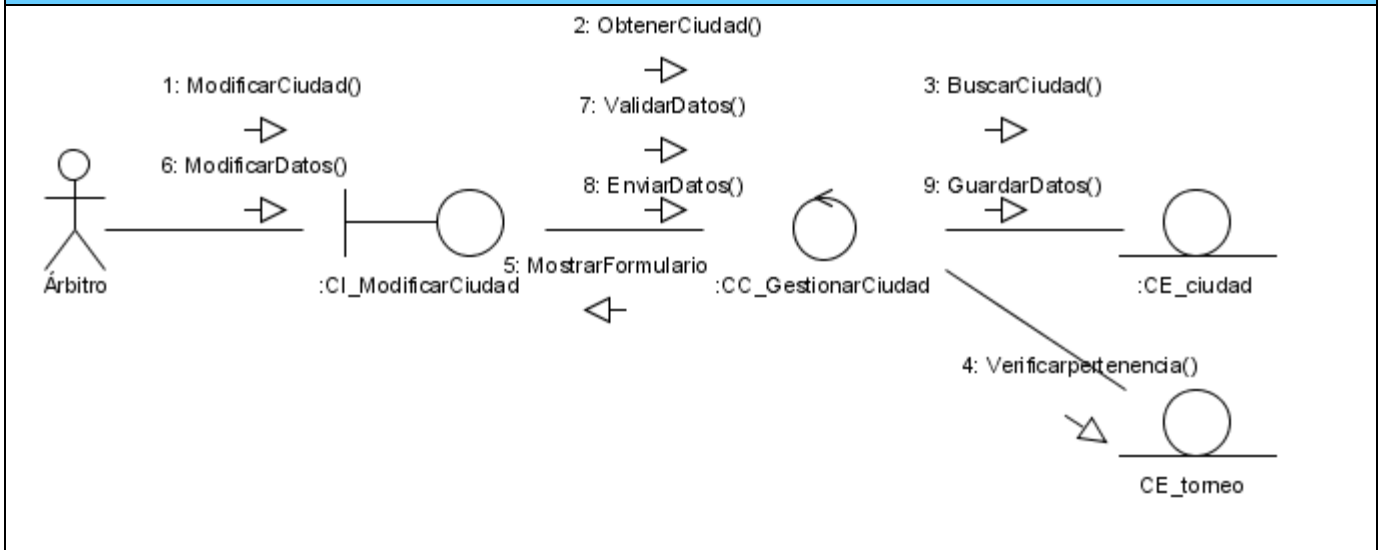


Diagrama de Colaboración del Análisis Sección: "Eliminar Ciudad"

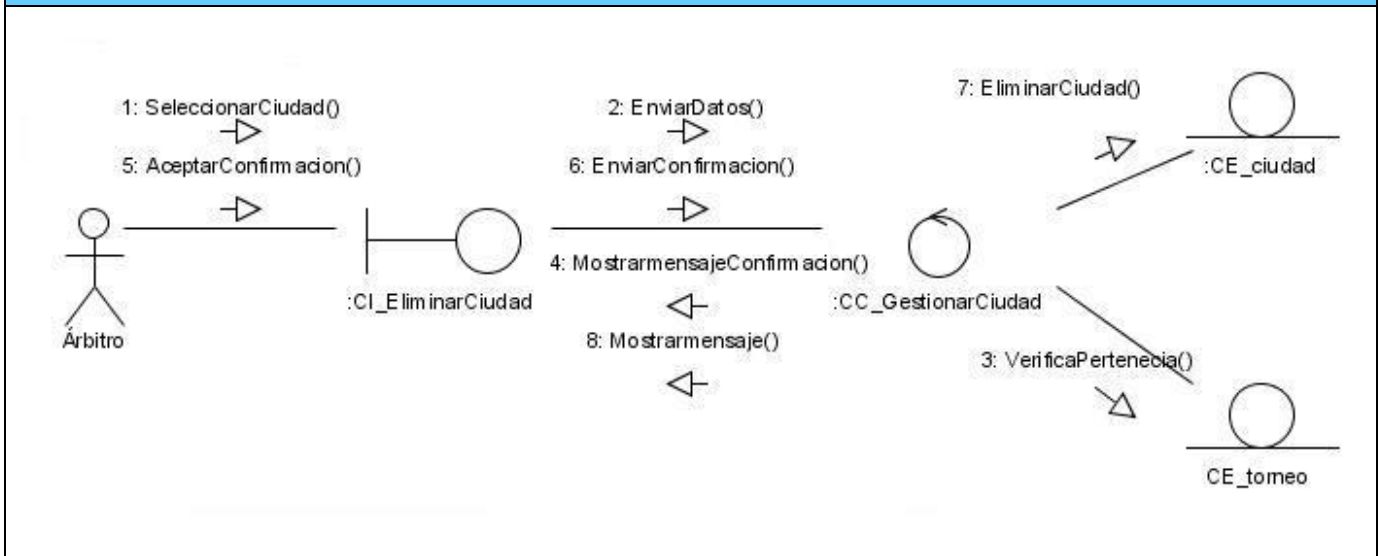




Figura 68: CU Gestionar Título del Árbitro

Diagrama de Colaboración del Análisis Sección: "Adicionar Título del Árbitro"

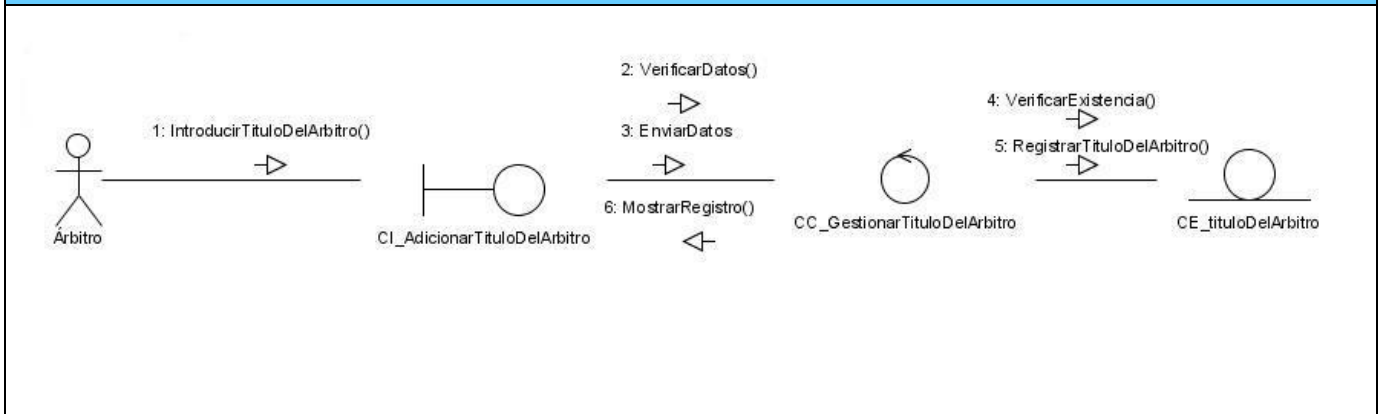


Diagrama de Colaboración del Análisis Sección: "Modificar Título del Árbitro"

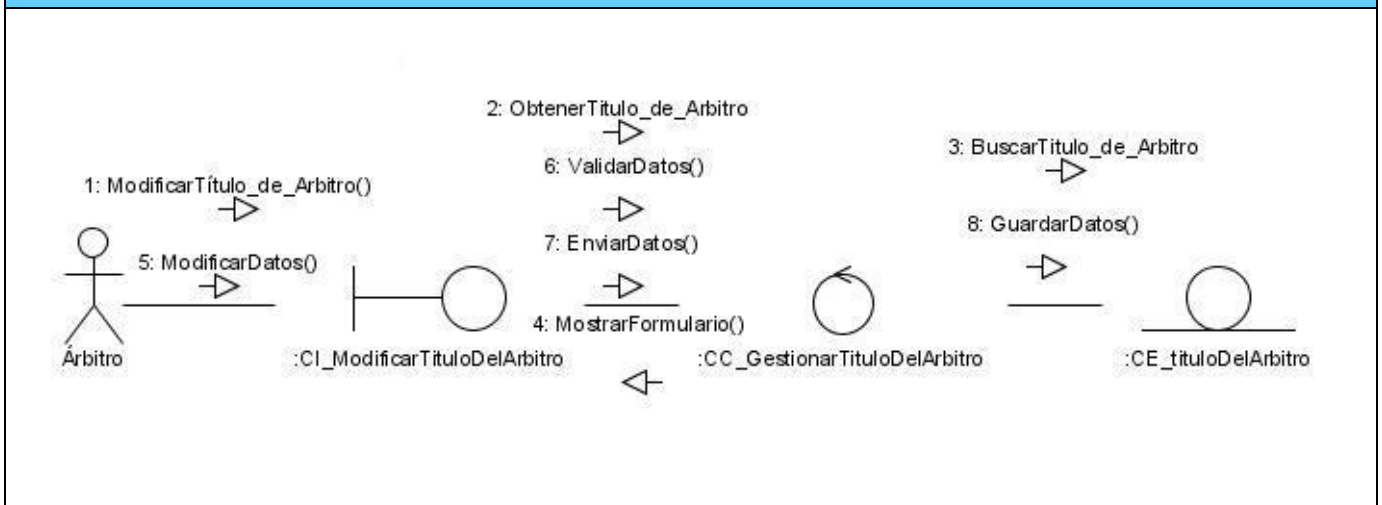




Diagrama de Colaboración del Análisis Sección: “Eliminar Título del Árbitro”

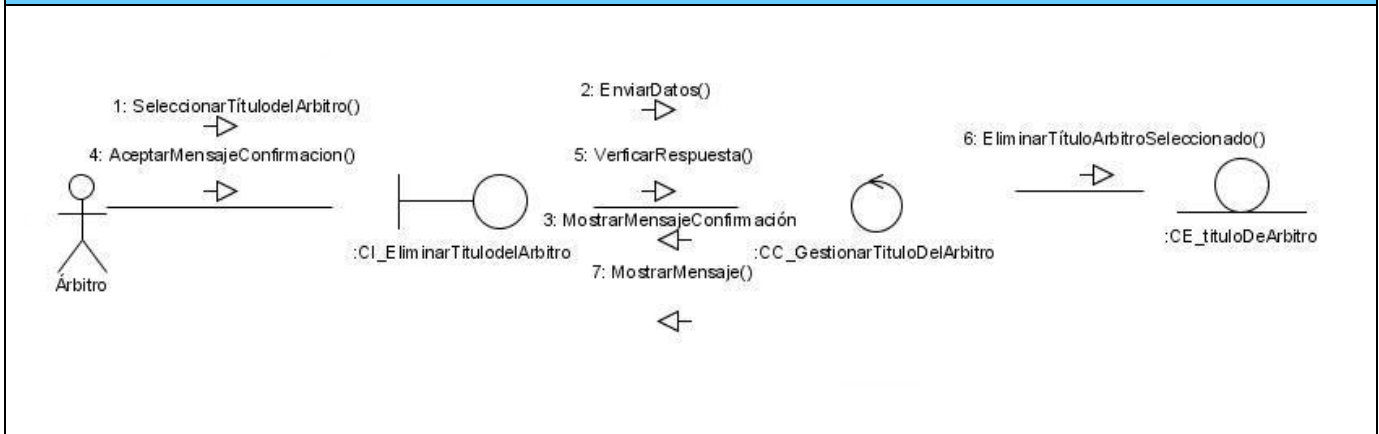


Figura 69: CU Gestionar Título del Jugador

Diagrama de Colaboración del Análisis Sección: “Adicionar Título del Jugador”

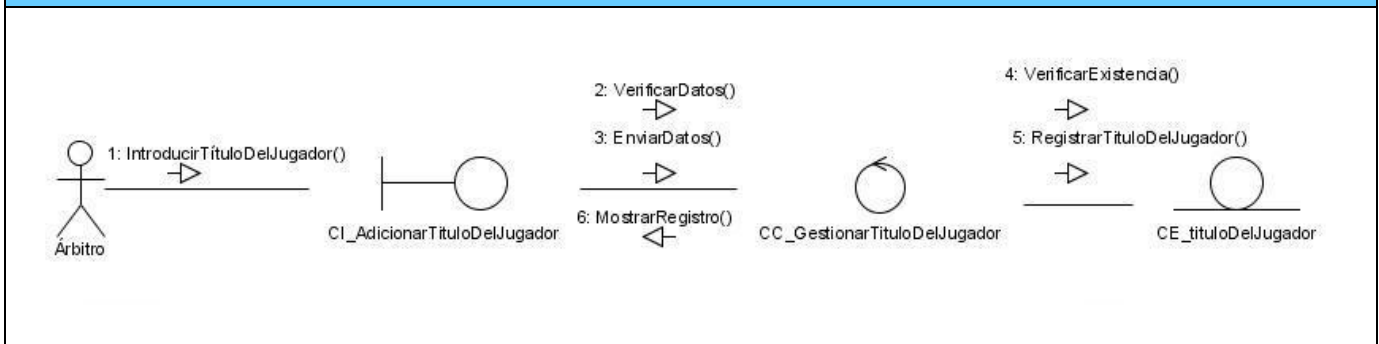




Diagrama de Colaboración del Análisis Sección: “ Modificar Título del Jugador”

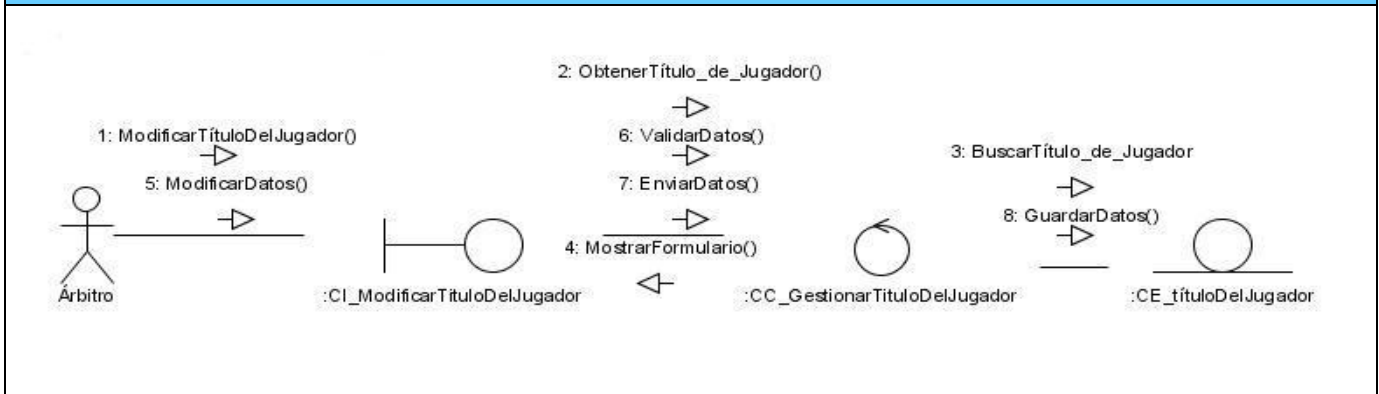
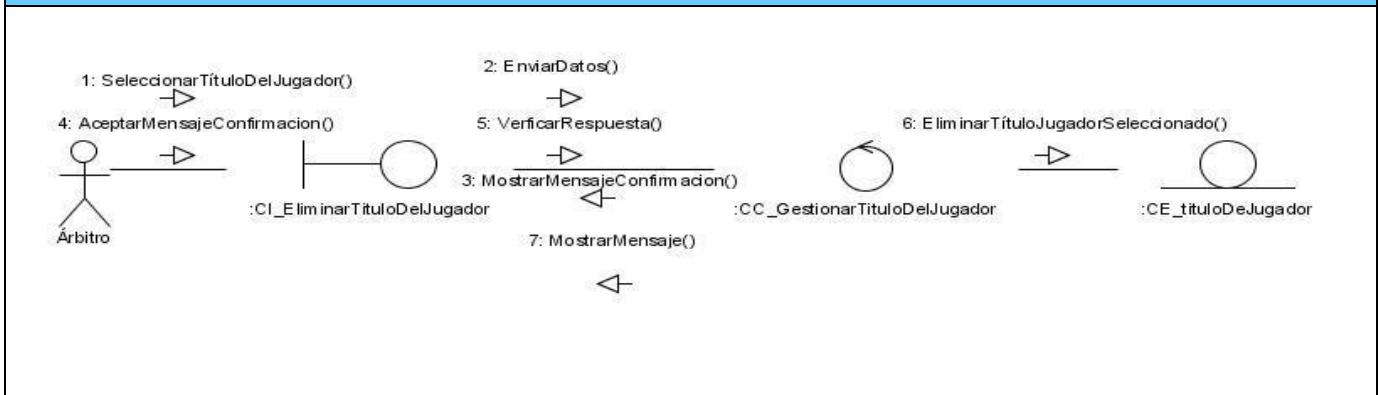
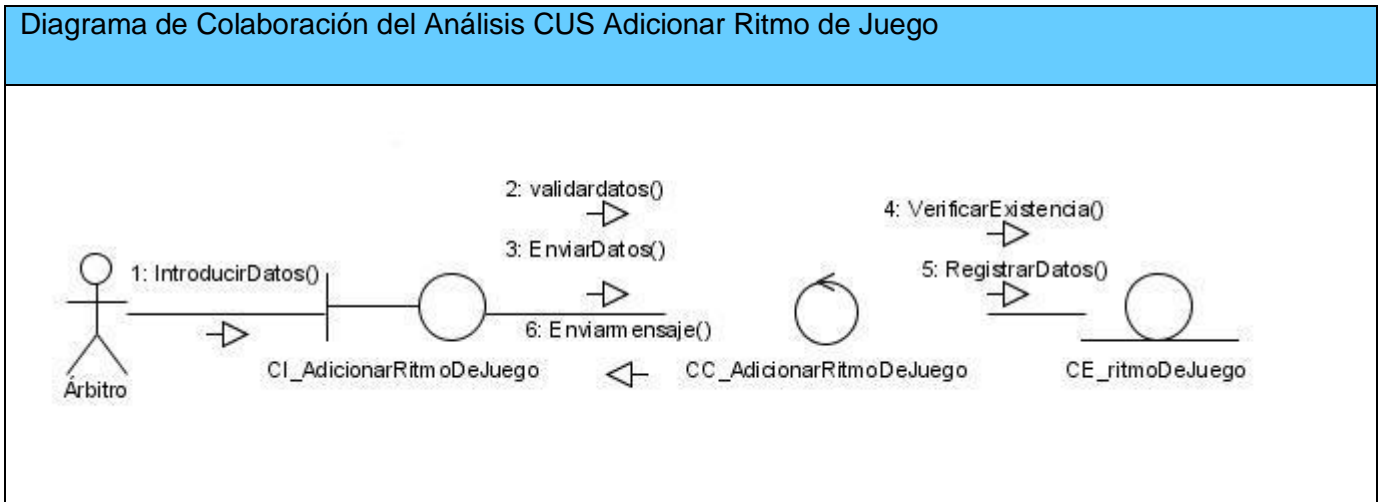


Diagrama de Colaboración del Análisis Sección: “ Eliminar Título del Jugador”

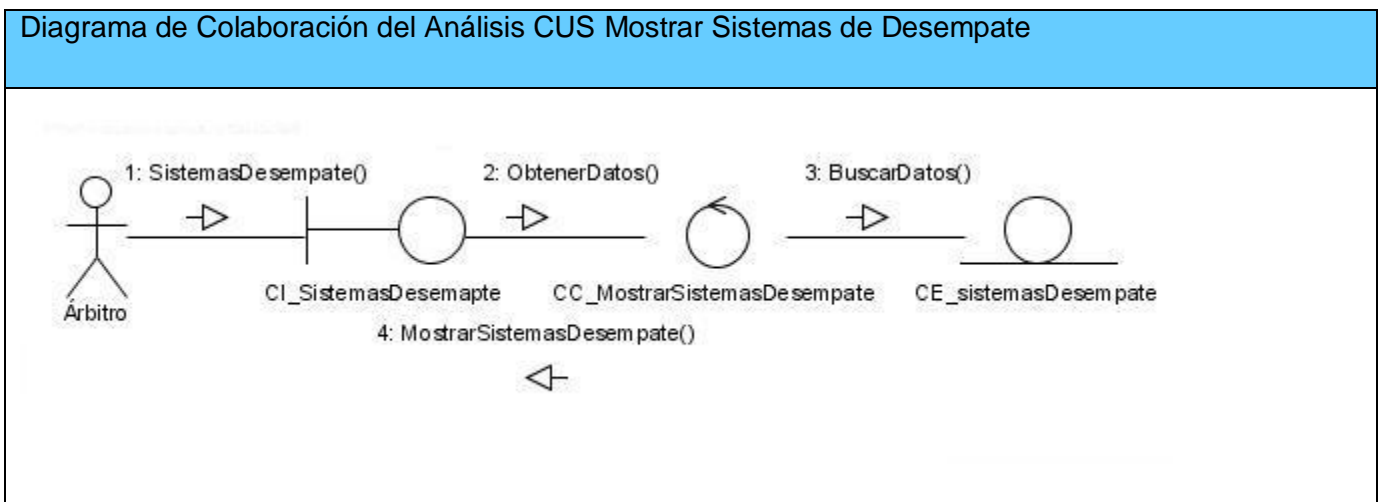




**Figura 70:** CU Adicionar Ritmo de Juego



**Figura 71:** CU Mostrar Sistemas de Desempate





**Figura 72:** CU Gestionar Jugadores del torneo

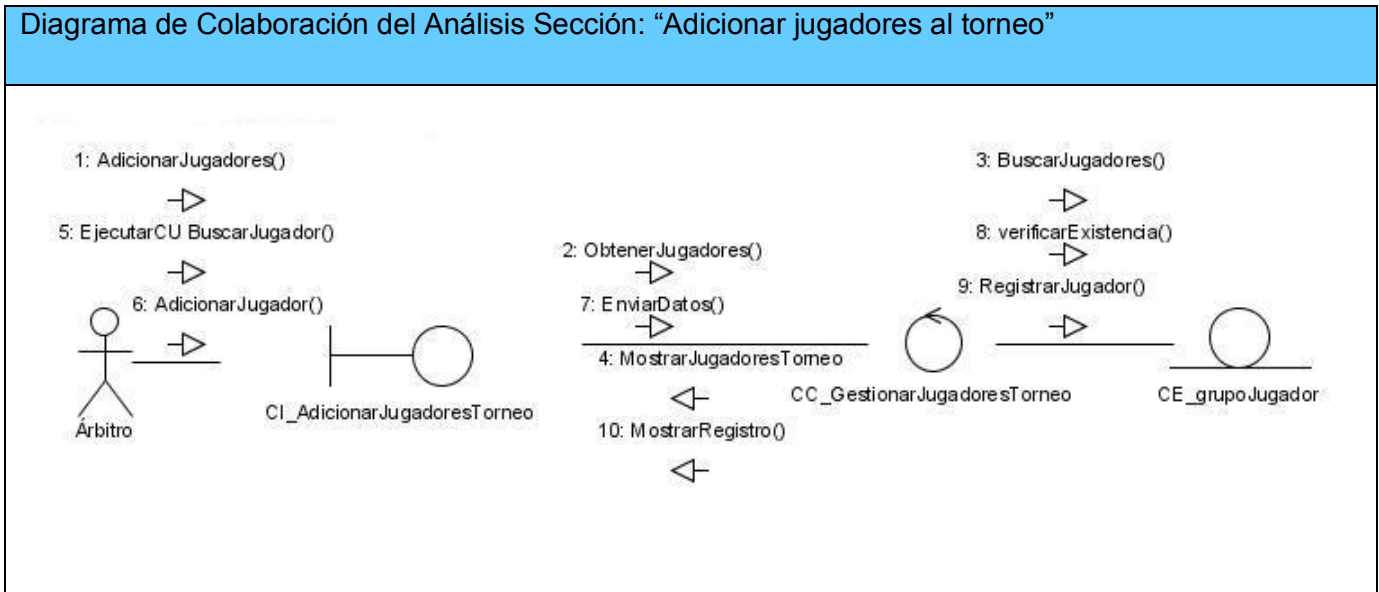




Diagrama de Colaboración del Análisis Sección: "Eliminar jugadores del torneo"

