

Universidad de las Ciencias Informáticas



Título: Infodrez - Módulo de Torneos Online.

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

Autor(es):

Raciel Garrido Torres
José Rodríguez Alarcón

Tutor(es):

Ing. Abel Meneses Abad

Co-tutor:

Ing. Alison Muñoz Capote

Consultante:

Msc. Isbel Herrera del Sol

Asesor:

MSc. Graciela González Pérez

DECLARACIÓN DE AUTORÍA

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

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

Autores:

Raciel Garrido Torres

José Rodríguez Alarcón

Tutor:

Abel Meneses Abad

Resumen

Este trabajo persigue como objetivo desarrollar un módulo para la plataforma Web Infodrez específicamente el Juego Online, que permita jugar torneos en el ajedrez en tiempo real con sistema suizo. Para una mejor comprensión del trabajo es necesario agregar que en el mundo ajedrecístico el juego por torneos rige los principales eventos a nivel mundial, definiendo de esta forma campeones regionales y mundiales en el ajedrez. Vale aclarar que este desarrollo creará los tipos de torneos existentes definidos según características como: el tiempo de juego, el número de rondas, la solución del pareo final y la realización de de los cruces. Para ello se ha realizado un estudio de estas características permitido automatizar determinados procesos de forma tal que cumpla las necesidades del usuario final, para jugar mediante la plataforma Infodrez en tiempo real.

En fin, este trabajo está enmarcado en un desarrollo Web creado como módulo o componente del juego de Ajedrez Online de la plataforma Infodrez que permita jugar en la modalidad de torneos específicamente el suizo.

Índice

Introducción	7
Capítulo 1: Fundamentación teórica	9
1.1- Introducción	9
1.2- Definición y características de Torneos de Ajedrez	10
1.2.1- ¿Qué es el ajedrez?	10
1.2.2- ¿Qué es un torneo de ajedrez?	10
1.3- Softwares para el juego de ajedrez a nivel internacional	11
1.3.1- Internet Chess Club (ICC)	11
1.3.2- Chessbase (playchess.com)	12
1.3.3- Free Internet Chess Server	12
1.3.4- International Correspondence Chess Federation (ICCF) ---	13
1.3.5- Planeta de ajedrez.com	13
1.3.6- MKGI Chess Club	13
1.3.7- AjedrezOnline.com	14
1.3.8- Chesshere.com	14
1.3.9- Queenalice.com	15
1.3.10- Chess17.com	15
1.3.11- UciDrez	15
1.4- Metodología RUP	16
1.5- Tecnologías utilizadas	16
1.5.1- Lenguaje PHP	16
1.5.2- Plataforma LAMP	17
1.6- Herramientas de desarrollo	22
1.6.1- Visual Paradigm	22
1.6.2- Quanta Plus	23
1.6.3- Zend Studio	24
1.7- Métodos de investigación	24
1.8- Cronograma	25

Capítulo 2: Características del sistema	26
2.1- Introducción	26
2.2- Flujo de procesos que realizan en un torneo	26
2.3- Análisis crítico de los procesos que se desarrollan	26
2.3.1- Objetos de automatización	27
2.4- Modelo de Negocio	27
2.4.1- Actores del negocio	27
2.4.2- Trabajadores del negocio	28
2.4.3- Reglas del negocio	28
2.4.4- Diagrama de Casos de Uso del Negocio	29
2.5- Especificación de los requerimientos del software	30
2.5.1- Requerimientos funcionales	30
2.5.2- Requerimientos no funcionales	33
2.6- Definición de los Casos de Uso	35
2.6.1- Definición de los actores del sistema	35
2.6.2- Definición de los Casos de Uso del sistema	35
2.6.3- Descripción de los Casos de Uso del sistema	40
2.6.4- Prototipos de Interfaz de usuario	63
Capítulo 3: Análisis y Diseño del sistema	76
3.1- Introducción	76
3.2- Análisis	76
3.2.1- Diagramas de Clases del Análisis	77
3.2.2- Diagramas de Interacción (Colaboración)	83
3.3- Diseño	90
3.3.1- Diagramas de Clases del Diseño	90
3.4- Diseño de la Base de Datos	97
3.4.1- Diagrama de Clases persistentes	98
3.4.2- Modelo de Base de Datos	99
3.4.3- Descripción de las tablas de la Base de Datos	100

Capítulo 4: Implementación	103
4.1- Diagrama de despliegue	103
4.2- Diagrama de componentes	104
Conclusiones	105
Recomendaciones	106
Bibliografía	107
Glosario	109

Introducción

El desarrollo de software para el ajedrez, es una rama de la informática con años de experiencia. Hoy se conocen versiones de clientes y servidores muy avanzados con equipos enteros de desarrollo del software y planes de desarrollo muy bien elaborados. Se conoce también los esfuerzos en materia de inteligencia artificial para el desarrollo de motores de juego similares a los empleados en la Computadora DeepBlue, tales como el GNU CHESS. La utilización de tecnologías Web para la implementación de software para el ajedrez en estos momentos está en un estado primitivo, los pocos desarrollos que existen a nivel mundial no cumplen con las principales funcionalidades de una aplicación de tal magnitud.

En la Universidad de las Ciencias Informáticas en el año 2005, en un intento por promover el juego del ajedrez se lanza un concurso para la realización de una aplicación para el Juego Online, el cual fue ganado por el proyecto Sherpa y se le dio el nombre de **UCIDrez**; software que serviría como base para la investigación realizada. En marzo de 2006 la facultad 10 migra completamente hacia sistemas libres, con navegadores web basados en políticas libres y con restricciones en su conjunto estándares; lo cual resultó un problema para los ajedrecistas de la facultad, pues el único sitio que existía en la universidad para el juego de ajedrez no era multiplataforma, solo era soportado por el IE el cual pertenece a tecnología propietaria. El proyecto Unicornios de la facultad 10, decide desarrollar una plataforma que permitiera el juego de ajedrez con respuesta en tiempo real, culminando la misma en abril del año 2007 después de un mes de desarrollo cuando salió la primera versión beta.

Después de varios meses de juego en dicha plataforma, los ajedrecistas y la cátedra de ajedrez deciden aumentar los servicios de dicha plataforma con la creación de torneos jugados en tiempo real automatizando el proceso de pareo y distribución de los jugadores en cada torneo para incrementar las funcionalidades de la plataforma. Por lo que se puede evidenciar que **la plataforma para el juego de ajedrez no cuenta con un módulo para el juego de usuarios en torneos como situación problemática a resolver**, que permita el juego en tiempo real entre usuarios en forma de torneo, lo que implicaría elevar la calidad, los conocimientos de los ajedrecistas y de usuarios finales del sitio en general.

El **problema científico** de la investigación es:

¿Cómo lograr una aplicación interactiva para el juego de ajedrez en la modalidad de torneos?

Considerando como **objeto de estudio**, software para ajedrez y centrándose en el **campo de acción** sobre aplicaciones web para jugar torneos de ajedrez.

Como **objetivo general** está identificado: Diseñar e implementar un módulo de torneos para el Juego Online en la plataforma Infodrez.

Para dar solución a la problemática se plantean los siguientes **objetivos específicos de la investigación**:

- 1- Determinar las principales características que tendría un sistema Web para jugar ajedrez en la modalidad de torneos.
- 2- Analizar sistemas implementados a nivel mundial.
- 3- Analizar y diseñar el componente torneos online.
- 4- Implementar y probar el módulo para el juego de ajedrez mediante torneos, con el pareo Suizo.

Idea a defender:

La implementación de un módulo para jugar ajedrez en la modalidad de torneos, sería beneficioso para el desarrollo del ajedrez en la Universidad de las Ciencias Informáticas.

Como vía para cumplir los objetivos propuestos se definieron las siguientes **tareas de investigación**:

- 1- Investigar las diferentes características para la organización de torneos, en eventos realizados en la Universidad de las Ciencias Informáticas.
- 2- Realizar un estudio de las principales herramientas y sus funcionalidades que implementan esta categoría del juego de ajedrez en el mundo.
- 3- Preparar el análisis del sistema basado en un levantamiento de requisitos, enmarcado en las necesidades de la Universidad de las Ciencias Informáticas.
- 4- Realizar la implementación del componente para el juego de torneos de ajedrez en tiempo real.

Capítulo 1: Fundamentación Teórica:

1.1 Introducción:

Desde los inicios de la Revolución cubana, el Comandante en Jefe Fidel Castro Ruz, abogó por el desarrollo del ajedrez en Cuba y el carácter masivo que este debía tener, promoviendo simultáneas gigantes con la participación en 1966 del campeón mundial Tigran Petrosian. Otro gran precursor fue el comandante Ernesto Guevara al cual le fue concedido el título de Caballero de la FIDE por su contribución al juego ciencia.

Teniendo en cuenta la gran importancia que tiene el ajedrez para la Revolución Cubana en la actualidad, en nuestro país se desarrollan programas y teleclases destinados a la enseñanza del juego ciencia para que sea llevado a cada ciudadano. Observando el auge creciente de este atractivo juego en la juventud y el proceso de informatización que se lleva a cabo en el país; podríamos optar por herramientas para aprovechar los avances de conectividad y crear una estructura que permitiera explotar dichos avances: ¿Qué software existen a nivel mundial que se puedan adaptar a las necesidades de informatización para contribuir a masificar el ajedrez en Cuba?. Sería necesario resaltar que el ajedrez en la informática tiene una larga historia y hoy en día esta muy mercantilizado, por lo que representaría una gran inversión para cualquier país.

Es requerido que en los programas de entrenamiento de trevejistas se efectúen topes para medir fuerzas y adquirir experiencia competitiva, pero Cuba es un país bloqueado, lo que trae consigo problemas económicos y la transportación hacia estos eventos sería muy costosa sin tener en cuenta locales de entrenamiento, y todo el aseguramiento que se debe tener para realizar cualquier evento deportivo. Dicho problema podría resolverse mediante el uso de la tecnología, ya que los jugadores no se tendrían que mover de sus localidades y podrían topar de forma plena con trevejistas de otras localidades y en ocasiones con jugadores de otras naciones minimizando los gastos del gobierno y brindando un mayor confort.

1.2 Definición y características de Torneos de ajedrez “el Sistema Suizo”:

1.2.1 ¿Qué es el ajedrez?

El ajedrez es un juego de mesa para dos personas. Más precisamente, es un juego de guerra, perteneciente a la misma familia del xiàngqí (ajedrez chino) y el shōgi (ajedrez japonés). Se cree que todos ellos provienen del chaturanga, que se practicaba en la India en el siglo VI.

Es uno de los juegos más populares del mundo. Se considera no sólo un juego, sino un arte, una ciencia y un deporte mental. Esto último es muy apropiado, dado que se juega a menudo de forma competitiva. La enseñanza del ajedrez puede ser útil como forma de desarrollar la mente. El ajedrez es jugado tanto recreativo como competitivamente en clubes, torneos, en Internet, e incluso por correo (ajedrez por correspondencia) (Wikipedia, 2008).

1.2.2 ¿Qué es un torneo de ajedrez?

Los torneos surgen en la época medieval, es el combate a caballo entre varias personas puestas en cuadrillas y bandos de una parte y otra. Estas cuadrillas batallaban y se herían entre sí dando vueltas en torno para perseguir cada cual a su contrario. Era la imitación de una batalla (Wikipedia, 2008).

Los torneos de ajedrez son justas entre jugadores de ajedrez, que se enfrentan en un lugar determinado, para determinar un ganador del torneo.

Clasificaciones generales de los torneos de ajedrez

Los torneos se pueden efectuar agrupando los jugadores de forma:

- Individuales

Cada jugador depende de forma exclusiva de lo que hagan en el torneo

- Por equipos

Se crean equipos de jugadores donde se mide el esfuerzo de todos para obtener la victoria.

Según el sistema de juego

- Round Robin

Sistema donde todos los participantes del torneo se enfrentan entre ellos en un número constante de oportunidades (generalmente, una o dos si es a doble vuelta).

- Suizo

Se predefine el número de rondas a jugar, se crean grupos de participantes con una misma puntuación para cada ronda, posibilitando una gran participación de jugadores y la definición de un ganador en pocas rondas.

1.3 Software para el juego de ajedrez a nivel internacional:

1.3.1 Internet Chess Clubs:

Fundado en 1995, el Internet Chess Club (ICC) fue pionero en ofrecer juegos online en la red. Sus raíces se remontan a un pequeño grupo de jugadores de ajedrez de la década de los 80. En 1995, después de estar algunos meses en internet, el número de usuarios había crecido de unos cientos a más de diez mil inscritos. Tras más de 10 años de actividad, el Internet Chess Club cuenta con más de 30.000 socios, lo que lo convierte en el mayor sitio de Internet con Servicios Premium de Ajedrez Online. Se realizan torneos periódicos restringidos para jugadores que paguen sus inscripciones donde participan jugadores con categoría de gran maestro. Entre sus principales funcionalidades tenemos:

- Juego online en tiempo real.
- Juego en la modalidad de torneos.
- Juego contra motores de ajedrez.
- Multilenguaje.
- Observar partidas online de grandes maestros.
- Medición de fuerza por sistema de elo.

El servidor está hospedado en los Estados Unidos, y se permite jugar una semana gratis, pero después es necesaria la compra del registro, solo los jugadores con título FIDE mayor que MF pueden jugar de forma gratis. Existen varios clientes para conectarse a él y jugar, los más famosos como el Fritz, no son libres. Está basado en java.

1.3.2 ChessBase (playchess.com)

El servidor de ChessBase, "Playchess.com" para algunos el servidor más popular y grande del mundo: se han disputado más de 230 millones de partidas; cuenta con 230.000 miembros registrados; se han celebrado más de 100.000 torneos y todas las noches en las horas punta están conectados entre 5.000 y 7.000 jugadores de todas partes del planeta, desde principiantes hasta los mejores ajedrecistas del mundo (entre ellos también Kasparov, que todavía sigue jugando con regularidad en Playchess.com). Entre las principales funcionalidades que ofrece están:

- Juego online en tiempo real.
- Juego en la modalidad de torneos en tiempo real.
- Observar juegos.
- Simultaneas.
- Ajedrez temático.
- Entrenamiento online.

El servidor está hospedado en los Estados Unidos, y su sistema de registro casi idéntico al de Internet Chess Clubs, una cuenta de cuesta **29,90€ a 43,95 US\$**.

1.3.3 Free Internet Chess Server (FICS)

Permite jugar ajedrez de forma gratis en internet. Creado por la comunidad de jugadores de ajedrez partidarios del software libre hoy en día cuenta con mas de 300 000 usuarios registrados, y es uno de los mas antiguos de su tipo en internet. Entre sus principales funcionalidades están:

- Juego online en tiempo real.
- Juego en la modalidad de torneos en tiempo real.
- Observar juegos.

El servidor esta basado en C++ totalmente libre y el registro en este es totalmente gratis, está hospedado en los Estados Unidos, y posee más de 4 clientes todos libres.

1.3.4 ICCF(International Correspondence Chess Federation)

Sitio oficial de la FIDE dedicado al ajedrez por correspondencia. Posee la funcionalidad del juego mediante email, además de gestionar el envío de tarjetas postales y cartas a cualquier lugar del mundo promoviendo el juego ciencia por correspondencia. Su sistema de rating elo está aprobado por la FIDE como elo oficial de ajedrez por correspondencia, teniendo grandes maestros y hasta campeones mundiales de ajedrez por correspondencia. Posee más de 100 000 miembros de 65 nacionalidades diferentes y se han realizado en el 18 campeonatos mundiales de ajedrez por correspondencia.

Entre sus principales funcionalidades están:

- Juego por correspondencia de partidas sueltas.
- Juego por correspondencia en la modalidad de torneos.
- Sistema de rating elo oficial de la FIDE.

El servidor web está hospedado en los Estados Unidos, y como requisito de registro está pertenecer al club de forma oficial. Basado en php y como cliente se utilizan navegadores web.

1.3.5 Planeta del ajedrez.com

Sitio dedicado al ajedrez que administra el marketing para la venta de clientes para conectar a un servidor privado de juegos de ajedrez online. Entre sus principales funcionalidades tenemos:

- Juego online en tiempo real.
- Juego en la modalidad de torneos por equipo e individual.

Se permite el registro gratis en este y está hospedado en España administrado por la FEDA.

1.3.6 MKGI Chess Club

Aplicación web creada por Yannick Kirschhoffer, dedicada al ajedrez postal o por correspondencia, que cuenta con más de diez mil usuarios en todo el mundo. Posibilita realizar partidas con alto grado de complejidad, el juego contra motores de ajedrez. Entre sus principales funcionalidades están:

- Juego online por correspondencia.
- Juego en la modalidad de torneos por correspondencia.
- Análisis de posiciones de juego según posibilidades de ataque.
- Juego contra motores de ajedrez.
- Medición de fuerza por sistema de elo

Es servidor está hospedado en los Estados Unidos. Su registro es gratis así como el código fuente del servidor que está basado en php.

1.3.7 AjedrezOnline.com

Sitio web dedicado al mundo del ajedrez. Propone una zona de juego donde varias centenas de jugadores, de todo nivel, se enfrentan midiendo fuerzas. Entre sus principales funcionalidades están:

- Juego online en tiempo real.
- Juego en la modalidad de torneos en tiempo real.
- Medición de fuerza por sistema de elo.

Permite registro gratis y el servidor está basado en java, así como sus clientes utilizando apples en la web.

1.3.8 Chesshere.com

Sitio web orientado al ajedrez por correspondencia aunque posee también un juego online con respuestas en tiempo real. Se han jugado mas de 370 000 partidas y cuenta con mas de 10 000 jugadores de todo el mundo. Entre las principales funcionalidades que implementa están:

- Juego online en tiempo real.
- Juego en la modalidad de torneos en tiempo real.
- Observar juegos.
- Juegos por correspondencia o ajedrez postal.

El servidor está hospedado en los Estados unidos, basado en php y ajax.

1.3.9 Queenalice.com

Sitio web dedicado al ajedrez por correspondencia de todos los niveles y edades. Entre sus principales funcionalidades están:

- Juegos por correspondencia o ajedrez postal.
- Juegos en la modalidad de torneos (Round Robin y Sistema Suizo)
- Multilenguaje (El servidor detecta la configuración de lenguaje del navegador).
- Navegador de aperturas ECO.

El servidor está hospedado en los Estados Unidos, basado en php, y permite registro gratis.

1.3.10 Chess17.com

Sitio web dedicado al ajedrez online, donde se han realizado más de 7 231 759 jugadas. Permite juegos en distintas modalidades basadas en el tiempo de juego, además de notificación por email. Entre las principales funcionalidades con que cuenta están:

- Juego online en tiempo real.
- Análisis estadístico de partidas.

El servidor está hospedado en los Estados Unidos, basado en php y permite jugar como invitado, pero para registrarse es necesaria la compra de un número de registro.

1.3.11 UciDrez

Sitio web creado en la Universidad de las Ciencias Informáticas, destinado a la promoción del ajedrez dentro de la Universidad y en Cuba, como parte de la política de práctica masiva del juego Ciencia. Cuenta con más de 2000 jugadores en 2 años de publicación llegando a ser un éxito para la comunidad ajedrecística que en la universidad radicaba. Entre las principales funcionalidades con que cuenta se pueden nombrar:

- Juego online en tiempo real.
- Observar partidas online.

- Medición de fuerza por sistema de elo.

El servidor estaba hospedado por la Universidad de las Ciencias Informáticas en Cuba, su registro era gratis. Estaba basado en .net e implementado para el navegador web Internet Explorer.

1.4 Metodología RUP:

El **Proceso Unificado Racional** (*Rational Unified Process* en inglés, habitualmente resumido como **RUP**) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. La metodología RUP es más apropiada para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios (Wikipedia, 2008).

Existen metodologías como es el caso de XP, SCRUM y otras que pudiéramos mencionar pero no se ajustan a la aplicación que se desea implementar, ya que se deben desarrollar un conjunto de metodologías muy amplias. XP y SCRUM se asocian más bien con procesos de desarrollo un poco más pequeños, están dentro del grupo de las metodologías ágiles.

1.5 Tecnologías utilizadas:

1.5.1 Lenguaje PHP

PHP es un lenguaje de programación interpretado usado normalmente para la creación de páginas web. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-processor" (inicialmente PHP Tools, o, Personal Home Page Tools).

Los principales usos del PHP son la programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, lo que amplía en gran medida sus posibilidades de conexión. Programación en consola, al estilo de Perl o Shell scripting. Creación de aplicaciones gráficas independientes del navegador, por

medio de la combinación de PHP y Qt/GTK+, lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado.

PHP ofrece muchas ventajas, es un lenguaje multiplataforma, tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL, puede 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, tiene una biblioteca nativa de funciones sumamente amplia e incluida, no requiere definición de tipos de variables y es capaz de controlar el manejo de excepciones (Wikipedia, 2008).

Otros de los posibles lenguajes a utilizar eran Perl, Java y ASP. En el caso de Perl aunque comparte casi las mismas características de PHP no posee un IDE (Interface Development Environment) de desarrollo reconocido, por lo que su programación y la corrección de errores se hace muy difícil. Java aunque es muy completo y bastante utilizado en la Web consume muchos recursos de cómputo para su desarrollo y requiere de una máquina virtual para su correcta ejecución, ya esto rompe con la concepción de aplicación ligera. ASP también se desechó por el hecho de que trabaja solo sobre plataformas propietarias y hasta cierto punto también consume muchos recursos (Experiencias de integrantes del proyecto Unicornios).

1.5.2 Plataforma LAMP

LAMP es un conjunto de aplicaciones que permite establecer una plataforma Web bien sea para desarrollo o para producción, y todas las aplicaciones de libre disposición en la red.

La "L" cubre la capa del sistema operativo con Linux. En la actualidad el Linux se ha posicionado fuertemente en el sector de los servidores y está siendo utilizado cada día más para soluciones de misión crítica.

La "A" denota Apache como servidor Web. El servidor Web simplemente se encarga de servir páginas estilo Web hacia quien lo solicita, que por lo general es un PC o dispositivo con un visor de Web. El

Apache es el servidor Web de mayor número de instalaciones en la red.

La plataforma LAMP incluye la "M" de MySQL, una base de datos de muy fácil manejo y de relativamente buen desempeño. Es importante también aquí recordar que también se cuenta con otra base de datos de excelentes características de estabilidad llamada Postgress la cual también es de carácter gratuito, y aunque un poco más compleja de manejar que MySQL, también tiene desarrolladas recientemente un número considerable de herramientas que facilitan su manejo.

El último componente está para los lenguajes de programación. Las ofertas que propone LAMP, todas iniciando por la "P" son PHP, Perl, y Python. De estos tres, el que hasta hace poco tenía la delantera y todavía presenta buena acogida entre programadores es el Perl.

Las soluciones que ofrecen las casas comerciales se encargan de hacer que los cuatro componentes: sistema operativo, servidor Web, base de datos y lenguaje de programación; funcionen sincronizada y adecuadamente dentro de un conjunto de equipos. La propuesta LAMP pretende ser la respuesta a las casas comerciales con software que hasta ahora ha dado buenos resultados en forma separada y que promete consolidar el soporte para cada una de las herramientas (base de datos y lenguajes en especial) y garantizar la facilidad de instalación de cada uno de los componentes para permitir una integración transparente (Daccach, 2008).

Existen otras plataformas muy conocidas que también se usan en el mundo de la Web, está el caso de WAMP, XATMJ y la WIMA. Se pudiera haber utilizado cualquiera de estas pero hasta cierto punto rompen con la concepción de software libre y van en contra del desarrollo de plataformas Web ligeras y usables, que son los tres puntos principales a mantener en la aplicación que se debe crear. Ahora se expondrán algunos aspectos por los cuales no se utilizaron ninguna de estas.

En el caso de WAMP, corresponde a la unión de los componentes **Windows** como sistema operativo, **Apache** como servidor Web, **MySQL** como gestor de Bases de Datos y **PHP** como lenguaje del lado del servidor. Con solo hecho de usar Windows no era la plataforma indicada, ya que no es un sistema libre. Tiene como ventaja la existencia de una aplicación con el mismo nombre, WAMP v5 que instala todos los paquetes completos y configura la interacción entre ellos, además de poseer una interfaz visual para las configuraciones. Tiene como desventaja el mal funcionamiento de los componentes cuando se instalan por separado, su configuración de hace muy engorrosa.

XATMJ es un tipo de plataforma que no es muy común encontrarla en los proveedores de hosting más populares. La **X** significa que puedes utilizar cualquier sistema operativo, usa **A**pache como servidor Web, **T**omcat que trabaja como contenedor de servlets (derivado de applets de Java, son pequeños programas hechos en Java que corren en el servidor), **M**ySQL el servidor de Bases de Datos y **J**SP las páginas Java que están alojadas en el servidor. Aunque Tomcat es muy fiable y Java funciona en cualquier sistema operativo que disponga de la máquina virtual, consume mucho recurso de hardware.

Por último tenemos WIMA, este sistema es de los más utilizados para desarrollar aplicaciones web basadas en ASP clásico y ASP.NET. También se puede instalar PHP o Perl bajo IIS, y es la forma en la que los proveedores de hosting ofrecen ASP y PHP en el mismo servidor. Usa como sistema operativo **W**indows, trabaja con **I**nternet Information Services (IIS) como servidor Web, **M**S SQL Server como gestor de bases de datos y **A**SP como lenguaje Web del lado del servidor, generalmente las versiones de IIS siempre vienen asociadas o en conjunto con las versiones de ASP.

Entre las ventajas que ofrece están las del servidor de Bases de Datos MS SQL Server, este posee un potente entorno gráfico de administración y puede ofrecer a los usuarios grandes cantidades de datos de forma simultánea, ASP.NET desde su versión 2.0 es muy completa para desarrollo. Las desventajas que posee ante todo es que no es libre, plataforma totalmente propietaria.

El servidor IIS no es multiplataforma, funciona solamente sobre Windows como mismo sucede con MS SQL Server. Limitaciones de IIS en las versiones que no son de la familia Windows Server y en sentido general la plataforma tiene algunas vulnerabilidades (Blanco, 2007).

1.5.2.1 Sistema operativo Debian GNU/Linux

Por se un proyecto nacido en la facultad 10 que atiende el software libre en la universidad, se decidió trabajar sobre herramientas libres y como sistema operativo Debian. **Debian GNU/Linux** es la principal distribución Linux del proyecto Debian, que basa su principio y fin en el software libre.