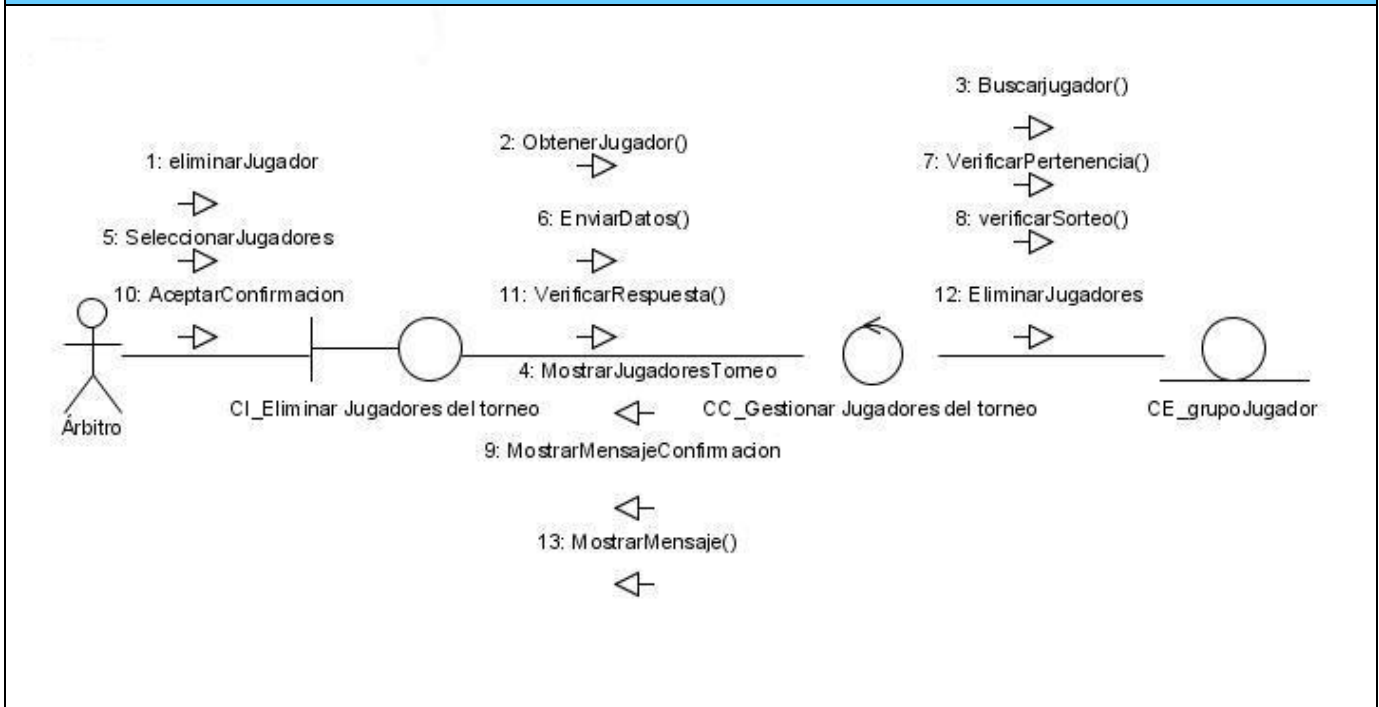




Figura 73: CU Elaborar Bases técnicas

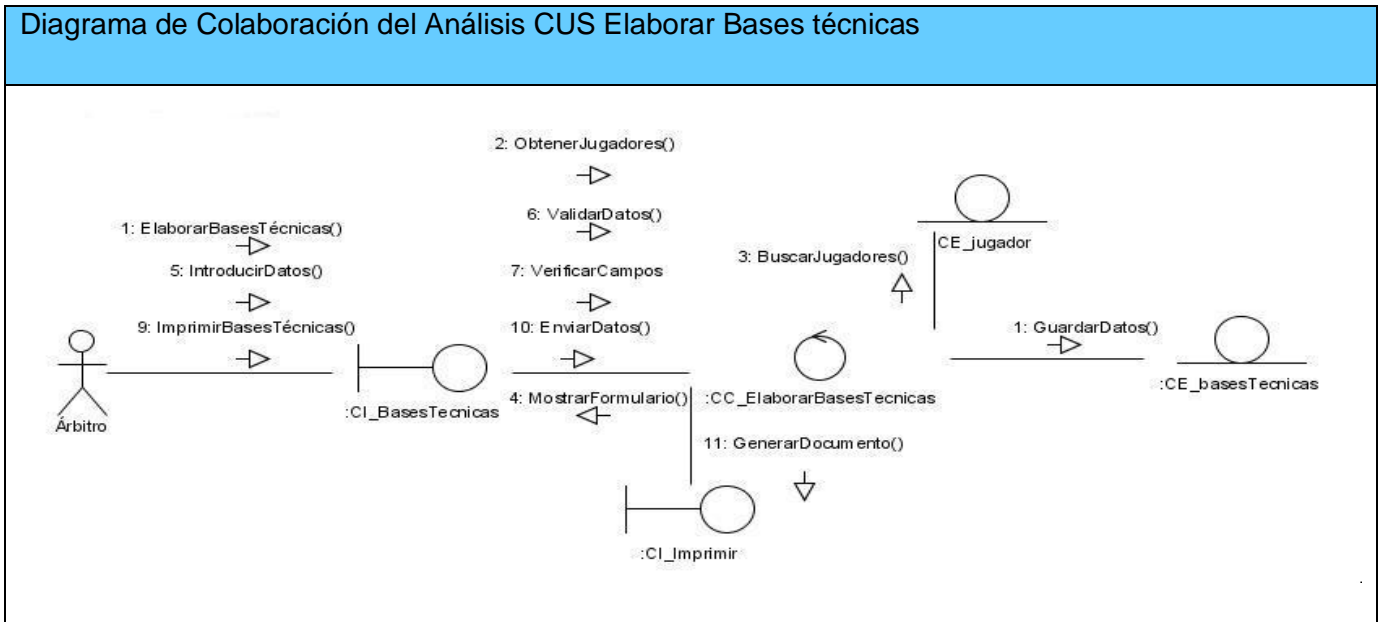


Figura 74: CU Cálculo del Rating Elo

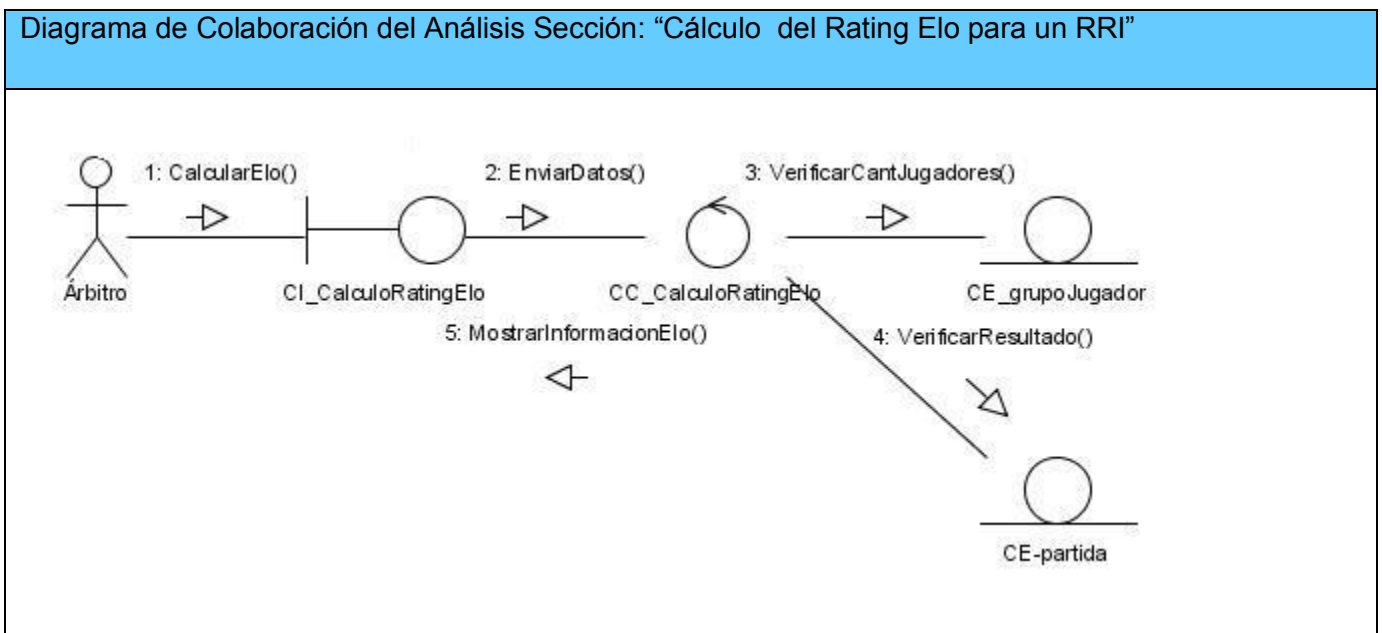






Diagrama de Colaboración del Análisis Sección: "Cálculo del Rating Elo para un RRE"

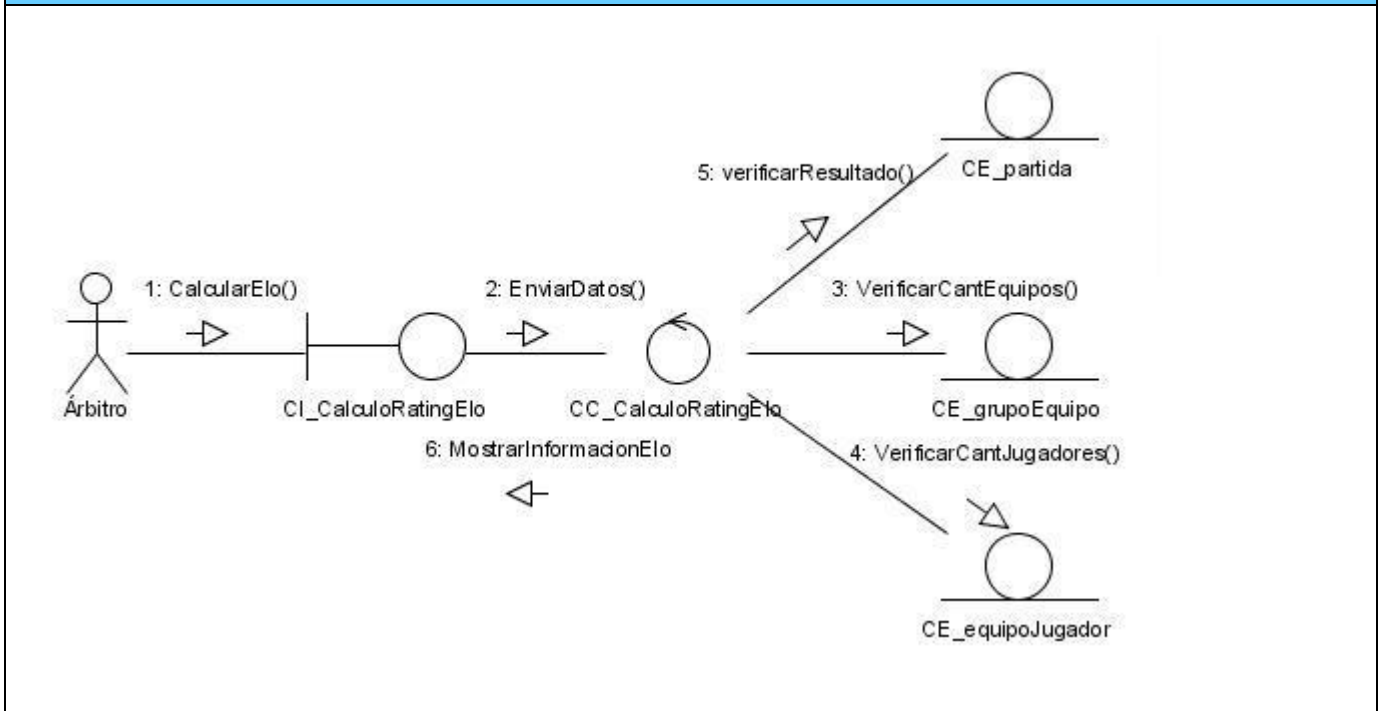




Figura 75: CU Mostrar Cuadro Sinóptico

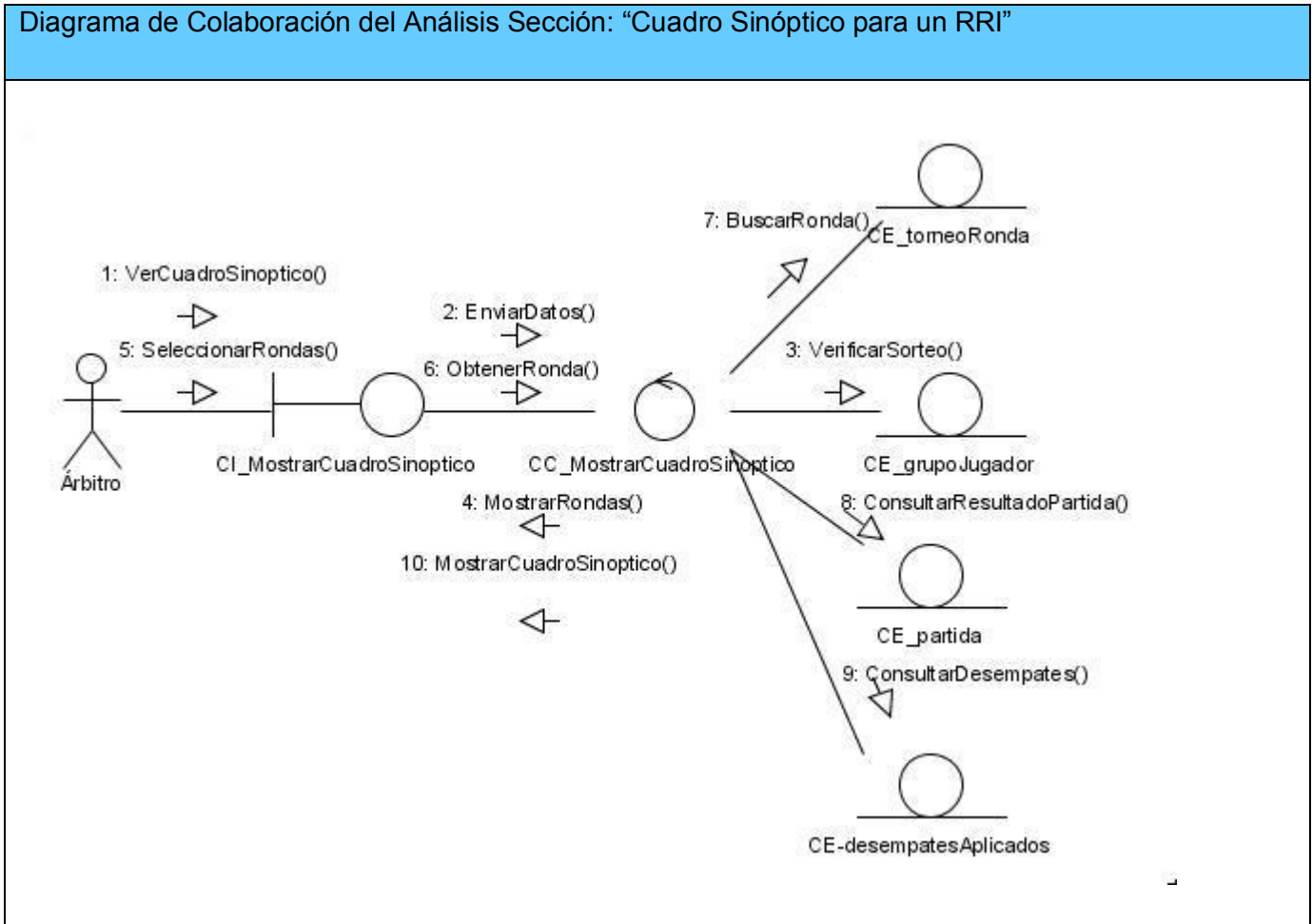
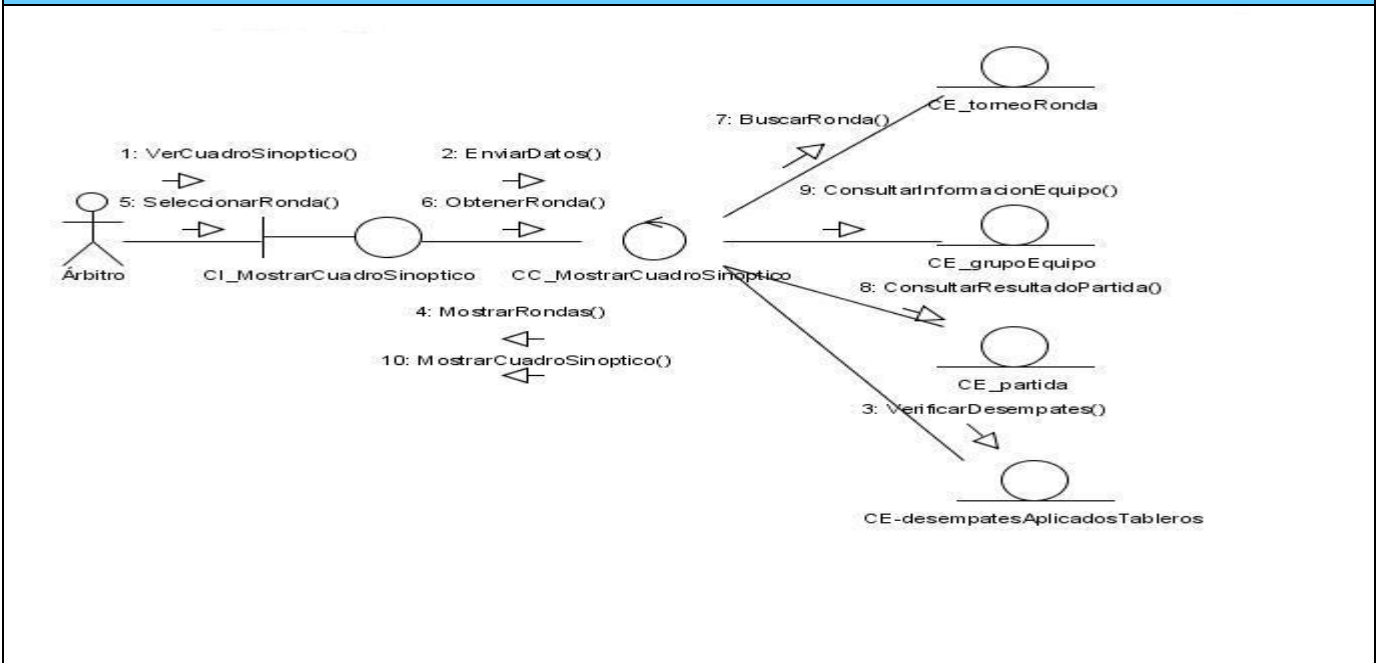


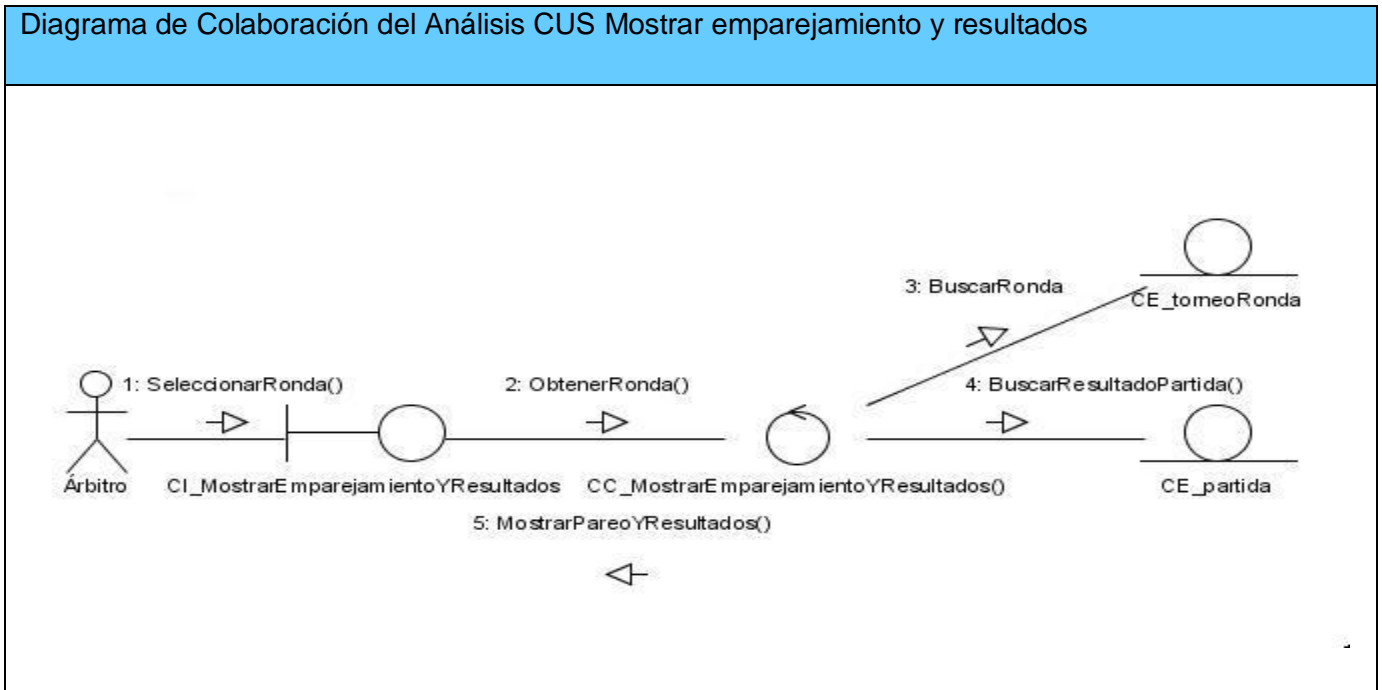


Diagrama de Colaboración del Análisis Sección: "Cuadro Sinóptico para un RRE"



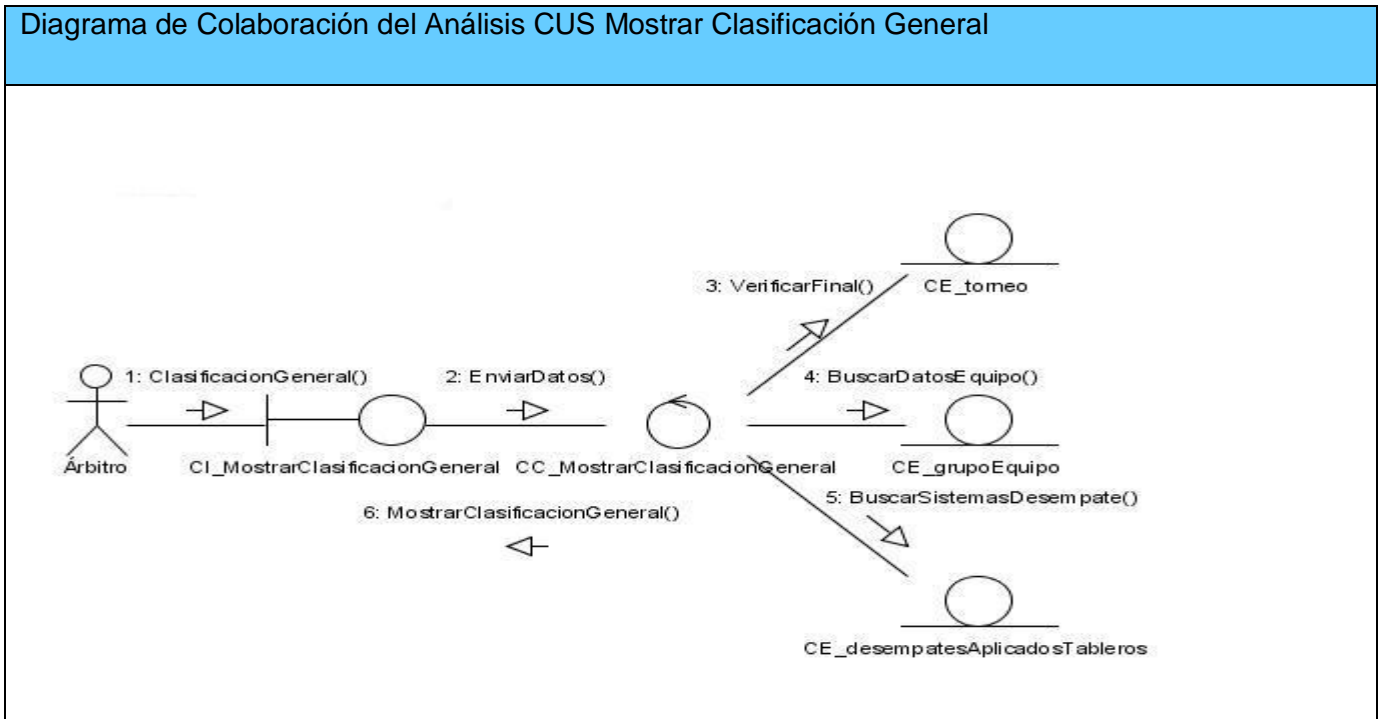


**Figura 76:** CU Mostrar emparejamiento y resultados





**Figura 77:** CU Mostrar Clasificación General





**Figura 78:** CU Gestionar equipos de un grupo

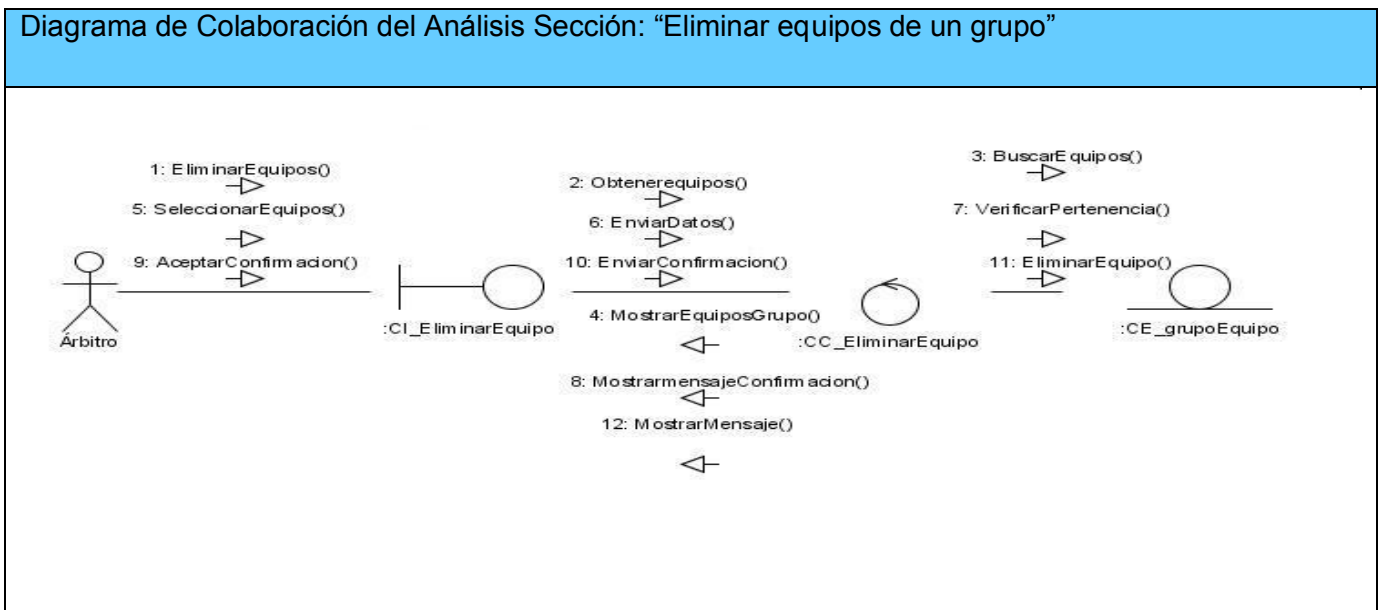
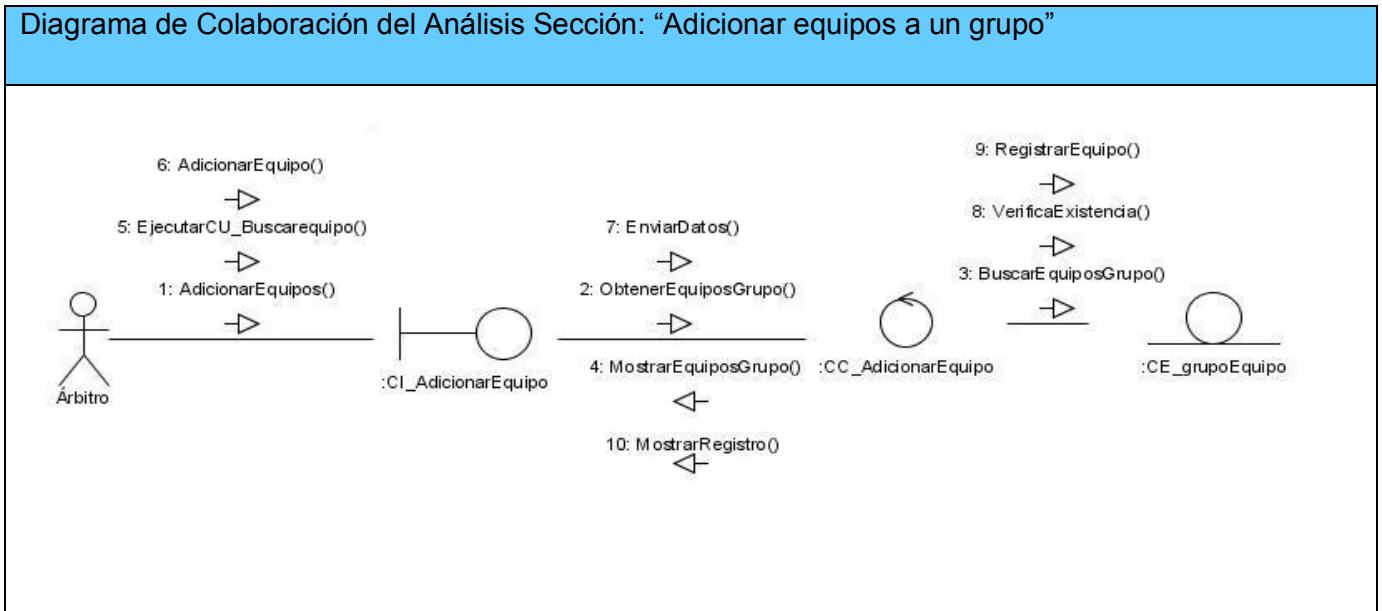




Figura 79: CU Buscar equipo

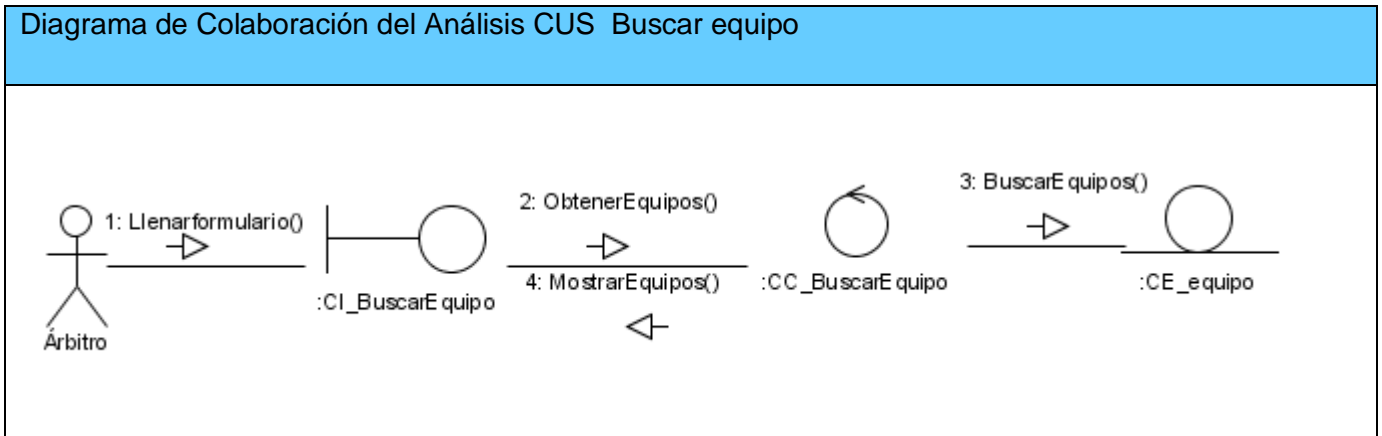


Figura 80: CU Modificar tableros

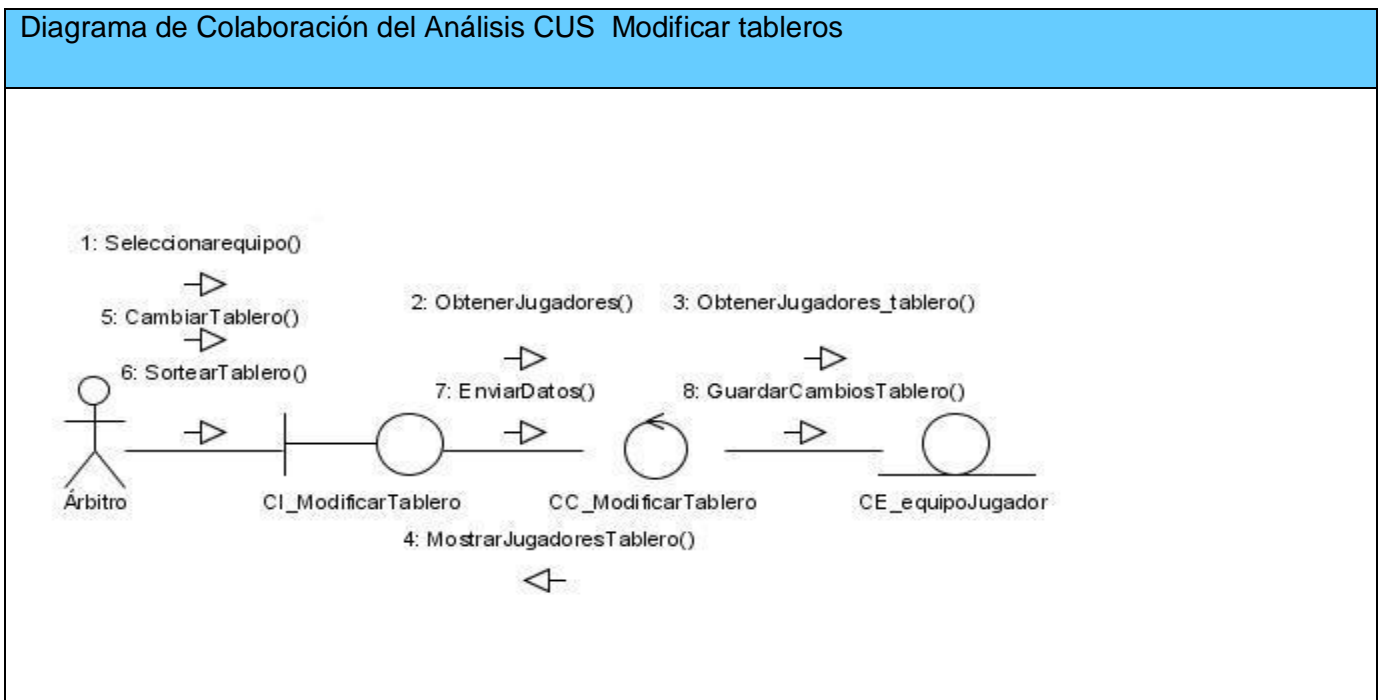
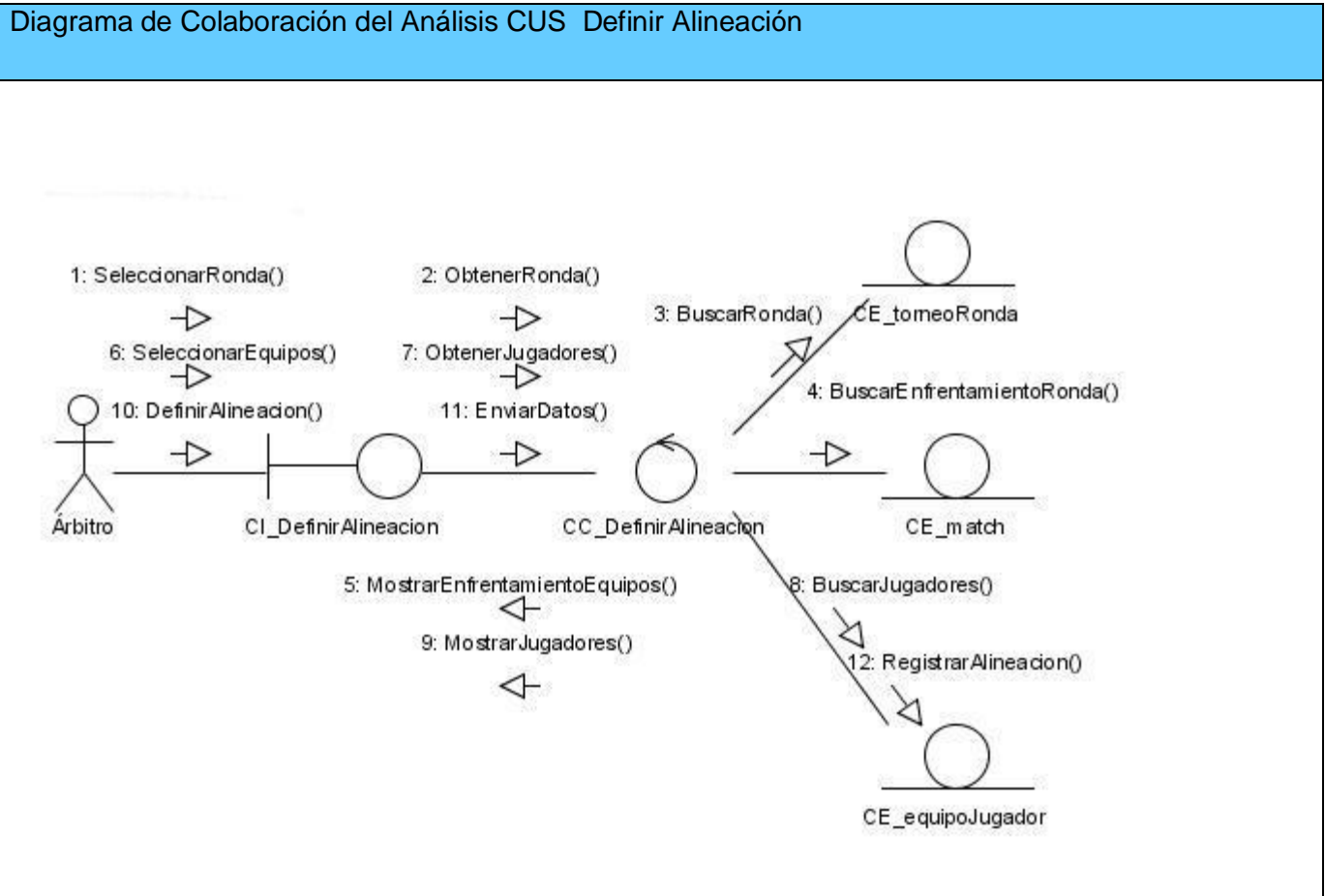




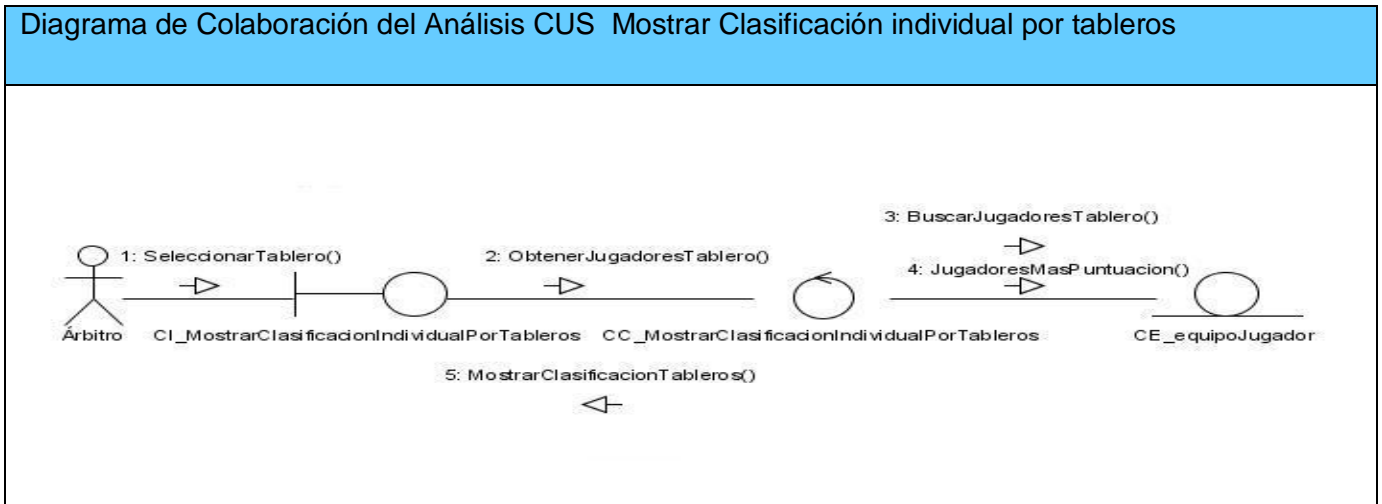
Figura 81: CU Definir Alineación



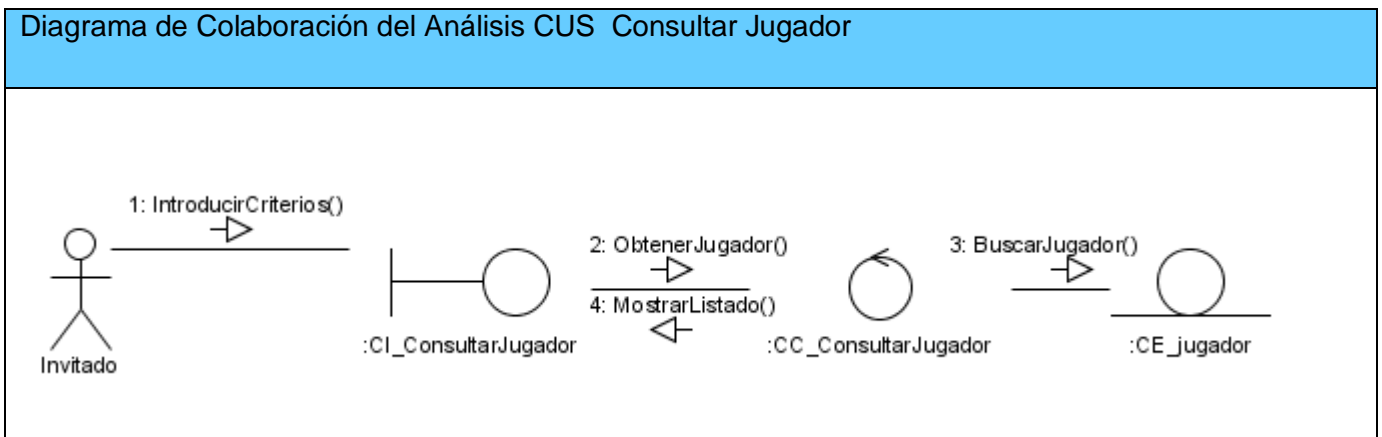




**Figura 82:** CU Mostrar Clasificación individual por tableros



**Figura 83:** CU Consultar Jugador





**Figura 84:** CU Gestionar imagen

Diagrama de Colaboración del Análisis Sección: "Adicionar imagen"

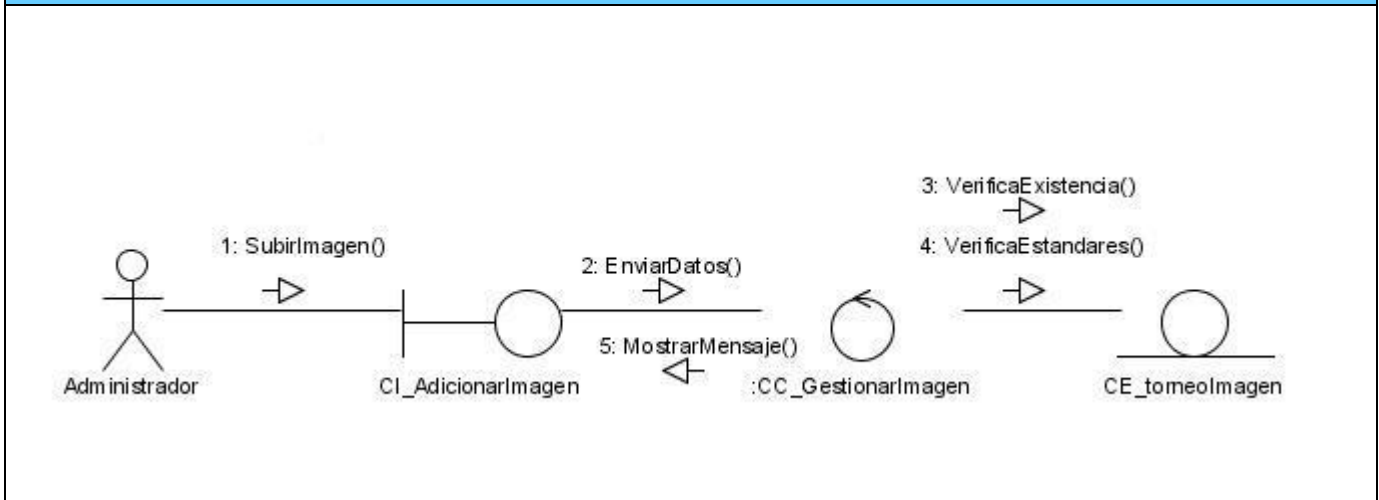
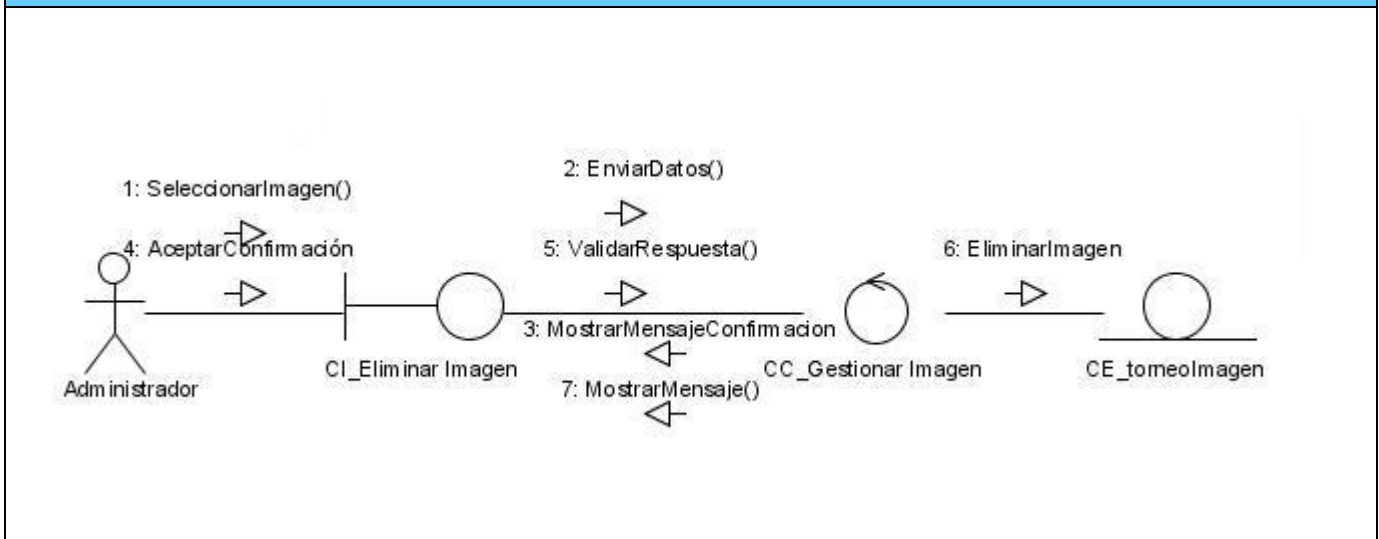


Diagrama de Colaboración del Análisis Sección: "Eliminar imagen"