Creada por el proyecto Debian en el año 1993, la organización responsable de la creación y mantenimiento de la misma distribución, centrado en el kernel Linux y utilidades GNU. Éste también mantiene y desarrolla sistemas GNU basados en otros núcleos (Debian GNU/Hurd, Debian GNU/NetBSD y Debian GNU/kFreeBSD).

Nace como una apuesta por separar en sus versiones el software libre del software no libre. El modelo de desarrollo es independiente a empresas, creado por los propios usuarios, sin depender de ninguna manera de necesidades comerciales. Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respete su licencia.

Al igual que todas las demás distribuciones de GNU/Linux comparte características que lo hacen muy popular.

Es muy robusto, estable y rápido: Ideal para servidores y aplicaciones distribuidas. A esto se añade que puede funcionar en máquinas con pocos recursos de hardware que no son de última generación.

Es libre: Esto implica no sólo la gratuidad del software, sino también que Linux es modificable y que Linux tiene una gran cantidad de aplicaciones libres en Internet. Todo ello arropado por la inmensa documentación que puede encontrarse en la Red.

No está restringido a personas con grandes conocimientos de informática: Los desarrolladores de Linux han hecho un gran esfuerzo por dotar al sistema de asistentes de configuración y ayuda, además de un sistema gráfico muy potente. Distribuciones Linux como Red Hat/Fedora tienen aplicaciones de configuración similares a las de Windows (Ciberaula, 2007).

No se utilizó otro sistema operativo (Mac OS, Windows o FreeBSD) ya que por lo general son propietarios y de los demás que son libres no tenemos mucha documentación al respecto, incluyendo a eso que algunos no poseen la estabilidad y rapidez de Debian GNU/Linux como es el caso de Windows, o son muy difíciles de adquirir por su precio como Macintosh.

1.5.2.2 Servidor Web Apache

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd (Servidor Web) 1.3, pero más tarde fue reescrito por completo.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, tiene amplia aceptación en la red: en el 2005,

Apache fue el servidor HTTP más usado, siendo el servidor empleado en el 48% de los sitios web en el mundo, actualmente su por ciento de uso es mucho más alto.

Apache ofrece muchas ventajas:

Modular: Acepta su integración con otros programa propios en él y paquetes de software con los que interactúa de forma estable respondiendo a cada petición del sistema.

Open source: Es una herramienta de código abierto.

Multi-plataforma: Es compatible con cualquier sistema operativo, una de sus forma mas estable y robusta la podemos ver cuando interactúa con el sistema operativo GNU/Linux para administración de sitios web.

Popular: La popularidad viene dada por su gran estabilidad, es una aplicación fácil de conseguir en la red, posee soporte técnico como la mayoría de las aplicaciones libres y mucha documentación en diferentes idiomas.

Gratuito: Es totalmente gratuito, por lo que no hay que pagar por su adquisición ni tiene una licencia por la que se deba pagar periódicamente a una compañía y productor (Wikipedia, 2008).

Otro servidor Web que se utiliza bastante en prestaciones de servicios el IIS, pero como ya se mencionó anteriormente es propietario y trabaja solo sobre sistemas operativos Microsoft Windows, por lo tanto no era la herramienta indicada.

1.5.2.3 Gestor de Bases de Datos MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones conocidas en todo el mundo. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. MySQL AB pertenece a Sun Microsystems desde enero de 2008.

La serie en desarrollo de MySQL Server actualmente, es la 5.1 al cual se añaden nuevas características en relación a la serie 5.0. La serie de producción actual de MySQL es 5.0, cuya penúltima versión estable es la 5.0.26 lanzada en octubre del 2006. Actualmente, se puede descargar la serie 5.0.27. La serie de producción anterior fue la 4.1, cuya versión estable es 4.1.7 lanzada en octubre de 2004. A estas versiones de producción sólo se arreglan problemas, es decir, ya no se añaden nuevas características. Y a las versiones anteriores solamente se les corrigen bugs críticos.

Según las cifras del fabricante, existen más de seis millones de copias de MySQL funcionando en la actualidad, lo que supera la base instalada de cualquier otra herramienta de bases de datos. El tráfico del sitio web de MySQL AB (compañía productora) superó en 2004 al del sitio de IBM.

Entre sus ventajas se pueden mencionar:

- 1- Completo soporte para operadores y funciones en cláusulas select y where.
- 2- Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación
- 3- Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- 4- Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- 5- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).
- 6- Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando named pipes y en sistemas Unix usando ficheros socket Unix.
- 7- En MySQL 5.0, los clientes y servidores Windows se pueden conectar usando memoria compartida.
- 8 - MySQL contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL (Wikipedia, 2008).

Existen otros gestores como es el caso PostgreSQL y Oracle que también son muy estables. No se trabajó con PostgreSQL porque las facilidades que ofrece trabajando con PHP son menores que con MySQL, no es tan ligero y consume mas recursos, podríamos decir que PHP y MySQL son la “pareja ideal” para desarrollo de plataformas Web libres. Oracle aunque tiene versiones para GNU/Linux es propietario, tampoco es tan ligero como MySQL (Comunidad de Fedora, 2007).

1.6 Herramientas de desarrollo

1.6.1 Visual Paradigm

El paradigma visual para UML es una herramienta del CASO de UML. La herramienta se diseña para los usuarios que están interesados en sistemas de software de gran escala basados en tecnología orientada a objetos. La herramienta apoya los estándares más recientes de las notaciones de Java y

de UML. Además, Vp-uml se encaja con un IDE de gran alcance. Las transiciones del análisis al diseño y entonces a la puesta en práctica se integran dentro de la herramienta. Ayuda al equipo de desarrollo de software, a documentar y cubrir el proceso de modelado, construcción y desarrollo de software, acelerando la contribución individual y a nivel de proyecto (Filehungry, 2007).

Otra herramienta líder en modelado, construcción y desarrollo de sistemas es la Suite Rational, posee una ayuda bastante completa en la que cualquier especialista podría aprender mucho, pero no tiene versiones funcionales para GNU/Linux. Visual Paradigm aunque es propietario al igual que Rational es multiplataforma y se hace muy fácil su uso por especialistas.

1.6.2 Quanta Plus

Quanta Plus (*Quanta+*) es una herramienta libre de desarrollo de páginas web diseñado para el proyecto KDE. Su versión actual es la 3.5. Forma parte del paquete kdewebdev. Entre sus características podemos decir que tiene asistentes para creación de tablas, enlaces y páginas en blanco, muestra resaltado de sintaxis de HTML, Javascript, CSS y otros más, contiene un analizador que informa acerca de la correcta creación de nuestras páginas.

Previsualización: Los documentos pueden ser previsualizados dentro de la aplicación usando el motor KHTML. Es posible preprocesar los documentos a través de un servidor web antes de previsualizar. Dentro de los editores HTML libres es el que más se destaca por todas las funcionalidades que posee, una de las más importantes es el avanzado autocompletamiento de código HTML que implementa, en este sentido es la mejor de las herramientas (Wikipedia, 2008).

El mundo del diseño Web afortunadamente posee muchos softwares para trabajar, entre ellos está el Adobe DreamWeaver, Microsoft Expression Web (conocido también como Microsoft FrontPage 2007), NVU, Bluefish y Web Composer. Adobe DreamWeaver es la herramienta líder en diseño, llegando a dominar el 90% de este mercado, es propietario. Microsoft Web Expression es una solución ofrecida por la compañía Microsoft, se destaca por el gran parecido con su sucesor Microsoft FrontPage 2003, también propietario. Por último tenemos a NVU, Bluefish y Web Composer que son muy similares, tienen como desventaja su pobre auto completamiento de HTML y Hojas de estilos (CSS) en la vista código, en este sentido el Quanta ofrece mayores facilidades (Wikipedia, 2008).

1.6.3 Zend Studio

El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. Podríamos decir que es la herramienta de desarrollo líder en la programación Web con lenguaje PHP.

El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración (Maestrosdelweb, 2006).

El Zend Studio no tiene muchos competidores que mencionar, en la actualidad es la herramienta líder en desarrollo de aplicaciones web dinámicas con PHP. Uno de los que más se asemeja es el Adobe DreamWeaver, pero este no está disponible para GNU/Linux y no posee auto completamiento script de PHP al nivel de Zend Studio, está más bien orientado a diseño. Tenemos también Aptana, este último aunque trabaja con PHP se especializa en JavaScript.

1.7 Métodos de investigación:

Analítico - sintético: Ha servido para extraer funcionalidades de programas creados por grupos de desarrollo de software de ajedrez, se llegó a las necesidades reales de la universidad siguiendo un desarrollo incremental basado en la evolución de sistemas para el juego de ajedrez.

Observación: Facilitó obtener las estadísticas iniciales de aceptación, de los principales software a nivel mundial, así como la extracción de parámetros tales como el uso de sistemas Operativos libres y navegadores empleados en la universidad.

Entrevista: Se creó una base de conocimientos apoyados en entrevistas realizadas en la Cátedra de Ajedrez de la universidad, facilitando la recopilación de requisitos y funciones del sistema, así como el proceso que se describe para la realización de torneos.

1.8 Cronograma

Actividad	Fecha	Responsable	Descripción
Modelamiento del negocio	01 al 29-02-2008	Raciel y Jose	Primer flujo de trabajo de la metodología RUP para la IGSW de la aplicación
Capítulo 1 de la tesis	10 y 13-02-2008	Raciel y Jose	Confección de capítulo 1 de la tesis en vista de presentarlo en el corte.
Segundo corte de tesis	14-02-2008	Raciel y Jose	
Requerimientos	1 al 15-03-2008	Raciel y Jose	Segundo flujo de trabajo de la metodología RUP para la IGSW de la aplicación
Análisis y Diseño	15 al 31-03-2008	Raciel y Jose	Tercer flujo de trabajo de la metodología RUP para la IGSW de la aplicación
Tercer corte de la tesis	12-04-2008	Raciel y Jose	
Capítulo 2 de la tesis	12-04-2008	Raciel y Jose	
Implementación	01 al 15-04-2008	Raciel y Jose	Cuarto flujo de trabajo de la metodología RUP para la IGSW de la aplicación
Prueba	01 al 15-04-2008	Raciel y Jose	Quinto flujo de trabajo de la metodología RUP para la IGSW de la aplicación

Capítulo 2: Características del Sistema: Módulo de Torneos Online.

2.1 Introducción.

En este capítulo se exponen los dos primeros flujos de trabajos de la metodología RUP, específicamente Modelamiento del negocio y Requerimientos. En el primer flujo es donde se genera todo lo referente a la dinámica del negocio, se obtienen los diagramas Casos de Uso del Negocio, los diagramas de actividades de cada Caso de Uso y el Modelo de objetos. Por otra parte en el Requerimientos se generan los Prototipos de Interfaz de usuario para cada Caso de Uso y un diagrama General de Interfaz para toda la aplicación, el diagrama de Casos de Uso del sistema, las descripciones de cada uno de ellos y los requisitos funcionales y no funcionales.

2.2 Flujo de procesos que se realizan en un torneo.

- Organizar y crear torneos que son actividades generalmente realizadas por el organizador o director del evento, donde se acuerdan las bases y los objetivos del torneo, los árbitros que serán invitados y los jugadores que estarán presentes en el mismo.
- Realizar pareos en las rondas que es dirigido por los árbitros auxiliares o adjuntos que estarán desarrollando el torneo, a grandes rasgos los pareos son la acción de decidir cuales serán las parejas de jugadores a enfrentarse en cada ronda teniendo en cuenta algunos parámetros necesarios.
- Jugar las partidas de cada ronda es una actividad realizada por cada jugador que se enfrenta en el torneo, llegando a un resultado final, victoria, derrota o tabla.
- Informar los resultados después de cada ronda y el resultado final del torneo es tarea de los árbitros involucrados, basado en esa información de siguen haciendo las rondas restantes.

2.3 Análisis críticos de los procesos que se ejecutan en la Cátedra de Ajedrez para organizar y desarrollar un torneo.

A la hora de organizar y desarrollar un torneo en la Cátedra de Ajedrez, el organizador le avisaba a los jugadores propuestos a participar por correo, informando sobre la fecha de comienzo del mismo y la fecha límite de la inscripción de en caso quedar plazas libres.

Ya en pleno desarrollo del torneo los árbitros deben tener control del listado de jugadores que participarán con su nombre, número que se le asignó, elo y federación o provincia a la que pertenece, todo esto es llevado manualmente. Cada vez que los jugadores terminan de jugar una ronda completa se deben registrar los resultados y realizar el pareo de la siguiente ronda partiendo de los resultados obtenidos.

Se informan también los resultados de cada ronda y el resultado final del torneo. Aclarar que cada uno de estos procesos se hacen manualmente dando margen a errores dentro de todo el proceso de desarrollo del torneo, pudiendo automatizar cada uno de ellos se resolvería un gran problema.

2.3.1 Objetos de automatización

- Crear torneos
- Invitar árbitros y jugadores al torneo.
- Realizar los pareos en cada ronda.
- Jugar las partidas en cada ronda.
- Publicar los resultados de las rondas y el resultado final.

2.4 Modelo de negocio.

2.4.1 Actores del negocio

Un actor del negocio es cualquier individuo grupo, organización, máquina o sistema de información externos con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio obteniendo un resultado que lo beneficia (Jacobson, 2004).

Nombre	Descripción
Organizador	Organiza el torneo, define sus bases y objetivos y gestiona los jugadores y árbitros que participarán.
Jugador	Es quien juega en las partidas de cada ronda y funciona como protagonista en la realización de los torneos.
Observador	Es aquel participante del torneo ya sea o no jugador que observa alguna de las partidas que se están jugando.

2.4.2 Trabajadores del negocio

Un trabajador del negocio representa una persona o un sistema automatizado (software) que actúa en el negocio realizando una o varias actividades comprendidas dentro del caso de uso, interactuando con otros trabajadores del negocio y manipulando entidades del negocio (Jacobson, 2004).

Nombre	Descripción
Árbitro principal	Llevando el control del listado de todos los participantes en el torneo. Informa sobre los resultados cada vez que el termina una ronda del torneo y tiene conocimiento de todo el proceso de pareo.
Árbitro	Es quien realiza los pareos para comenzar cada ronda, realiza un monitoreo constante de todas las partidas atento a que no se comentan violaciones, en caso de materializarse alguna debe expulsar al infractor del evento. Pude funcionar en ocasiones como observador.

2.4.3 Reglas del negocio

- 1- El número de rondas que se van a jugar se anuncia por anticipado.
- 2- Se deben fijar los objetivos: El organizador, Rango de Elo que participará, Categoría del torneo y si habrá o no aumento de categoría para los jugadores participantes.
- 3- A partir de los objetivos que definen las bases: Participantes, Calidad o Nivel del torneo (tener en cuenta para esto el Elo promedio de los jugadores), Fecha de inicio y fin del torneo, Nombre del mismo, Calendario de actividades, Invitar a los jugadores, Premios, Definir árbitro, Formas de inscripción y fecha de inicio y fin de la misma y Sistema de desempate.
- 4- El organizador y el árbitro principal son los únicos encargados de hacer la gestión de los árbitros adjuntos o auxiliares.
- 5- Solo el organizador puede gestionar los jugadores que participarán en el torneo.
- 6- Dos jugadores pueden jugar entre ellos solamente una vez.
- 7- Los jugadores se emparejan con otros de la misma puntuación, o la más cercana.
- 8- Cuando es posible, a un jugador se le dan el mismo número de veces las piezas blancas y negras.
- 9- Cuando es posible, a un jugador se le da el color contrario de piezas de la ronda anterior.
- 10- Si en cualquier ronda el número de participantes es impar, se concede el descanso al jugador de menor rango en el último grupo de puntuación.

11- Un jugador sólo puede descansar una vez. Un jugador que ha ganado un punto por incomparecencia, no puede descansar posteriormente.

12- La clasificación final se obtiene agregando los puntos ganados: 1 punto por una victoria, 0.5 puntos por unas tablas, y 0 por una derrota. Un jugador cuyo oponente no se presenta a una partida prevista, recibe un punto.

13- Se debe calcular el Elo final de cada jugador para obtener la variación que ha tenido.

14- Al final de la ronda las tarjetas de los jugadores deben quedar actualizadas.

15- Al jugador que haya descansado se le anota un punto en esa ronda. Se considera que no tiene ni adversario ni color en esa ronda.

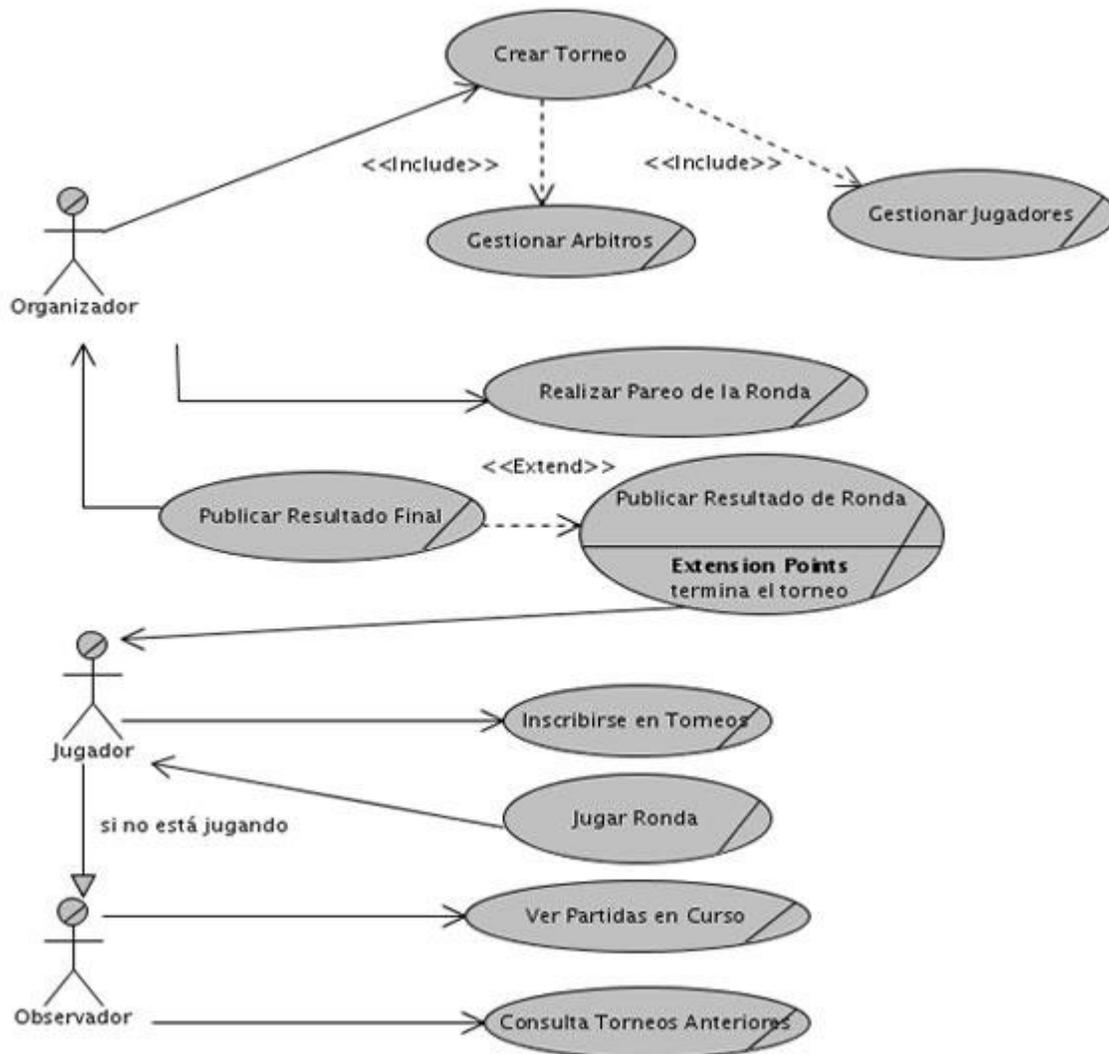
16- El jugador que gane la partida informa al árbitro el resultado, en caso de empate informa el de las piezas blancas.

17- Si es un jugador con partida en curso no debe observar otra partida, o sea, no puede desempeñarse como observador (Herrera del Sol, 2007).

2.4.4 Diagrama de Casos de Uso del Negocio

Un caso de uso del negocio representa un proceso dentro del negocio que se estudia, por lo que se corresponde con una secuencia de acciones con un orden lógico y que producen un resultado observable para ciertos actores del negocio (Jacobson, 2004).

A continuación se muestra el diagrama de Casos de Uso del Negocio, donde se representa una vista general de la dinámica del negocio a través de relaciones entre Casos de uso y actores.



2.5 Especificación de los requisitos del software.

2.5.1 Requerimientos funcionales.

1. Gestionar torneos.

1.1. Crear torneo

1.1.1 Adicionar los objetivos.

1.1.2 Adicionar las bases.

1.1.3 Notificar a todos los jugadores en cada entrada al sistema hasta cumplida la fecha de comienzo del torneo.

1.2. Editar información de torneos.

1.3. Eliminar torneos creados.

1.4. Listar los torneos.

2. Crear listado de árbitros que participaran en el torneo, según el criterio de estos.

2.1. Mostrar listado de árbitros registrados

2.1.1 Mostrar usuario.

2.1.2 Mostrar datos estadísticos de juego del árbitro.

2.2. Invitar árbitros a participar en el torneo.

2.2.1 Mostrar mensaje de invitación para torneo con envío de notificación.

2.2.2 Enviar notificación a árbitro auxiliar de conformidad o inconformidad con la invitación.

3. Crear listado de jugadores a invitar en el torneo.

3.1. Realizar una búsqueda dirigida de jugadores.

3.1.1 Por usuario.

3.1.2 Por ocupación.

3.1.3 Por área.

3.1.4 Ordenados por elo según tiempo de juego.

3.2. Elegir jugadores e invitarlos a participar en el torneo.

3.2.1 Mostrar mensaje de invitación para torneo con envío de notificación.

3.2.2 Permitir responder la notificación al jugador de conformidad o inconformidad con la invitación.

4. Gestionar Jugadores que participaran en el torneo.

4.1. Adicionar un nuevo jugador.

4.2. Listar jugadores que inscritos en el torneo.

4.3. Eliminar jugadores de la lista de participantes de torneo.

5. Gestionar árbitros auxiliares que participaran en el torneo.

5.1. Adicionar un nuevo árbitro.

5.2. Listar árbitros del torneo.

5.3. Eliminar árbitros de la lista del torneo.

6. Realizar pareo de jugadores por sistema suizo.

6.1. Notificar resultado de pareo a los participantes en torneo.

7. Comenzar partida de torneo.

- 7.1. Crear las partidas según el pareo realizado para la ronda.
- 7.2. Guardar datos de partida al concluir esta.

8. Publicar resultados de cada ronda.

- 8.1. Enviar notificación por ronda de los resultados de esta.
- 8.2. Si es la ronda final publicar los resultados de la ronda final y guardar el torneo para que se consultado en cualquier momento.

9. Permitir observar a otros jugadores las partidas en curso.

- 9.1. Permitir a jugadores que están observando partidas realizar comentarios de la partida.

10. Permitir ver a cualquier usuario los torneos anteriores.

- 10.1. Mostrar listado de torneos anteriores.
- 10.2. Mostrar datos del torneo seleccionado.
 - 10.2.1. Mostrar rondas de torneo.
 - 10.2.2. Mostrar partidas de ronda.

Glosario de términos

Las **bases de un torneo** se definen en los siguientes puntos:

1. Cómo participar:

- 1.1 Inscripciones.
 - 1.1.1 reglas de inscripción.

2. Calendario del torneo:

- Define fecha de comienzo del torneo.
- Define las fechas en que se jugaran cada ronda.

3. Sistema de juego:

- 3.1. Sistema de torneo.
 - Tipo de torneo a jugar.
 - Cantidad de rondas a jugar.

3.2. Ritmo de las partidas.

- Tiempo en el cual se juega la partida.
- Incremento por cada jugada.

3.3. Desarrollo del torneo

3.3.1 Sistema de desempate

4. Premio:

- Definir premio a los lugares (puede ser un aumento de elo mayor y gana privilegios)

5. Cuestiones técnicas:

5.1 Reglas de la partida.

- Desconexión de un jugador.
- Jugador que no entra a jugar la ronda.
- Reglas generales del comportamiento en el torneo

2.5.2 Requerimientos no funcionales.

1- 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.
- Sistemas Operativos GNU/Linux (recomendado) o Microsoft Windows.

Para la interpretación por clientes (Navegadores Web):

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

2- 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.

3- 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.

4- Diseño e 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.

5- Seguridad:

Verificar autenticidad y niveles de privilegio del usuario antes de hacer cualquier operación dentro del sistema. Verificar 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 al acceso de usuarios del dominio.

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

6- Político – Culturales:

Este módulo junto a Infodrez fueron creados inicialmente como una necesidad en la Universidad, por tener probabilidades de llegar a ser un producto nacional y de competir en el mercado internacional, debe ajustarse a las características de otros softwares con este fin que están actualmente predominando en el ajedrez mundial.

7- Soporte:

Legales: La plataforma está basada en la licencia GNU/GPL.

Confiabilidad: Realiza control de errores y recuperación ante fallos.

2.6 Definición de los Casos de Uso.

2.6.1 Definición de actores del sistema.

Actores	Justificación
Árbitro principal	Es el usuario encargado de realizar casi todas las operaciones en el torneo, hace la gestión de torneos, la gestión e invitación de los árbitros auxiliares, la gestión e invitación de los jugadores que participarán en el torneo y da a conocer los resultados finales del torneo una vez que termina.
Árbitro auxiliar	Es el encargado junto al árbitro principal de llevar a cabo y chequear el correcto desarrollo de cada torneo, da a conocer los resultados de cada ronda a los participantes en el torneo y controla que no se cometan violaciones mientras se realizan las partidas.
Jugador	Es quien materializa junto a su contrario las partidas de cada ronda y se mantiene en constante intercambio con el árbitro principal y el árbitro auxiliar, en algunos casos puede funcionar como observador, ya que cuando termina su partida puede observar las de otros jugadores.
Observador	Es aquel participante del torneo que observa partidas, que en algunos casos puede ser jugador, consulta además datos de los torneos anteriores.

2.6.2 Definición de Casos de Uso del sistema.

Quedarán representados los requisitos funcionales del sistema mediante un diagrama de casos de uso. Para ello se definió de acuerdo a lo planteado en los epígrafes anteriores, cuales serían los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades los definimos a continuación:

Casos de uso:

CUS-1	Gestionar torneos
Actor	Árbitro principal.
Resumen	Al árbitro principal hace todo lo relacionado con la gestión de torneos, puede crear torneos, editar torneos ya publicados o eliminar torneos registrados.