**Anexo 7: Diagramas de Clases del Diseño**



Figura 85: CU Autenticar usuario

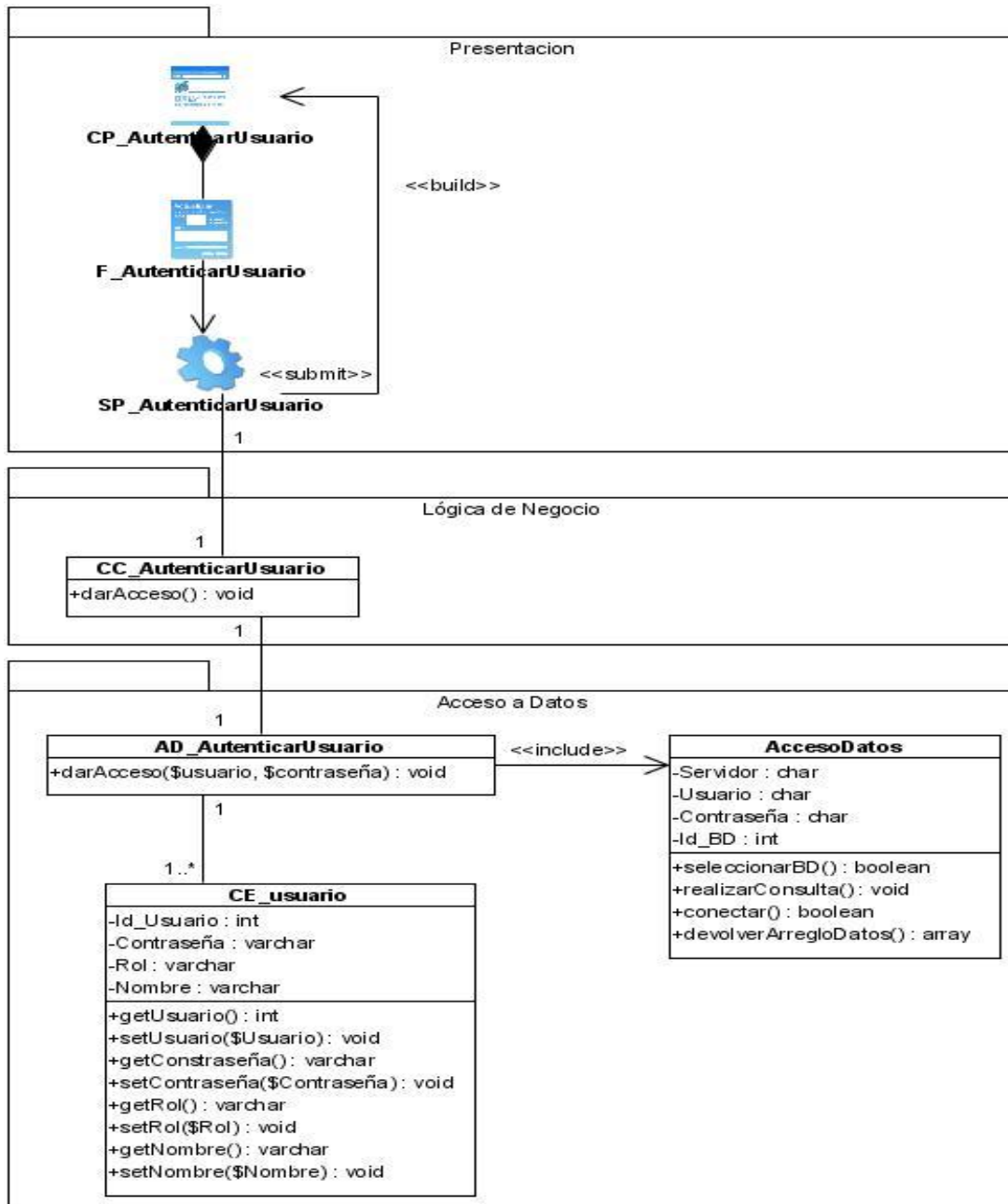


Figura 86: CU Gestionar usuario

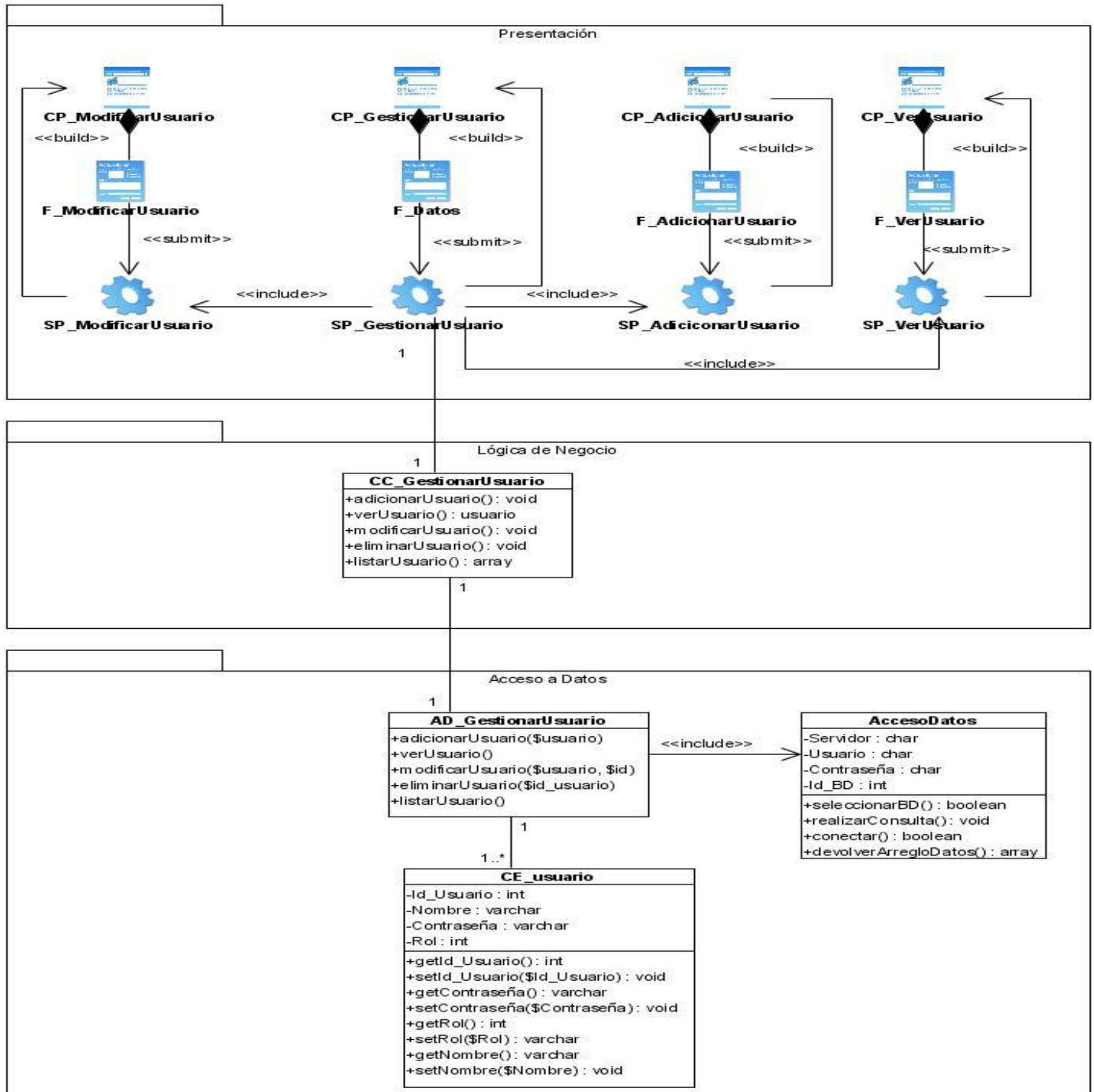




Figura 87: CU Gestionar Ciudad

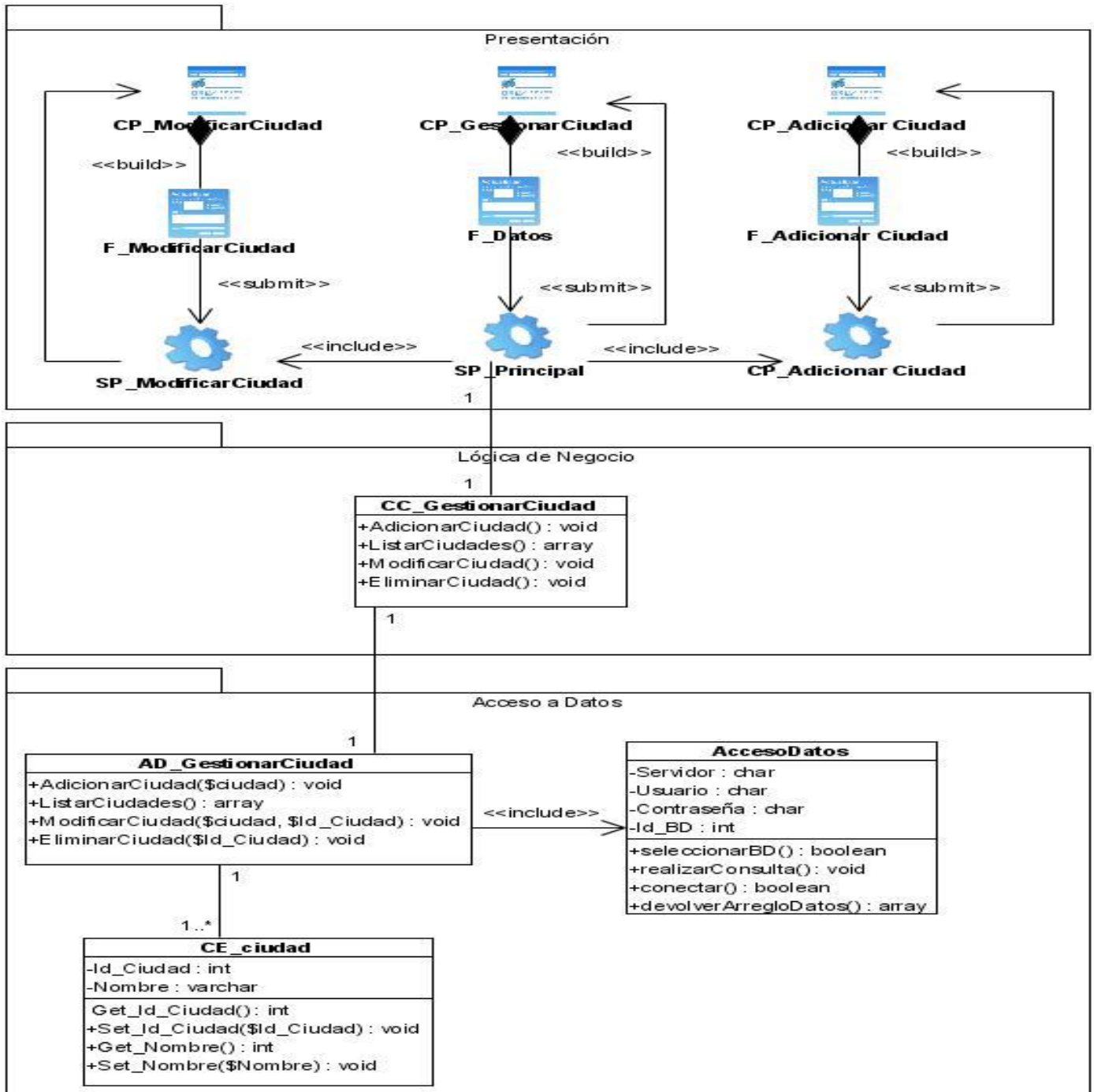




Figura 88: CU Gestionar Federación

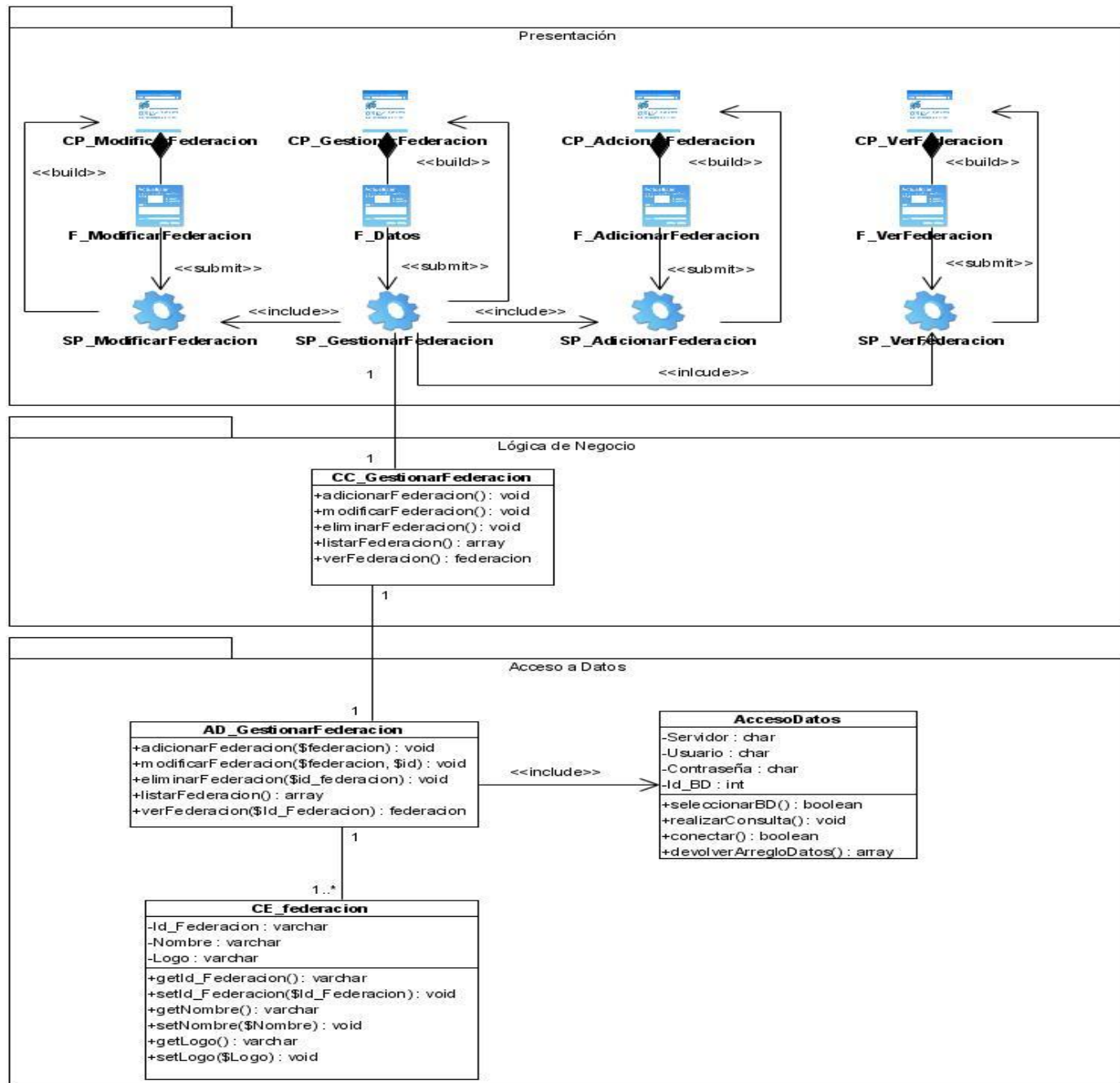




Figura 89: CU Gestionar Título del Árbitro

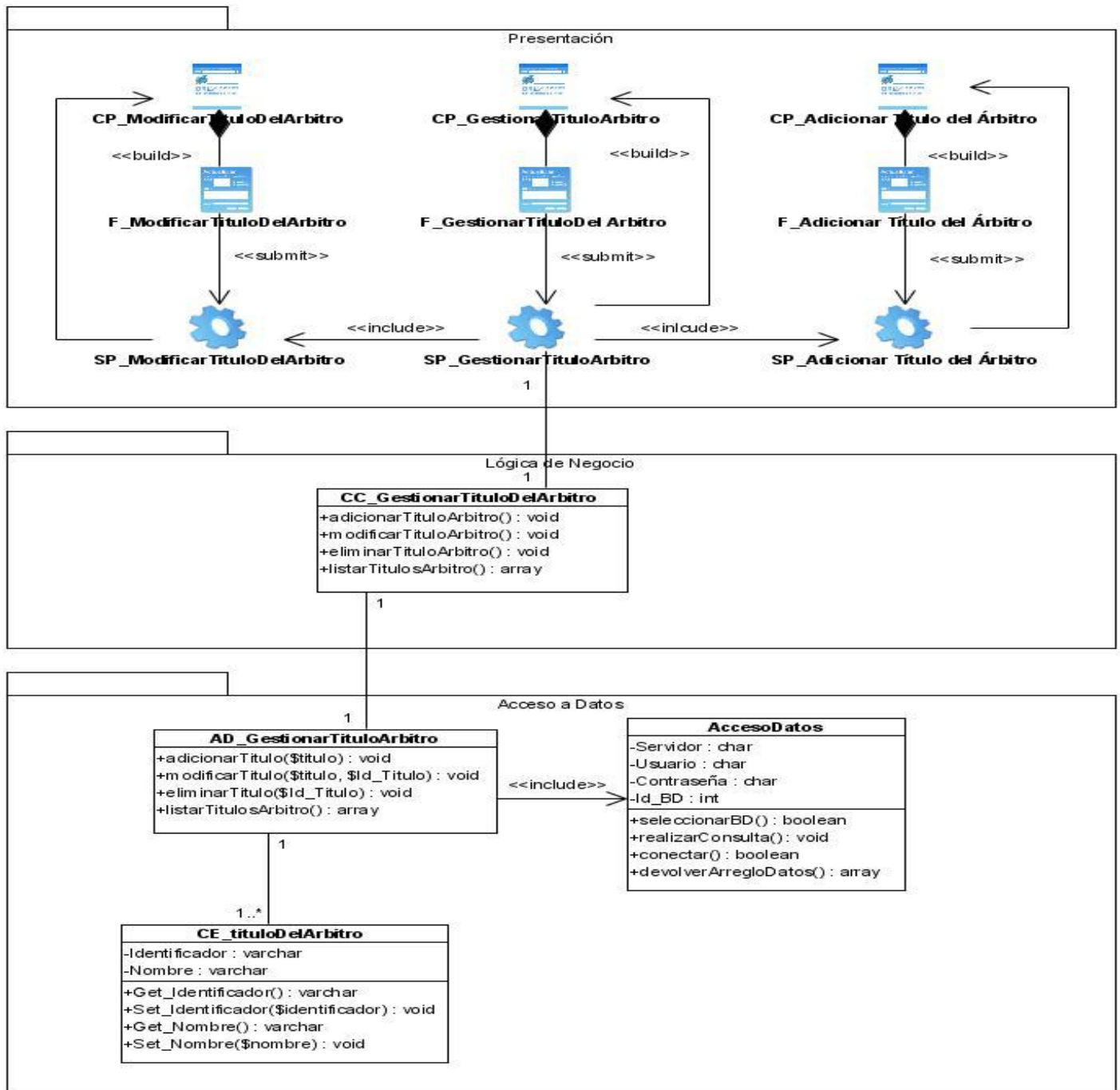




Figura 90: CU Gestionar Título del Jugador

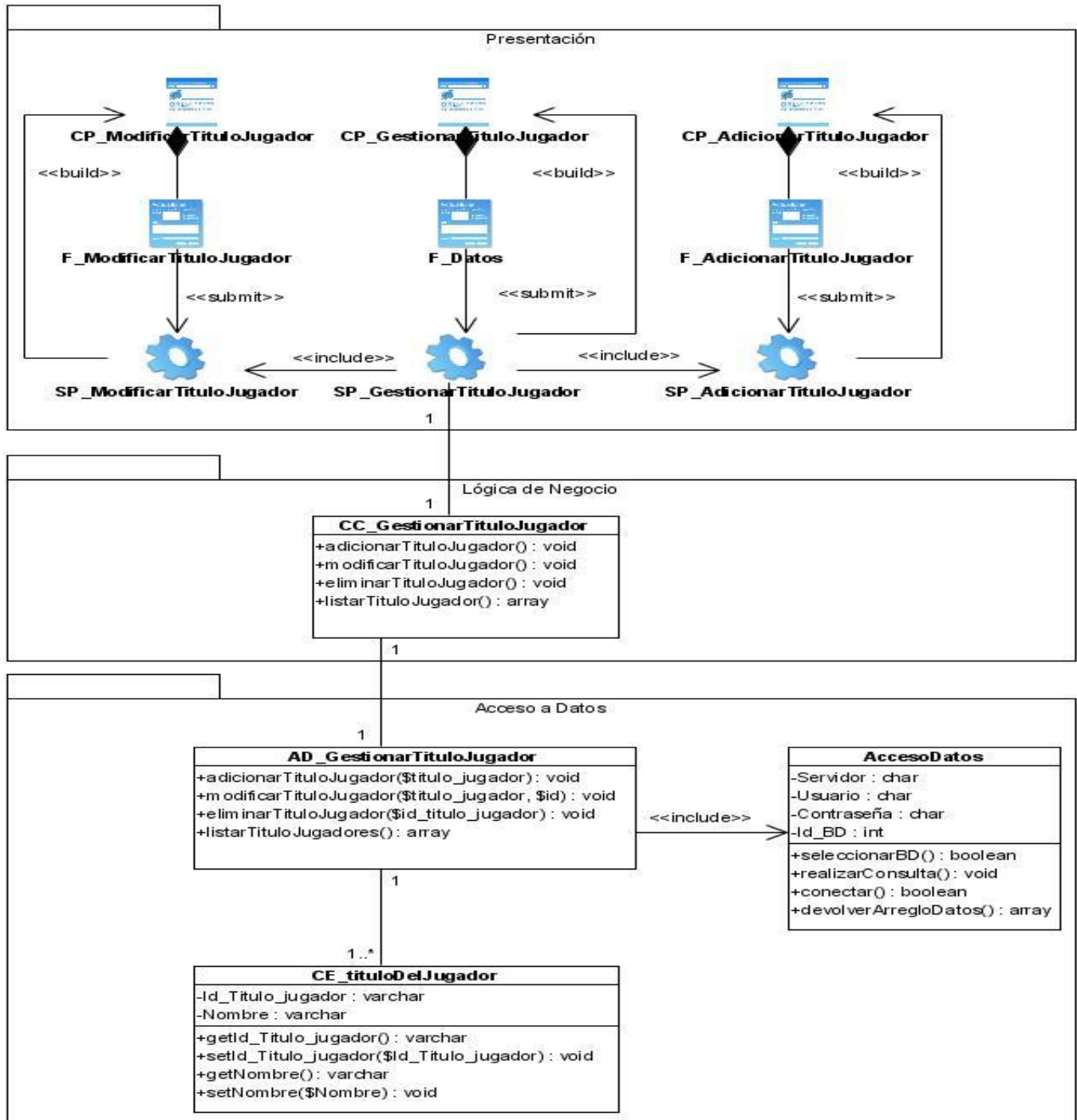






Figura 91: CU Adicionar Ritmo de Juego

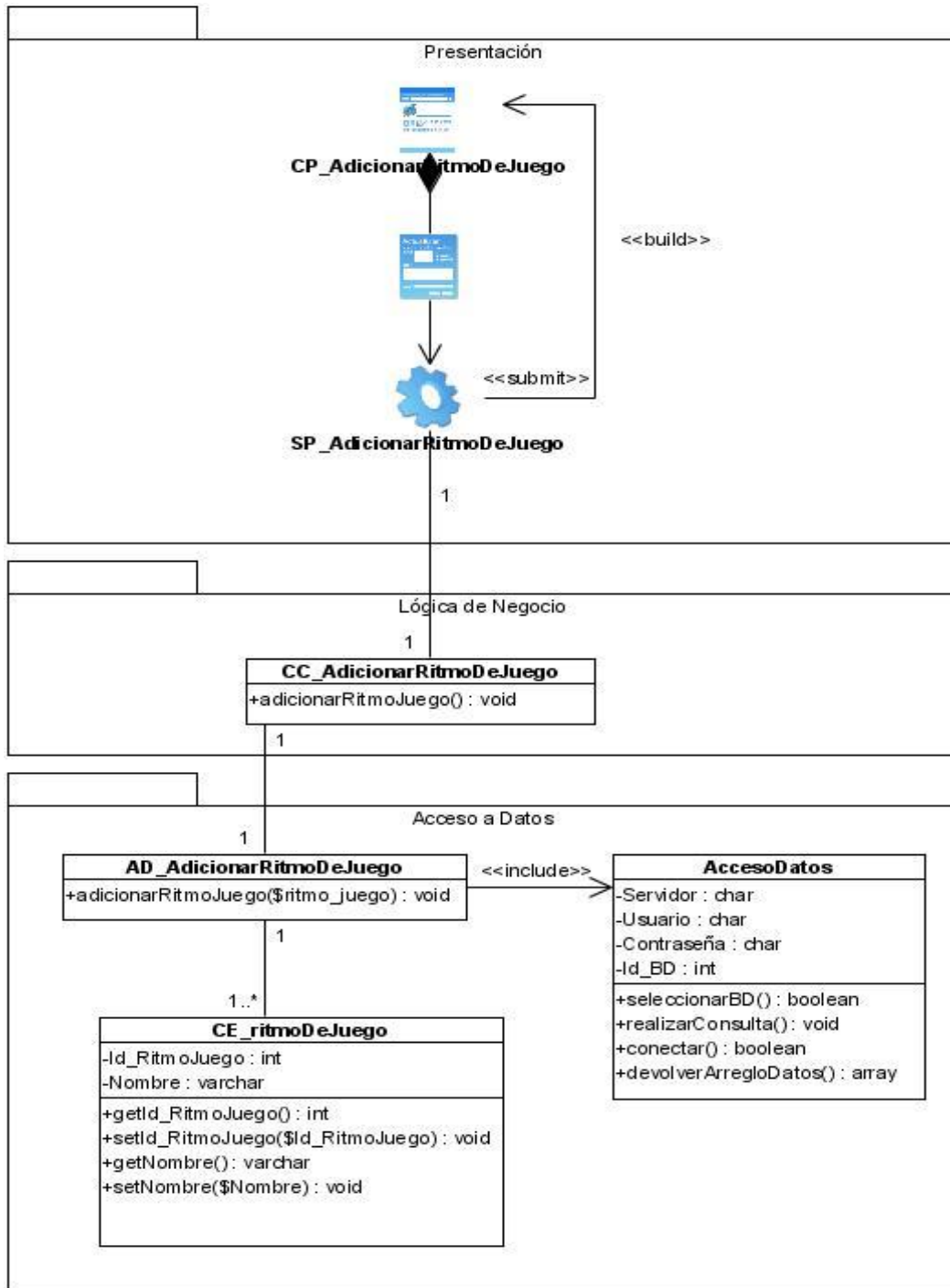




Figura 92: CU Mostrar Ritmo de Juego

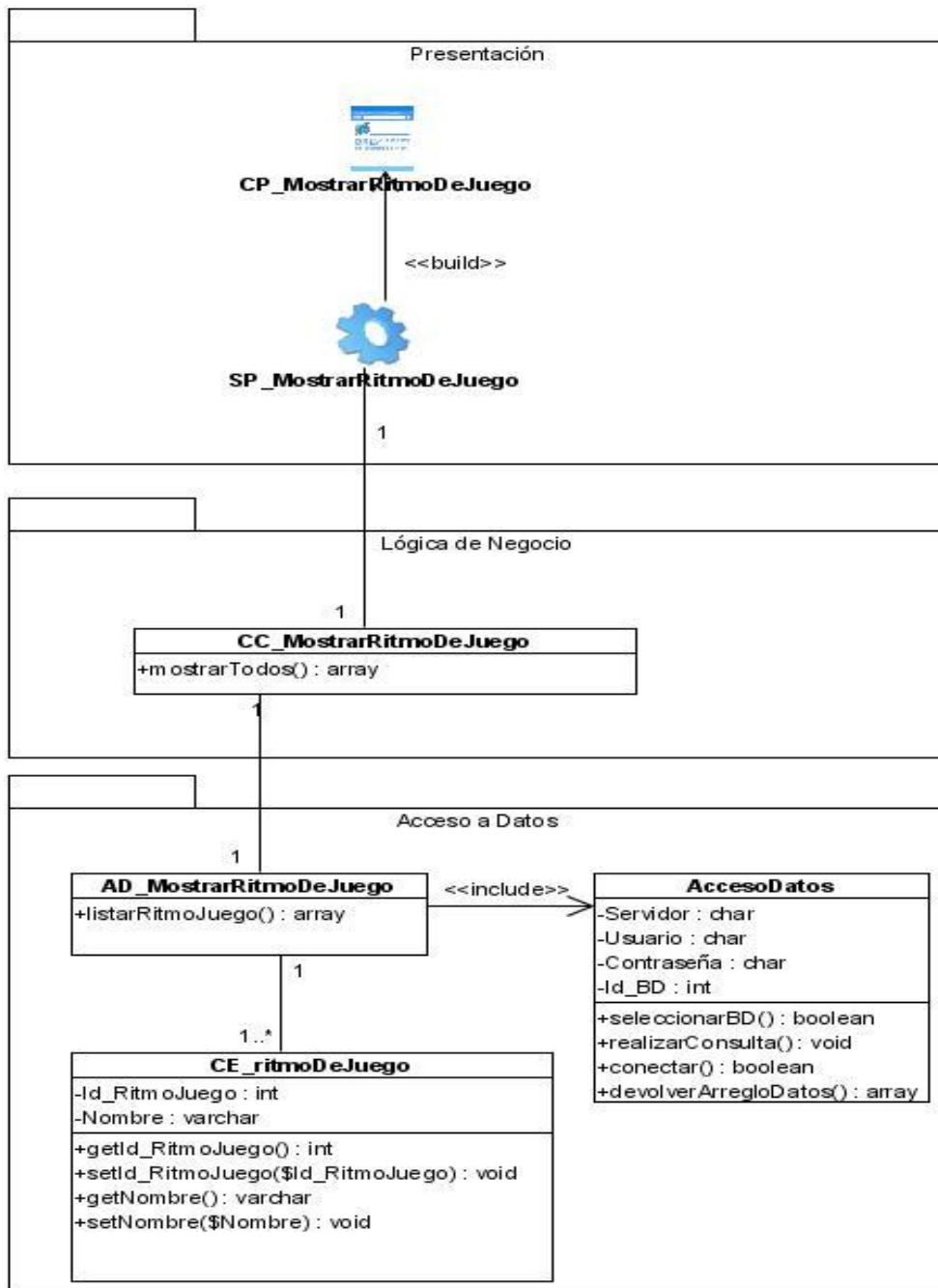




Figura 93: CU Mostrar Sistemas de desempate

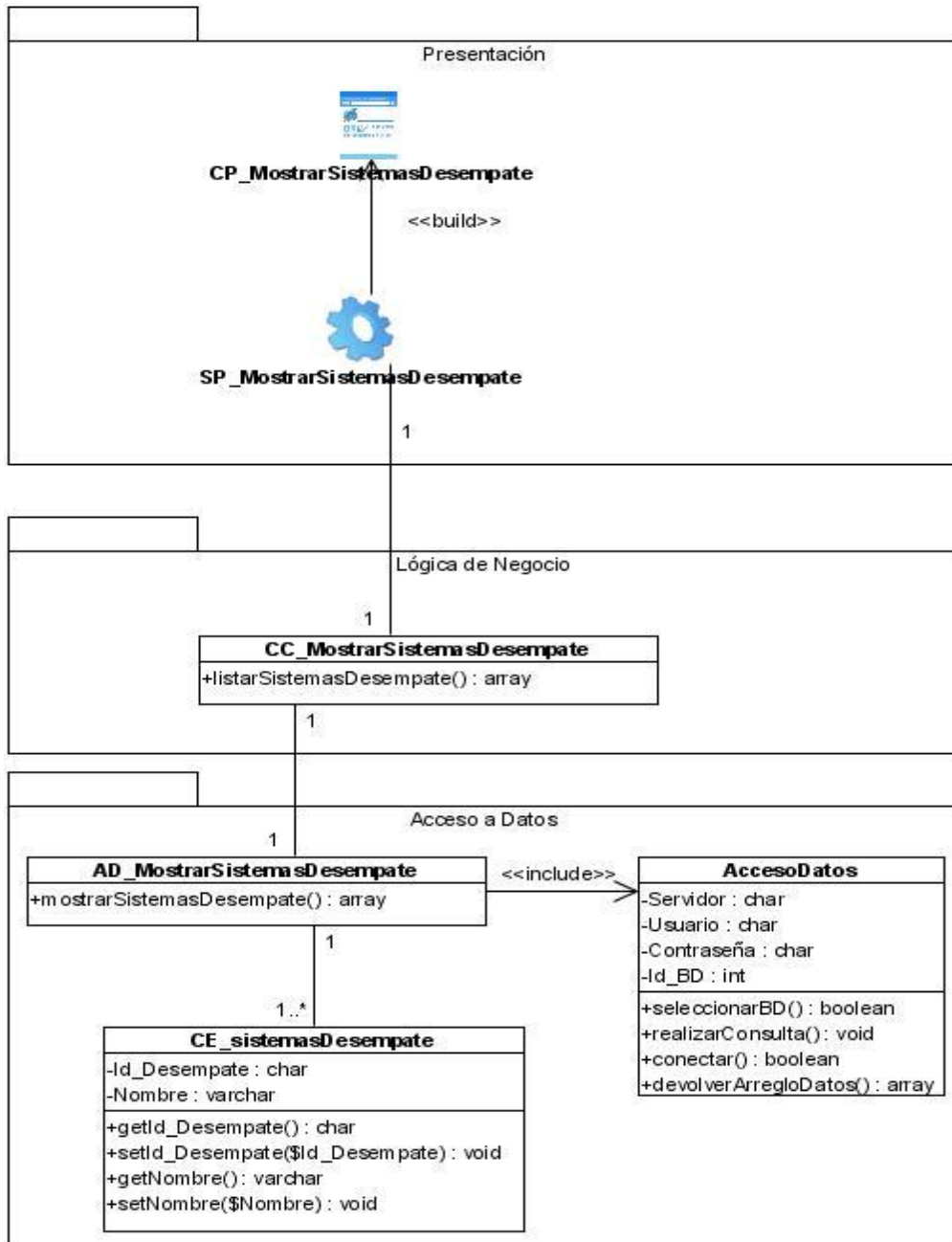




Figura 94: CU Gestionar jugadores del torneo

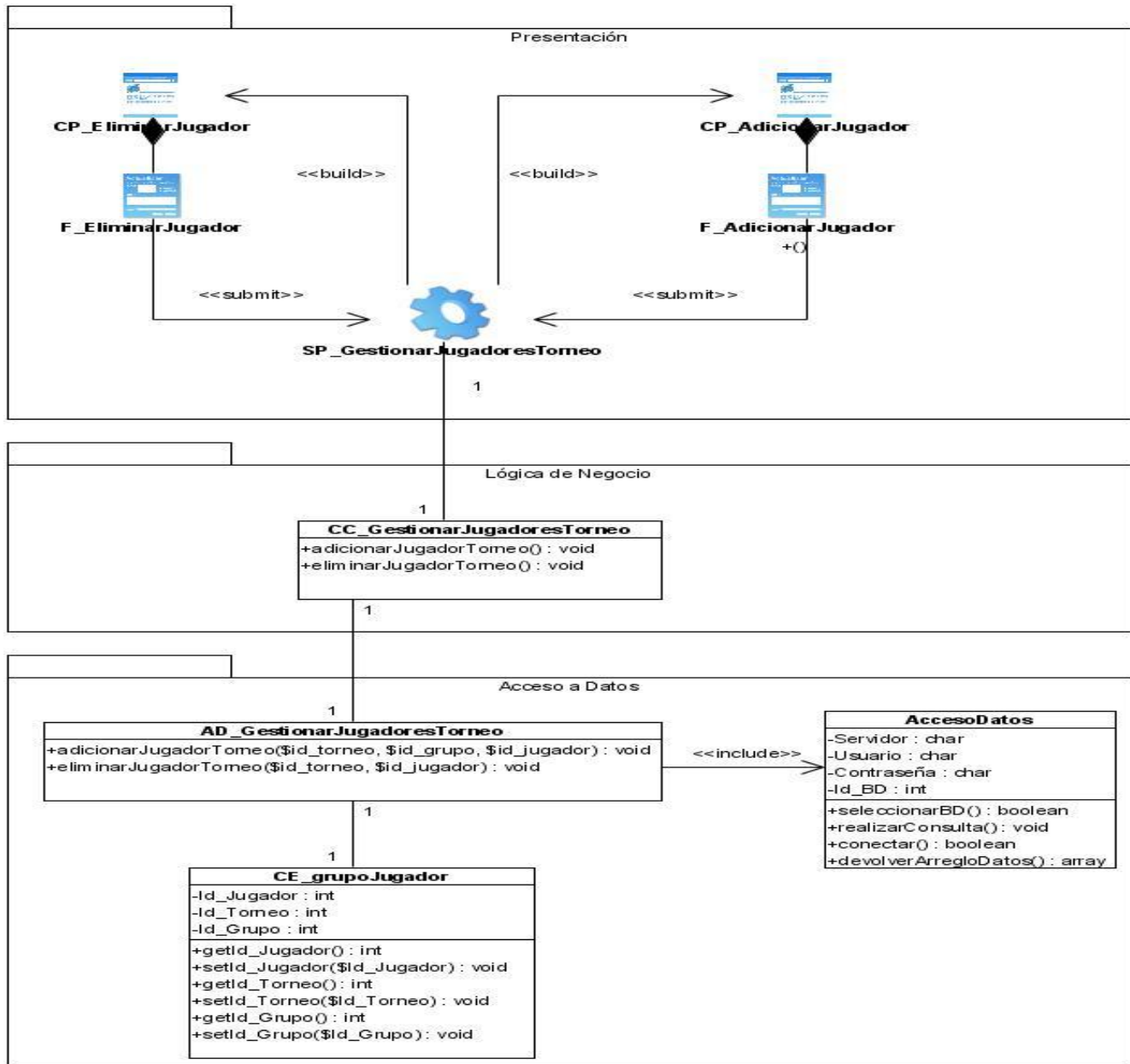




Figura 95: CU Buscar jugador

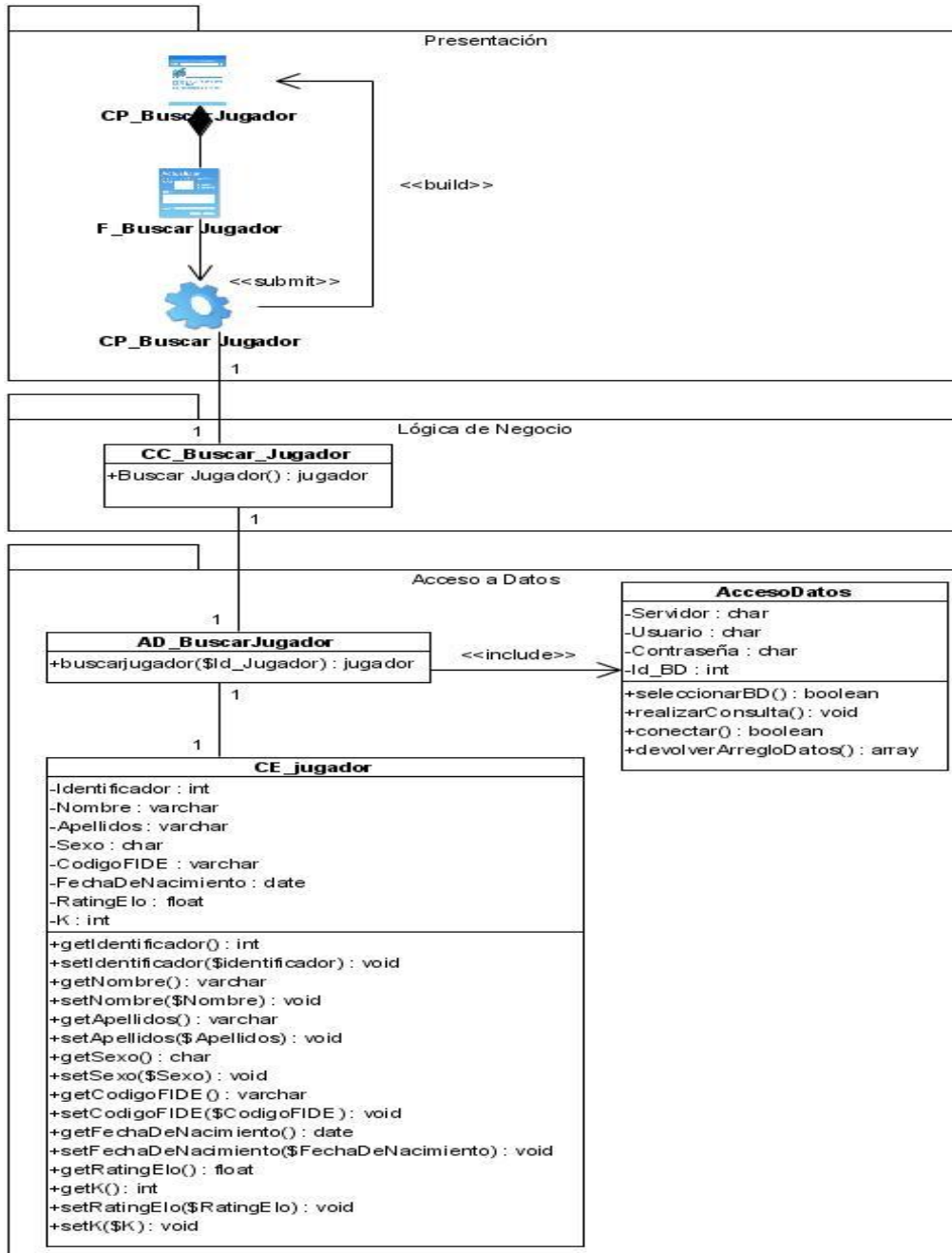




Figura 96: CU Definir Sistemas de Desempate

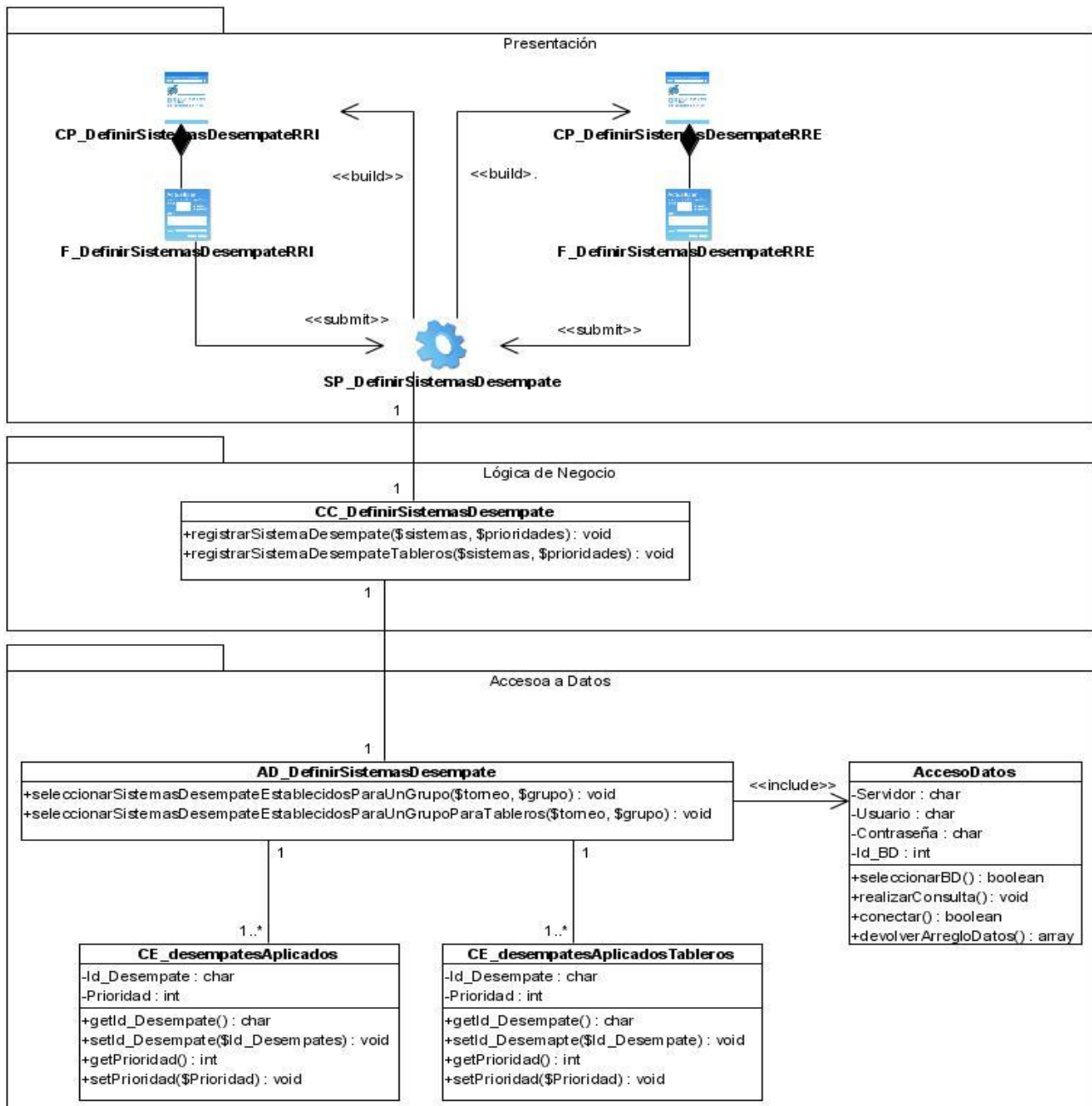






Figura 98: CU Gestionar Ronda

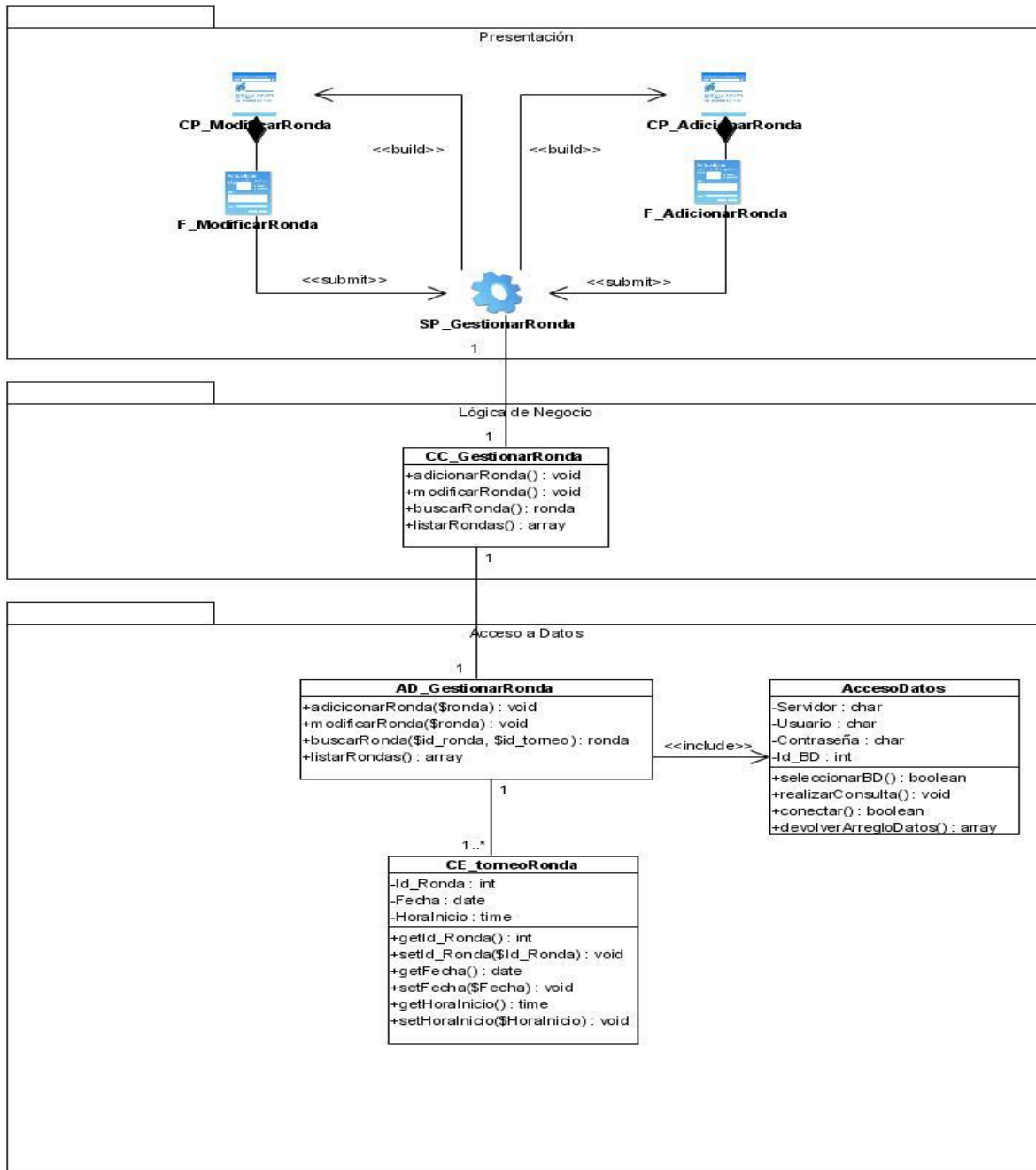






Figura 99: CU Gestionar Partida

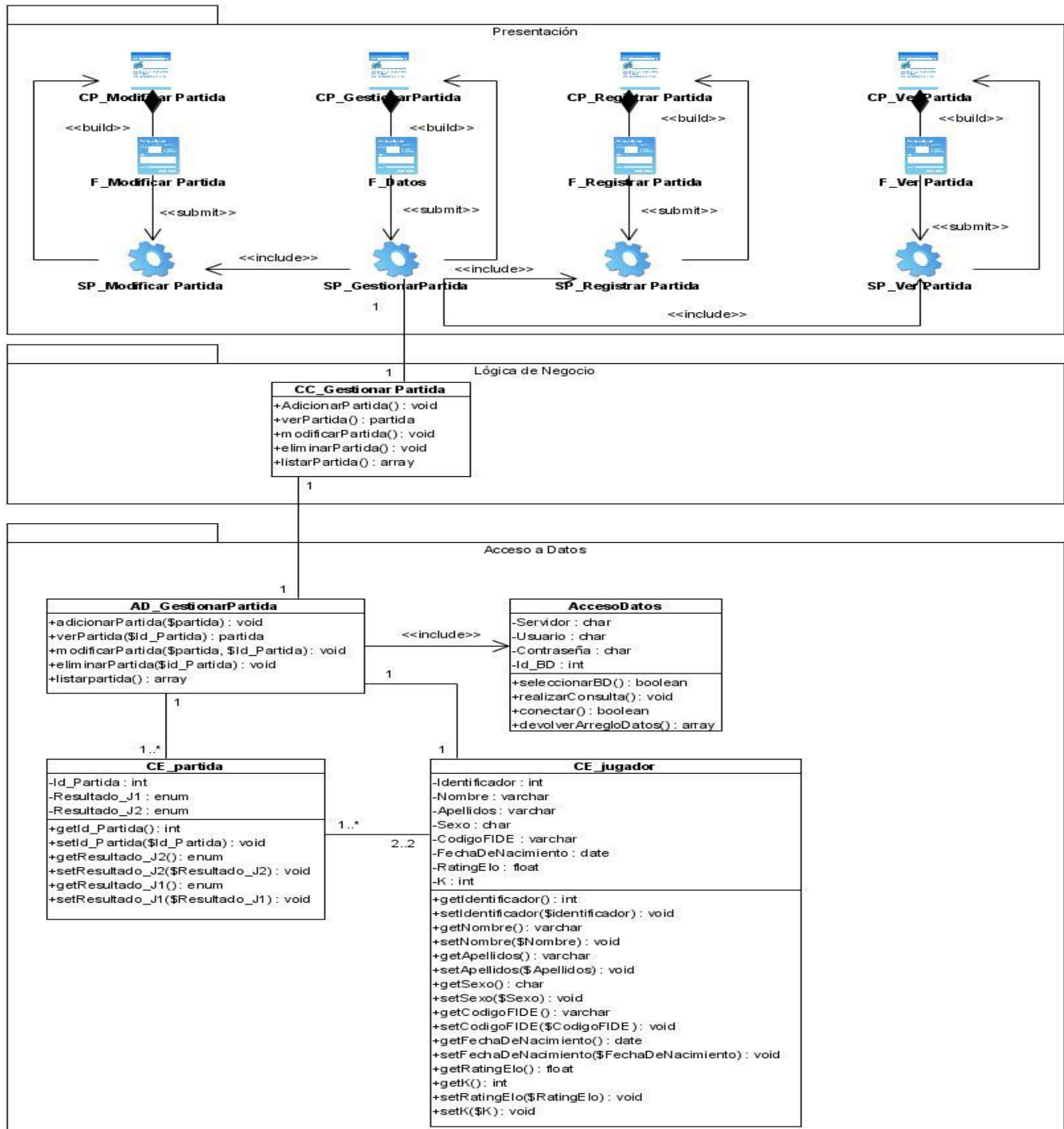




Figura 100: CU Cálculo del Rating Elo

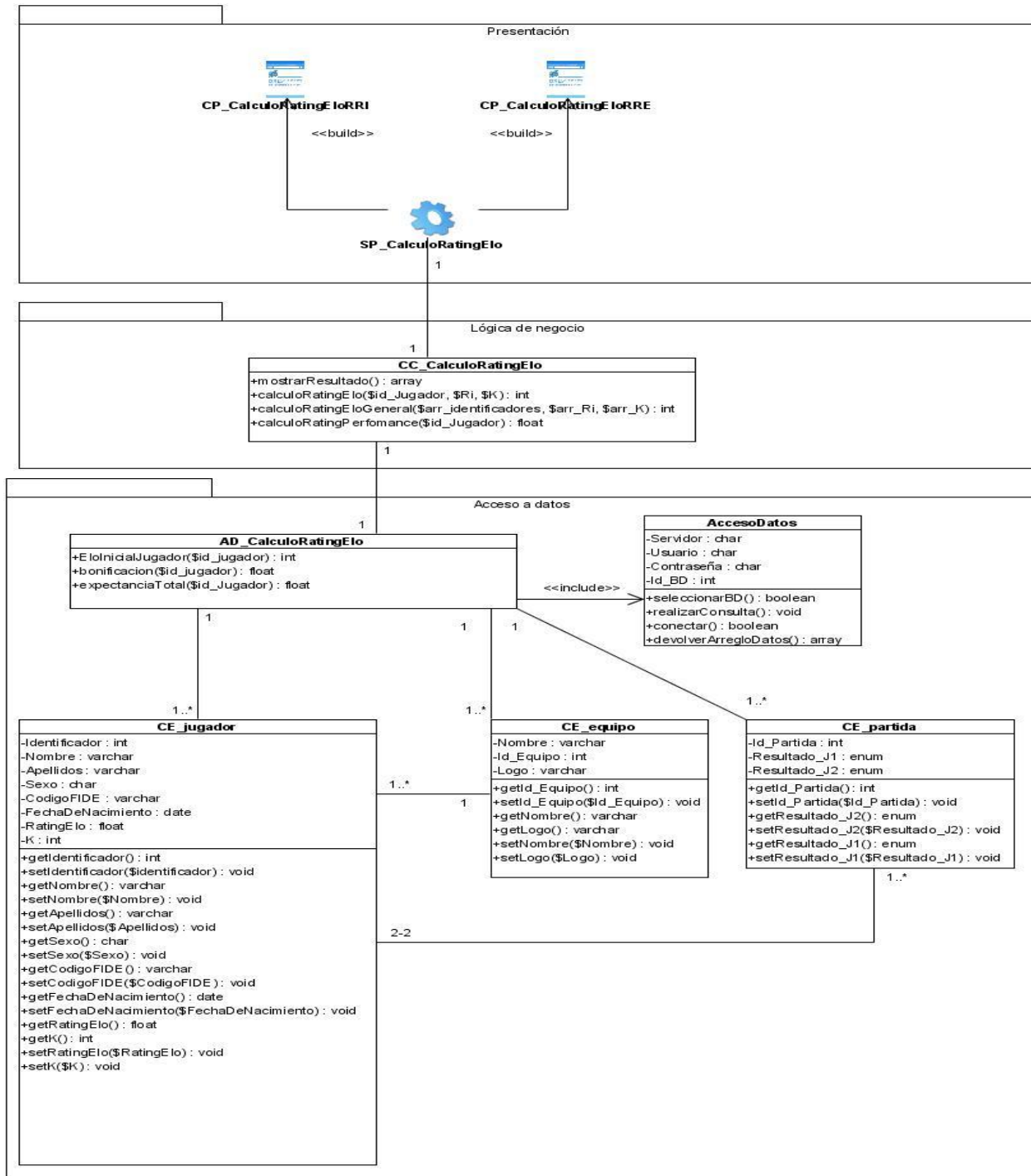




Figura 101: CU Mostrar Cuadro Sinóptico

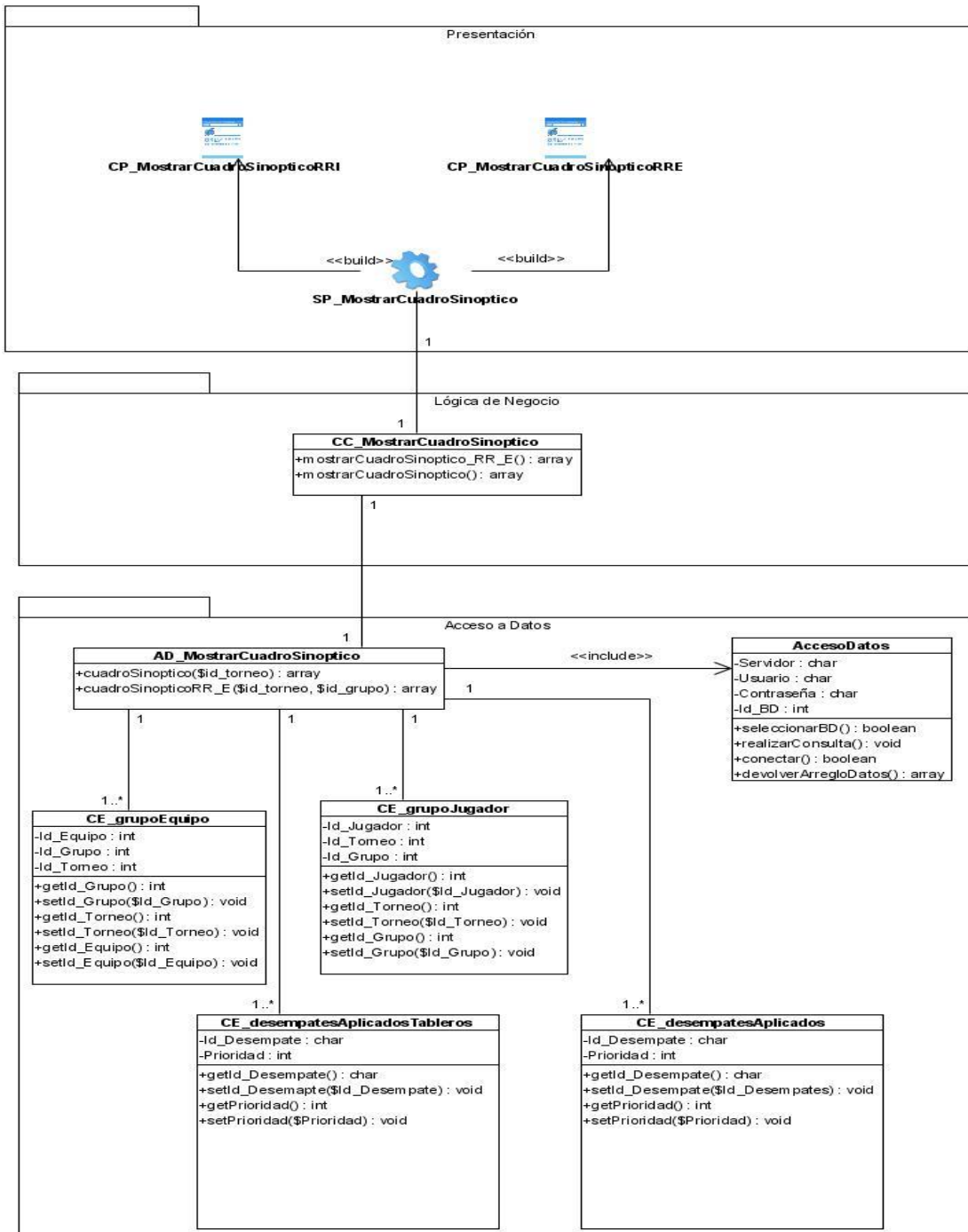




Figura 102: CU Mostrar emparejamiento y resultados

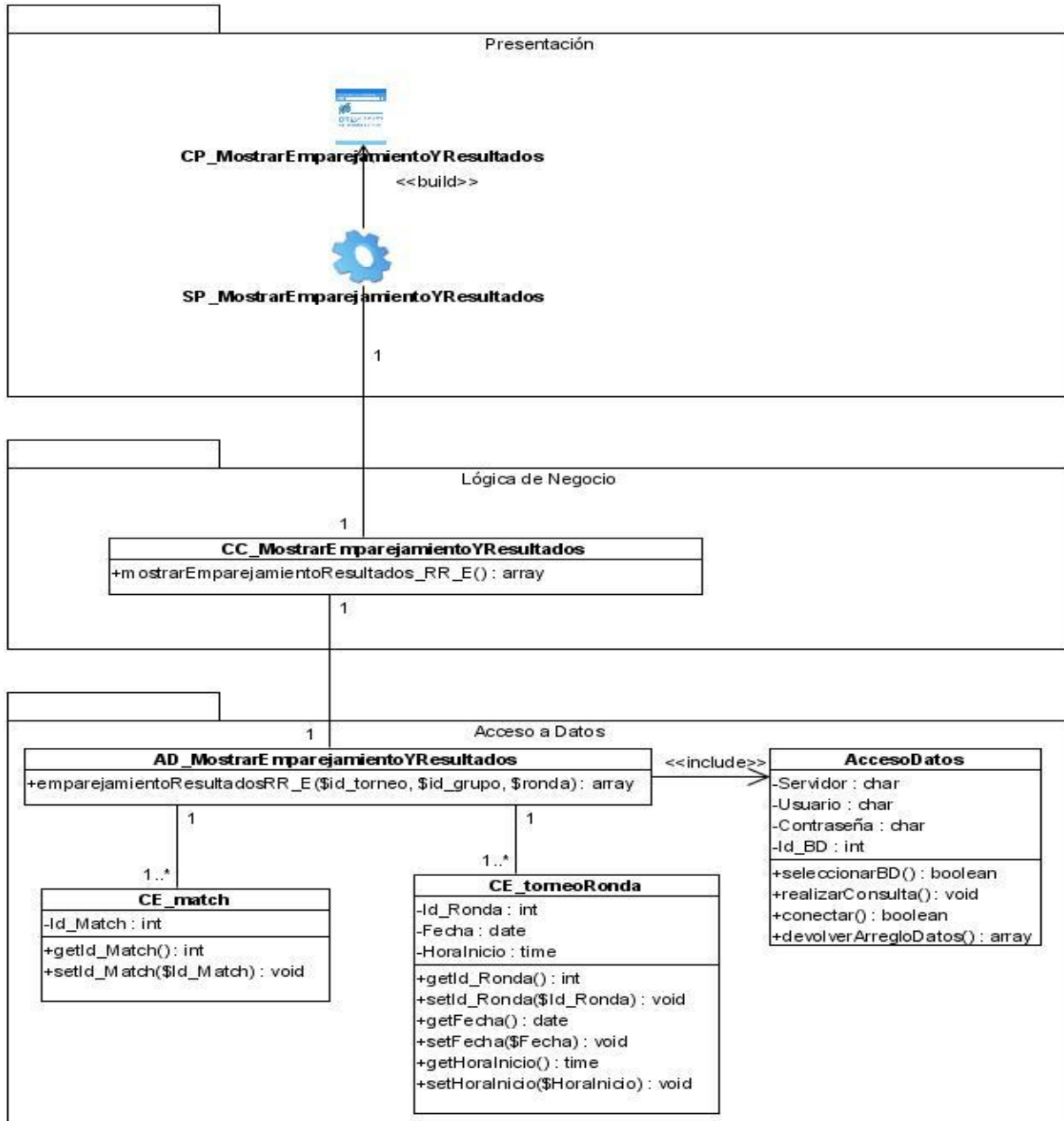




Figura 103: CU Mostrar Clasificación General

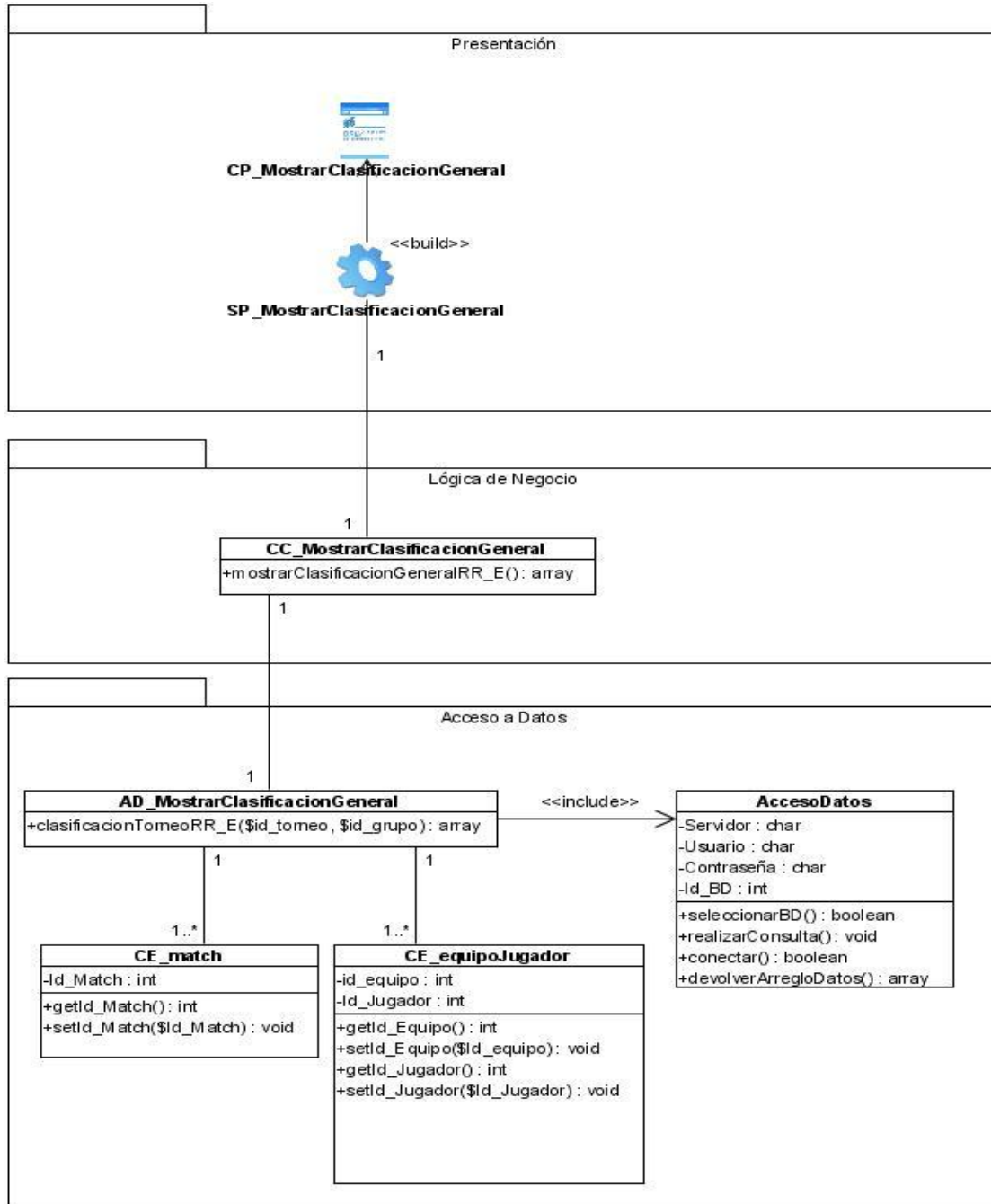




Figura 104: CU Gestionar Grupo

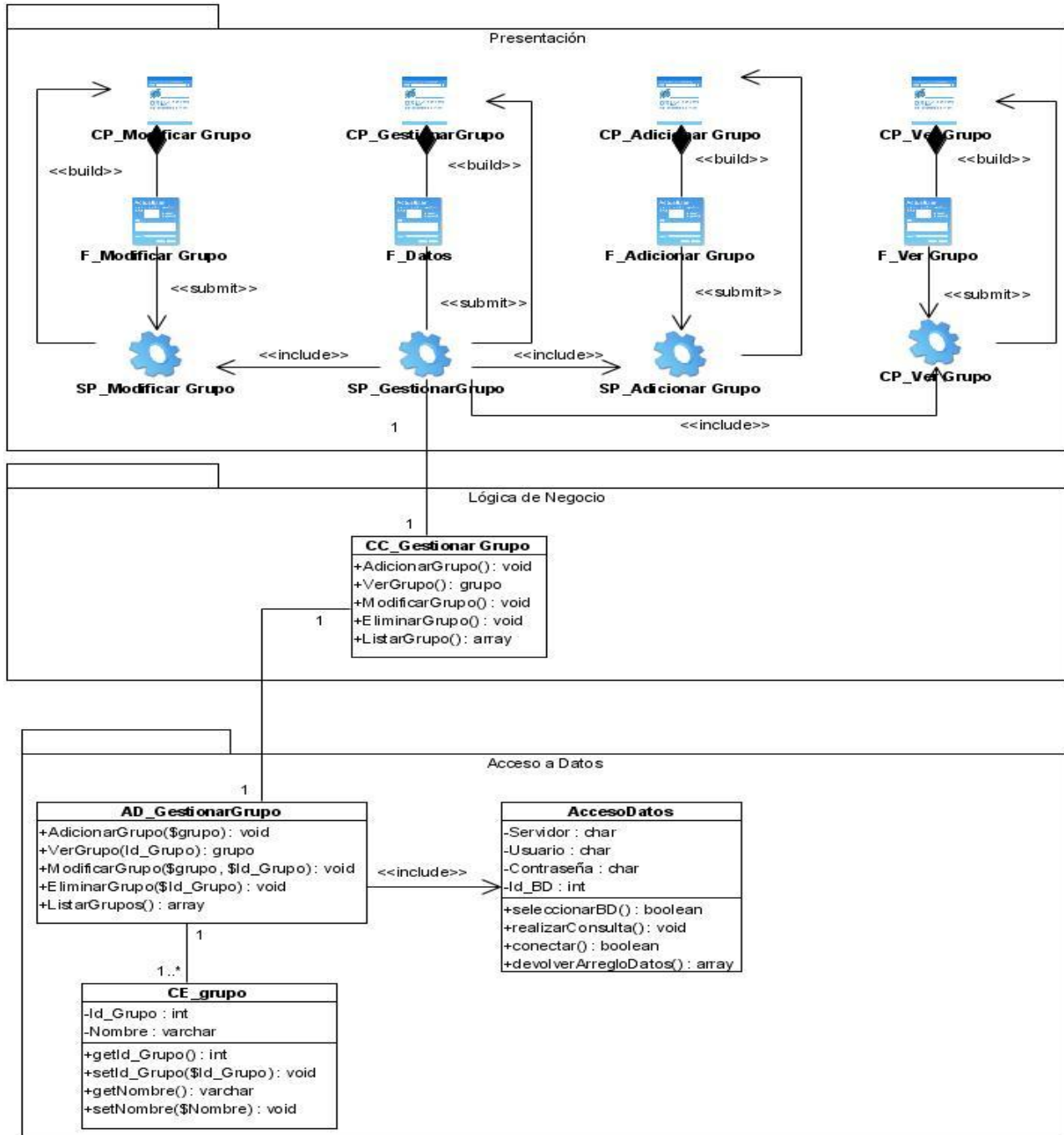




Figura 105: CU Gestionar equipos de un grupo

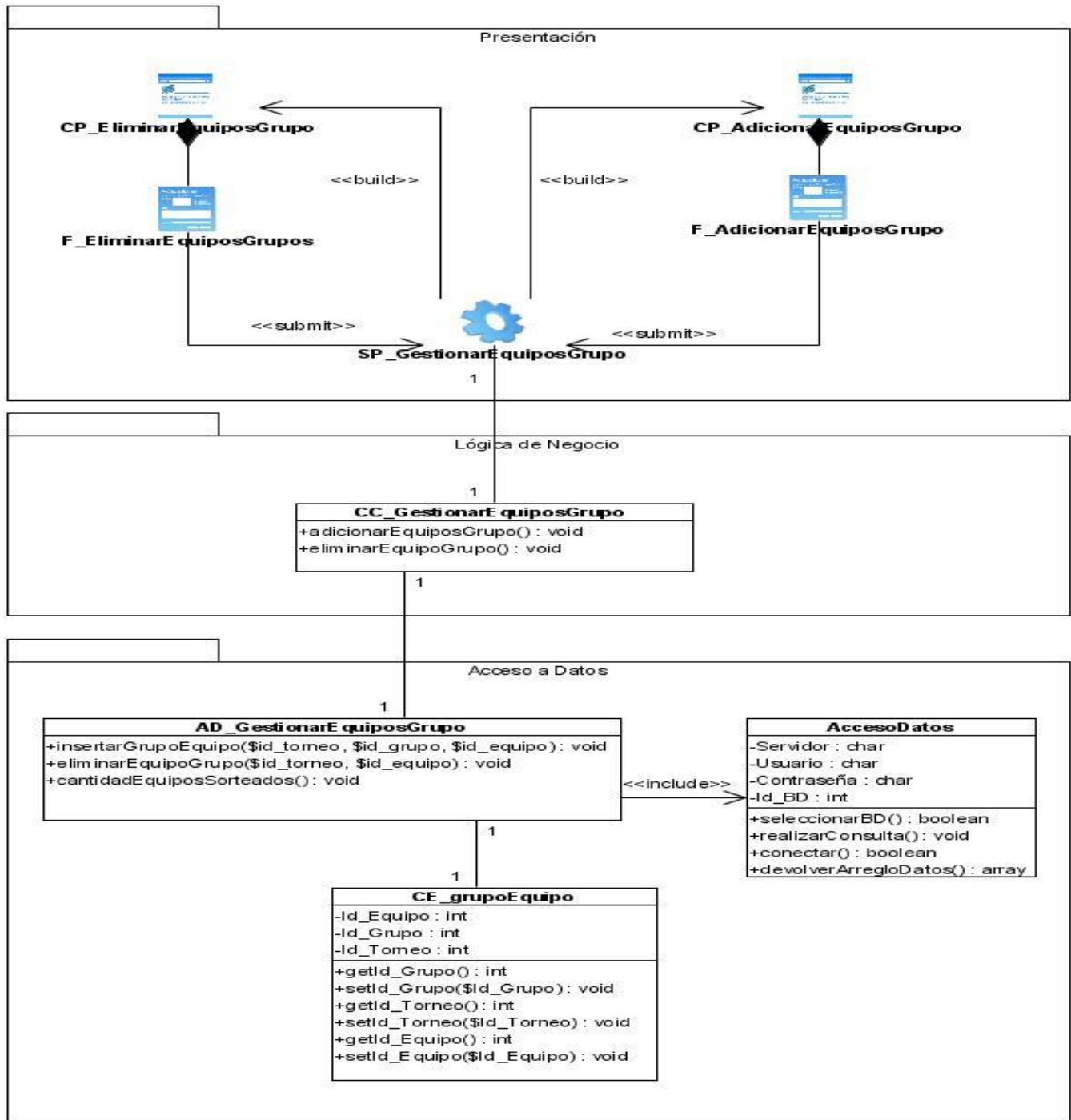




Figura 106: CU Buscar equipo

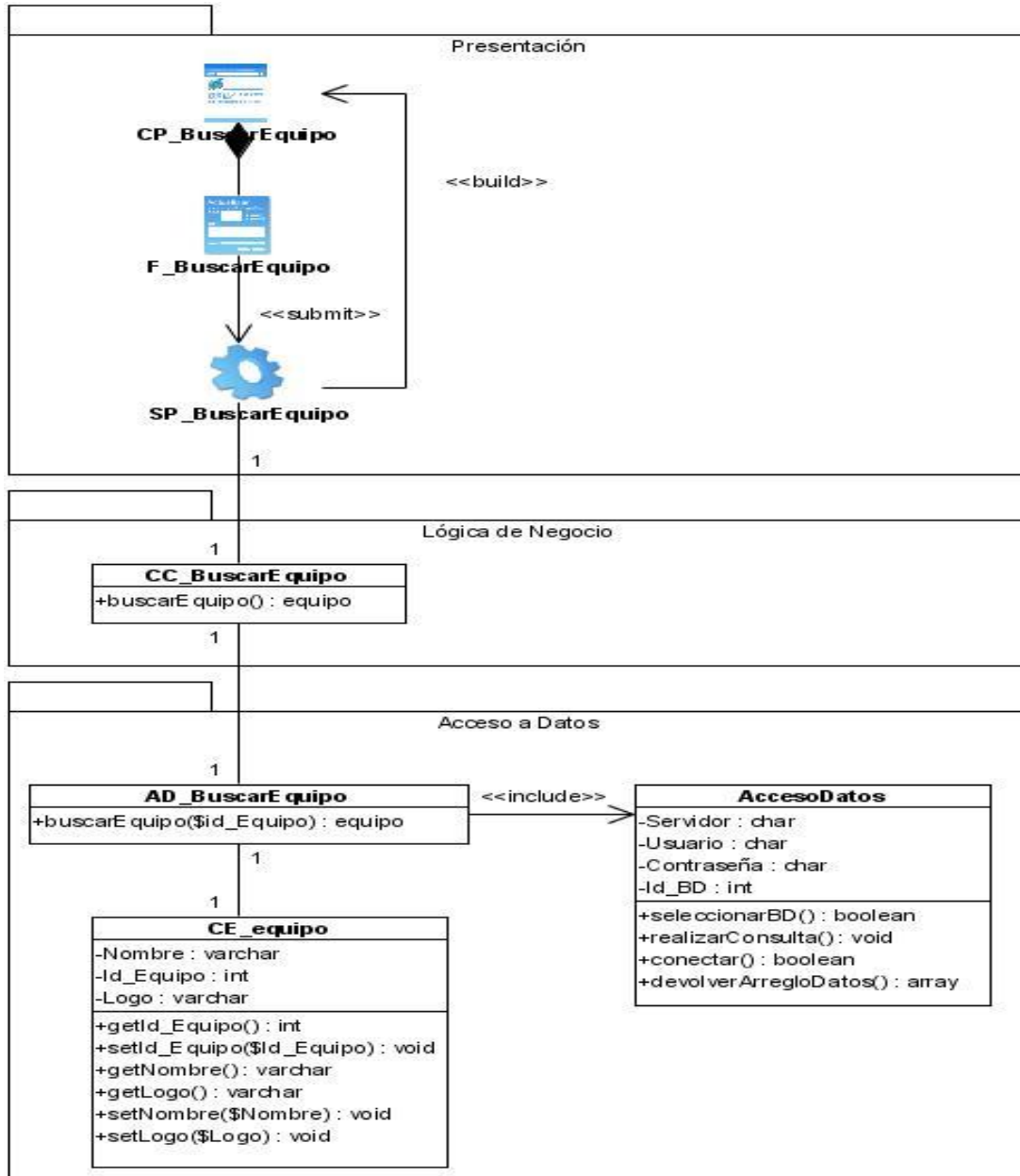






Figura 107: CU Gestionar jugadores de un equipo

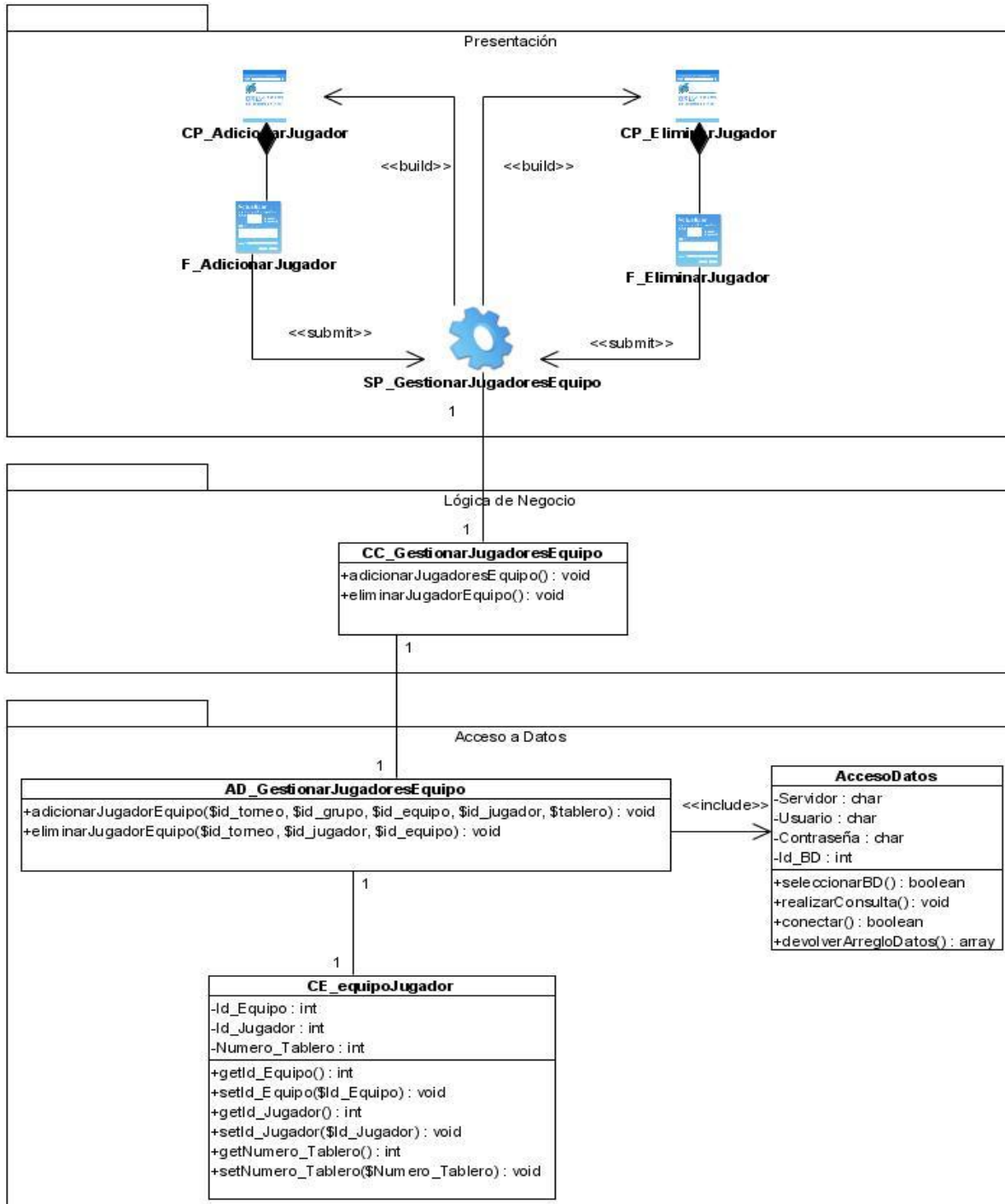




Figura 108: CU Modificar tableros

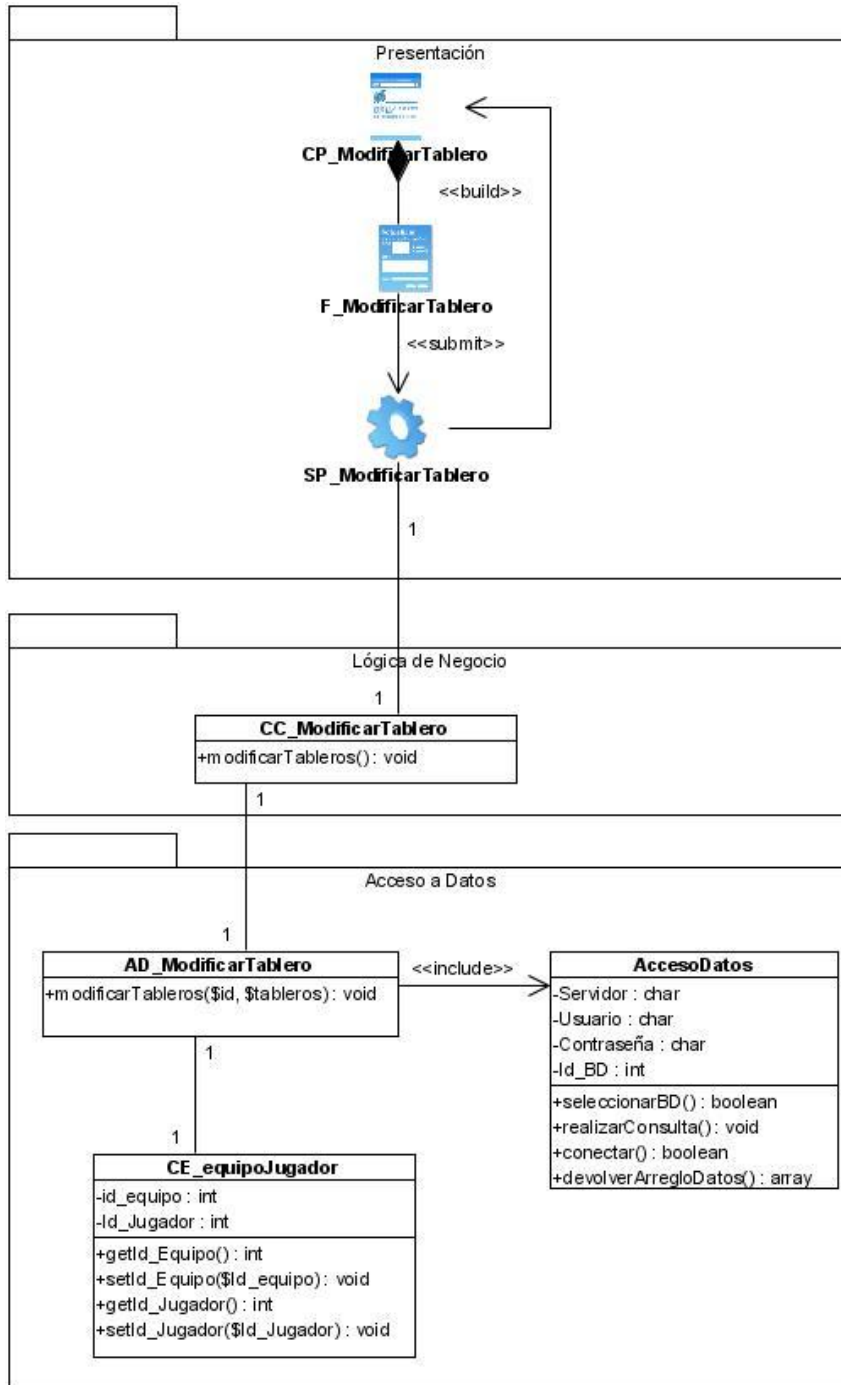




Figura 109: CU Definir Alineación

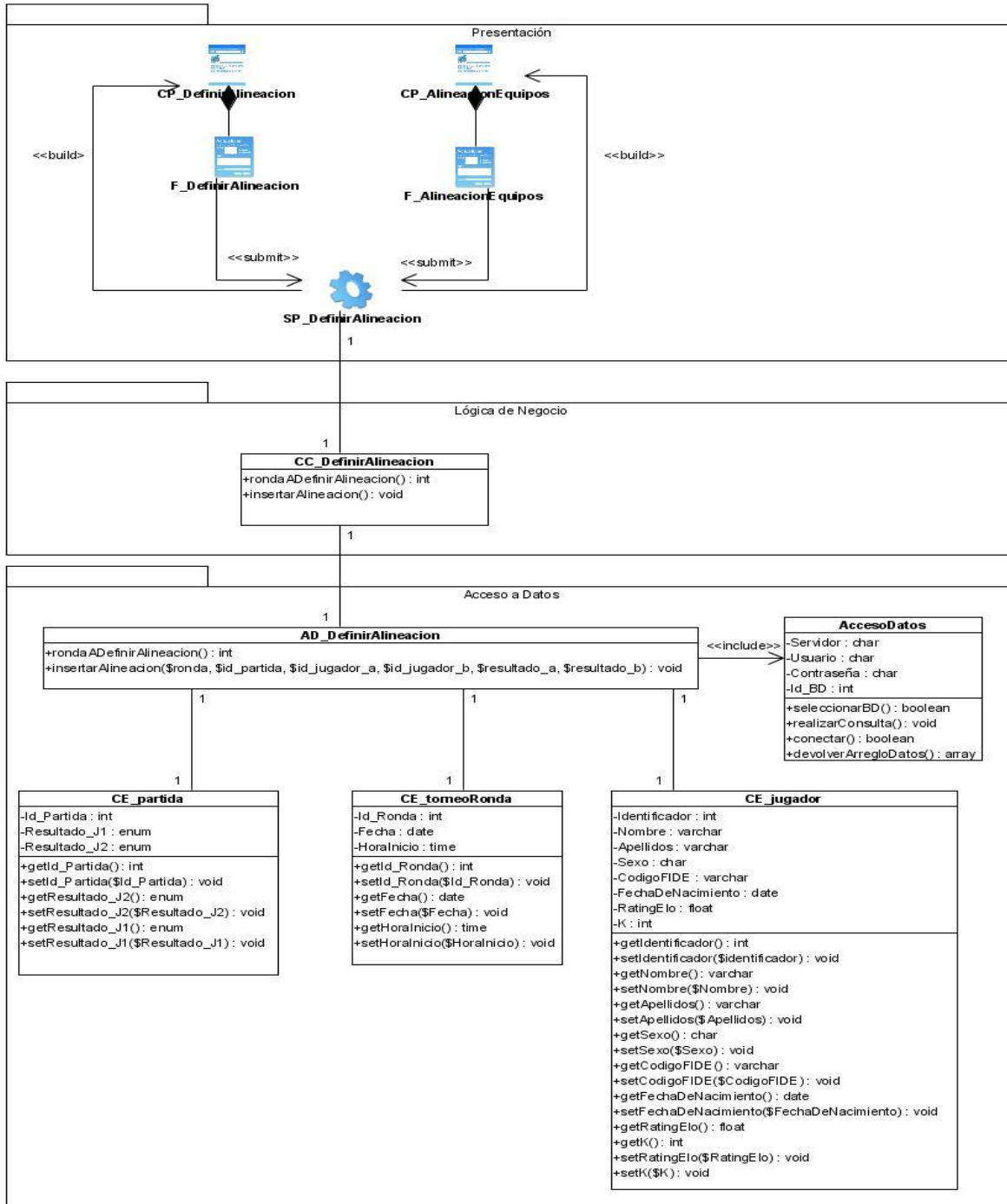




Figura 110: CU Mostrar clasificación individual por tableros

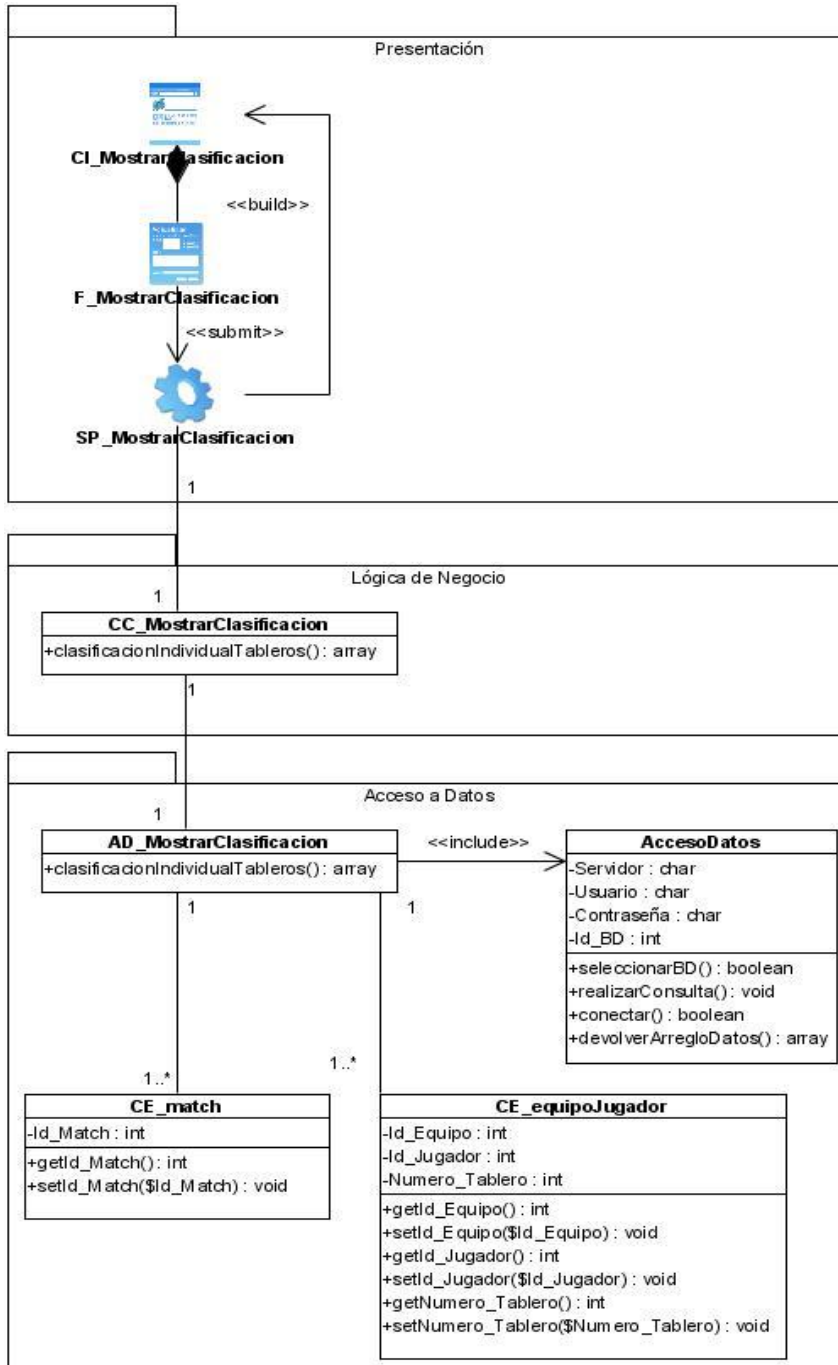




Figura 111: CU Consultar jugador

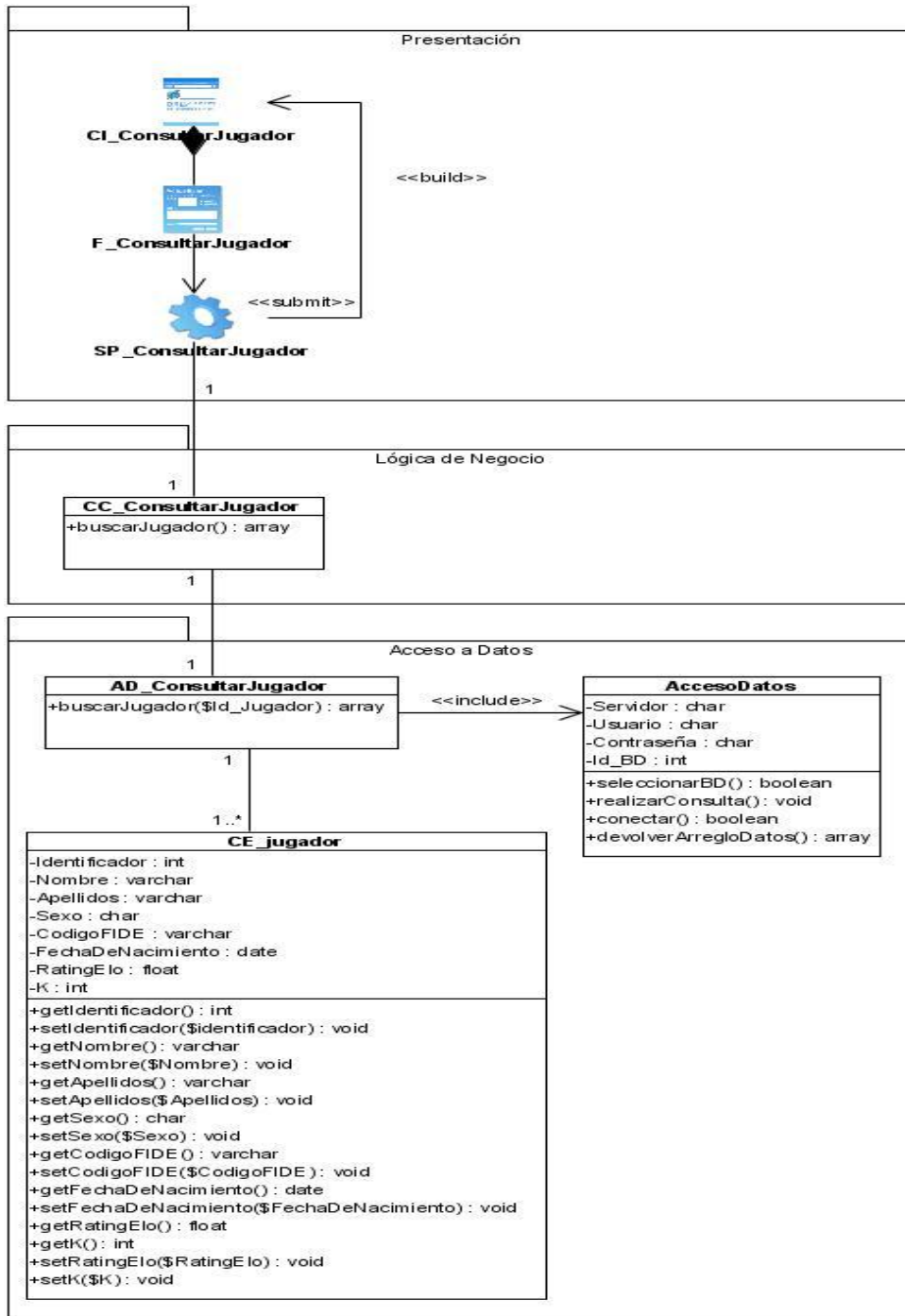




Figura 112: CU Consultar partida

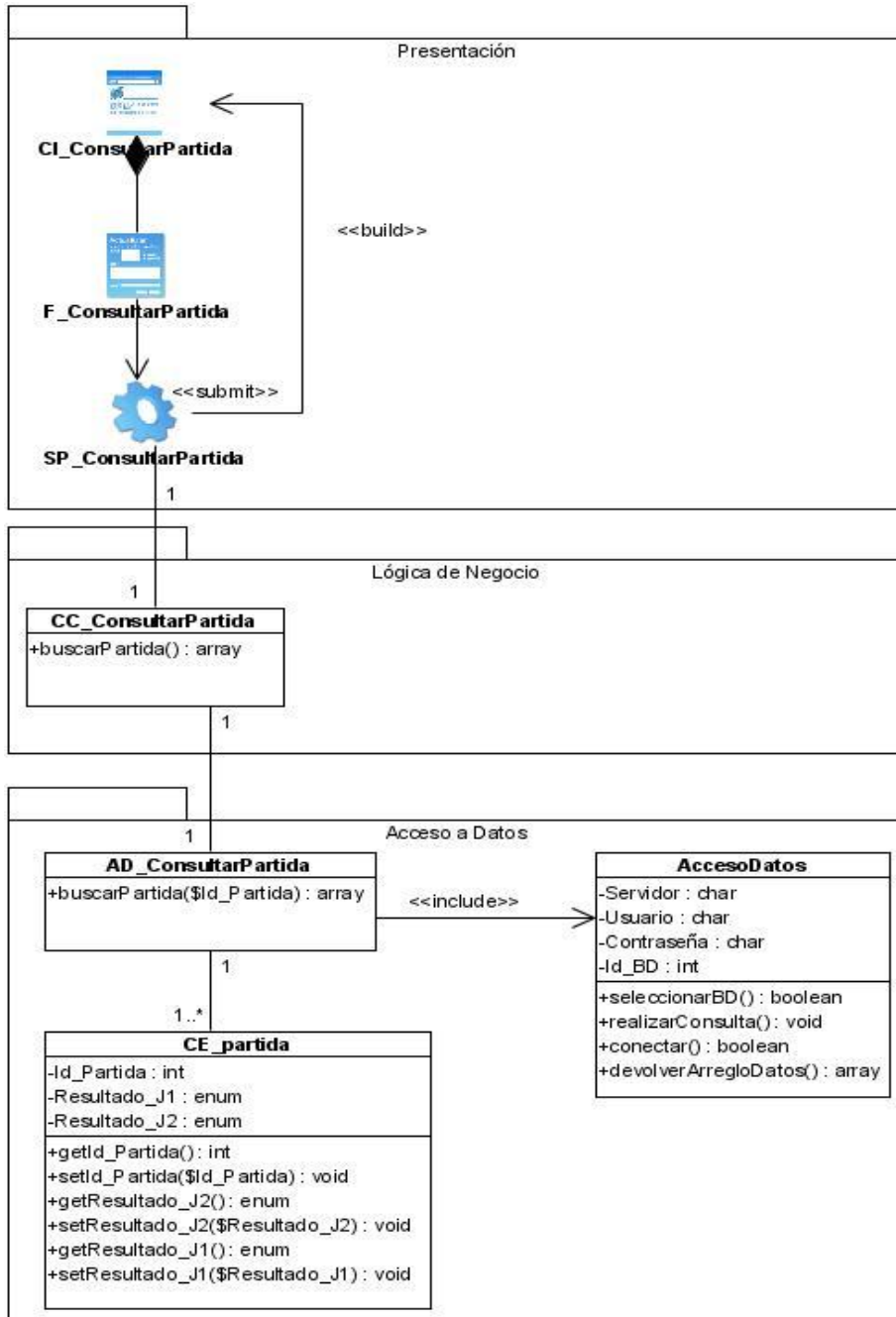




Figura 113: CU Gestionar Imagen

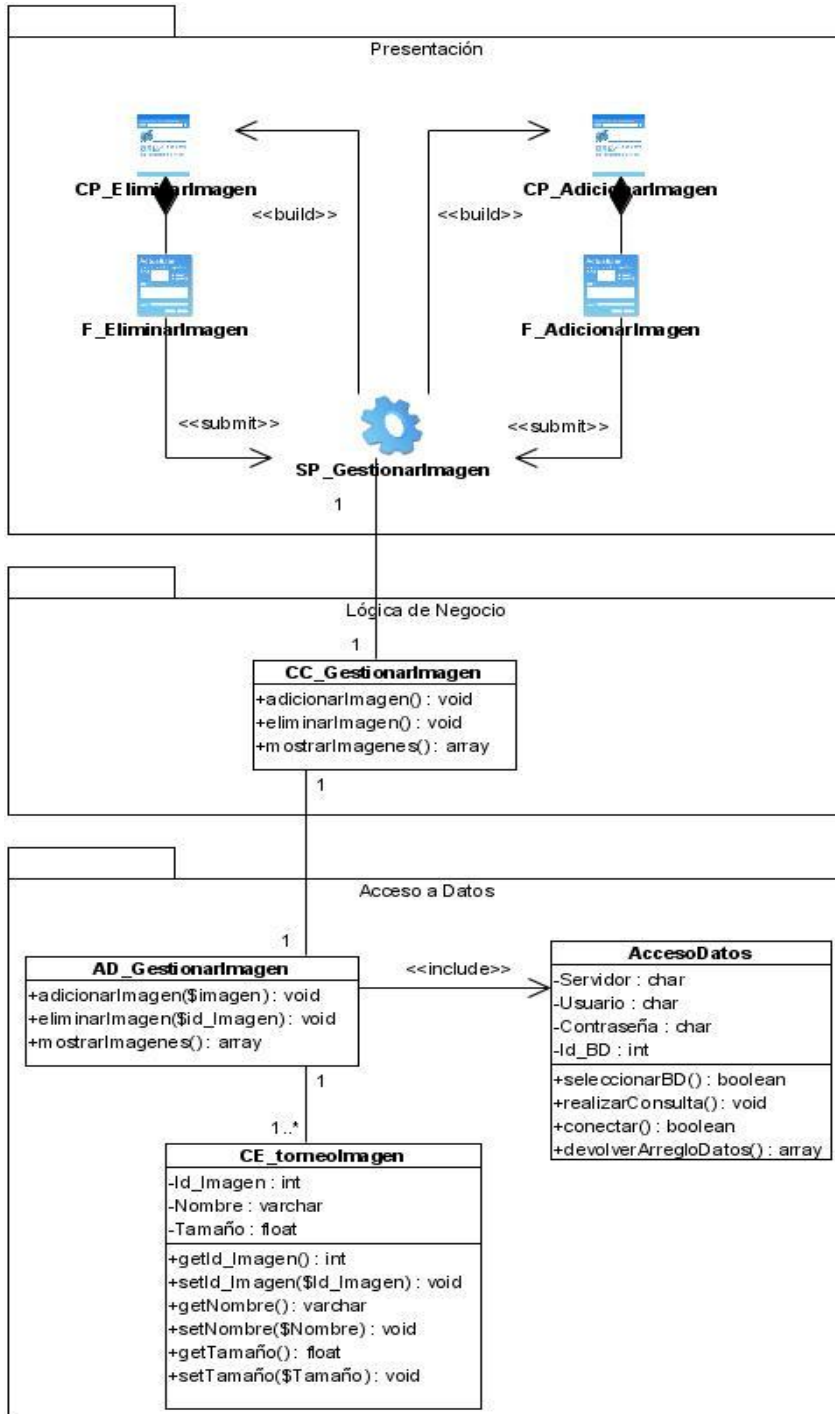
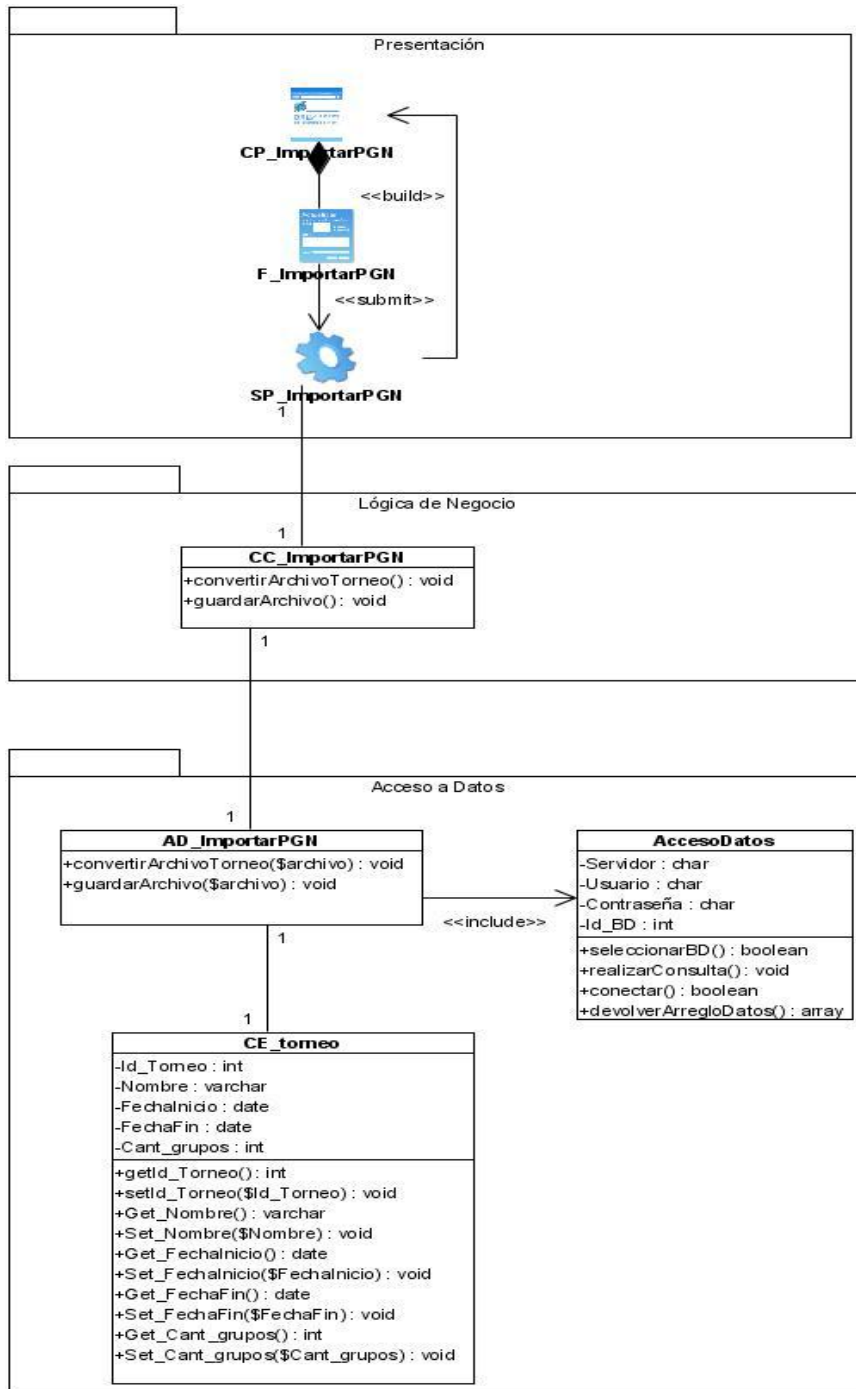




Figura 114: CU Importar PGN







**Anexo 8: Descripciones de las clases de diseño Web**

**Tabla 51:** Descripción de la clase CC\_GestionarUsuario

<b>Nombre</b>	CC_GestionarUsuario	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarUsuario ()	
	modificarUsuario()	
	verUsuario()	
	listarUsuarios()	
	eliminarUsuario()	
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a los usuarios del sistema.	

**Tabla 52:** Descripción de la clase CC\_AdicionarRitmoDeJuego

<b>Nombre</b>	CC_AdicionarRitmoDeJuego
<b>Tipo de Clase</b>	Controladora



Atributo	Tipo
<b>Responsabilidad</b>	
	adicionarRitmoDeJuego ()
<b>Descripción</b>	Es la clase que se encarga de adicionar ritmos de juego

Tabla 53: Descripción de la clase CC\_AutenticarUsuario

<b>Nombre</b>	CC_AutenticarUsuario	
<b>Tipo de Clase</b>	Controladora	
Atributo	Tipo	
<b>Responsabilidad</b>		
	darAcceso ()	
<b>Descripción</b>	Es la clase que se encarga de permitir el acceso a los usuarios del sistema	

Tabla 54: Descripción de la clase CC\_BuscarEquipo

<b>Nombre</b>	CC_BuscarEquipo
<b>Tipo de Clase</b>	Controladora



Atributo	Tipo
<b>Responsabilidad</b>	
	buscarEquipo ()
<b>Descripción</b>	Es la clase que se encarga de buscar equipos en la BD

Tabla 55: Descripción de la clase CC\_BuscarJugador

<b>Nombre</b>	CC_BuscarJugador	
<b>Tipo de Clase</b>	Controladora	
Atributo	Tipo	
<b>Responsabilidad</b>		
	buscarJugador ()	
<b>Descripción</b>	Es la clase que se encarga de buscar jugadores en la BD	

Tabla 56: Descripción de la clase CC\_ConsultarInformacionTorneo

<b>Nombre</b>	CC_ConsultarInformacionTorneo	
<b>Tipo de Clase</b>	Controladora	
Atributo	Tipo	



<b>Responsabilidad</b>	
	buscarTorneo()
<b>Descripción</b>	Es la clase que se encarga de mostrar la información del torneo

Tabla 57: Descripción de la clase CC\_CalculoRatingElo

<b>Nombre</b>	CC_CalculoRatingElo	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	mostrarResultados()	
	calculoRatingElo(\$arr_identificadores,\$arr_Ri,\$_arr_K)	
	calculoRatingEloGeneral(\$Id_jugador,\$Ri,\$K)	
	calculoRatingPerformance(\$Id_jugador)	
<b>Descripción</b>	Es la clase que se encarga de gestionar el cálculo del Elo	

Tabla 58: Descripción de la clase CC\_DefinirAlineacion

<b>Nombre</b>	CC_DefinirAlineacion
---------------	----------------------



<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	rondaADefinirAlineacion()	
	insertarAlineacion()	
<b>Descripción</b>	Es la clase que se encarga de definir la alineación entre los equipos	

Tabla 59: Descripción de la clase CC\_DefinirSistemasDesempate

<b>Nombre</b>	CC_DefinirSistemasDesempate	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	registrarSistemaDesempate(\$sistemas, \$prioridades)	
	registrarSistemaDesempateTableros(\$sistemas, \$prioridades)	
<b>Descripción</b>	Es la clase que se encarga de definir los sistemas de desempate a utilizar en el torneo	

Tabla 60: Descripción de la clase CC\_ElaborarBasesTecnicas



<b>Nombre</b>	CC_ElaborarBasesTecnicas	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	crearBases(\$nombre)	
	buscarJugadoresTorneos()	
	desempatesSeleccionados()	
<b>Descripción</b>	Es la clase que se encarga de elaborar el documento que contiene las bases técnicas del torneo	

Tabla 61: Descripción de la clase CC\_GestionarCiudad

<b>Nombre</b>	CC_GestionarCiudad	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarCiudad()	
	modificarCiudad()	



	eliminarCiudad()
	listarCiudades()
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a las ciudades

**Tabla 62:** Descripción de la clase CC\_GestionarFederacion

<b>Nombre</b>	CC_GestionarFederacion	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarFederacion()	
	modificarFederacion()	
	eliminarFederacion()	
	listarFederaciones()	
	verFederacion()	
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a las federaciones	

**Tabla 63:** Descripción de la clase CC\_GestionarTituloDelArbitro



<b>Nombre</b>	CC_GestionarTituloDelArbitro	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarTituloDelArbitro()	
	modificarTituloDelArbitro ()	
	eliminarTituloDelArbitro ()	
	listarTituloDelArbitro ()	
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a los títulos de los árbitros	

**Tabla 64:** Descripción de la clase CC\_GestionarTituloDelJugador

<b>Nombre</b>	CC_GestionarTituloDelJugador	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		