CUS-2	Gestionar árbitros auxiliares
Actor	Árbitro principal.
Resumen	Al árbitro principal registra a los árbitros que desee en los torneos y los elimina en caso de ser necesario, en ambos casos se le informa a los árbitros auxiliares sobre la decisión tomada.

CUS-3	Invitar árbitros auxiliares
Actor	Árbitro principal.
Resumen	Al árbitro principal invita a los árbitros auxiliares que el desea que participen en el torneo, el sistema le envía una notificación si desea participar, si acepta lo inscribe en el torneo, en caso que no confirme y simplemente no responda, se elimina de forma automática del listado de árbitros del torneo.

CUS-4	Gestionar jugadores
Actor	Árbitro principal.
Resumen	Al árbitro principal registra a los jugadores que desee en los torneos y los elimina en caso de ser necesario, en ambos casos se le informa a los jugadores sobre la decisión tomada.

CUS-5	Invitar jugadores
Actor	Árbitro principal.
Resumen	Al árbitro principal invita a los jugadores que el desea que participen en el torneo, el sistema le envía una notificación si desea participar, si acepta lo inscribe en el torneo, en caso que no confirme y simplemente no responda, se elimina de forma automática del listado de jugadores del torneo.

CUS-6	Realizar pareo de ronda
Actor	Árbitro principal.
Resumen	Para el caso de la primera ronda el jugador de más fuerza informa el color de pieza que desea usar en la primera ronda, sino responde en el tiempo establecido, el sistema le solicita al árbitro principal el color de pieza para que lo elija en lugar del jugador. A partir del color elegido hace la división en grupos y la asignación de partidas. Las rondas restantes se hacen basadas en los resultados de las partidas.

CUS-7	Publicar resultado de ronda
Actor	Árbitro auxiliar.
Resumen	El árbitro auxiliar accede a la sección donde está la tabla de resultados y verá el estado de las partidas en cada ronda, ya una vez terminada la ronda publica los resultados de la misma, esto genera un envío de notificaciones a los jugadores y demás participantes en el torneo informando sobre la publicación del resultado.

CUS-8	Publicar resultado final
Actor	Árbitro principal.
Resumen	El árbitro principal accede a la sección donde está la tabla de resultados, ya una vez registrados los resultados de todas las rondas publica el resultado final, esto genera un envío de notificaciones a los jugadores y demás participantes en el torneo informando sobre la publicación del resultado.

CUS-9	Enviar notificación
Actor	Árbitro principal y árbitro auxiliar.
Resumen	El sistema detecta cuando se realiza un evento que requiere del envío de una notificación. Identifica el evento, cual es el texto del mensaje indicado y lo envía a sus destinatarios. Las acciones que generan envíos de notificaciones generalmente son realizadas por el árbitro principal y el árbitro auxiliar.

CUS-10	Jugar partida de ronda
Actor	Árbitro principal y jugador.
Resumen	Una vez que el sistema hace el pareo se muestran los resultados a todos los participantes. El árbitro principal informa a los jugadores que en poco tiempo deben comenzar a jugar las partidas de la ronda. Después de un tiempo determinado manda a jugar generando un mensaje de notificación, cuando el jugador acepta dicho mensaje entra al tablero a jugar con su contrario, juega la partida y se registra la misma en la Base de Datos. Si el jugador no juega la partida por no presentación o porque sencillamente no quiere jugar entonces pierde.

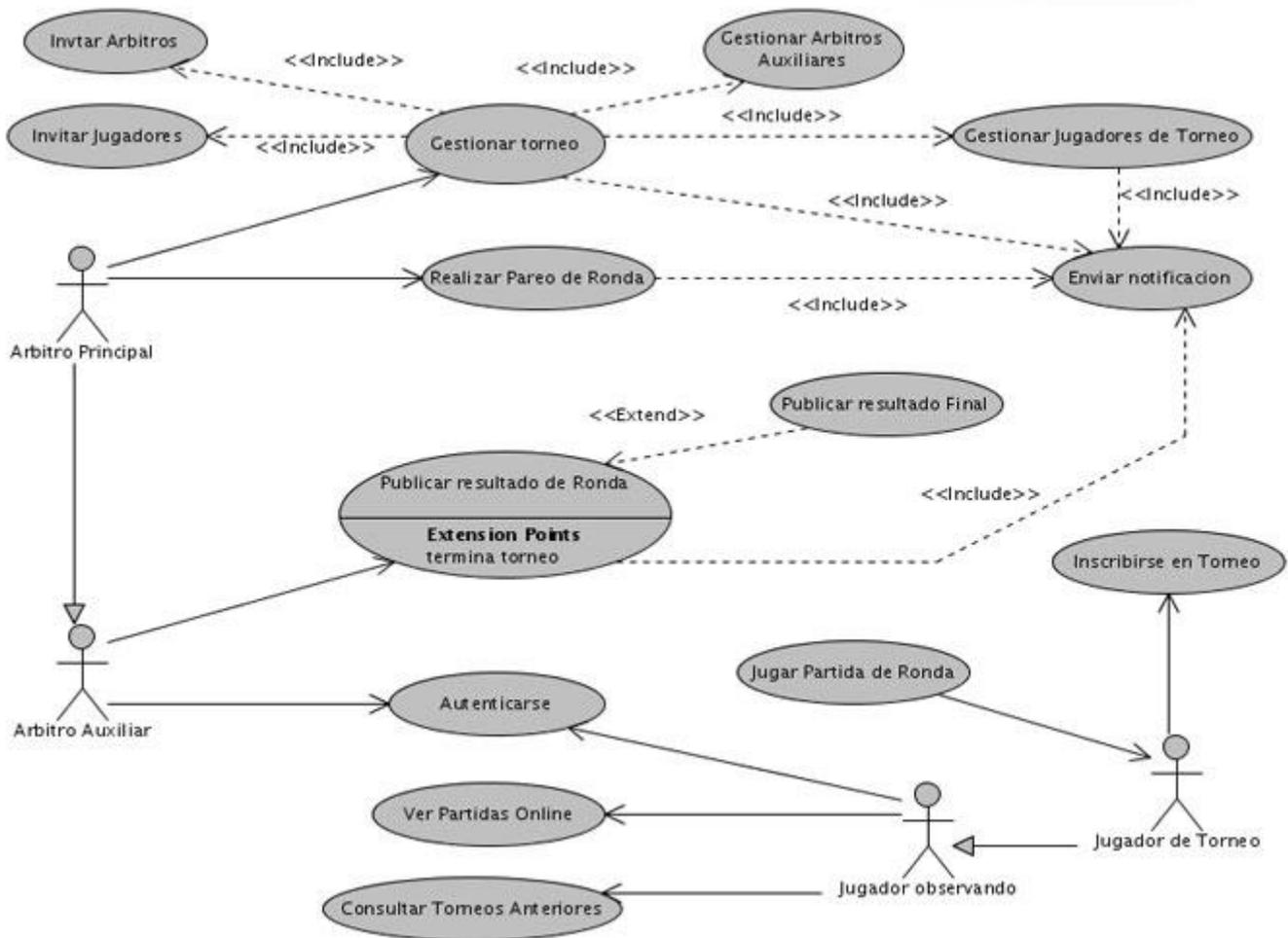
CUS-11	Inscribirse en torneo
Actor	Jugador.
Resumen	El jugador accede a la sección donde está el listado de torneos activos para inscribirse (esto es solo en caso de quedar plazas libre para el torneo), si cumple con las bases del torneo se le mostrará la opción "Registrarse", en caso contrario se le mostrará un mensaje informándole que no puede registrarse en el torneo porque no cumple con las bases. Cuando el jugador se inscribe se genera un mensaje de notificación informándole al árbitro principal.

CUS-12	Autenticarse
Actor	Árbitro auxiliar y observador.
Resumen	El usuario entra su usuario y contraseña en el sistema, si es usuario registrado, no está expulsado del torneo y el usuario y la contraseña son correctos podrá acceder al sistema. Si alguna de estas condiciones es incorrecta el sistema no le permite acceder y le informa sobre la situación.

CUS-13	Ver partidas online
Actor	Observador.
Resumen	El participante que desee observar una partida selecciona la que desee y la amplía en un visualizador de partidas. Si el observador es un jugador debe haber terminado su partida en esa ronda para poder observar otra. El árbitro auxiliar siempre está observando que no se cometa ninguna violación, en caso de que se haga alguna expulsa del torneo a quien la hizo.

CUS-14	Consultar torneos anteriores
Actor	Observador.
Resumen	El participante que desee ver datos de torneos anteriores los solicita al sistema, se le mostrará el listado de los torneos anteriores, con los nombres de los jugadores que participaron, los resultados de cada ronda y los PGN de cada partida. Después estudia la partida que desee.

Diagrama de Casos de Uso:



2.6.3 Descripción de los Casos de Uso del sistema.

En este subepígrafe se presenta la descripción detallada de cada Casos de Uso:

Caso de uso 1: Gestionar torneos

Prioridad: Crítico

Actores: Árbitro principal (inicia)

Propósito: Crear torneos estableciendo inicialmente los objetivos del mismo y la definición de las bases.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
<p>1- Árbitro principal: Si desea crear un torneo introduce los datos teniendo en cuenta toda la información necesaria. Esta información son los objetivos y las bases.</p> <p>Para los objetivos es necesario conocer:</p> <ul style="list-style-type: none">- El Organizador del torneo.- El Rango de elo que participará.- Categoría del torneo.- Si habrá o no aumento de categoría para los jugadores participantes. <p>Para las bases se debe saber:</p> <ul style="list-style-type: none">- Cantidad de participantes.- Calidad o nivel del torneo (se mide por el elo promedio de los jugadores).- Fecha de inicio y fin del torneo.- Nombre del torneo.- Calendario de actividades.- Premios.	

<ul style="list-style-type: none"> - Formas de inscripción. - Fecha de inicio y fin de la inscripción. - Sistema de desempate. 	<p>2- Registra el torneo y emite un mensaje informando que la operación de registro tuvo éxito.</p> <p>3- Ejecuta el caso de uso Enviar notificación enviando un mensaje a los jugadores y árbitros propuestos.</p> <p>Mensaje: <i>Hola a todos, hemos organizado un nuevo torneo, más adelante seguiremos informando al respecto.</i></p>
	<p>4- El caso de uso termina.</p>
<p>Cursos alternativos</p>	
<p>3- Árbitro principal: Corrige el error, introduce nuevamente los datos del torneo y se ejecuta la actividad 3 del curso normal.</p>	<p>2- Si la operación no tuvo éxito emite un mensaje informado donde está el error.</p>
<p>Cursos alternativos</p>	
<p>1- Árbitro principal: Edita los datos del torneo y guarda los cambios.</p>	<p>2- Actualiza los cambios y emite un mensaje informando que la operación de actualización tuvo éxito.</p> <p>3- Ejecuta el caso de uso Enviar notificación enviando un mensaje a los jugadores y árbitros propuestos.</p> <p>Mensaje: <i>Hemos hechos algunos cambios en el torneo, es necesario que usted los conozca ya que ha sido invitado a participar.</i></p>
	<p>4- El caso de uso termina.</p>
<p>Cursos alternativos</p>	
<p>1- Árbitro principal: Manda a eliminar.</p>	<p>2- Emite un mensaje de advertencia preguntando sobre por confirmar o cancelar eliminación.</p> <p>3- Si confirma, ejecuta el caso de uso Enviar</p>

	<p>notificación enviando un mensaje a los jugadores y árbitros propuestos.</p> <p>Mensaje: Hemos cancelado el torneo, en próximos eventos lo tendremos a usted en cuenta. Disculpe las molestias que esto pueda ocasionarle.</p>
	4- El caso de uso termina.

Referencias:

Listado de requerimientos:

- Se debe crear el torneo con todos los objetivos y las bases definidas.
- El sistema debe emitir una respuesta informando sobre el registro correcto o incorrecto del torneo.

Casos de uso asociados: Gestionar árbitros auxiliares (include), Gestionar jugadores (include) y Enviar notificación (include).

Precondiciones: El árbitro es designado como Árbitro principal y este debe tener bien definido todos aspectos de los objetivos y las bases del torneo antes de proceder al registro en el sistema, deben estar logueado.

Poscondiciones: El torneo queda registrado con todos los parámetros necesarios.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

Caso de uso 2: Gestionar árbitros auxiliares

Prioridad: Secundario

Actores: Árbitro principal (inicia)

Propósito: Obtener los nombres que los árbitros que participarán en el evento.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
1- Árbitro principal: Solicita el listado de los árbitros auxiliares que participarán en el torneo.	2- Muestra el listado de los árbitros del torneo.

<p>3- Árbitro principal: Selecciona los árbitros auxiliares que eliminará del torneo.</p> <p>4- Árbitro principal: Manda a eliminar.</p>	<p>5- Muestra un mensaje preguntando por confirmación, si acepta elimina al arbitro del torneo.</p> <p>6- Ejecuta el caso de uso Enviar notificación para informar al árbitro.</p> <p>Mensaje: Hola “nombre”, su participación en el torneo ha sido cancelada. Disculpe las molestias que esto le pueda ocasionar.</p>
	<p>8- El caso de uso termina.</p>
<p>Curso alternativo</p>	
	<p>6- Muestra un mensaje preguntando por confirmación, si cancela mantiene al árbitro en el torneo.</p>
	<p>7- El caso de uso termina.</p>
<p>Curso alternativo</p>	
<p>1- Árbitro principal: Realiza búsqueda de árbitros.</p> <p>2- Árbitro principal: Manda a listar según los parámetros.</p> <p>4- Árbitro auxiliar: Selecciona los árbitros que registrará en el torneo</p> <p>5- Árbitro auxiliar: Manda a registrar.</p>	<p>3- Muestra el listado de árbitros según la búsqueda.</p> <p>6- Registra al árbitro en el torneo.</p> <p>7- Ejecuta el caso de uso Enviar notificación para informar al árbitro.</p> <p>Mensaje: Hola “nombre”, ha sido registrado para participar en nuestro torneo. Muchas gracias por desear trabajar junto a nosotros.</p>
	<p>8- El caso de uso termina.</p>

Referencias:

Listado de requerimientos:

- El sistema debe emitir mensajes informando al árbitro auxiliar sobre su eliminación del torneo.
- Debe quedar confeccionado el listado final de los árbitros que participarán en el torneo.

Casos de uso asociados: Enviar notificación (include).

Precondiciones: El sistema debe tener el listado de los árbitros que participarán en el torneo.

Poscondiciones: El listado queda actualizado después de las operaciones pertinentes.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

Caso de uso 3: Invitar árbitros auxiliares

Prioridad: Secundario

Actores: Árbitro principal (inicia)

Propósito: Tener conocimiento de todos los árbitros auxiliares que están dispuestos a participar en el evento.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
1- Árbitro principal: Recibe el listado de los árbitros auxiliares propuestos a participar.	
2- Árbitro principal: Solicita una búsqueda de los árbitros auxiliares en el sistema.	
3- Árbitro principal: Introduce los parámetros necesarios para la búsqueda.	
	4- Muestra el listado de los árbitros que cumplan los parámetros.
5- Árbitro principal: Selecciona los árbitros	

alternativo.

Referencias:

Listado de requerimientos:

- El sistema debe emitir mensajes de invitaciones a los árbitros y si alguno no está disponible debe informar sobre esta situación el árbitro principal.

- Debe quedar confeccionado el listado final de los árbitros que participarán en el torneo.

Casos de uso asociados: Enviar notificación (include).

Precondiciones: El árbitro principal debe conocer inicialmente la lista de los árbitros auxiliares propuestos.

Poscondiciones: Queda confeccionado el listado final de los árbitros auxiliares que participarán en el torneo.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

Caso de uso 4: Gestionar jugadores

Prioridad: Secundario

Actores: Árbitro principal (inicia)

Propósito: Obtener los nombres de los jugadores que participarán en el evento..

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
1- Árbitro principal: Solicita el listado de los jugadores que participarán en el torneo.	2- Muestra el listado de los jugadores del torneo.
3- Árbitro principal: Selecciona los jugadores que eliminará del torneo.	

<p>4- Árbitro principal: Manda a eliminar.</p>	<p>5- Muestra un mensaje preguntando por confirmación, si acepta elimina al jugador del torneo.</p> <p>6- Ejecuta el caso de uso Enviar notificación para informar al árbitro.</p> <p>Mensaje: Hola “nombre”, su participación en el torneo ha sido cancelada. Disculpe las molestias que esto le pueda ocasionar.</p>
	<p>7- El caso de uso termina.</p>
<p>Cursos alternativos</p>	
	<p>5- Muestra un mensaje preguntando por confirmación, si cancela mantiene al jugador en el torneo.</p>
	<p>7- El caso de uso termina.</p>
<p>Curso alternativo</p>	
<p>1- Árbitro principal: Realiza búsqueda de jugadores.</p> <p>2- Árbitro principal: Manda a listar según los parámetros.</p> <p>4- Árbitro auxiliar: Selecciona los jugadores que registrará en el torneo</p> <p>5- Árbitro auxiliar: Manda a registrar.</p>	<p>3- Muestra el listado de jugadores según la búsqueda.</p> <p>6- Registra al jugador en el torneo.</p> <p>7- Ejecuta el caso de uso Enviar notificación para informar al árbitro.</p> <p>Mensaje: Hola “nombre”, ha sido registrado para participar en nuestro torneo. Muchas gracias por desear trabajar junto a nosotros.</p>
	<p>7- El caso de uso termina.</p>

Referencias:

Listado de requerimientos:

- El sistema debe emitir mensajes informando al jugador sobre su eliminación del torneo.
- Debe quedar confeccionado el listado final de los jugadores que participarán en el torneo.

Casos de uso asociados: Enviar notificación (include).

Precondiciones: El sistema debe tener el listado de los jugadores que participarán en el torneo.

Poscondiciones: El listado queda actualizado después de las operaciones pertinentes.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

Caso de uso 5: Invitar jugadores

Prioridad: Secundario

Actores: Árbitro principal

Propósito: Tener conocimiento de todos los jugadores que están dispuestos a participar en el evento.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
1- Árbitro principal: Recibe el listado de los jugadores propuestos a participar. 2- Árbitro principal: Solicita una búsqueda de los jugadores en el sistema. 3- Árbitro principal: Introduce los parámetros necesarios para la búsqueda.	4- Muestra el listado de los jugadores que cumplan los parámetros.
5- Árbitro principal: Selecciona los jugadores en correspondencia con la lista de propuestas. 6- Árbitro principal: Ya una vez seleccionado el jugador, le envía una notificación.	7- Ejecuta el caso de uso Enviar notificación para verificar la disponibilidad. Mensaje: Hola "nombre", en el mensaje anterior

Referencias:Listado de requerimientos:

- El sistema debe emitir mensajes de invitaciones a los jugadores y si alguno no está disponible debe informar sobre esta situación el árbitro principal.
- Debe quedar confeccionado el listado final de los jugadores que participarán en el torneo, considerando que los jugadores que confirmaron su participación son el total necesario para desarrollar el evento, de no ser así, se deja abierta una sección de inscripciones.

Casos de uso asociados: Enviar notificación (include).

Precondiciones: El árbitro principal debe tener inicialmente la lista de los jugadores propuestos.

Poscondiciones: Queda confeccionado el listado final de los jugadores que participarán en el torneo, en caso de que no sean el total de los jugadores necesarios para desarrollar el evento, sería solo el listado de los jugadores disponibles ya inscritos por el sistema (en ambos casos la lista va a contener datos tales como: Nombre(s) y Apellidos, Sexo, Elo, Título, Federación y Fecha de nacimiento).

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

Caso de uso 6: Realizar pareo de ronda

Prioridad: Crítico

Actores: Árbitro principal (inicia)

Propósito: Realizar el pareo correcto para cada ronda del torneo.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
1- Árbitro principal: Si es la primera ronda solicita la lista de los jugadores.	2- Muestra la lista ordenada por fuerza y de los que no se conozca su elo, ordenados alfabéticamente. 3- Verifica la lista y selecciona al jugadores de

	<p>más fuerza y le pregunta con qué color de pieza desea jugar.</p> <p>4- Se ejecuta el caso de uso Enviar notificación para solicitar el color de pieza.</p> <p>Mensaje: Por ser usted el jugador(a) de mayor elo del torneo debe seleccionar el color de pieza con que desea jugar para la primera ronda, piezas <u>blancas</u> o <u>negras</u>.</p>
<p>5- Jugador: Selecciona el color de pieza.</p>	<p>6- Captura y guarda el color de pieza seleccionado.</p> <p>7- Se ejecuta el caso de uso Enviar notificación sobre la selección al árbitro principal.</p> <p>Mensaje: Ha decidido jugar con las piezas <u>blancas</u>.</p>
<p>8- Árbitro principal: Verifica la respuesta.</p>	
<p>9- Árbitro principal: Manda a realizar la división en grupos.</p>	
	<p>10- Realiza la división en grupos, los de mayor fuerza en la primera mitad y el resto en la segunda mitad.</p>
	<p>11- Asigna los colores al resto de los jugadores a partir del seleccionado por el jugador de más fuerza o por el árbitro.</p>
	<p>12- Si no hay jugador bye se hace las asignaciones de las partidas.</p>
	<p>13- Muestra resultado de las asignaciones de partidas.</p>
<p>14- Árbitro principal: Observa el resultado.</p>	
<p>15- Árbitro principal: Ordena notificar el resultado a los jugadores y árbitros auxiliares.</p>	<p>16- Ejecuta el caso de uso Enviar notificación</p> <p>Mensaje: Han quedado confeccionadas las</p>

	asignaciones de partidas de esta ronda, usted mismo puede <u>verlas</u> .
	17- El caso de uso termina.
Cursos alternativos	
5- Jugador: No selecciona el color de pieza.	6- Si hasta un tiempo determinado el jugador no ha seleccionado un color de pieza se ejecuta el caso de uso Enviar notificación informando al árbitro principal. Mensaje: El jugador no ha seleccionado el color de pieza en el tiempo establecido, usted está autorizado a hacerlo para que pueda comenzar el torneo, <u>blancas o negras</u> .
7- Árbitro principal: Selecciona el color de pieza y se realiza la actividad 9 del curso normal.	
Cursos alternativos	
1- Si no es la primera ronda manda a ejecutar a actividad 12 del flujo normal.	
Cursos alternativos	
	12- Si hay jugador bye descansa (los jugadores que quedan como bye descansan solo una vez durante el torneo), se crean las partidas y se realiza la actividad 13 del curso normal.

Referencias:

Listado de requerimientos:

- El sistema debe emitir mensajes a los jugadores y árbitros de como quedó la asignación de las partidas correspondientes a cada ronda.
- Deben quedar clara las asignaciones de partidas correspondientes a cada ronda.

Casos de uso asociados: Enviar notificación (include).

Precondiciones: El árbitro principal debe tener inicialmente la lista final de los jugadores que participarán en el torneo, esto es solo para el caso de la primera ronda, para las restantes debe tener además los resultados de la ronda anterior.

Poscondiciones: Quedan hecha las asignaciones de partidas.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

Caso de uso 7: Publicar resultado de ronda

Prioridad: Crítico

Actores: Árbitro auxiliar (inicia)

Propósito: Mostrar los resultados de la ronda a todos los participantes en el torneo.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
1- Árbitro auxiliar: Solicita resultado de ronda.	2- Muestra los resultados de la ronda
3- Árbitro auxiliar: Observa los datos y manda a publicarlos.	4- Actualiza las posiciones de los jugadores. 5- Confecciona la tabla de resultados y la publica. 6- Ejecuta el caso de uso Enviar notificación para informar el resultado de la ronda. Mensaje: Ya está disponible el <u>resultado de la ronda “#” del torneo.</u>
	6- El caso de uso termina

Referencias:

Listado de requerimientos:

- El sistema debe emitir mensajes a los jugadores sobre los resultados de la ronda.
- Deben quedar publicado el resultado de la ronda.

Casos de uso asociados: Publicar resultado final (extend) y Enviar notificación (include).

Precondiciones: Tener registrado el resultado de la ronda.

Poscondiciones: Queda publicado el resultado de la ronda y listo para la próxima ronda que se efectuará.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

Caso de uso 8: Publicar resultado final

Prioridad: Crítico

Actores: Árbitro principal (inicia)

Propósito: Mostrar el resultado final del torneo teniendo en cuenta los resultados de todas las rondas.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
1- Árbitro principal: Solicita todos los resultados generales.	2- Muestra todos los resultados.
3- Árbitro principal: Manda a publicar los resultados finales.	4- Realiza el cálculo final de elo. 5- Actualiza las posiciones de los jugadores. 6- Confecciona la tabla de resultados y la publica. 7- Ejecuta el caso de uso Enviar notificación para informar el resultado final del torneo. Mensaje: Ya está disponible el <u>resultado final del torneo</u> .
	8- El caso de uso termina

Referencias:

Listado de requerimientos:

- El sistema debe emitir mensajes a los jugadores y árbitros sobre el resultado final.
- Deben quedar publicado el resultado final del torneo.

Casos de uso asociados: Enviar notificación (include).

Precondiciones: Están registrados los resultados de todas las rondas.

Poscondiciones: Queda publicado el resultado final del torneo.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

Caso de uso 9: Enviar notificación

Prioridad: Auxiliares

Actores: Árbitro principal y árbitro auxiliar

Propósito: Mostrar mensajes de información a cada uno de los involucrados en el torneo cada vez que se vaya a realizar una actividad que requiera de una notificación.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
	1- Identifica la actividad realizada. 2- Selecciona el texto del mensaje que corresponde a la actividad siempre que esta lo requiera. 3- Envía el mensaje a su destinatario.
	4- El caso de uso termina.

Referencias:

Listado de requerimientos:

- El sistema debe emitir mensajes a los jugadores y árbitros cada vez que se realice alguna actividad dentro del torneo que requiera de notificaciones.

Casos de uso asociados: -

Precondiciones: Se realizó una actividad que requiere de una notificación.

Poscondiciones: El mensaje llega a su destino final.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

Caso de uso 10: Jugar partida de ronda

Prioridad: Crítico

Actores: Árbitros principal (inicia) y Jugador

Propósito: Jugar las partidas asignadas a cada pareja de jugadores.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
1- Árbitro principal: Informa a los jugadores que deben tomar posición para comenzar la partida.	2- Ejecuta el caso de uso Enviar notificación. Mensaje: Debe estar listo para jugar, en segundos comenzarán las partidas de esta ronda.
3- Árbitro principal: Da la orden de comenzar a jugar.	4- Ejecuta el caso de uso Enviar notificación. Mensaje: Ya puede comenzar a jugar. Datos de la partida. <u>Aceptar</u> o <u>Cancelar</u> .
5- Jugador: Juega si acepta la partida que le corresponde.	6- Registra los resultados de la partida.
	7- El caso de uso termina
Cursos alternativos	
5- Jugador: Si cancela la partida y no la juega pierde.	6- Muestra un mensaje informándole al jugador que ha perdido la partida.
	7- El caso de uso termina.