	adicionarTituloDelJugador ()
	modificarTituloDelJugador ()
	eliminarTituloDelJugador ()
	listarTituloDelJugador ()
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a los títulos de los jugadores

Tabla 65: Descripción de la clase CC\_GestionarGrupo

<b>Nombre</b>	CC_GestionarGrupo	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarGrupo ()	
	modificarGrupo ()	
	eliminarGrupo ()	
	listarGrupo ()	
	verGrupo ()	
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a los grupos	

**Tabla 66:** Descripción de la clase CC\_GestionarImagen

<b>Nombre</b>	CC_GestionarImagen	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarImagen ()	
	eliminarImagen ()	
	mostrarImagen()	
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a las imágenes del torneo	

**Tabla 67:** Descripción de la clase CC\_GestionarJugadoresEquipo

<b>Nombre</b>	CC_GestionarJugadoresEquipo	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarJugadoresEquipo ()	



	eliminarJugadoresEquipo ()
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a los jugadores de un equipo

**Tabla 68:** Descripción de la clase CC\_GestionarJugadoresTorneo

<b>Nombre</b>	CC_GestionarJugadoresTorneo	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarJugadoresTorneo ()	
	eliminarJugadoresTorneo ()	
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a los jugadores de un torneo	

**Tabla 69:** Descripción de la clase CC\_GestionarPartida

<b>Nombre</b>	CC_GestionarPartida	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	



<b>Responsabilidad</b>	
	adicionarPartida ()
	verPartida ()
	modificarPartida ()
	eliminarPartida ()
	listarPartida()
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a las partidas

Tabla 70: Descripción de la clase CC\_GestionarRonda

<b>Nombre</b>	CC_GestionarRonda	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarRonda ()	
	modificarRonda ()	
	buscarRonda ()	
	listarRonda ()	



<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a las rondas de un torneo
--------------------	---

**Tabla 71:** Descripción de la clase CC\_GestionarEquiposGrupo

<b>Nombre</b>	CC_GestionarEquiposGrupo	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarEquipos grupo ()	
	eliminarEquiposGrupo ()	
<b>Descripción</b>	Es la clase que se encarga de gestionar todo lo referente a los equipos de un grupo	

**Tabla 72:** Descripción de la clase CC\_ImportarPGN

<b>Nombre</b>	CC_ImportarPGN	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	



<b>Responsabilidad</b>	
	convertirArchivoTorneo()
	guardarArchivo()
<b>Descripción</b>	Es la clase que se encarga de cargar un archivo PGN e interpretar su información convirtiéndola en un torneo

Tabla 73: Descripción de la clase CC\_ModificarTablero

<b>Nombre</b>	CC_ModificarTablero	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	modificarTableros ()	
<b>Descripción</b>	Es la clase que se encarga de modificar los tableros de los jugadores en un equipo	

Tabla 74: Descripción de la clase CC\_MostrarClasificaciónGeneral

<b>Nombre</b>	CC_MostrarClasificaciónGeneral
---------------	--------------------------------



<b>Tipo de Clase</b>	Controladora
<b>Atributo</b>	<b>Tipo</b>
<b>Responsabilidad</b>	
	mostrarClasificaciónGeneralRR_E ()
<b>Descripción</b>	Es la clase que se encarga de mostrar el resultado de los enfrentamientos entre los equipos