Referencias:

Listado de requerimientos:

- Informarle a cada jugador quien es su contrario.
- Deben quedar registrados los resultados de las partidas.

Casos de uso asociados: -

Precondiciones: Se debe haber realizado un pareo antes.

Poscondiciones: Quedan registrados los datos de las partidas y listos para enviar los resultados.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

Caso de uso 11: Inscribirse en torneo

Prioridad: Secundario

Actores: Jugador

Propósito: Inscribirse para participar en el torneo.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
1- Jugador: Solicita la inscripción.	2- Verifica que los datos del jugador estén acordes con las bases del torneo. 3- Si los datos están acordes lo inscribe. 4- Le muestra un mensaje informándole que la inscripción de realizó satisfactoriamente. 5- Ejecuta el caso de uso Enviar notificación para informar sobre inscripción al árbitro principal. Mensaje: El jugador "nombre" se ha inscrito en el torneo.
	7- El caso de uso termina
Cursos alternativos	

	3- Si los datos no están acordes y le muestra un mensaje informándole que no puede inscribirse porque no cumple con las bases del torneo.
	4- El caso de uso termina

Referencias:

Listado de requerimientos:

- El jugador que entra al sistema para inscribirse no estaba en la lista de propuestas a jugar en el torneo inicialmente, estas son inscripciones para plazas sobrantes.
- Informarle al árbitro principal sobre las nuevas inscripciones.

Casos de uso asociados: Enviar notificación (include)

Precondiciones: El torneo posee plazas sobrantes.

Poscondiciones: Queda adicionado un nuevo jugador al listado.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

Caso de uso 12: Autenticarse

Prioridad: Secundario

Actores: Árbitro auxiliar (Inicia), Jugador (inicia), Árbitro principal (inicia)

Propósito: Entrar al sistema para desempeñar el rol que le corresponde.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
1- Jugador, árbitro auxiliar o principal: Accede al sistema vía Web.	
2- Jugador, árbitro auxiliar o principal: Introduce usuario y contraseña.	
	3- Verifica si el usuario y la contraseña son correctos y que no esté en la lista de expulsados.

	4- Si son correctos y no está en la lista de expulsados entra al sistema.
	5- El caso de uso termina
Cursos alternativos	
5- Jugador, árbitro auxiliar o principal: Corrige los datos incorrectos e intenta de nuevo entrar al sistema hasta introducir el usuario y la contraseña indicados.	4- Si los datos son erróneos le muestra un mensaje informándole que el usuario o la contraseña son incorrectos. 6- Se ejecuta la actividad 4 del curso normal.
Cursos alternativos	
	4- Si los datos están correctos y está en la lista de expulsados no le permite entrar al sistema y le muestra un mensaje informándole que no puede entrar porque ha sido expulsado del torneo.
	5- El caso de uso termina.

Referencias:

Listado de requerimientos:

- Se debe mostrar siempre un mensaje en caso de error.

Casos de uso asociados: -

Precondiciones: Usuario sin autenticar.

Poscondiciones: Usuario autenticado y dentro del sistema.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

Caso de uso 13: Ver partidas online

Prioridad: Auxiliares

Actores: Observador (inicia)

Propósito: Observar partidas de los jugadores del torneo.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
1- Observador: Selecciona la partida que desea ver.	2- Le muestra la partida activa en un visualizador.
3- Observador: Observa la partida por un tiempo determinado.	
4- Árbitro auxiliar: Verifica que el observador no cometa ningún tipo de infracción.	
5- Observador: Si no comete ningún tipo de infracción puede ver tantas partidas desee.	
6- El caso de uso termina	
Cursos alternativos	
5- Observador: Comete algún tipo de infracción. 6- Árbitro auxiliar: Lo expulsa del torneo.	7- Lo registra en la lista de usuarios expulsados.

Referencias:

Listado de requerimientos:

- Si el observador es un jugador del torneo debe primeramente haber terminado su partida.

Casos de uso asociados: -

Precondiciones: Usuarios observando partidas.

Poscondiciones: En caso de que alguno de los que observa cometa una infracción es expulsado del torneo y registrado en la lista de expulsión.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no

sea grande y total disponibilidad de la aplicación.

Caso de uso 14: Consultar torneos anteriores

Prioridad: Auxiliares

Actores: Observador (inicia)

Propósito: Consultar datos y estadísticas de torneos anteriores.

Descripción:

Flujo de trabajo	
Acción del actor	Respuesta del sistema
1- Observador: Solicita el listado de los torneos anteriores.	2- Muestra el listado de los torneos anteriores.
3- Observador: Consulta el listado de los torneos anteriores.	
4- Observador: No solicita más información.	
5- El caso de uso termina.	
Cursos alternativos	
4- Observar: Solicita el listado de los jugadores que participó en cada uno de ellos.	5- Muestra el listado de los jugadores de torneos anteriores.
6- Observador: Consulta el listado de los jugadores.	
7- Observador: No solicita más información.	
8- El caso de uso termina.	
Cursos alternativos	
7- Observador: Solicita los datos de las partidas y las rondas en cada torneo.	8- Muestra los datos de las rondas y las partidas.
9- Observador: Consulta los datos de las rondas y las partidas.	

10- Observador: Observa la partida seleccionada.	
11- El caso de uso termina.	

Referencias:

Listado de requerimientos:

- Si el observador es un jugador del torneo debe primeramente haber terminado su partida.

Casos de uso asociados: -

Precondiciones: Solicitud del observador.

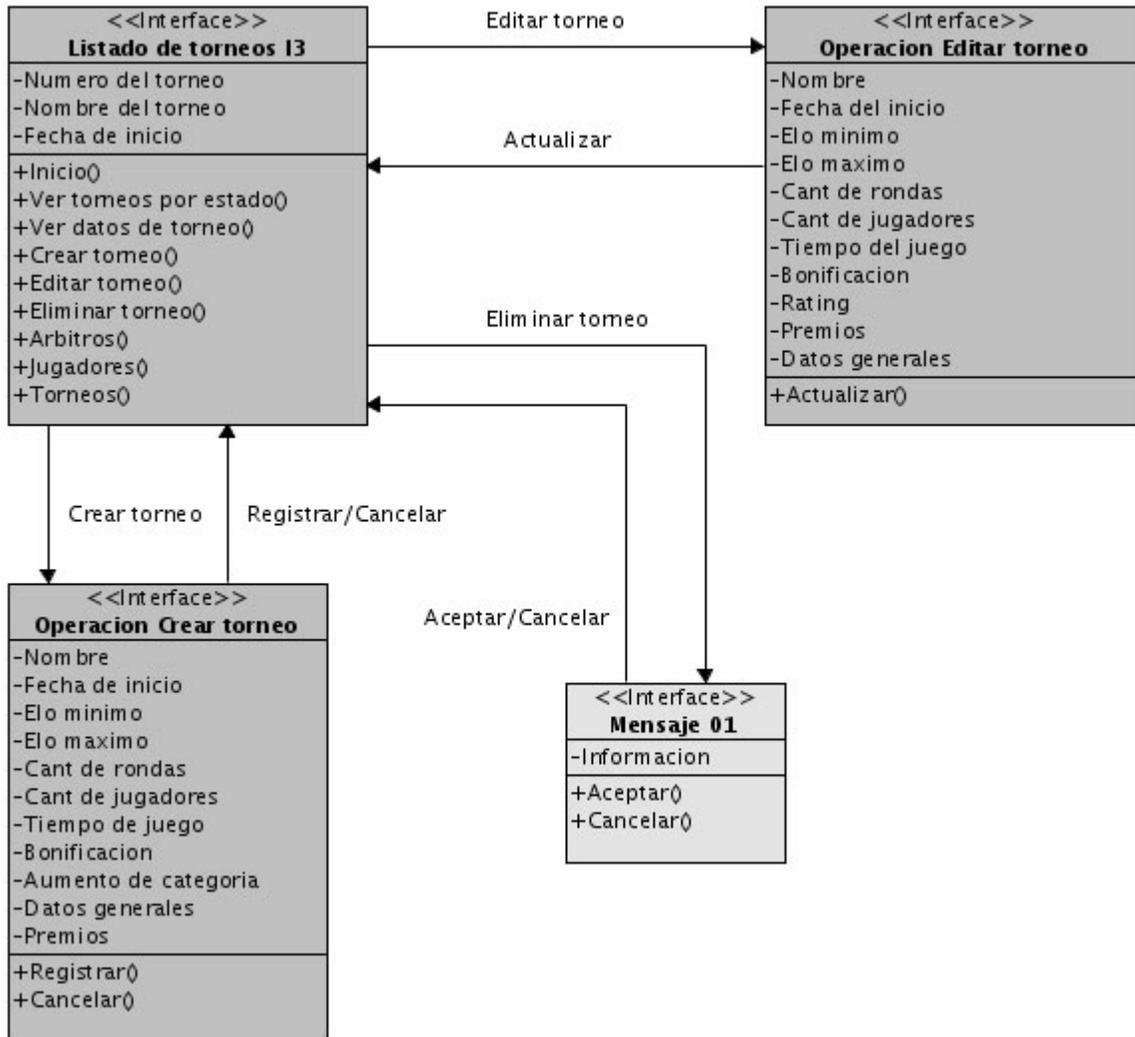
Poscondiciones: Datos consultados por parte del observador.

Requerimientos especiales: Debe existir una conexión de red estable aunque el ancho de banda no sea grande y total disponibilidad de la aplicación.

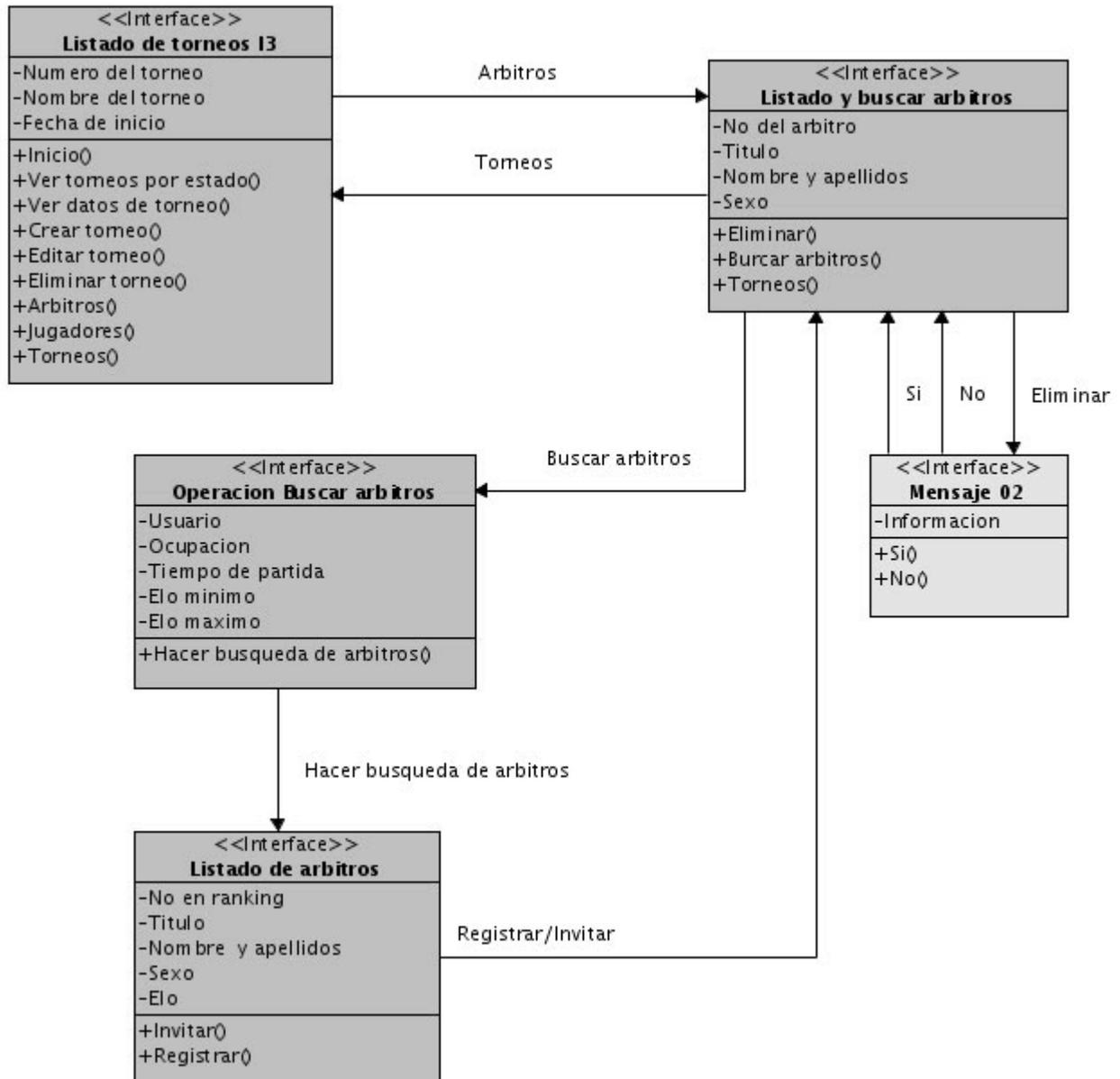
2.6.4 Prototipos de Interfaz de usuario.

A continuación se muestran los Prototipos de Interfaz de Usuarios para cada uno de los Casos de Uso incluyendo también el modelo general:

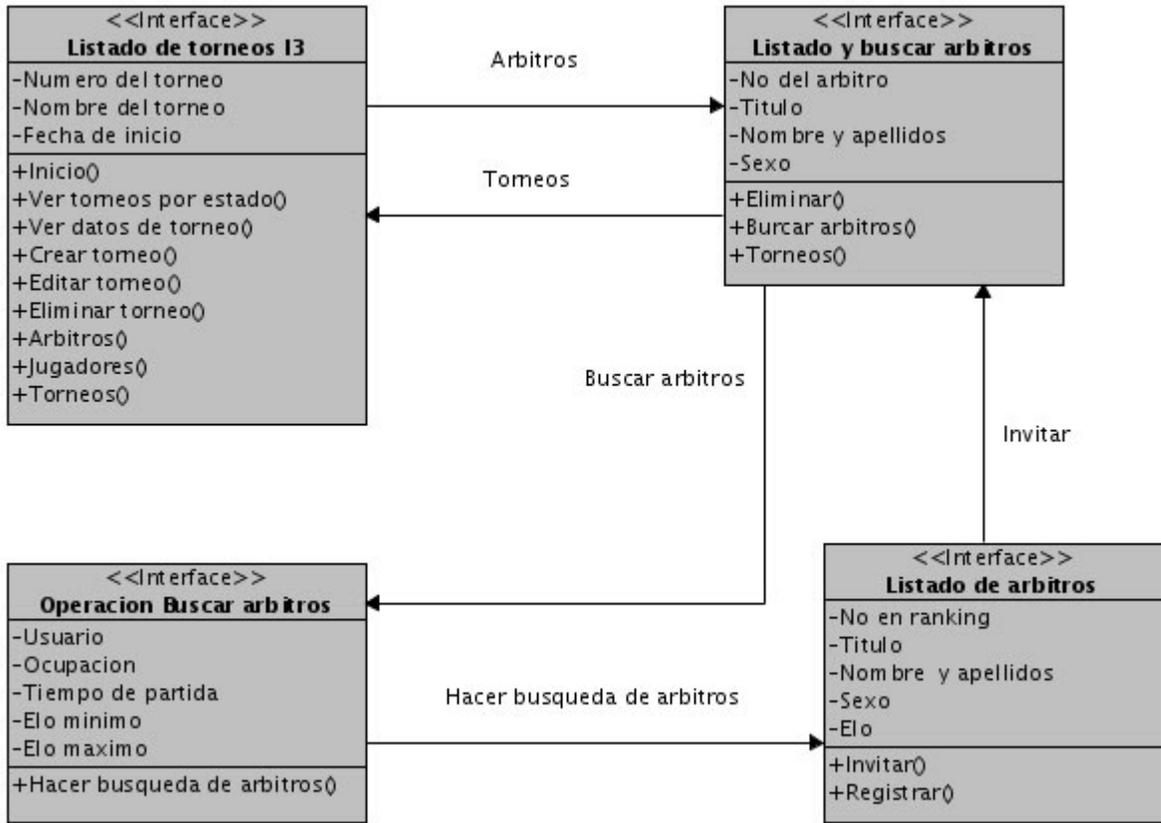
CUS-1: Gestionar torneos.



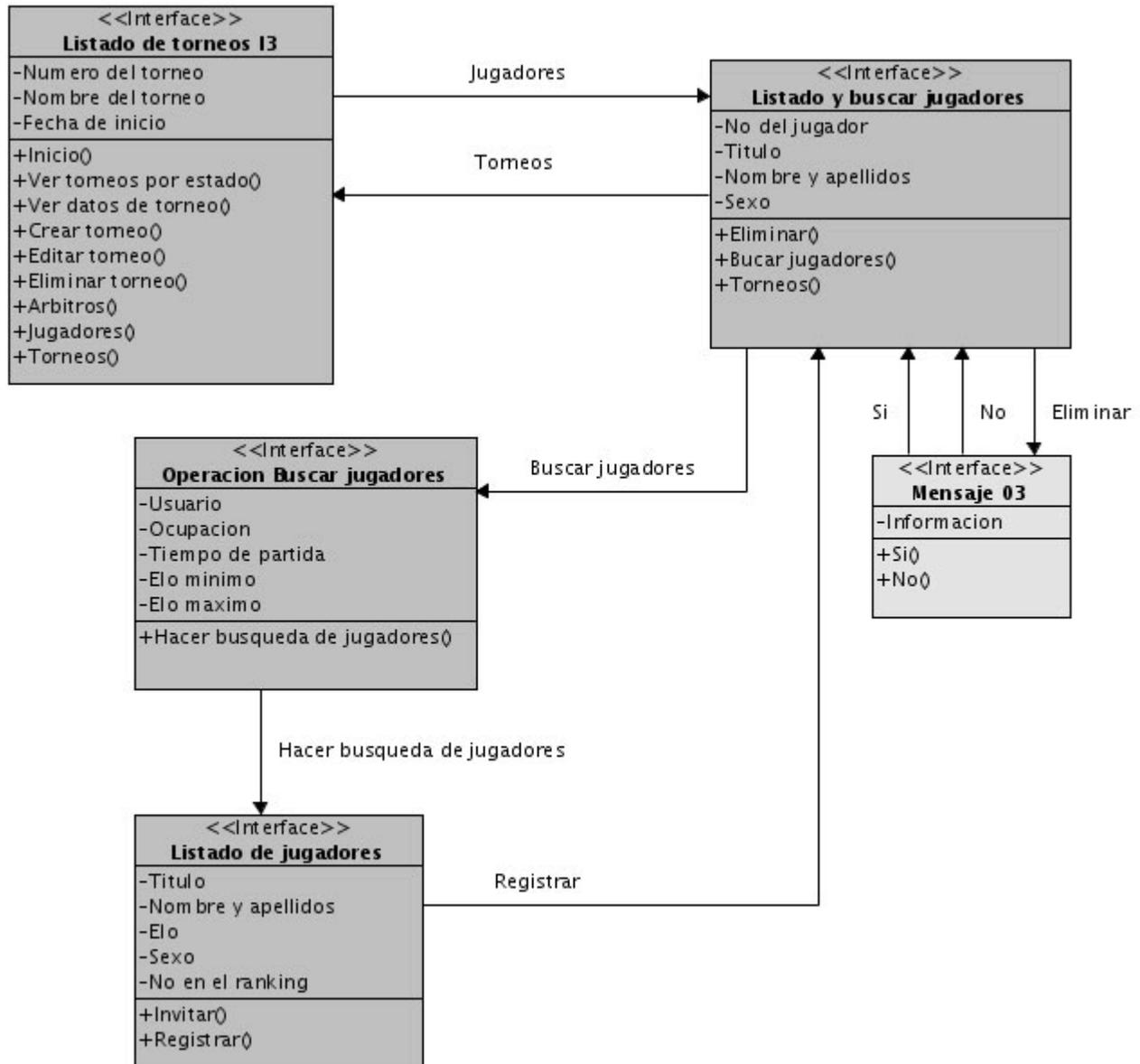
CUS-2: Gestionar árbitros auxiliares



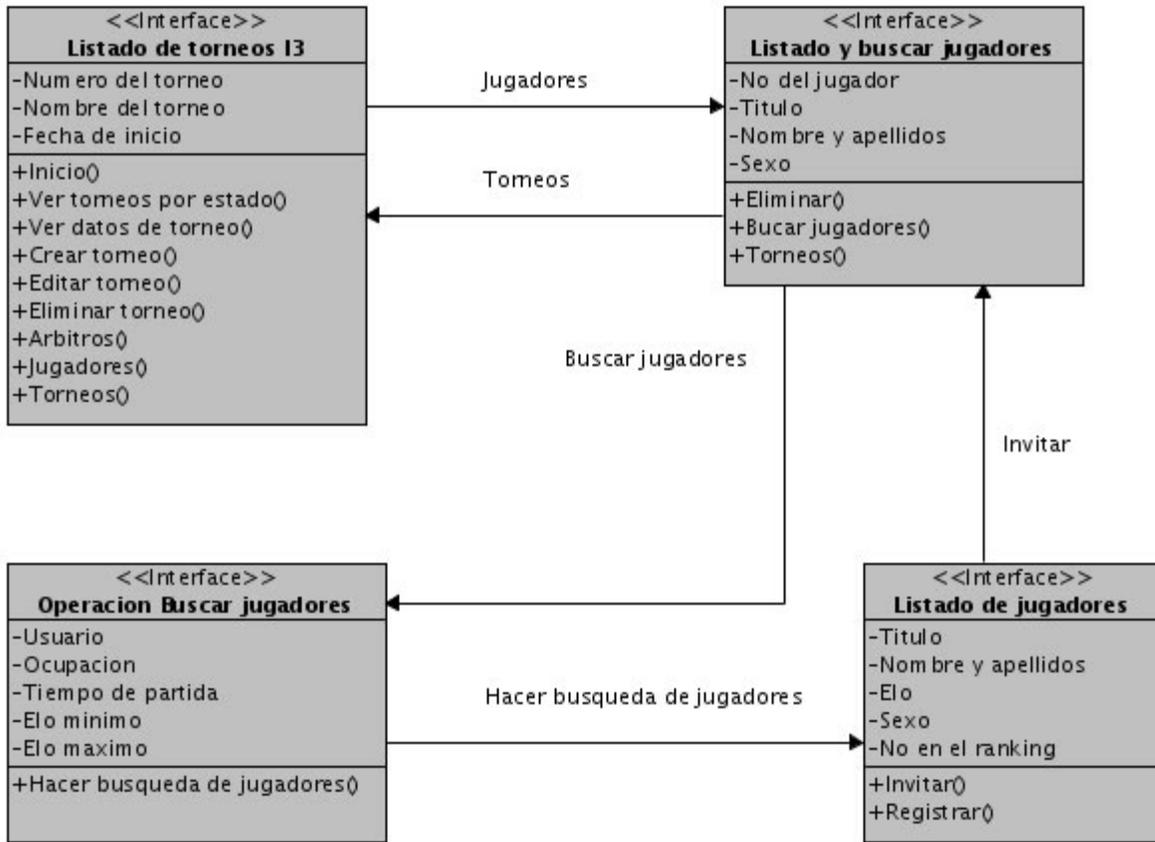
CUS-3: Invitar árbitros auxiliares



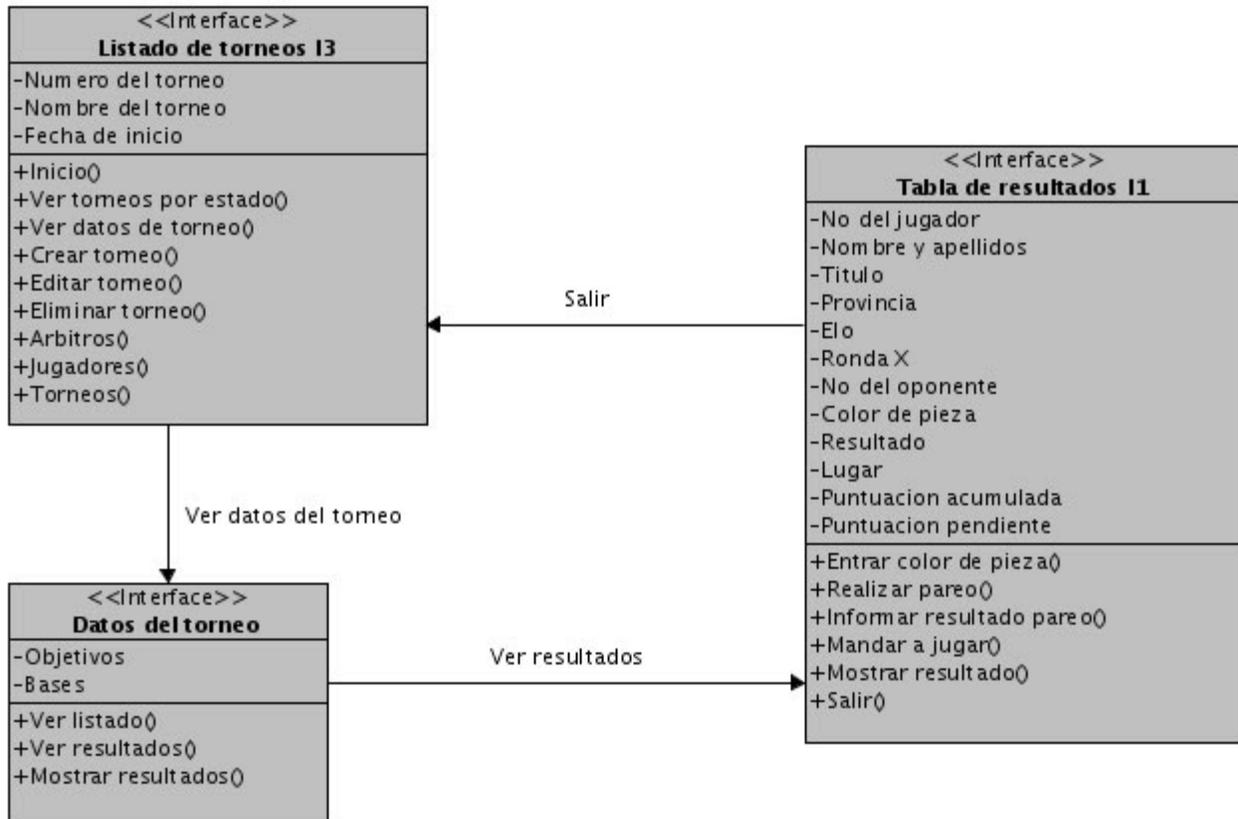
CUS-4: Gestionar jugadores



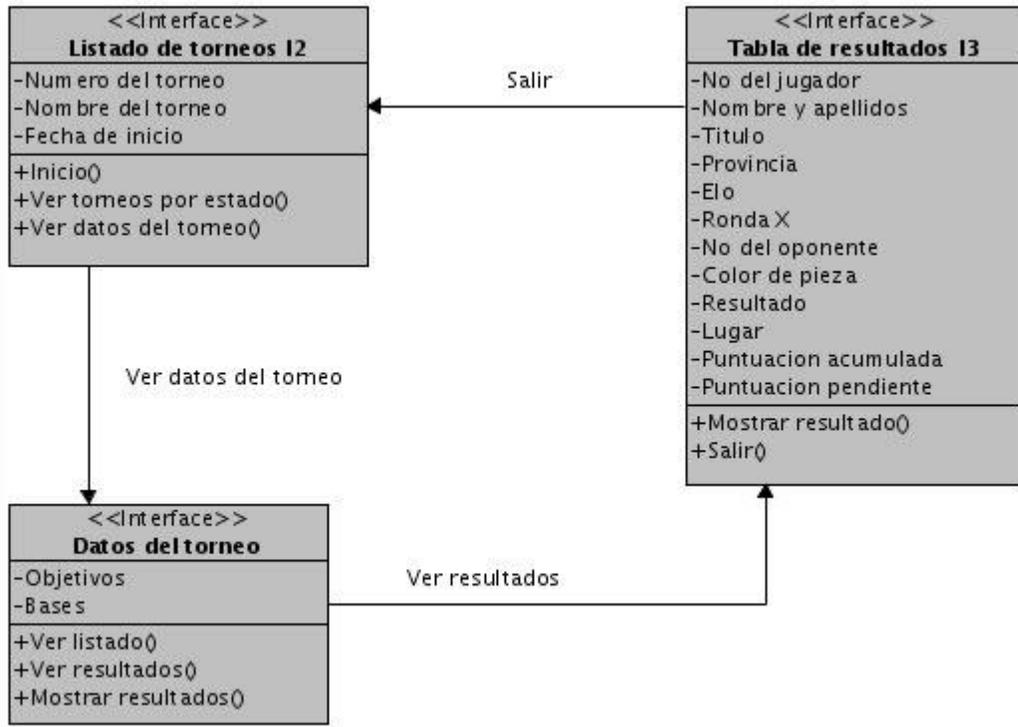
CUS-5: Invitar jugadores



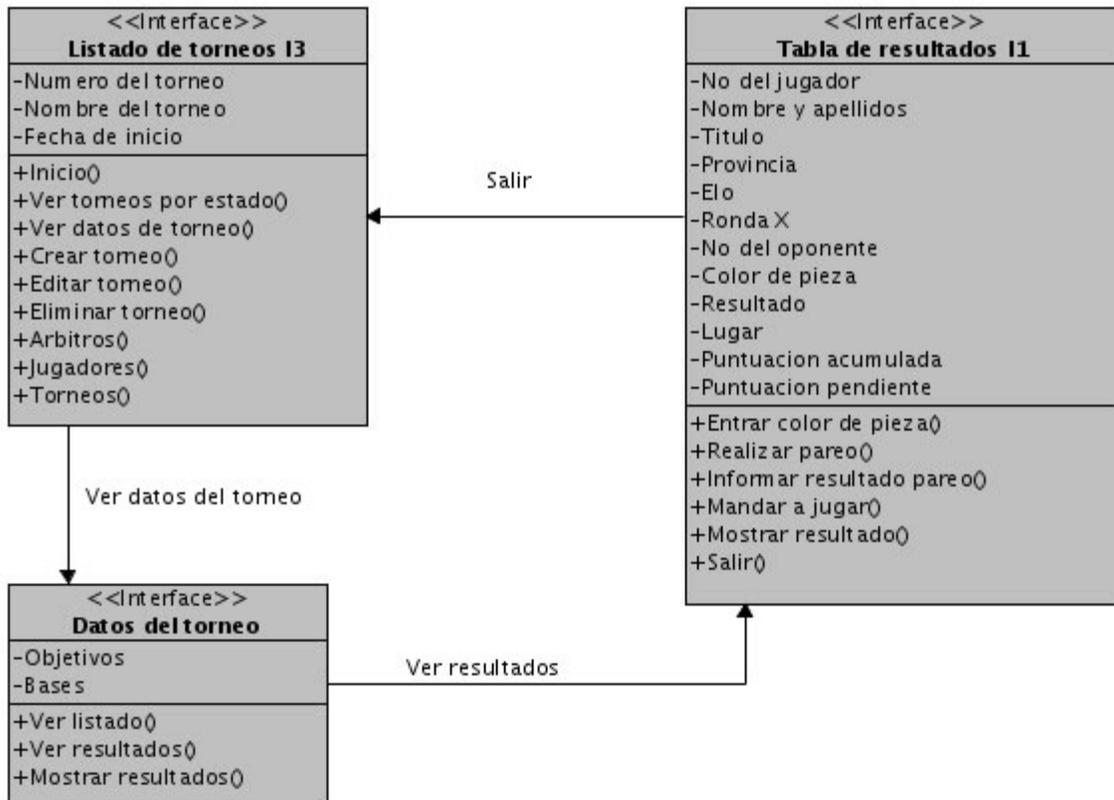
CUS-6: Realizar pareo de ronda



CUS-7: Publicar resultado de ronda



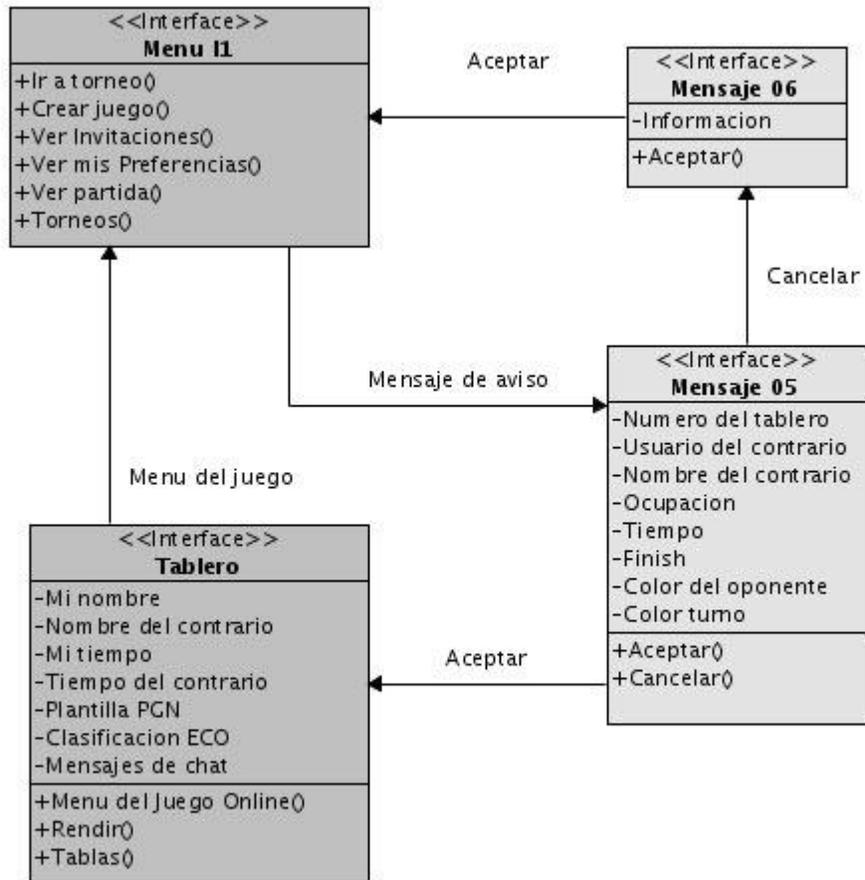
CUS-8: Publicar resultado final



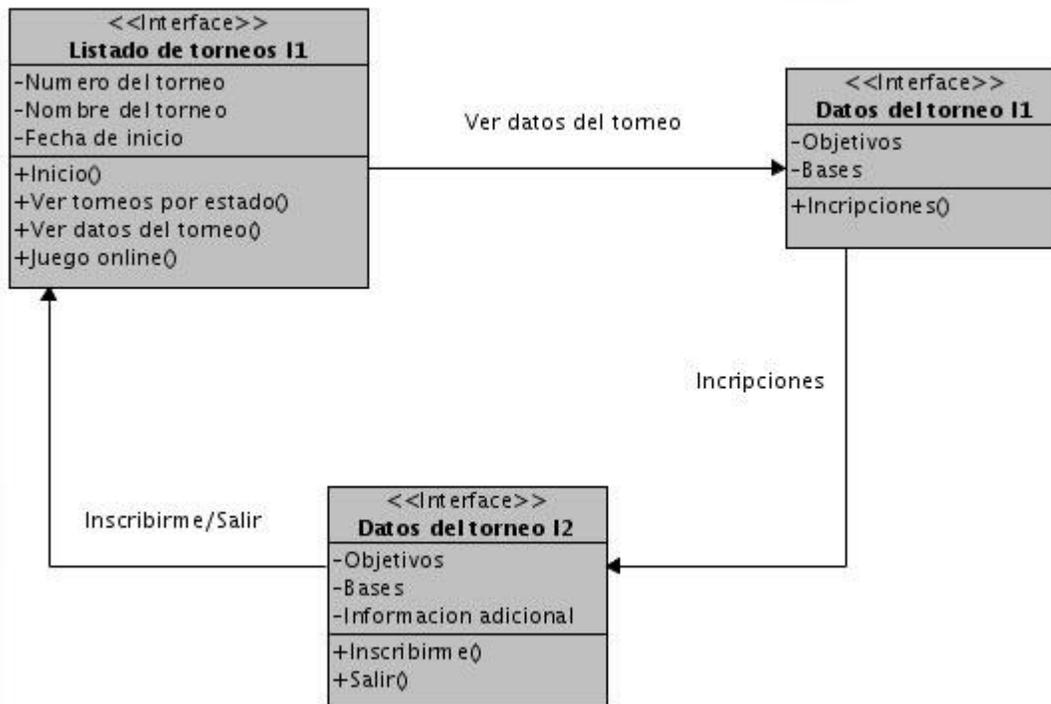
CUS-9: Enviar notificación

No genera interfaz, lo hace internamente el sistema.

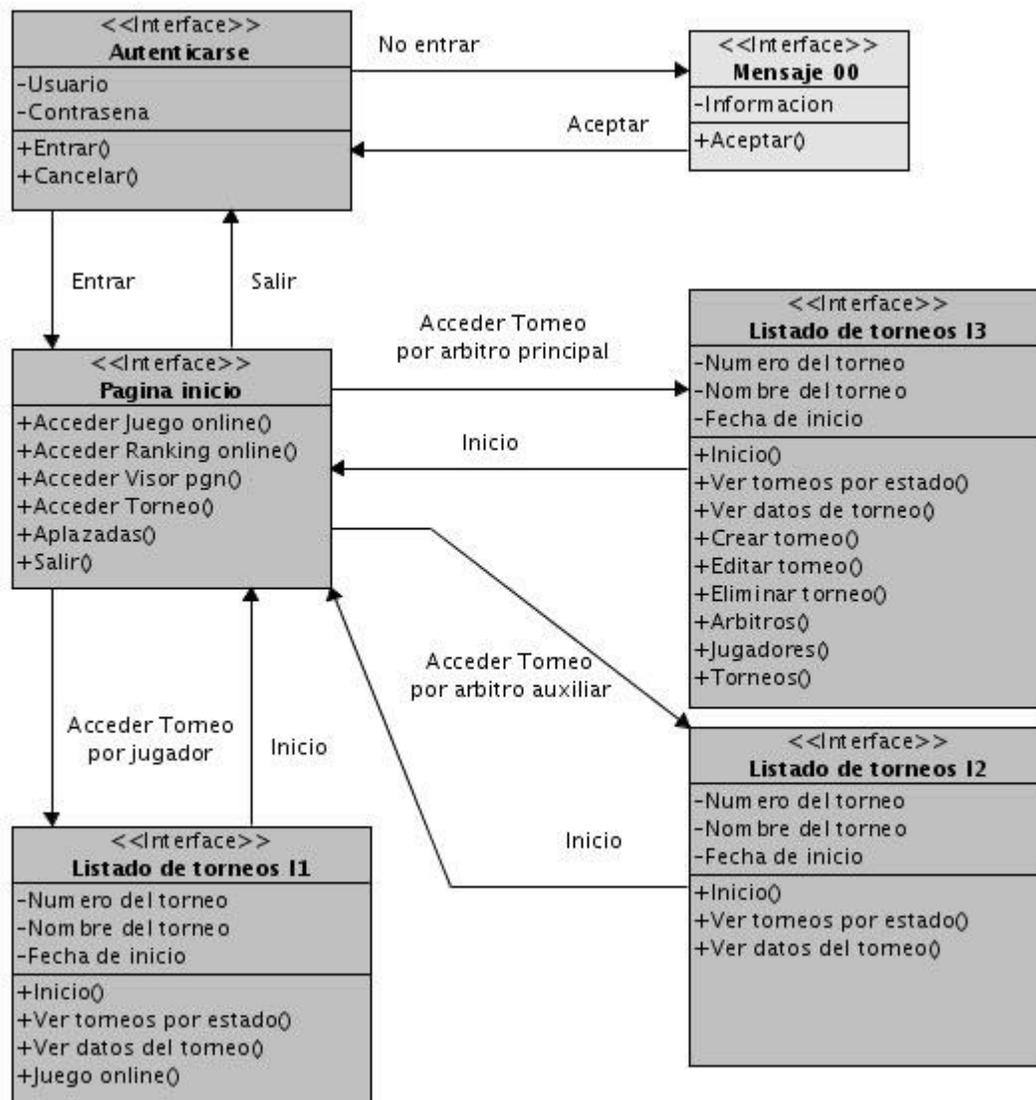
CUS-10: Jugar partida de ronda



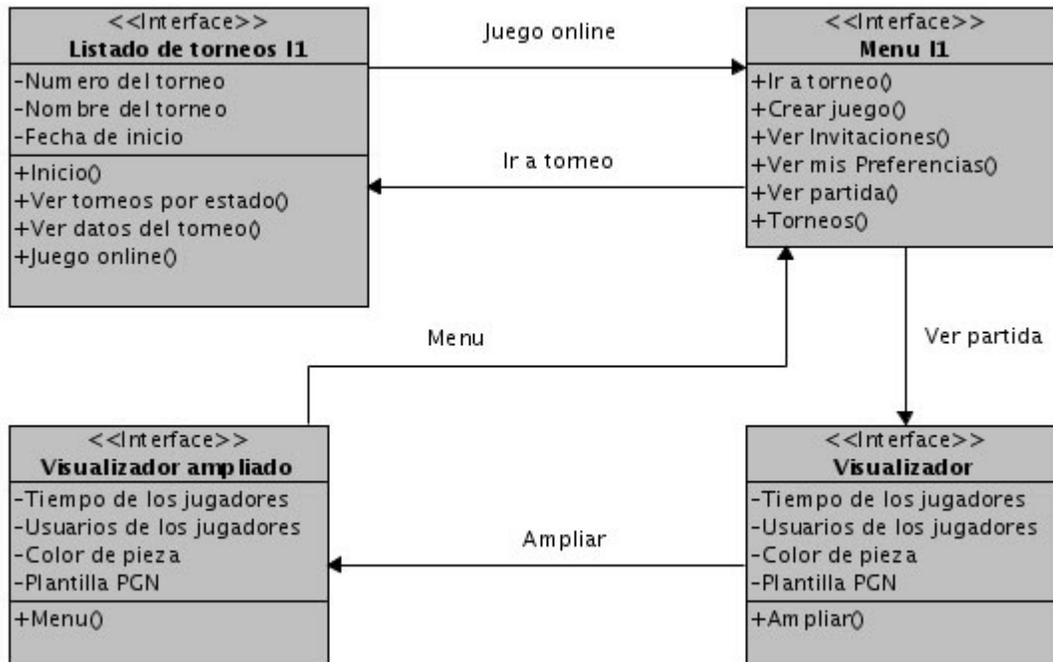
CUS-11: Inscribirse en torneo



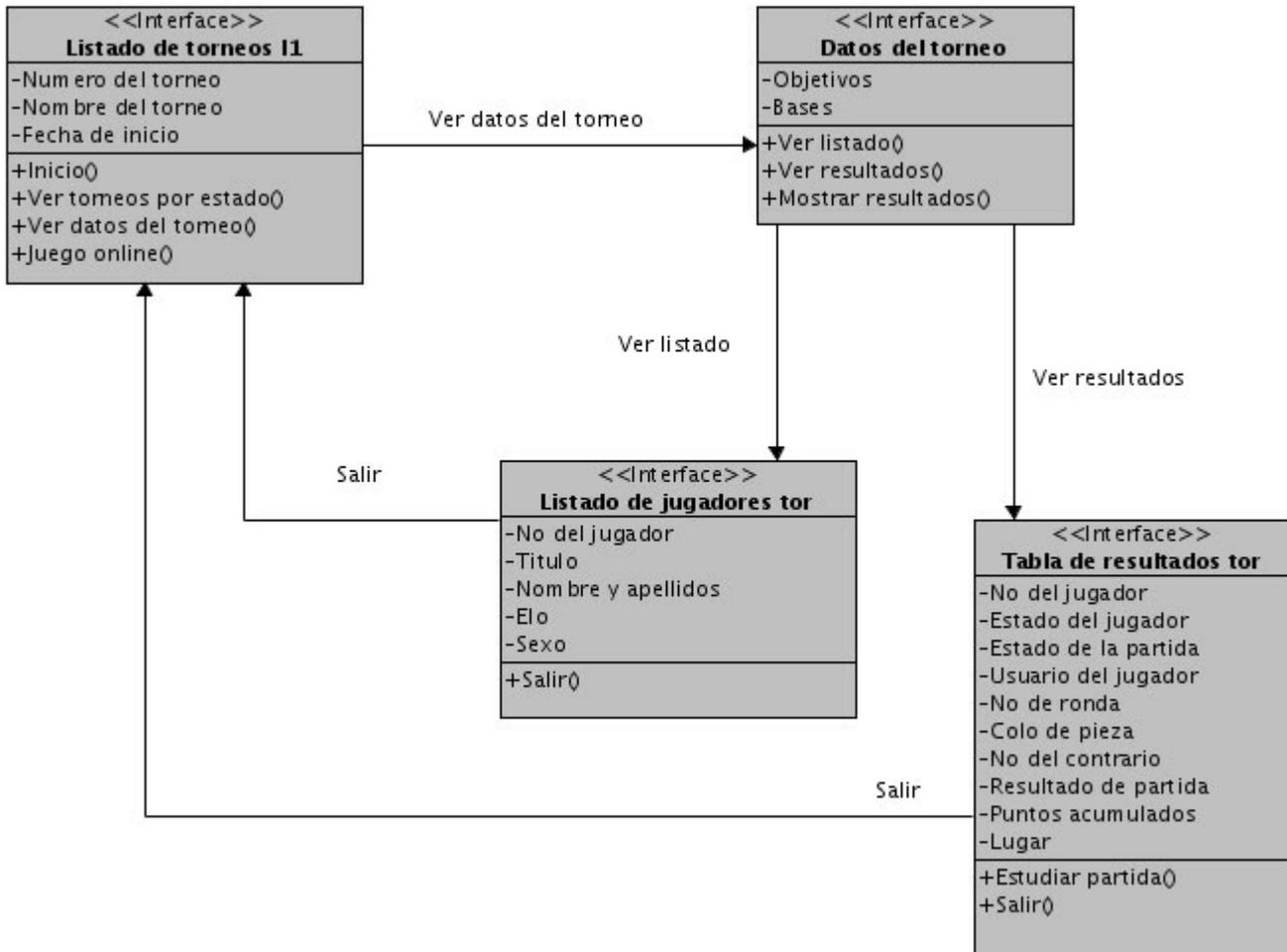
CUS-12: Autenticarse



CUS-13: Ver partidas online



CUS-14: Consultar torneos anteriores



Capítulo 3: Análisis y diseño del sistema

3.1- Introducción.

En este capítulo se trata la parte correspondiente al Análisis contenido dentro del flujo Análisis y diseño. 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 presentará también el Diseño que viene siendo una especie de refinamiento del Análisis que tiene en cuenta también los requisitos no funcionales.

3.2- Análisis.

Para una mejor comprensión de esta parte, se mostrarán cada uno de los diagramas generados dentro del análisis, específicamente los diagramas de clases y los diagramas de colaboración, compuestos por clases interfaz, clases controladoras y clases entidades. ¿Qué son cada una de estas clases y que elementos manejan?

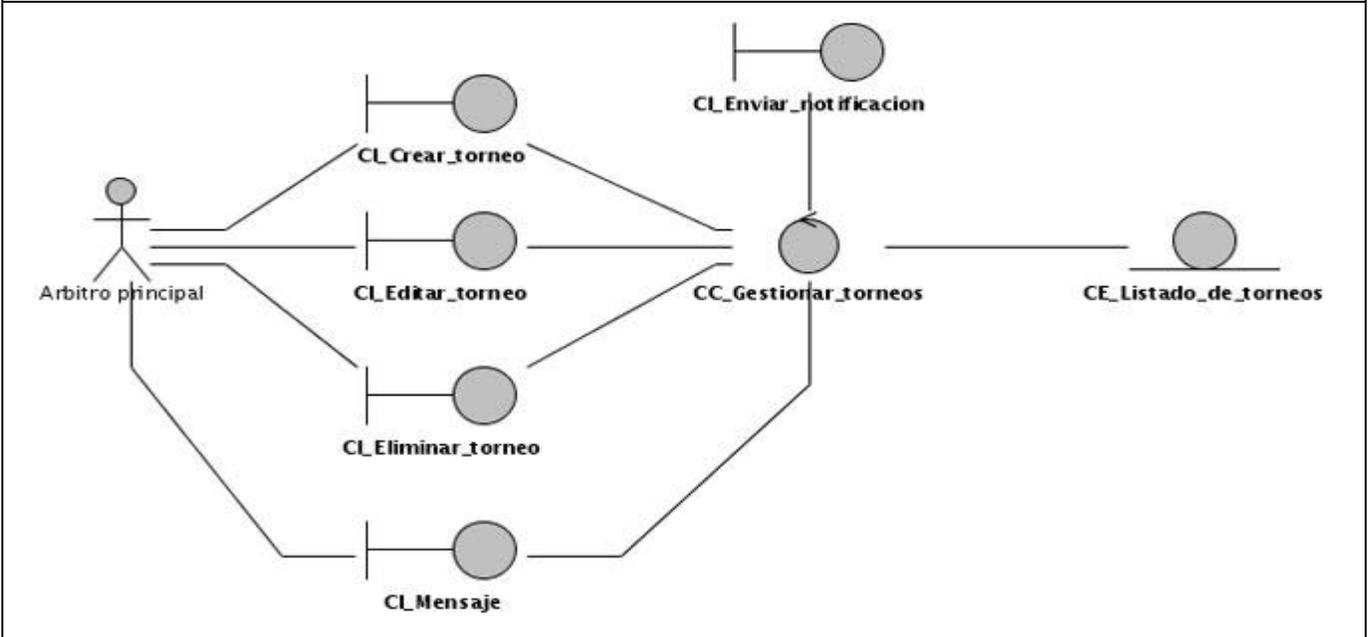
Clases interfaz: Modelan la interacción entre el sistema y sus actores.

Clases controladoras: Coordinan la realización de uno o unos casos de uso controlando las actividades de los objetos que implementan la funcionalidad del caso de uso y la interacción entre las clases interfaz y las clases entidades.

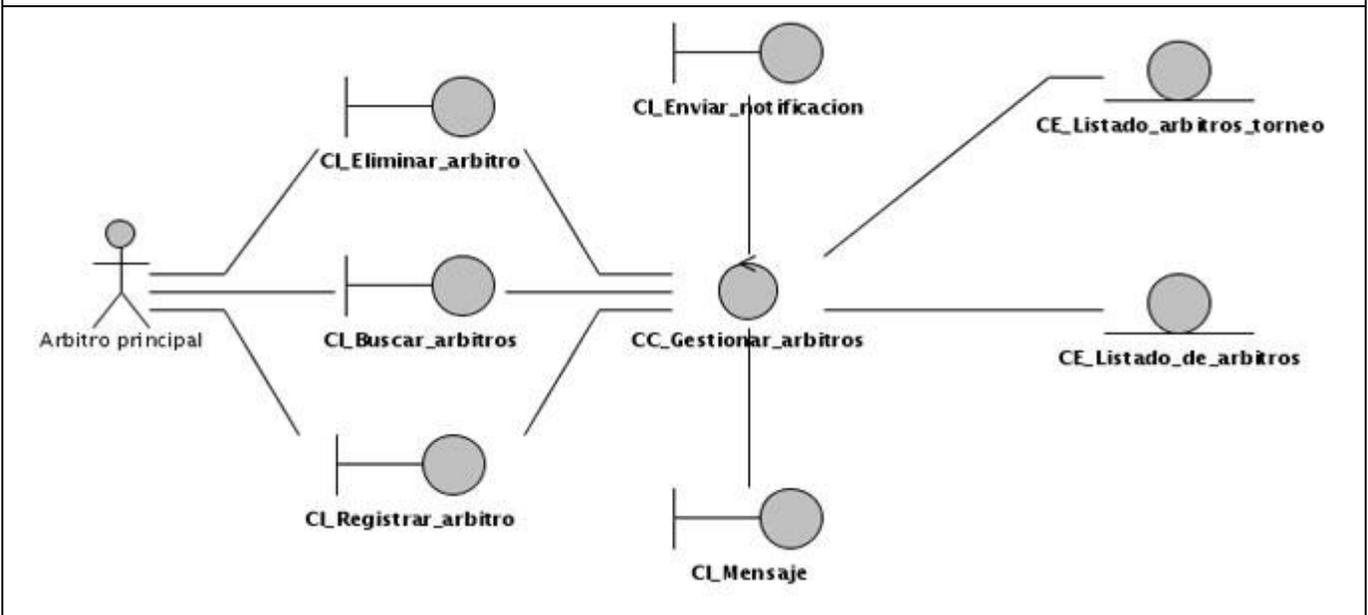
Clases entidades: Modelan información que posee larga vida y que es a menudo persistente (Jacobson, 2004).