**Tabla 75:** Descripción de la clase CC\_MostrarCuadroSinoptico

<b>Nombre</b>	CC_MostrarCuadroSinoptico
<b>Tipo de Clase</b>	Controladora
<b>Atributo</b>	<b>Tipo</b>
<b>Responsabilidad</b>	
	mostrarCuadroSinoptico_RR_E()
	mostrarCuadroSinoptico()
<b>Descripción</b>	Es la clase que se encarga de mostrar el resultado por rondas de los equipos con los sistemas de desempate utilizados ordenados por prioridad



**Tabla 76:** Descripción de la clase CC\_MostrarRitmoDeJuego

<b>Nombre</b>	CC_MostrarRitmoDeJuego	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	mostrarTodos()	
<b>Descripción</b>	Es la clase que se encarga de mostrar los ritmos de juegos existentes en la base de datos	

**Tabla 77:** Descripción de la clase CC\_MostrarSistemasDesempate

<b>Nombre</b>	CC_MostrarSistemasDesempate	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	listarSistemasDesempate()	
<b>Descripción</b>	Es la clase que se encarga de mostrar los sistemas de desempates registrados en la base de datos	



**Tabla 78:** Descripción de la clase CC\_RealizarEmparejamiento

<b>Nombre</b>	CC_RealizarEmparejamiento	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	realizarPareo()	
	realizarPareoEquipo()	
	pareoNumSorteo()	
	pareoNumSorteoEquipo()	
<b>Descripción</b>	Es la clase que se encarga de realizar el emparejamiento	

**Tabla 79:** Descripción de la clase CC\_RealizarSorteo

<b>Nombre</b>	CC_RealizarSorteo	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	sortear()	



	sortearEquipo()
	obtenerJugadores()
	deshacerSorteo()
	deshacerSorteoEquipo()
<b>Descripción</b>	Es la clase que se encarga de realizar el sorteo de los jugadores y los equipos según el tipo de torneo

**Tabla 80:** Descripción de la clase CC\_MostrarEmparejamientoYResultados

<b>Nombre</b>	CC_MostrarEmparejamientoYResultados	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	mostrarEmparejamientoResultados_RR_E()	
<b>Descripción</b>	Es la clase que se encarga de mostrar los sistemas de desempates registrados en la base de datos	

**Tabla 81:** Descripción de la clase CC\_ConsultarPartida

<b>Nombre</b>	CC_ConsultarPartida
---------------	---------------------



<b>Tipo de Clase</b>	Controladora
<b>Atributo</b>	<b>Tipo</b>
<b>Responsabilidad</b>	
	buscarPartida()
<b>Descripción</b>	Es la clase que se encarga de mostrar los datos de una partida seleccionada

Tabla 82: Descripción de la clase CC\_ConsultarJugador

<b>Nombre</b>	CC_ConsultarJugador
<b>Tipo de Clase</b>	Controladora
<b>Atributo</b>	<b>Tipo</b>
<b>Responsabilidad</b>	
	buscarJugador()
<b>Descripción</b>	Es la clase que se encarga de mostrar los datos de un jugador seleccionado

Tabla 83: Descripción de la clase CC\_MostrarClasificacion



<b>Nombre</b>	CC_MostrarClasificacion	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	clasificacionIndividualTableros()	
<b>Descripción</b>	Es la clase que se encarga de mostrar la clasificación individual por tableros de los jugadores de un equipo	

Tabla 84: Descripción de la clase AccesoDatos

<b>Nombre</b>	AccesoDatos	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
Servidor	char	
Usuario	char	
Contraseña	char	
Id_BD	int	
<b>Responsabilidad</b>		
	ejecutarConsulta()	



	seleccionarBD()
	conectar()
	DevolverArregloDatos()
<b>Descripción</b>	Clase encargada de realizar la conexión a la base de datos y ejecutar las consultas.