3.2.1- Diagramas de Clases del Análisis.

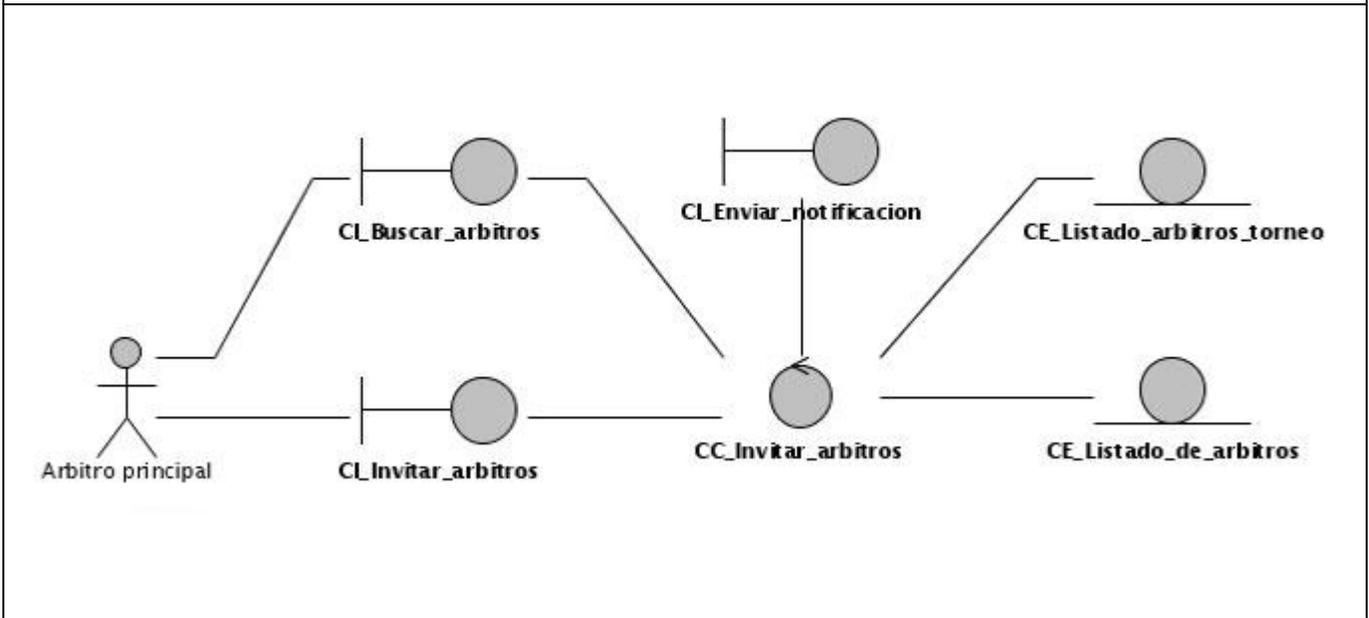
3.2.1.1- Caso de Uso: Gestionar torneos



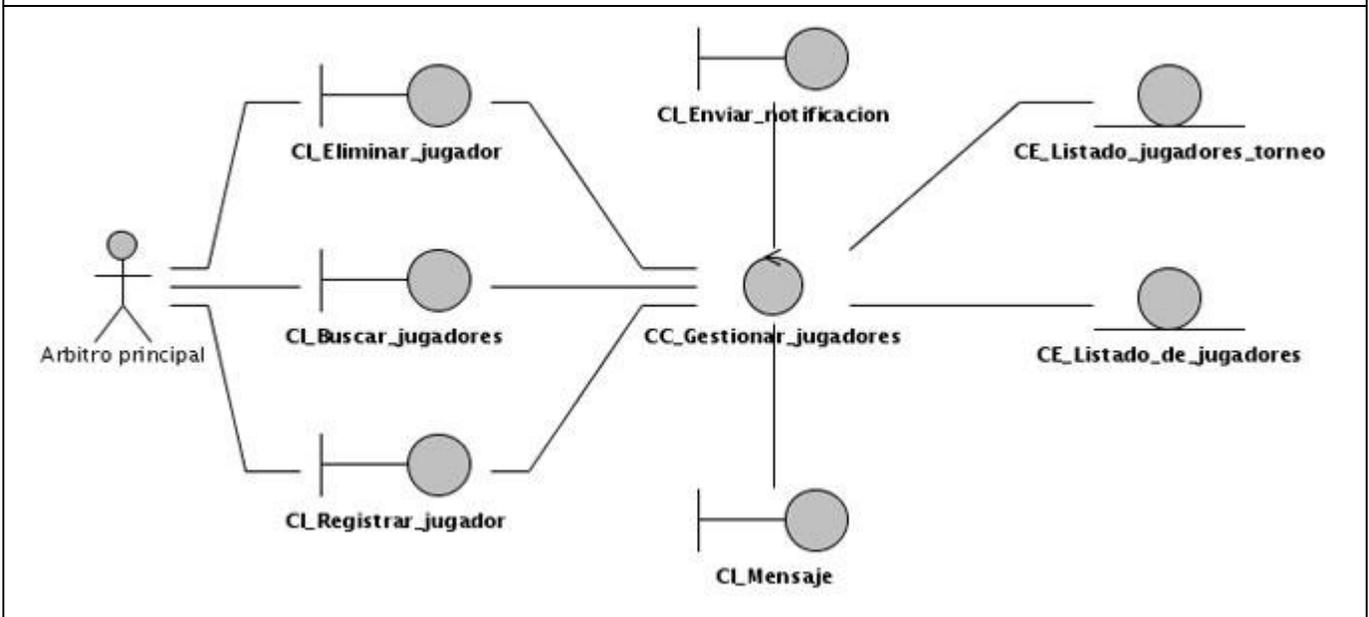
3.2.1.2- Caso de Uso: Gestionar árbitros auxiliares



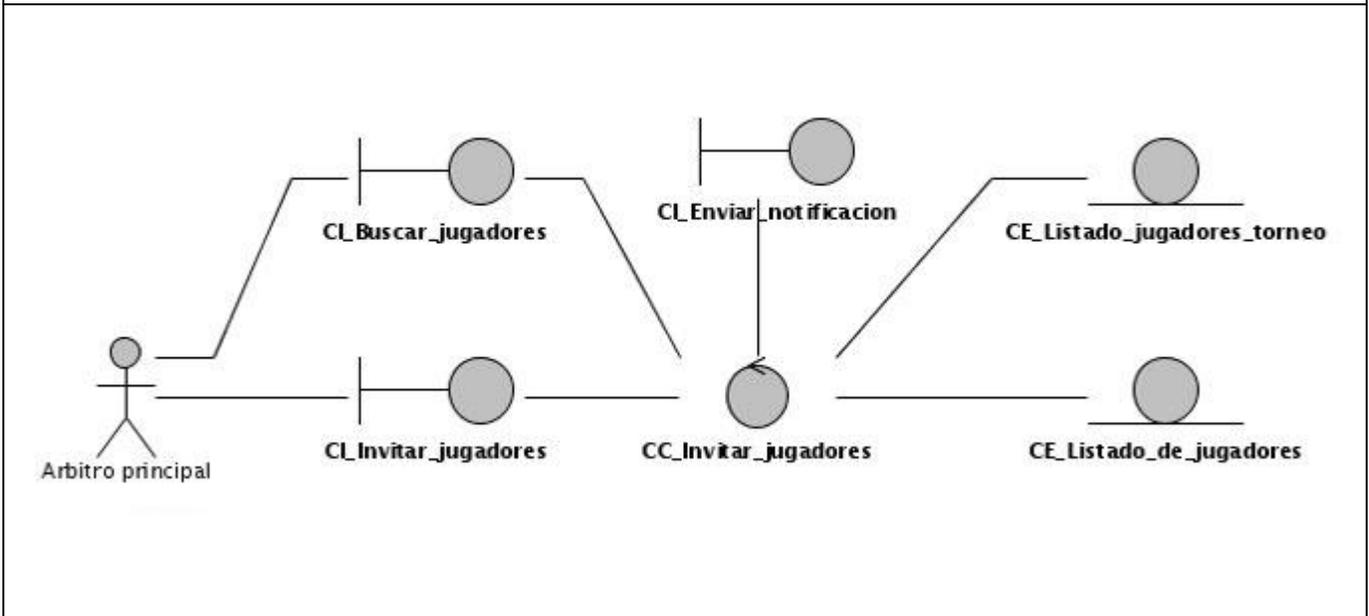
3.2.1.3- Caso de Uso: Invitar árbitros auxiliares



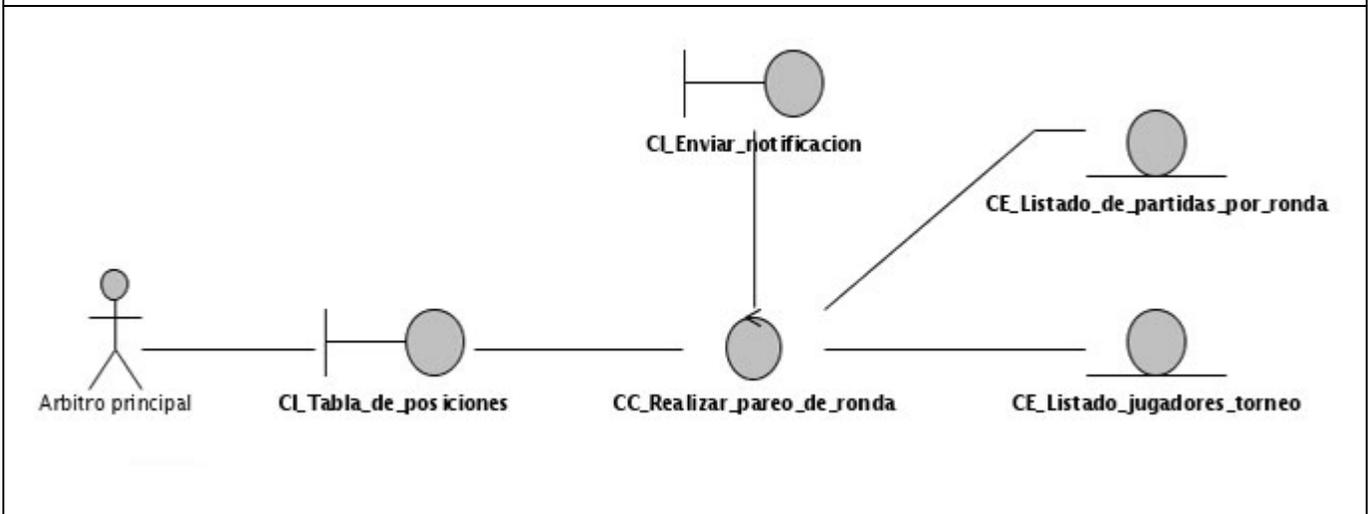
3.2.1.4- Caso de Uso: Gestionar jugadores



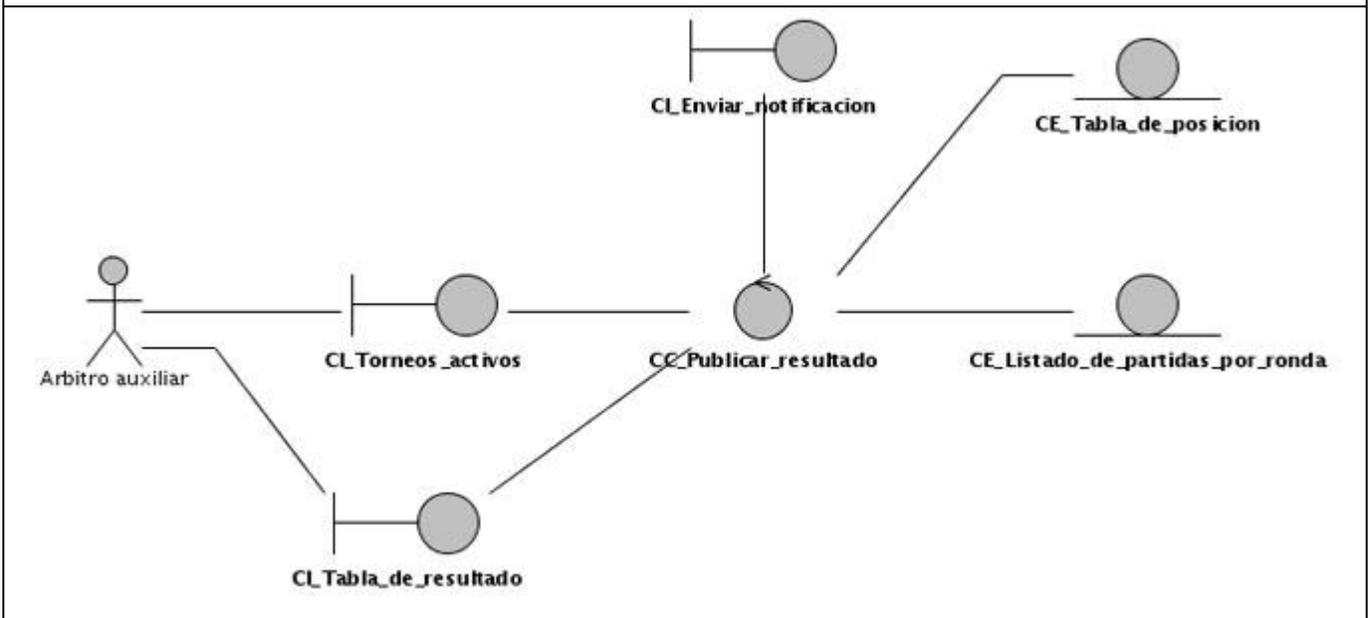
3.2.1.5- Caso de Uso: Invitar jugadores



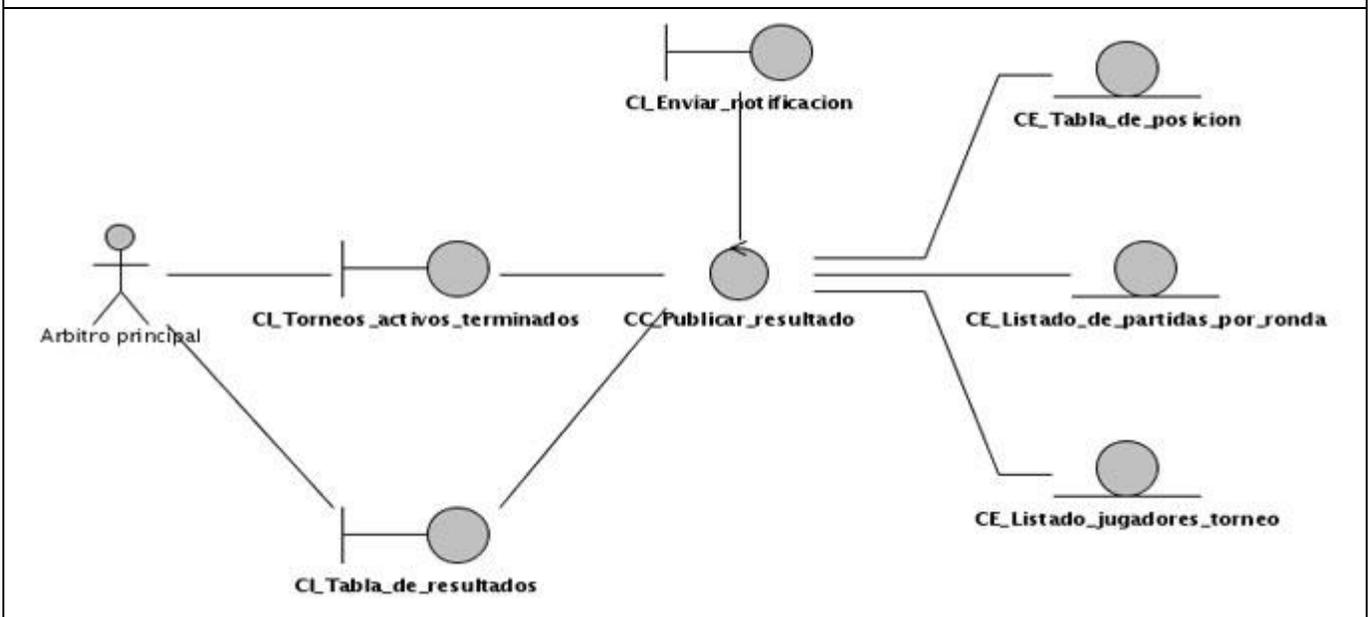
3.2.1.6- Caso de Uso: Realizar pareo de ronda



3.2.1.7- Caso de Uso: Publicar resultado de ronda



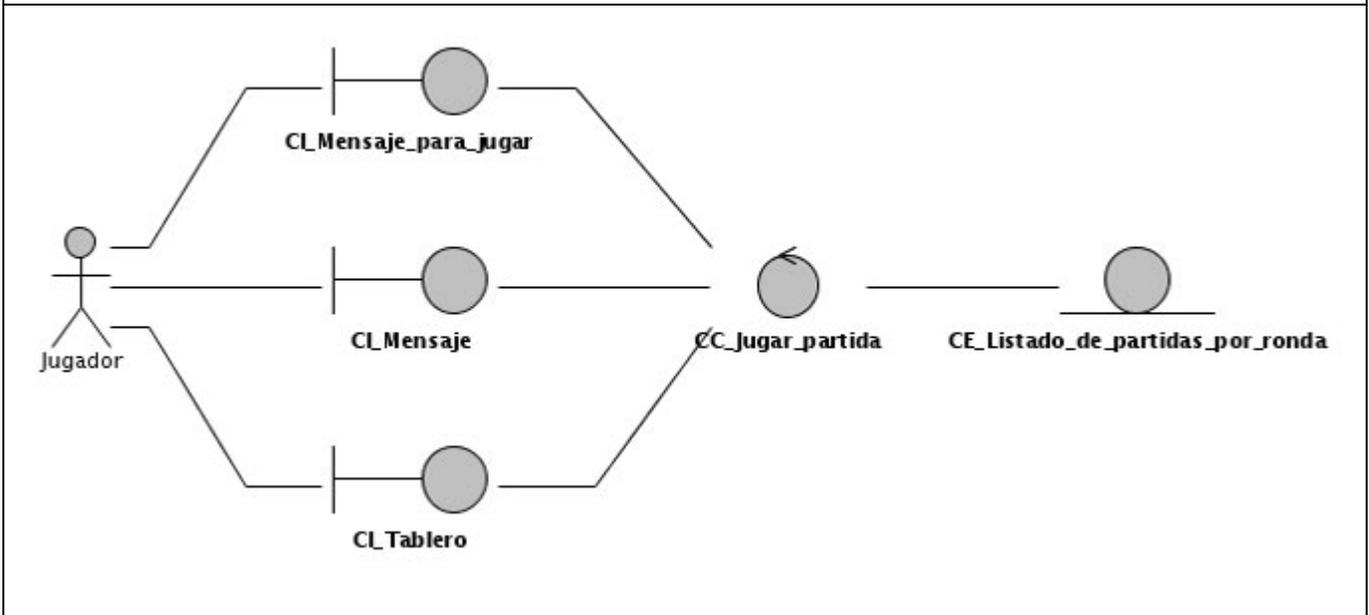
3.2.1.8- Caso de Uso: Publicar resultado final



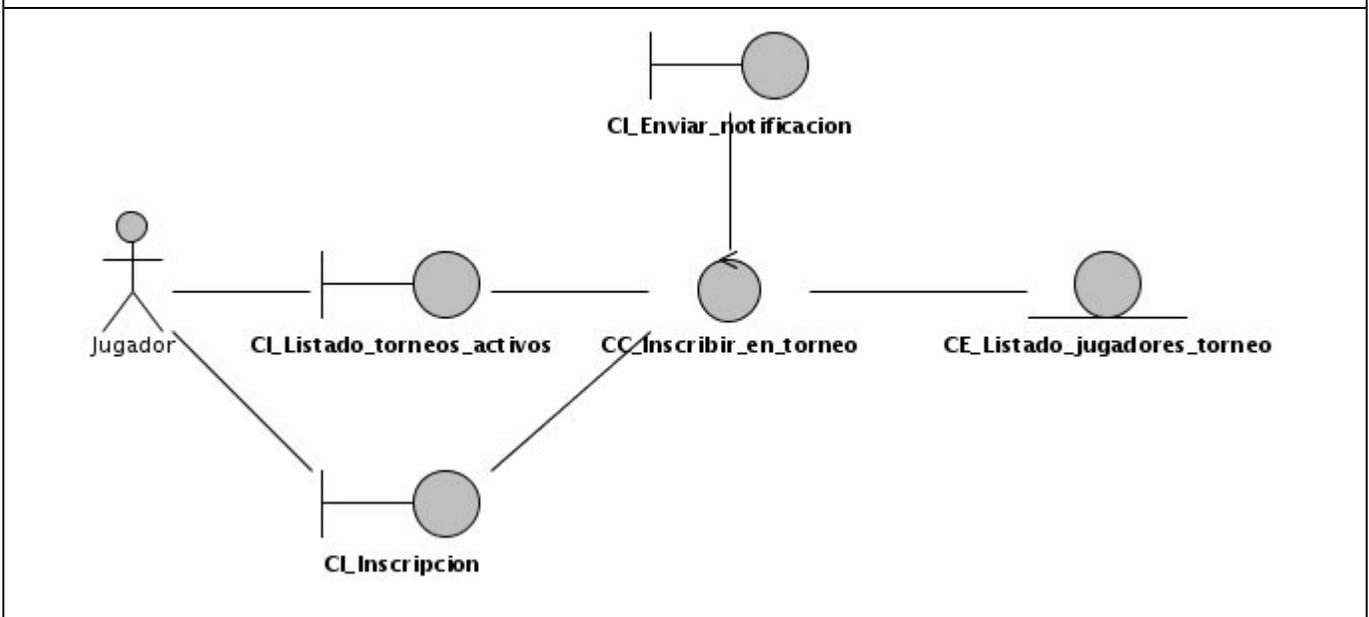
3.2.1.9- Caso de Uso: Enviar notificación

No genera diagrama

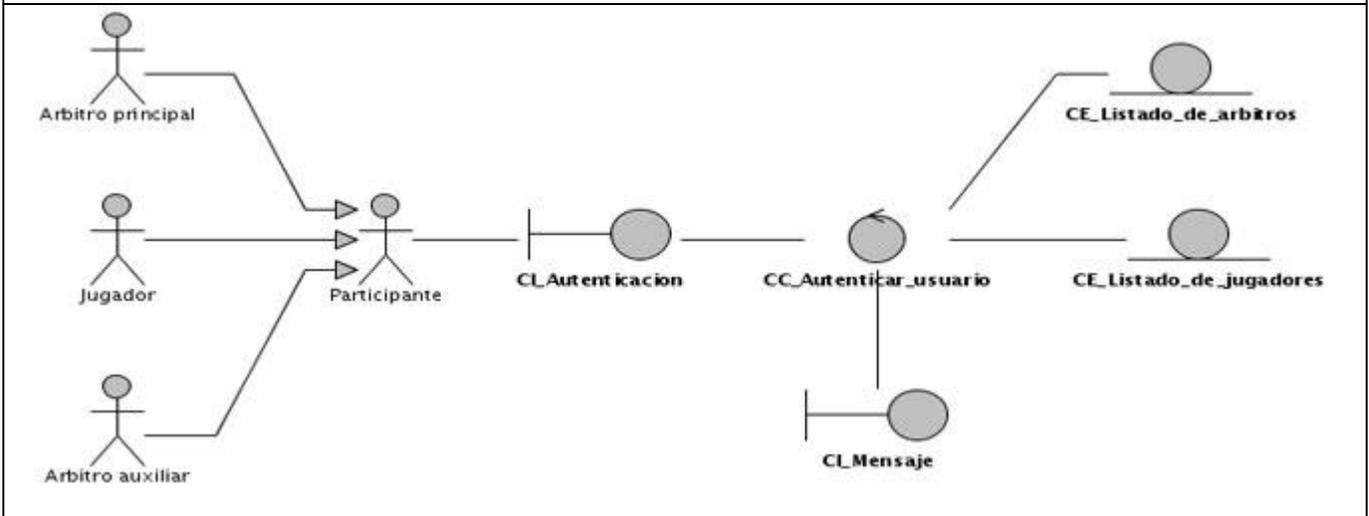
3.2.1.10- Caso de Uso: Jugar partida de ronda



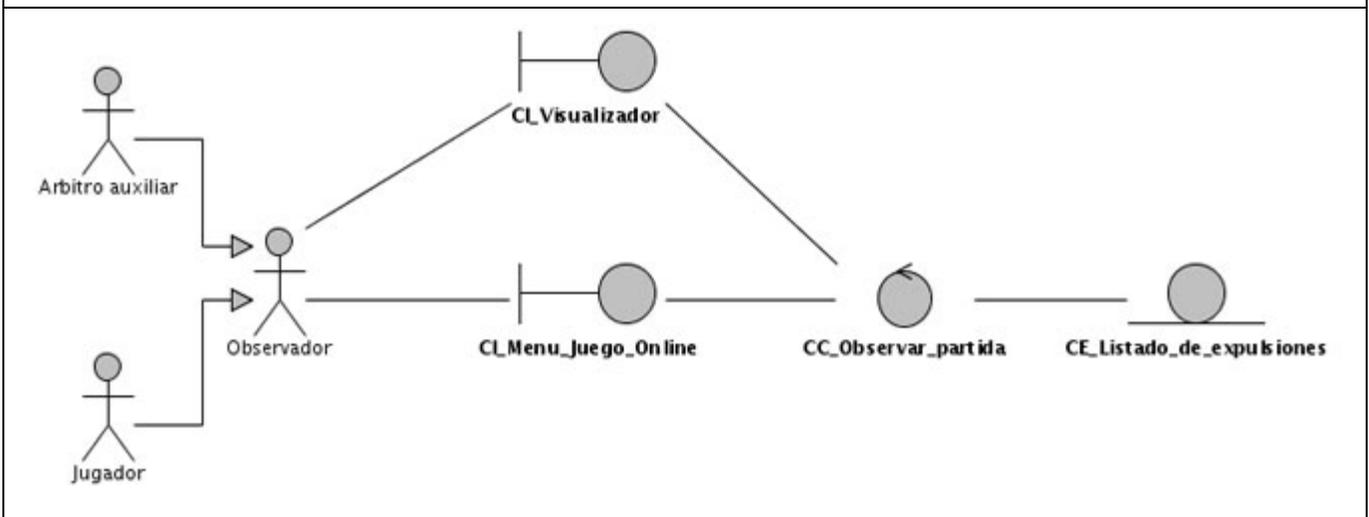
3.2.1.11- Caso de Uso: Inscribirse en torneo



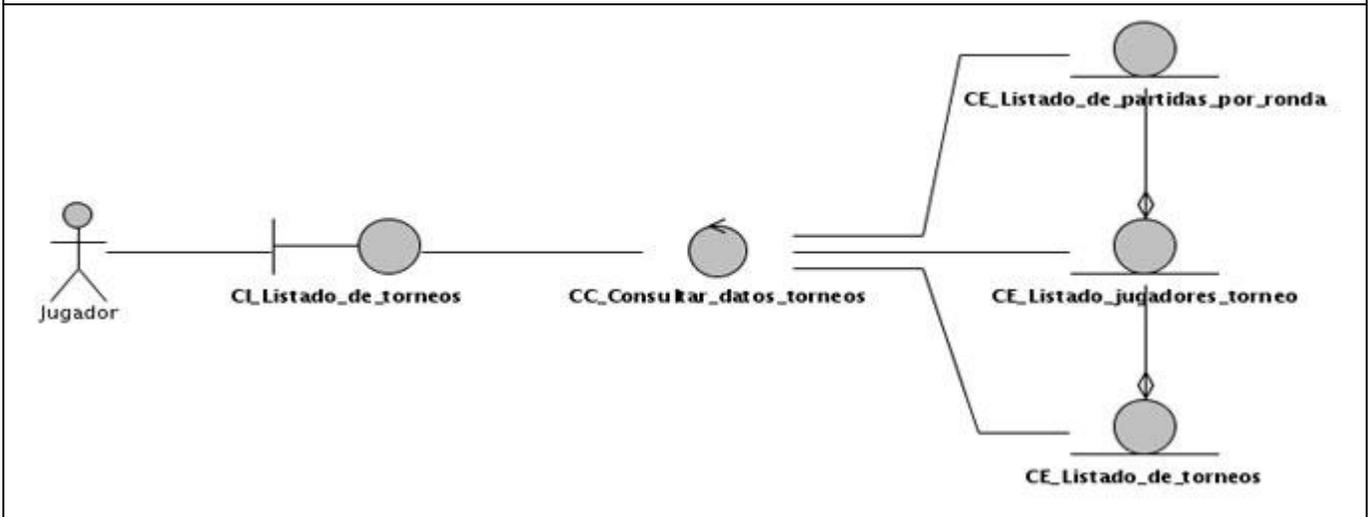
3.2.1.12- Caso de Uso: Autenticarse



3.2.1.13- Caso de Uso: Ver partidas online

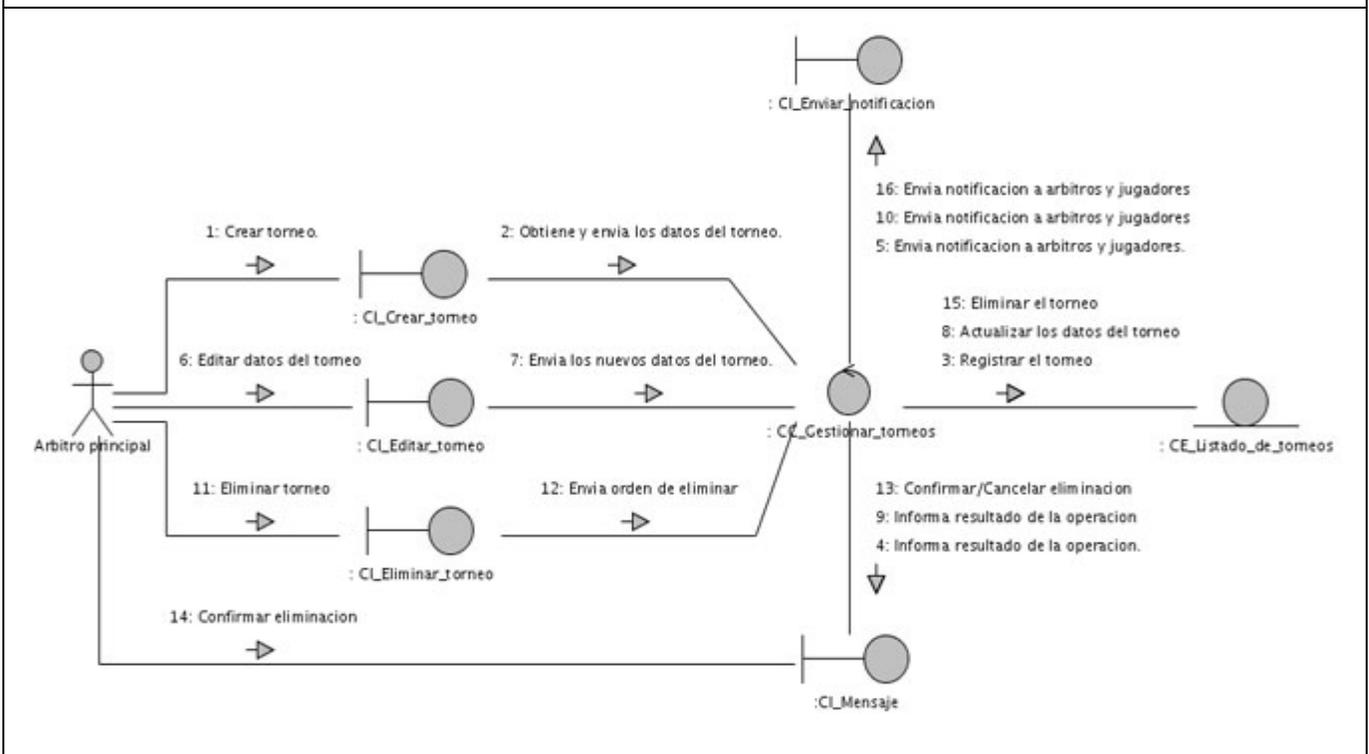


3.2.1.14- Caso de Uso: Consultar torneos anteriores

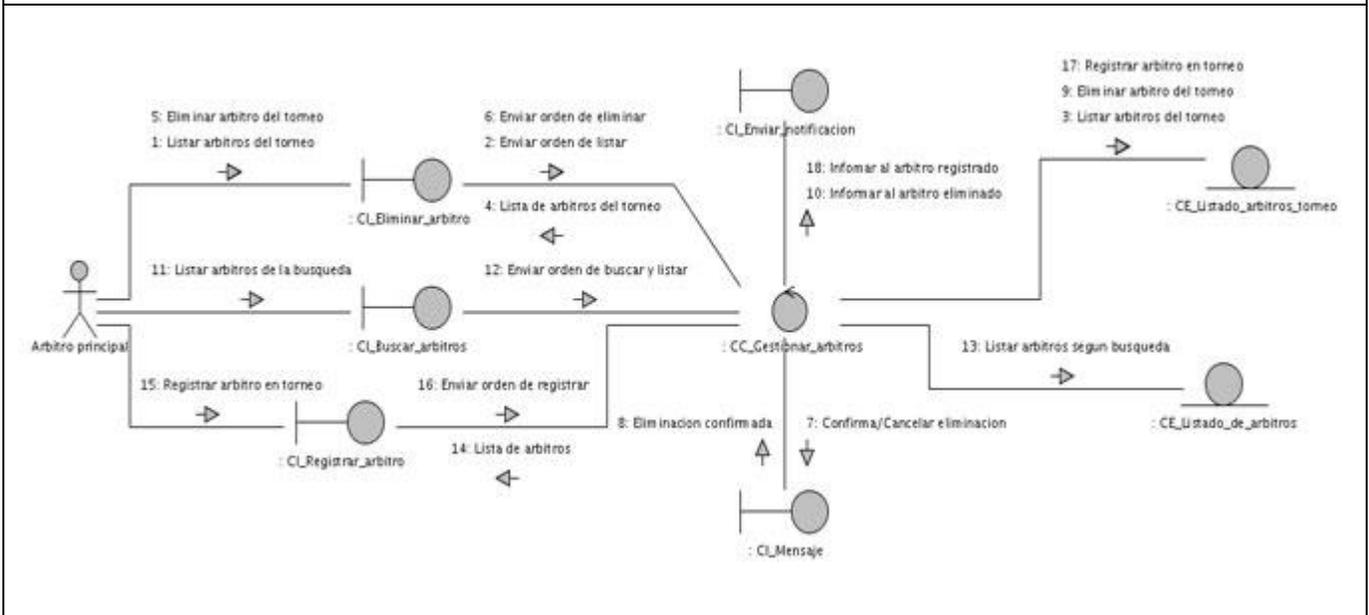


3.2.2- Diagramas de Interacción (Colaboración).

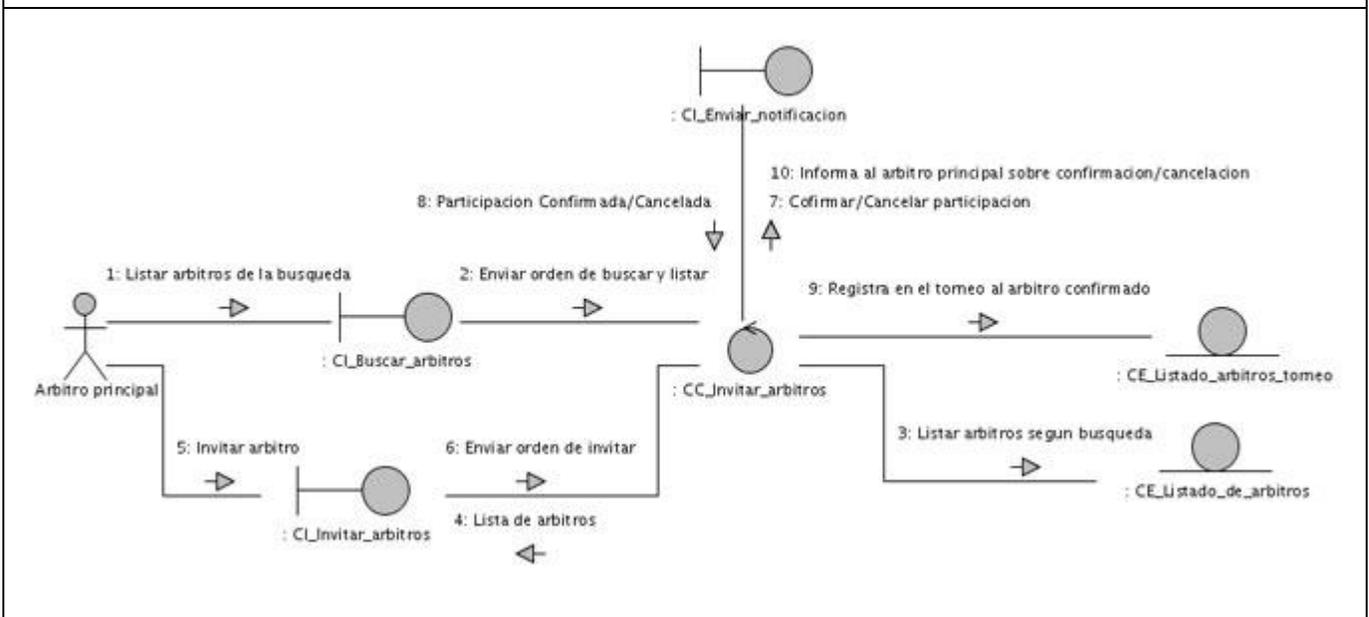
3.2.2.1- Caso de Uso: Gestionar torneos



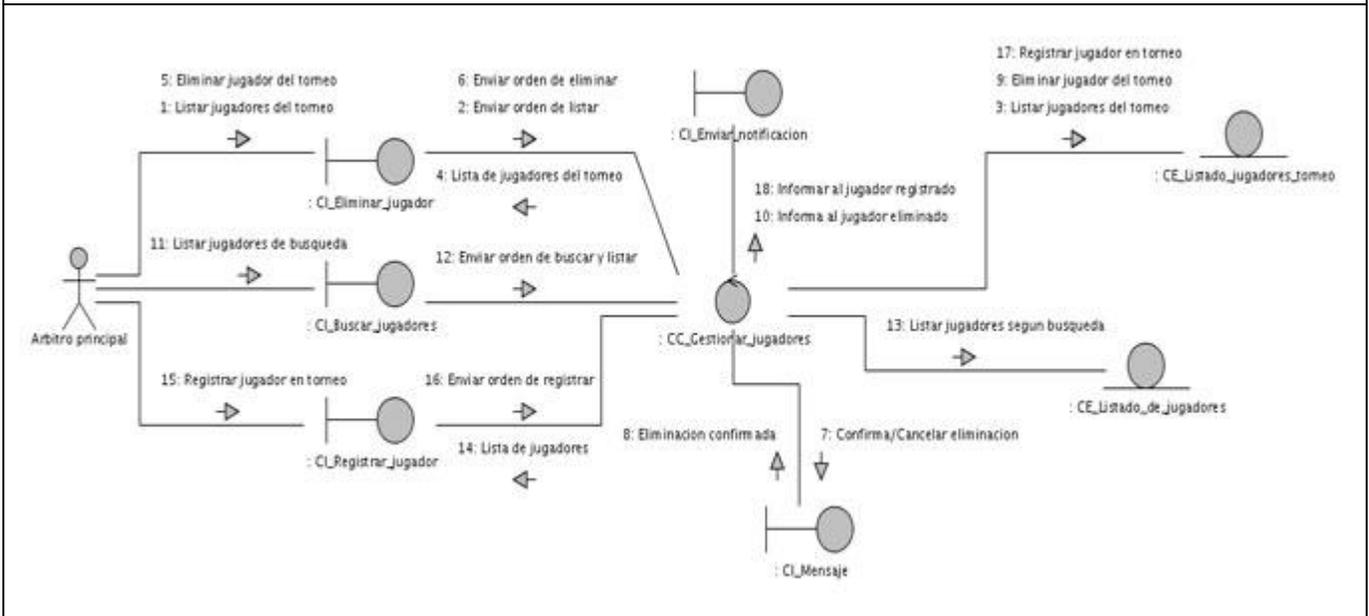
3.2.2.2- Caso de Uso: Gestionar árbitros auxiliares



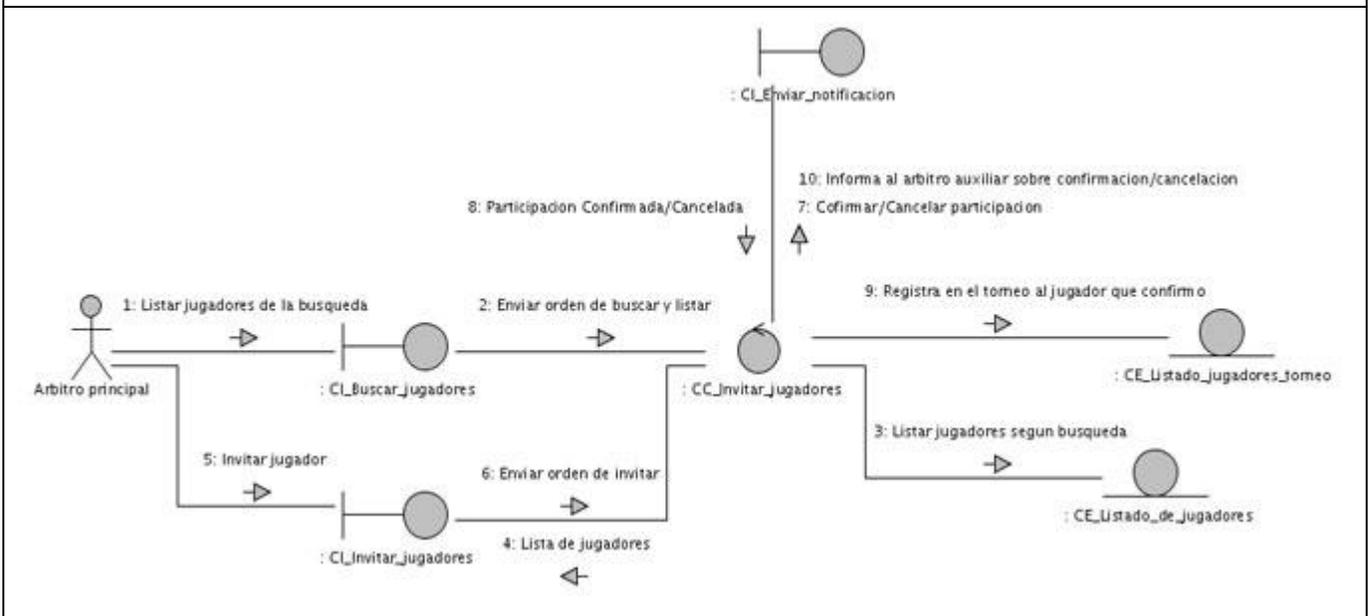
3.2.2.3- Caso de Uso: Invitar árbitros auxiliares



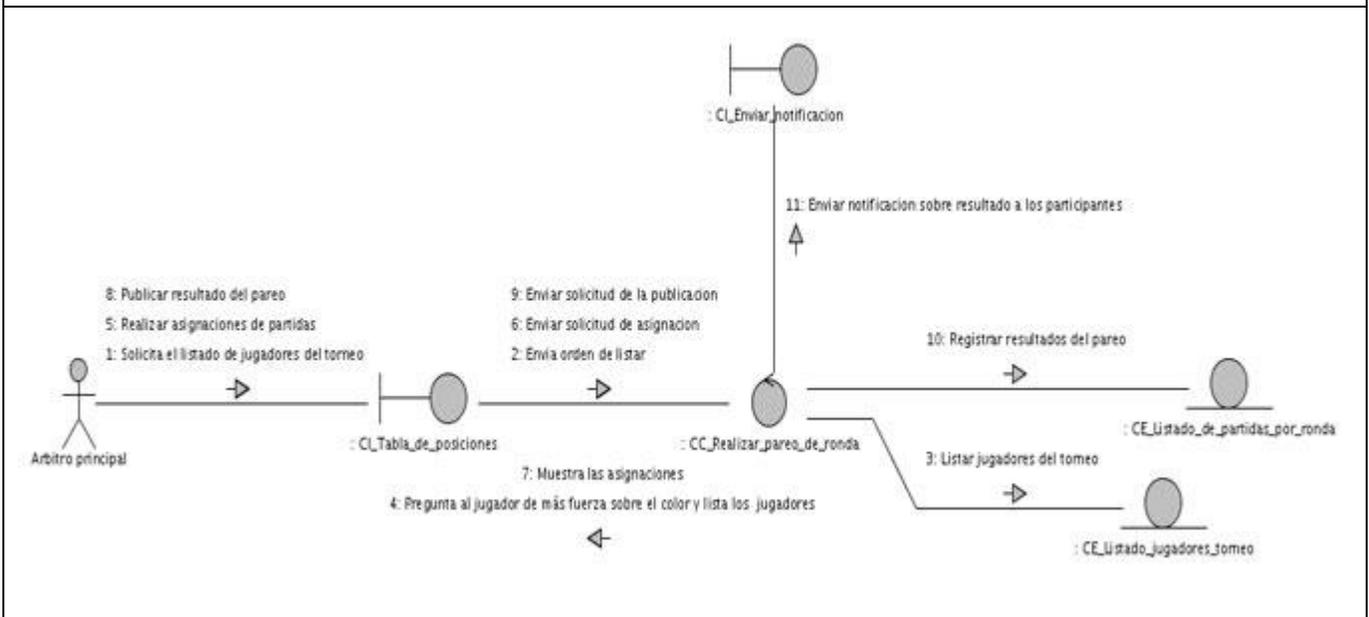
3.2.2.4- Caso de Uso: Gestionar jugadores



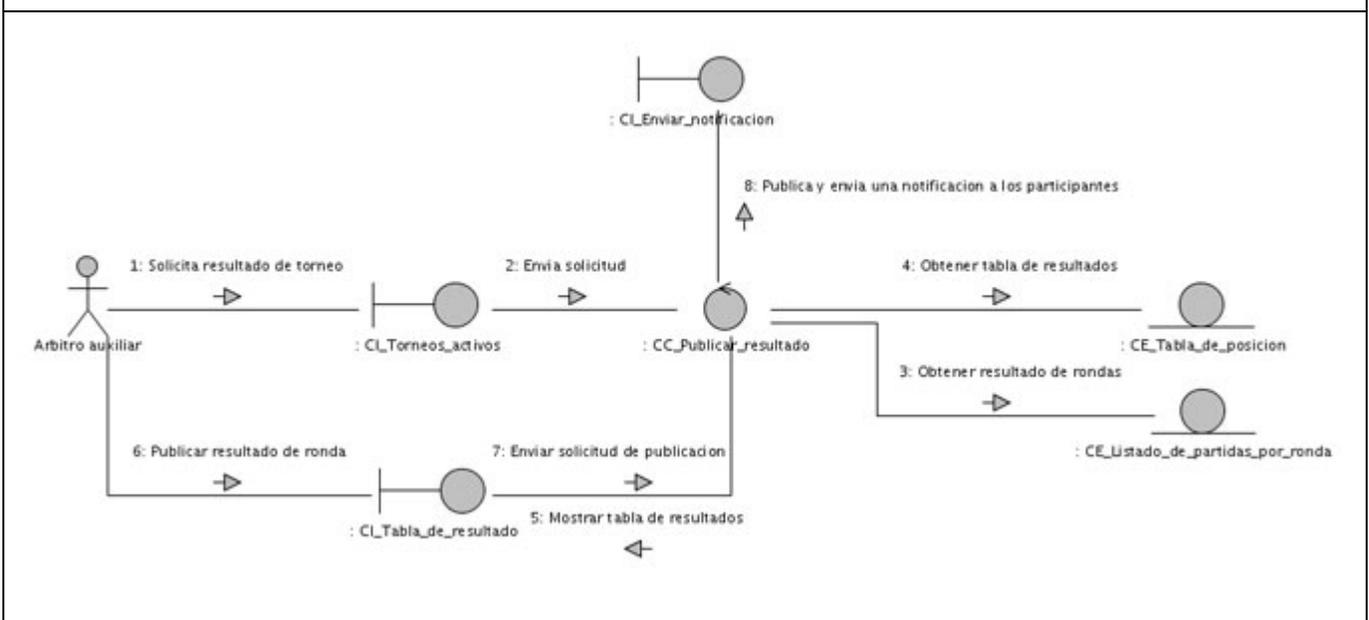
3.2.2.5- Caso de Uso: Invitar jugadores



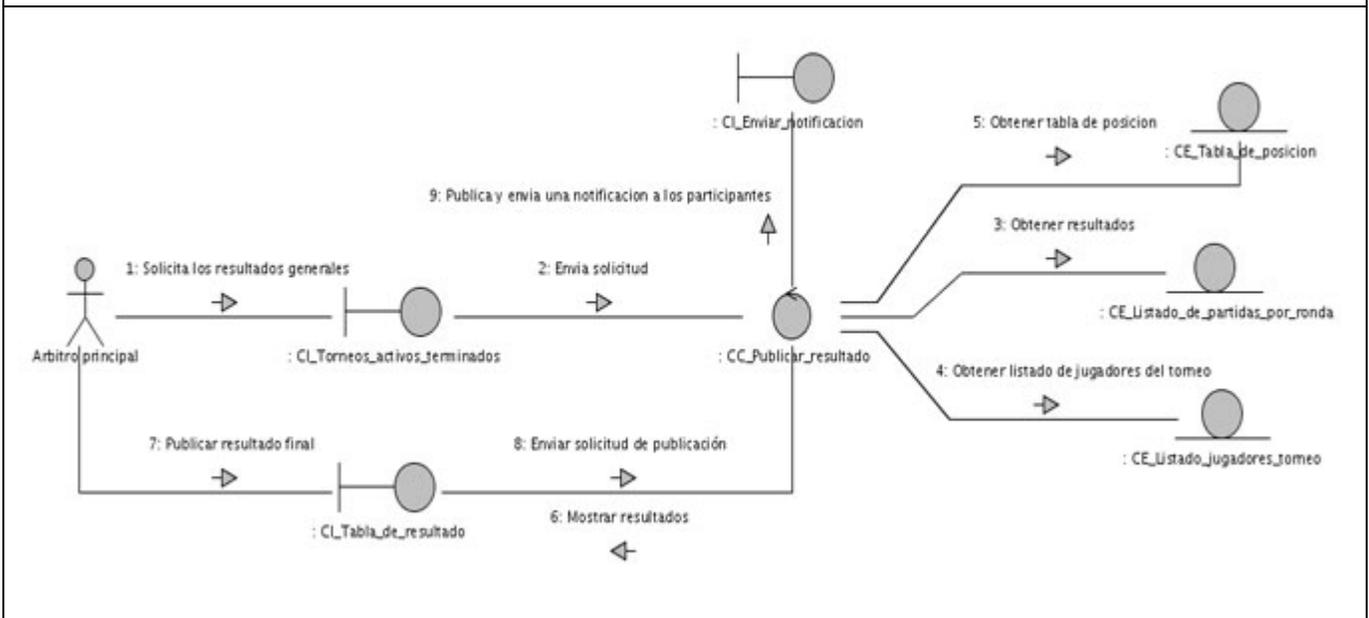
3.2.2.6- Caso de Uso: Realizar pareo de ronda