**Tabla 85:** Descripción de la clase AD\_AdicionarRitmoDeJuego

<b>Nombre</b>	AD_AdicionarRitmoDeJuego	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarRitmoJuego(\$ritmo_juego)	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Adicionar Ritmo de Juego.	

**Tabla 86:** Descripción de la clase AD\_AutenticarUsuario

<b>Nombre</b>	AD_AutenticarUsuario
<b>Tipo de Clase</b>	AD



Atributo	Tipo
<b>Responsabilidad</b>	
	darAcceso(\$usuario, \$contraseña)
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Autenticar Usuario.

Tabla 87: Descripción de la clase AD\_ BuscarEquipo

<b>Nombre</b>	AD_ BuscarEquipo	
<b>Tipo de Clase</b>	AD	
Atributo	Tipo	
<b>Responsabilidad</b>		
	buscarEquipo(\$id_Equipo)	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Buscar Equipo.	

Tabla 88: Descripción de la clase AD\_ BuscarJugador

<b>Nombre</b>	AD_ BuscarJugador
---------------	-------------------



<b>Tipo de Clase</b>	AD
<b>Atributo</b>	<b>Tipo</b>
<b>Responsabilidad</b>	
	buscarJugador(\$id_Jugador)
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Buscar Jugador.

**Tabla 89:** Descripción de la clase AD\_ConsultarInformaciónDelTorneo

<b>Nombre</b>	AD_ConsultarInformaciónDelTorneo
<b>Tipo de Clase</b>	AD
<b>Atributo</b>	<b>Tipo</b>
<b>Responsabilidad</b>	
	buscarTorneo(\$Id_Torneo)
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Consultar Información del Torneo.

**Tabla 90:** Descripción de la clase AD\_CalculoRatingElo



<b>Nombre</b>	AD_CalculoRatingElo	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	EloInicialJugador(\$id_jugador)	
	bonificacion(\$id_jugador)	
	expectanciaTotal(\$id_Jugador)	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Calculo del Rating Elo.	

Tabla 91: Descripción de la clase AD\_DefinirAlineacion

<b>Nombre</b>	AD_DefinirAlineacion	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	rondaADefinirAlineacion()	
	+insertarAlineacion(\$ronda, \$id_partida, \$id_jugador_a, \$id_jugador_b,	





	\$resultado_a, \$resultado_b)
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Definir Alineación.

**Tabla 92:** Descripción de la clase AD\_DefinirSistemasDesempate

<b>Nombre</b>	AD_DefinirSistemasDesempate	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	seleccionarSistemasDesempateEstablecidosParaUnGrupo(\$torneo, \$grupo)	
	seleccionarSistemasDesempateEstablecidosParaUnGrupoParaTableros(\$torneo, \$grupo)	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Definir Sistemas de Desempate.	

**Tabla 93:** Descripción de la clase AD\_ElaborarBasesTecnicas

<b>Nombre</b>	AD_ElaborarBasesTecnicas
<b>Tipo de Clase</b>	AD



Atributo	Tipo
<b>Responsabilidad</b>	
	buscarJugadoresTorneoOrdenados(\$id_torneo)
	desempatesAplicados(\$id_torneo, \$id_grupo)
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Elaborar Bases Técnicas.

Tabla 94: Descripción de la clase AD\_GestionarCiudad

<b>Nombre</b>	AD_GestionarCiudad	
<b>Tipo de Clase</b>	AD	
Atributo	Tipo	
<b>Responsabilidad</b>		
	adicionarCiudad(\$ciudad)	
	modificarCiudad (\$ciudad,\$Id_Ciudad)	
	eliminarCiudad (\$Id_Ciudad)	
	listarCiudades()	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso	



	Gestionar Ciudad.
--	-------------------

**Tabla 95:** Descripción de la clase AD\_GestionarFederacion

<b>Nombre</b>	AD_GestionarFederacion	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarFederacion (\$federacion)	
	modificarFederacion (\$federacion,\$Id_Federacion)	
	eliminarFederacion (\$Id_Federacion)	
	listarFederaciones()	
	verFederacion(\$Id_Federacion)	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Federacion.	

**Tabla 96:** Descripción de la clase GestionarGrupo

<b>Nombre</b>	AD_GestionarGrupo
<b>Tipo de Clase</b>	AD



Atributo	Tipo
<b>Responsabilidad</b>	
	adicionarGrupo (\$grupo)
	modificarGrupo (\$grupo,\$Id_Grupo)
	eliminarGrupo (\$Id_Grupo)
	listarGrupos()
	verGrupo()
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Grupo.

Tabla 97: Descripción de la clase AD\_GestionarImagen

<b>Nombre</b>	AD_GestionarImagen	
<b>Tipo de Clase</b>	AD	
Atributo	Tipo	
<b>Responsabilidad</b>		
	adicionarImagen (\$imagen)	
	eliminarImagen(\$id_Imagen)	



	mostrarImágenes()
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Imagen.

**Tabla 98:** Descripción de la clase AD\_GestionarJugadoresEquipo

<b>Nombre</b>	AD_GestionarJugadoresEquipo	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarJugadorEquipo(\$id_torneo,\$id_grupo, \$id_equipo, \$id_jugador, \$tablero)	
	eliminarJugadorEquipo(\$id_torneo, \$id_jugador, \$id_equipo)	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Jugadores de un Equipo.	

**Tabla 99:** Descripción de la clase AD\_GestionarJugadoresTorneo

<b>Nombre</b>	AD_GestionarJugadoresTorneo
<b>Tipo de Clase</b>	AD



Atributo	Tipo
<b>Responsabilidad</b>	
	adicionarJugadorTorneo(\$id_torneo, \$id_grupo, \$id_jugador)
	eliminarJugadorTorneo(\$id_torneo, \$id_jugador)
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Jugadores de un Torneo.

Tabla 100: Descripción de la clase AD\_GestionarPartida

<b>Nombre</b>	AD_GestionarPartida	
<b>Tipo de Clase</b>	AD	
Atributo	Tipo	
<b>Responsabilidad</b>		
	adicionarPartida(\$partida)	
	modificarPartida(\$partida,\$Id_Partida)	
	verPartida(\$Id_Partida)	
	eliminarPartida(\$Id_Partida)	
	listarPartidas()	



<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Partida.
--------------------	---

**Tabla 101:** Descripción de la clase AD\_GestionarRonda

<b>Nombre</b>	AD_GestionarRonda	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarRonda(\$ronda)	
	modificarRonda(\$ronda)	
	buscarRonda(\$id_ronda, \$id_torneo)	
	listarRondas()	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Ronda.	

**Tabla 102:** Descripción de la clase AD\_GestionarTituloJugador

<b>Nombre</b>	AD_GestionarTituloJugador
<b>Tipo de Clase</b>	AD



Atributo	Tipo
<b>Responsabilidad</b>	
	adicionarTituloJugador(\$Titulo)
	modificarTituloJugador(\$titulo, \$id_Titulo)
	eliminarTituloJugador(\$id_titulo)
	listarTitulosJugadores()
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Título del Jugador.

Tabla 103: Descripción de la clase AD\_GestionarTituloArbitro

<b>Nombre</b>	AD_GestionarTituloArbitro	
<b>Tipo de Clase</b>	AD	
Atributo	Tipo	
<b>Responsabilidad</b>		
	adicionarTituloArbitro(\$Titulo)	
	modificarTituloArbitro(\$titulo, \$id_Titulo)	
	eliminarTituloArbitro (\$id_titulo)	





	listarTitulosArbitros()
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Título del Árbitro.

**Tabla 104:** Descripción de la clase AD\_GestionarUsuario

<b>Nombre</b>	AD_GestionarUsuario	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	adicionarUsuario(\$usuario)	
	verUsuario()	
	modificarUsuario(\$usuario, \$id_usuario)	
	eliminarUsuario(\$id_usuario)	
	listarUsuarios()	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Usuario.	

**Tabla 105:** Descripción de la clase AD\_GestionarEquiposGrupo



<b>Nombre</b>	AD_GestionarEquiposGrupo	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	insertarGrupoEquipo(\$id_torneo, \$id_grupo, \$id_equipo)	
	eliminarEquipoGrupo(\$id_torneo, \$id_equipo)	
	cantidadEquiposSorteados()	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Equipos de un Grupo.	

Tabla 106: Descripción de la clase AD\_ImportarPGN

<b>Nombre</b>	AD_ImportarPGN	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	convertirArchivoTorneo(\$archivo)	
	guardarArchivo(\$archivo)	



<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Importar PGN.
--------------------	--

Tabla 107: Descripción de la clase AD\_ModificarTablero

<b>Nombre</b>	AD_ModificarTablero	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	modificarTableros(\$id_Jugador, \$tableros)	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Modificar Tablero.	

Tabla 108: Descripción de la clase AD\_MostrarClasificaciónGeneral

<b>Nombre</b>	AD_MostrarClasificaciónGeneral	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		



	clasificacionTorneoRR_E(\$id_torneo, \$id_grupo)
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Mostrar Clasificación General del Torneo.

**Tabla 109:** Descripción de la clase AD\_MostrarCuadroSinoptico

<b>Nombre</b>	AD_MostrarCuadroSinoptico	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	cuadroSinoptico(\$id_torneo)	
	cuadroSinopticoRR_E(\$id_torneo, \$id_grupo)	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Mostrar Cuadro Sinóptico.	

**Tabla 110:** Descripción de la clase AD\_MostrarRitmoDeJuego

<b>Nombre</b>	AD_MostrarRitmoDeJuego	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	



<b>Responsabilidad</b>	
	listarRitmoJuego()
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Mostrar Ritmo de Juego.

**Tabla 111:** Descripción de la clase AD\_MostrarSistemasDesempate

<b>Nombre</b>	AD_MostrarSistemasDesempate	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	mostrarSistemasDesempate()	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Mostrar Sistemas de Desempate.	

**Tabla 112:** Descripción de la clase AD\_MostrarEmparejamientoYResultados

<b>Nombre</b>	AD_MostrarEmparejamientoYResultados
<b>Tipo de Clase</b>	AD



Atributo	Tipo
<b>Responsabilidad</b>	
	emparejamientoResultadosRR_E(\$id_torneo, \$id_grupo, \$ronda)
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Mostrar Emparejamiento y Resultados.

Tabla 113: Descripción de la clase AD\_RealizarEmparejamiento

<b>Nombre</b>	AD_RealizarEmparejamiento	
<b>Tipo de Clase</b>	AD	
Atributo	Tipo	
<b>Responsabilidad</b>		
	pareoRoundRobin()	
	pareoRoundRobinEquipo()	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Realizar Emparejamiento.	

Tabla 114: Descripción de la clase AD\_RealizarSorteo



<b>Nombre</b>	AD_RealizarSorteo	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	sorteoRoundRobin()	
	sorteoRoundRobinEquipo()	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Realizar Sorteo.	

Tabla 115: Descripción de la clase AD\_ConsultarPartida

<b>Nombre</b>	AD_ConsultarPartida	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		



---

	buscarPartida(\$Id_Partida)
	mostrarPartida()
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Consultar Partida.

**Tabla 116:** Descripción de la clase AD\_ConsultarJugador

<b>Nombre</b>	AD_ConsultarJugador	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	





<b>Responsabilidad</b>	
	buscarJugador(\$Id_Jugador)
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Consultar Jugador

Tabla 117: Descripción de la clase AD\_MostrarClasificacion

<b>Nombre</b>	AD_MostrarClasificacion	
<b>Tipo de Clase</b>	AD	
<b>Atributo</b>	<b>Tipo</b>	
<b>Responsabilidad</b>		
	clasificacionIndividualTableros()	
<b>Descripción</b>	Clase encargada de controlar el acceso a los datos del caso de uso Mostrar Clasificación Individual por tableros	

Tabla 118: Descripción de la clase CE\_ritmoDeJuego

<b>Nombre</b>	CE_ritmoDeJuego
---------------	-----------------



<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_RitmoJuego	int	
Nombre	varchar	
<b>Responsabilidad</b>		
	getId_RitmoJuego ()	
	setId_RitmoJuego (\$Id_RitmoJuego)	
	getNombre()	
	setNombre(\$Nombre)	
<b>Descripción</b>	Clase entidad que representa la tabla RitmoDeJuego en la base de datos	

Tabla 119: Descripción de la clase CE\_usuario

<b>Nombre</b>	CE_usuario	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_Usuario	int	
Nombre	varchar	
Rol	int	
Contraseña	varchar	



Responsabilidad	
	getId_Usuario()
	setUsuario(\$Usuario)
	getConstraseña()
	setContraseña(\$Contraseña)
	getRol()
	setRol(\$Rol)
	getNombre()
	setNombre(\$Nombre)
Descripción	Clase entidad que representa la tabla Usuario en la base de datos

Tabla 120: Descripción de la clase CE\_partida

Nombre	CE_partida	
Tipo de Clase	Entidad	
Atributo	Tipo	
Id_Partida	int	
Resultado_J1	int	
Resultado_J2	int	
Responsabilidad		



	getId_Partida()
	setId_Partida(\$Id_Partida)
	getResultado_J1()
	setResultado_J1(\$Resultado_J1)
	getResultado_J2()
	setResultado_J2(\$Resultado_J2)
<b>Descripción</b>	Clase entidad que representa la tabla Partida en la base de datos

Tabla 121: Descripción de la clase CE\_desempatesAplicados

<b>Nombre</b>	CE_desempatesAplicados	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_Desempate : char	char	
Prioridad	int	
<b>Responsabilidad</b>		
	getId_Desempate()	
	setId_Desempate(\$Id_Desempates)	
	getPrioridad()	
	setPrioridad(\$Prioridad)	



<b>Descripción</b>	Clase entidad que representa la tabla SistemasDesempate en la base de datos
--------------------	---

**Tabla 122:** Descripción de la clase CE\_desempatesAplicadosTableros

<b>Nombre</b>	CE_desempatesAplicadosTableros	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_Desempate : char	char	
Prioridad	int	
<b>Responsabilidad</b>		
	getId_Desempate()	
	setId_Desempate(\$Id_Desempates)	
	getPrioridad()	
	setPrioridad(\$Prioridad)	
<b>Descripción</b>	Clase entidad que representa la tabla SistemasDesempate en la base de datos	

**Tabla 123:** Descripción de la clase CE\_ciudad

<b>Nombre</b>	CE_ciudad
---------------	-----------



<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>		<b>Tipo</b>
-Id_Ciudad		int
Nombre		varchar
<b>Responsabilidad</b>		
	getId_Ciudad()	
	setId_Ciudad(\$Id_Ciudad)	
	getNombre()	
	setNombre(\$Nombre)	
<b>Descripción</b>	Clase entidad que representa la tabla Ciudad en la base de datos	

Tabla 123: Descripción de la clase CE\_tipoTorneo

<b>Nombre</b>	CE_tipoTorneo	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>		<b>Tipo</b>
Id_tipoTorneo		int
Nombre		varchar
<b>Responsabilidad</b>		



	getId_TipoTorneo()
	setId_TipoTorneo(\$Id_Grupo)
	getNombre()
	setNombre(\$Nombre)
<b>Descripción</b>	Clase entidad que representa la tabla tipoTorneo en la base de datos

**Tabla 124:** Descripción de la clase CE\_torneoImagen

<b>Nombre</b>	CE_torneoImagen	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_Imagen	int	
Nombre	varchar	
Tamaño	float	
<b>Responsabilidad</b>		
	getId_Imagen()	



	setId_Imagen(\$Id_Imagen)
	getNombre()
	setNombre(\$Nombre)
	getTamaño()
	setTamaño(\$Tamaño)
<b>Descripción</b>	Clase entidad que representa la tabla TorneoImagen en la base de datos

**Tabla 125:** Descripción de la clase CE\_torneoRonda

<b>Nombre</b>	CE_torneoRonda	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_Ronda	int	
Fecha	date	





Horainicio	time
<b>Responsabilidad</b>	
	getId_Ronda()
	setId_Ronda(\$Id_Ronda)
	getFecha()
	setFecha(\$Fecha)
	getHorainicio()
	setHorainicio(\$Horainicio)
<b>Descripción</b>	Clase entidad que representa la tabla TorneoRonda en la base de datos

**Tabla 126:** Descripción de la clase CE\_tituloDelJugador

<b>Nombre</b>	CE_tituloDelJugador	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_Titulo	varchar	
Nombre	varchar	



Responsabilidad	
	getId_Titulo ()
	setId_Titulo (\$Id_Titulo_jugador)
	getNombre()
	setNombre(\$Nombre)
Descripción	Clase entidad que representa la tabla TituloDelJugador en la base de datos

Tabla 127: Descripción de la clase CE\_tituloDelArbitro

Nombre	CE_tituloDelArbitro	
Tipo de Clase	Entidad	
Atributo	Tipo	
Id_Titulo	varchar	
Nombre	varchar	
Responsabilidad		
	getId_Titulo ()	
	setId_Titulo (\$Id_Titulo_arbitro)	
	getNombre()	
	setNombre(\$Nombre)	



<b>Descripción</b>	Clase entidad que representa la tabla TituloDelArbitro en la base de datos
--------------------	--

**Tabla 128:** Descripción de la clase CE\_match

<b>Nombre</b>	CE_match	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_Match	int	
<b>Responsabilidad</b>		
	getId_Match()	
	setId_Match(\$Id_Match)	
<b>Descripción</b>	Clase entidad que representa la tabla Match en la base de datos	

**Tabla 129:** Descripción de la clase CE\_sistemasDesempate

<b>Nombre</b>	CE_sistemasDesempate	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_Desempate : char	char	
Nombre	varchar	



Responsabilidad	
	getId_Desempate()
	setId_Desempate(\$Id_Desempates)
	getNombre()
	setNombre(\$Nombre)
Descripción	Clase entidad que representa la tabla SistemasDesempate en la base de datos

Tabla 130: Descripción de la clase CE\_basesTecnicas

Nombre	CE_basesTecnicas	
Tipo de Clase	Entidad	
Atributo	Tipo	
Id_Bases	int	
Listajugadores	jugador	
SistemasJuego	string	
Premio	string	
Tribunal	string	
SistemasDesempate	string	
PersonalTecnico	string	



Responsabilidad	
	getId_Bases()
	setId_Bases(\$Id_Bases)
	getListaJugadores()
	setListaJugadores(\$ListaJugadores)
	getReglas()
	setReglas(\$Reglas)
	getSistemasJuego()
	setSistemasJuego(\$SistemasJuego)
	getPremios()
	setPremios(\$Premios)
	getTribunal()
	setTribunal(\$Tribunal)
	getPersonalTecnico()
	setPersonalTecnico(\$PersonalTecnico)
<b>Descripción</b>	Clase entidad que representa la tabla batesTecnicas en la base de datos

Tabla 131: Descripción de la clase CE\_grupo

<b>Nombre</b>	CE_grupo
---------------	----------



<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_Grupo	int	
Nombre	varchar	
<b>Responsabilidad</b>		
	getId_Grupo()	
	setId_Grupo(\$Id_Grupo)	
	getNombre()	
	setNombre(\$Nombre)	
<b>Descripción</b>	Clase entidad que representa la tabla Grupo en la base de datos	

Tabla

132: Descripción de la clase CE\_federacion

<b>Nombre</b>	CE_federacion	
<b>Tipo de Clase</b>	Entidad	
<b>Atributo</b>	<b>Tipo</b>	
Id_Federacion	varchar	
Nombre	varchar	
Logo	varchar	



<b>Responsabilidad</b>	
	getId_Federacion()
	setId_Federacion(\$Id_Federacion)
	getNombre()
	setNombre(\$Nombre)
	getLogo()
	setLogo(\$Logo)
<b>Descripción</b>	Clase entidad que representa la tabla Federacion en la base de datos



### Glosario de Términos

#### A

Algoritmo: Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

Archivos PGN: archivos de texto plano (.txt) que son susceptibles de ser leídos por los programas especializados en ajedrez. Se pueden abrir y editar con el «*Bloc de notas*» sin que se añada ningún tipo de código de programa que distorsione la lectura por los programas de ajedrez. Se trata de una extensión de código libre.

Ámbito: Espacio comprendido dentro de límites determinados.

Autenticación: en la seguridad de ordenador, la autenticación es el proceso de intento de verificar la identidad digital del remitente de una comunicación como una petición para conectarse. El remitente siendo autenticado puede ser una persona que usa un ordenador, un ordenador por sí mismo o un programa del ordenador. En un web de confianza, "autenticación" es un modo de asegurar que los usuarios son quién ellos dicen que ellos son - que el usuario que intenta realizar funciones en un sistema es de hecho el usuario que tiene la autorización para hacer así.

#### B

Bancos de datos: Una aplicación informática que recoge toda la información disponible procedente de diferentes fuentes por medio de un tratamiento de datos electrónico, la registra uniformemente, y la mantiene lista para su utilización en cualquier momento.

Bean (JavaBean): es un componente hecho en software que se puede reutilizar y que puede ser manipulado visualmente por una herramienta de programación en lenguaje Java.

#### C

C: lenguaje de programación orientado a la implementación de Sistemas Operativos, concretamente Unix.

Clase interfaz: Modelan la interacción entre el sistema y sus actores





Clase control: Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

Clase entidad: Modelan información que posee larga vida y que es a menudo persistente.

Client Page: representa una página Web con formato HTML. Son interpretadas por el navegador.

Cohesión: Acción y efecto de reunirse o adherirse las cosas entre sí o la materia de que están formadas.

Conciso: forma de expresar un concepto con exactitud.

CU: Caso de uso

CVS (Concurrent Versions System): es una aplicación informática que implementa un sistema de control de versiones, mantiene el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto (de programa) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren.

### D

Debian: es una comunidad conformada por desarrolladores y usuarios, que mantiene un sistema operativo GNU basado en software libre precompilado y empaquetado, en un formato sencillo en múltiples arquitecturas de computador y en varios núcleos.

### E

Estándar de facto: patrón o norma que se caracteriza por no haber sido consensuada ni legitimada por un organismo de estandarización al efecto, sino que se trata de una norma generalmente aceptada y ampliamente utilizada por iniciativa propia de un gran número de interesados.

EJB (Enterprise JavaBeans): son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE de Oracle - Sun Microsystems (ahora JEE 5.0).

### F



Form: colección de elementos de entrada que son parte de una página cliente. Sus atributos son los elementos de entrada del formulario.

### I

IEEE: corresponde a las siglas de The Institute of Electrical and Electronics Engineers, el Instituto de Ingenieros Eléctricos y Electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización.

### J

Java: lenguaje de programación orientado a objetos

### M

Mítico(a): (adjetivo) fabuloso, mitológico, legendario.

### O

OMG (Object Management Group): es un consorcio dedicado al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos, tales como UML, XMI, CORBA.

ORM: es un componente de software que permite trabajar con los datos persistidos como si ellos fueran parte de una base de datos orientada a objetos.

### P

Proceso de negocio: conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido.

Proliferación: reproducirse en formas similares.

Prototipo: modelo (representación, demostración o simulación) fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas.

### R



RRI: Round Robin Individual

RRE: Round Robin por Equipos

### **S**

Server Page: representa la página Web que tiene código que se ejecuta en el servidor.

Sistema de clasificación: Sistema mediante el cual a partir de los resultados de los jugadores durante el torneo, permite realizar una competencia entre los mismos a partir de los parámetros de clasificación, ya sea clasificación general del evento, clasificación por puntuación o clasificación por rendimiento.

Subversion: es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS

### **U**

URL (Uniform Resource Locator): es un localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

### **X**

XMI o XML Metadata Interchange (XML de Intercambio de Metadatos): es una especificación para el Intercambio de diagramas.