3.2.2.7- Caso de Uso: Publicar resultado de ronda



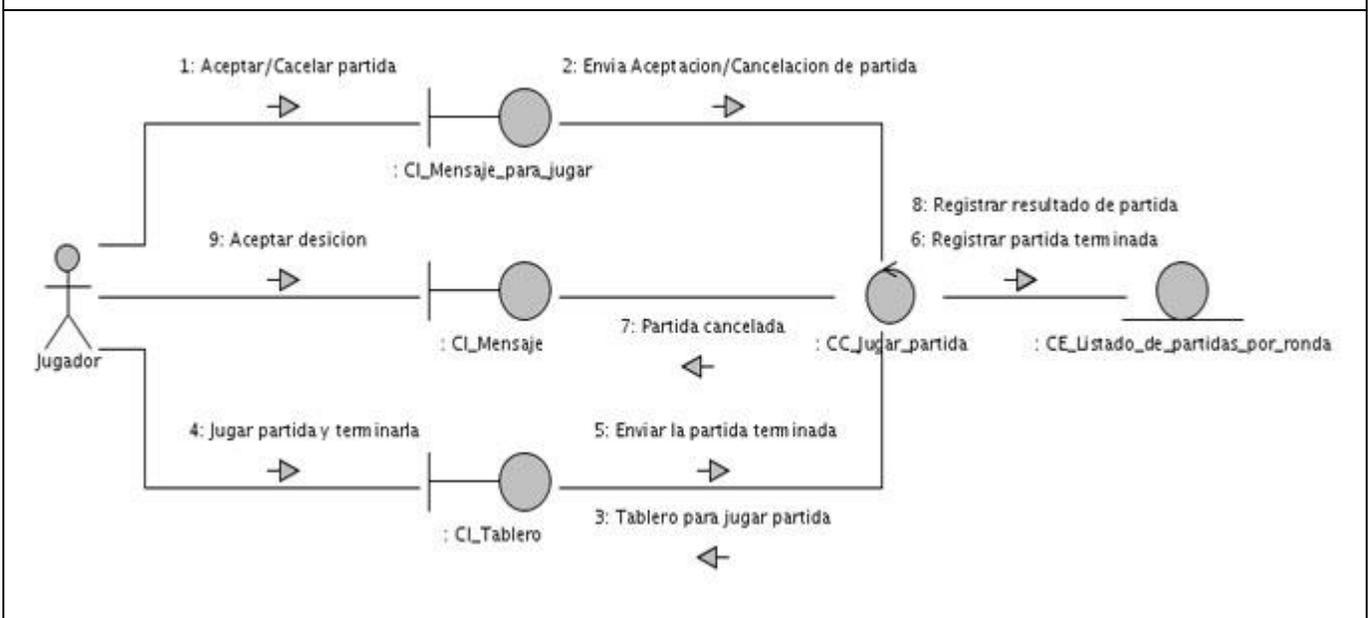
3.2.2.8- Caso de Uso: Publicar resultado final



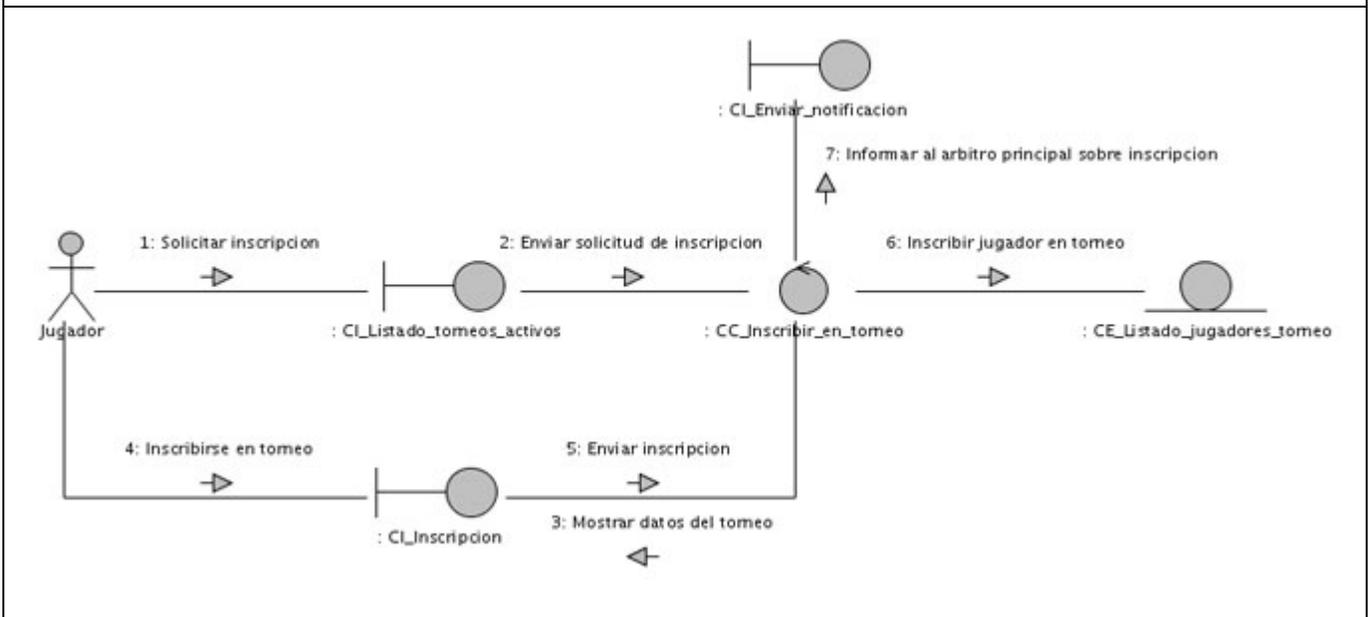
3.2.2.9- Caso de Uso: Enviar notificación

No genera diagrama

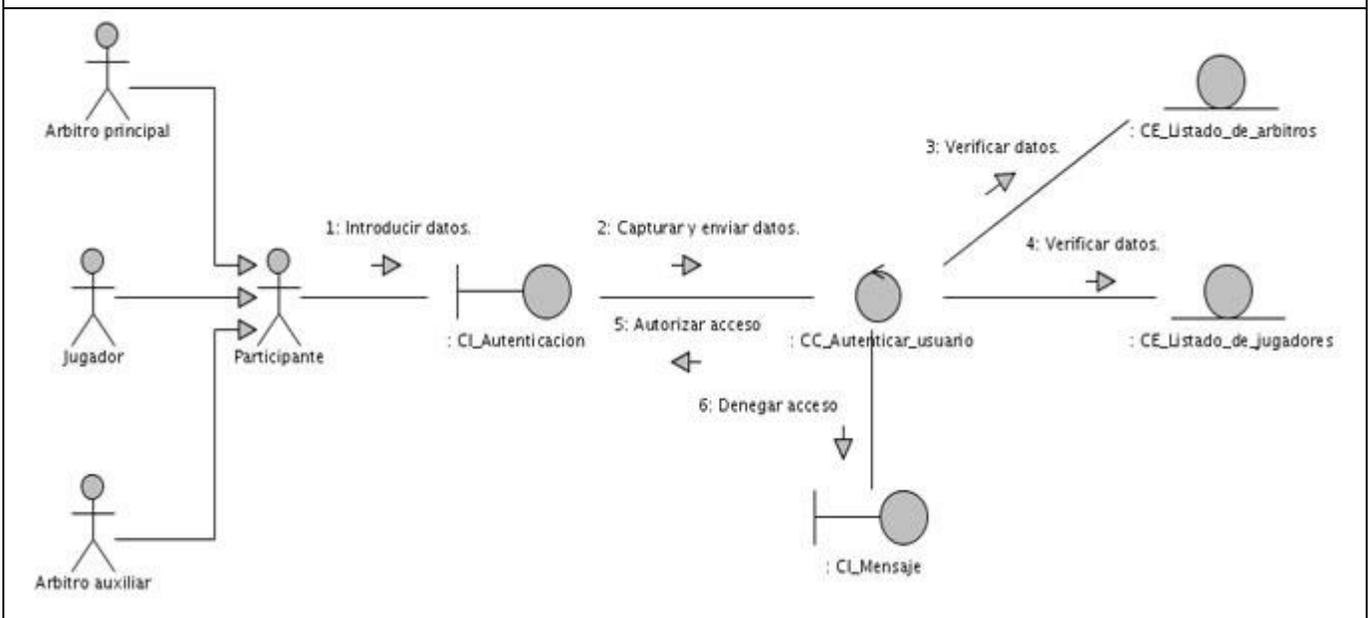
3.2.2.10- Caso de Uso: Jugar partida de ronda



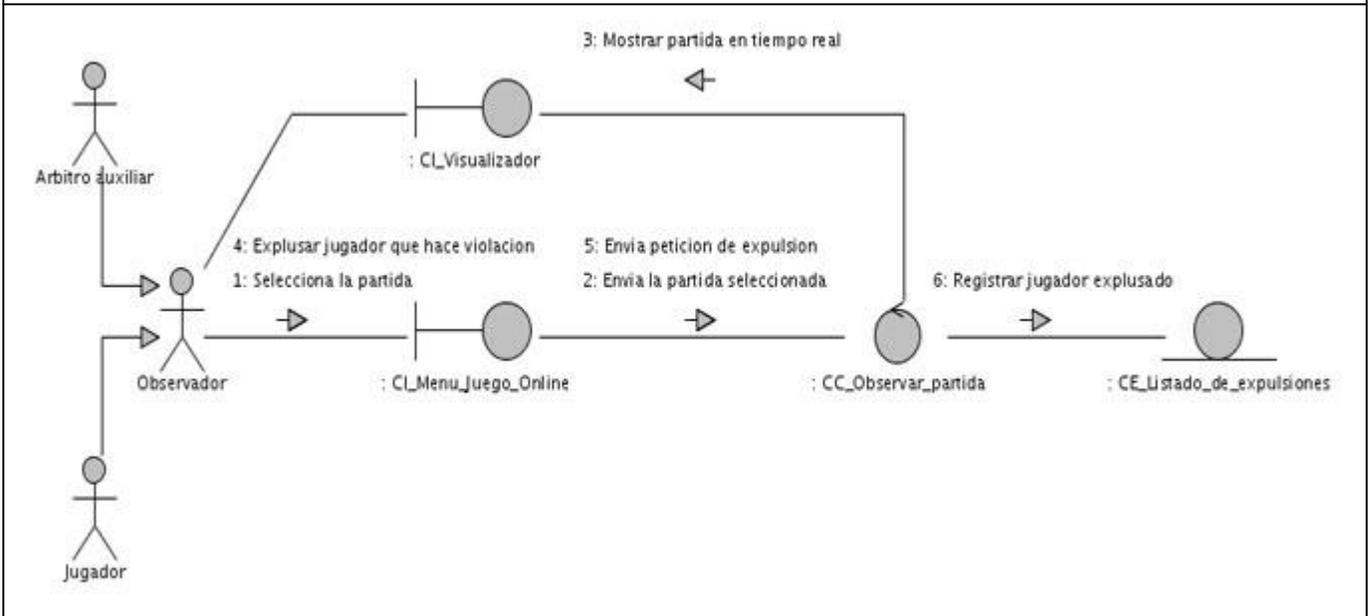
3.2.2.11- Caso de Uso: Inscribirse en torneo



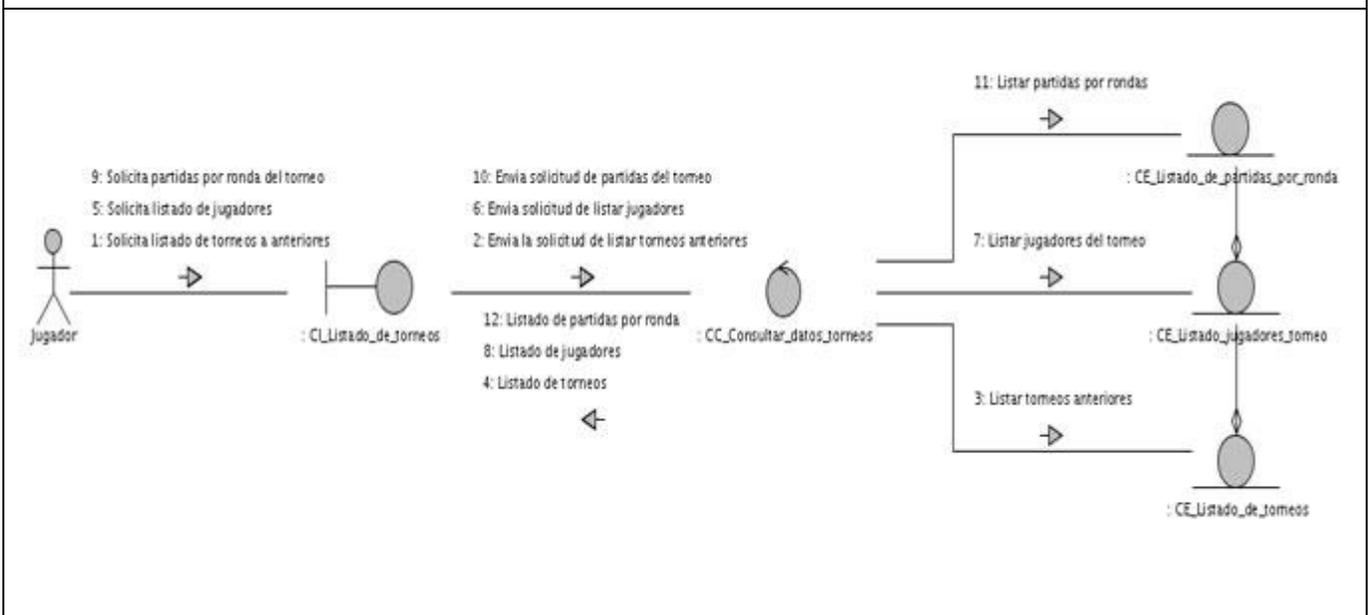
3.2.2.12- Caso de Uso: Autenticarse



3.2.2.13- Caso de Uso: Ver partidas online



3.2.2.14- Caso de Uso: Consultar torneos anteriores



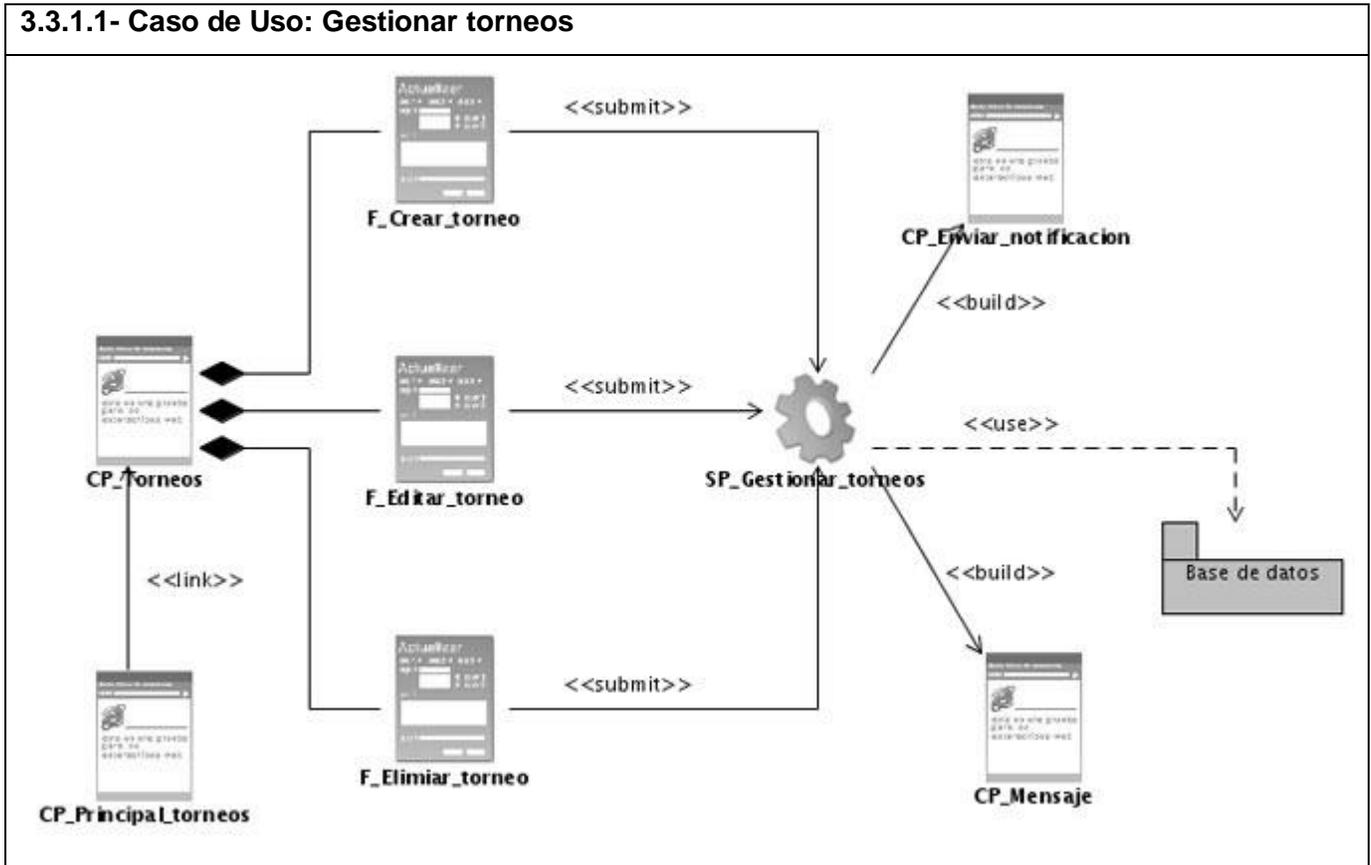
3.3- Diseño.

En el diseño modelamos el sistema y encontramos su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requisitos. Además impone una estructura del sistema que debemos esforzarnos por conservar lo más fielmente posible cuando demos forma al sistema.

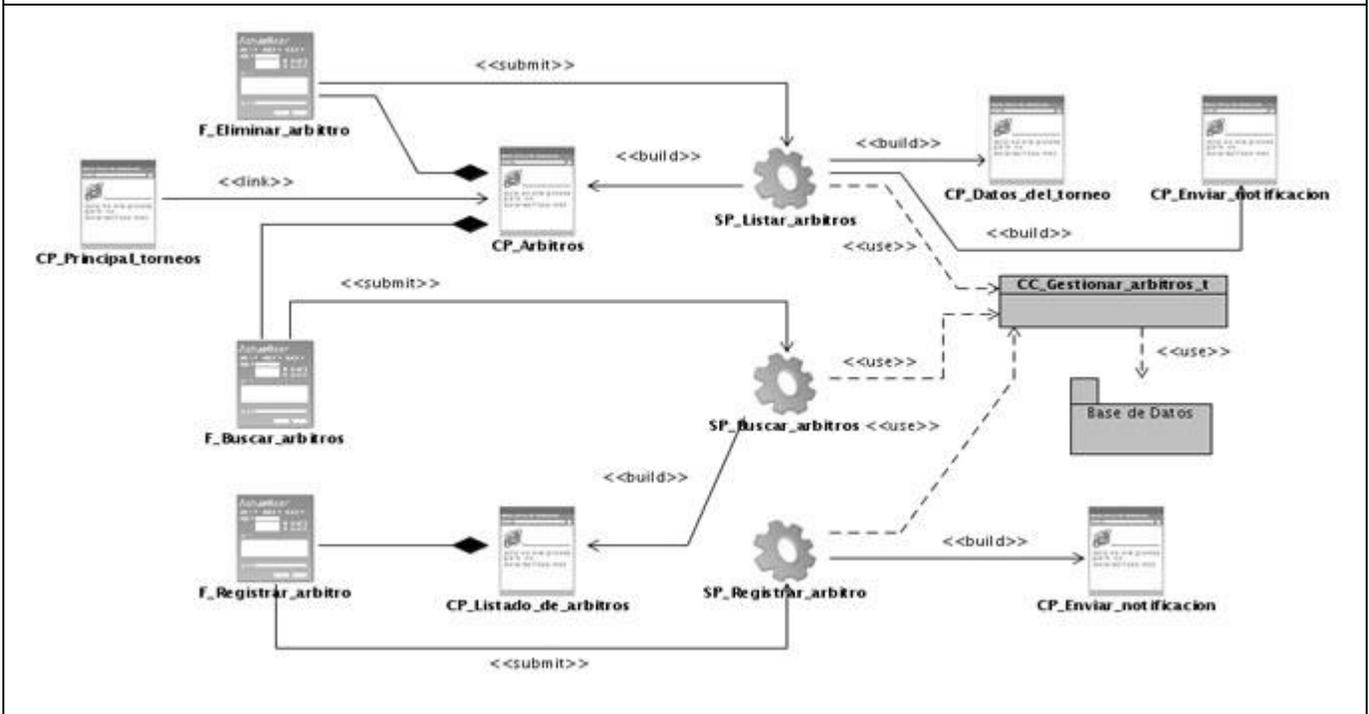
Permite además explotar todas las ventajas de la programación orientada a objeto. En este subepígrafe se presentan principalmente los diagramas y artefactos generados en el diseño, que no son más que el diagrama de clases con estereotipos web, en modelo de datos con sus respectivas descripciones y el diagrama de clases persistentes.

3.3.1- Diagrama de Clases del Diseño.

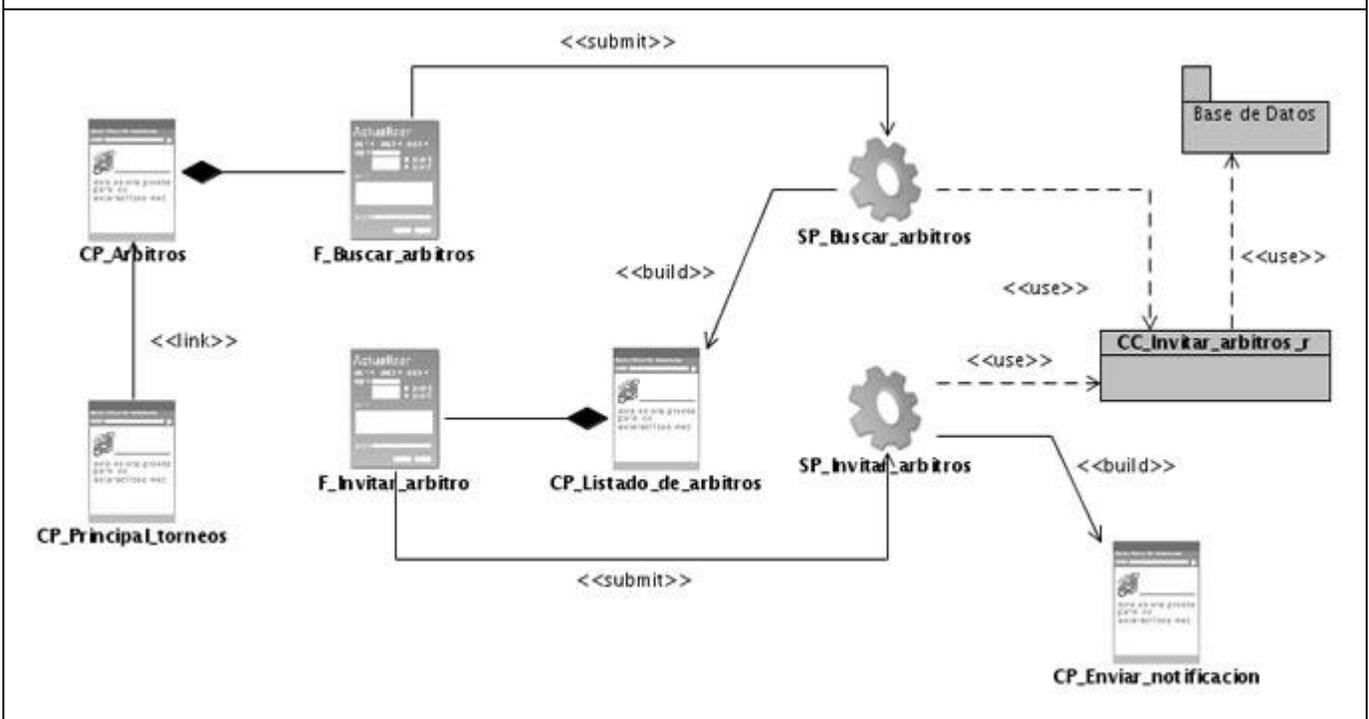
3.3.1.1- Caso de Uso: Gestionar torneos



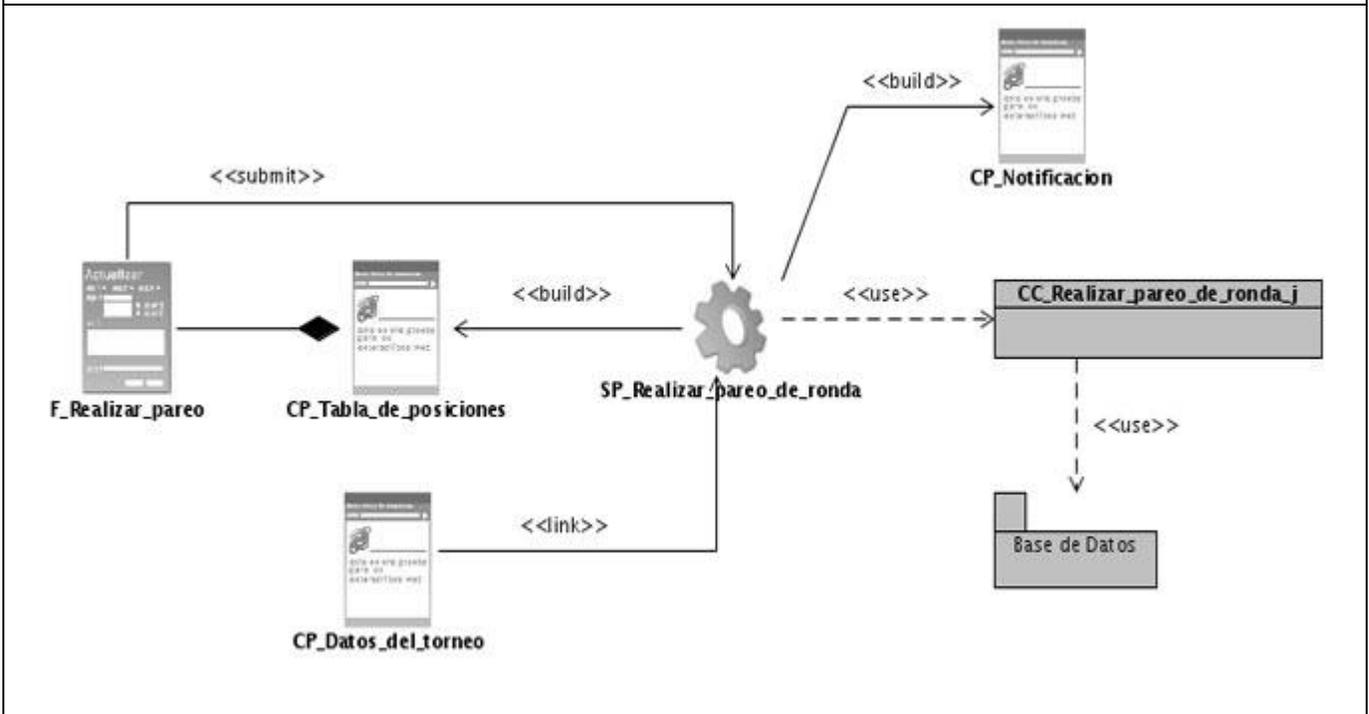
3.3.1.2- Caso de Uso: Gestionar árbitros auxiliares



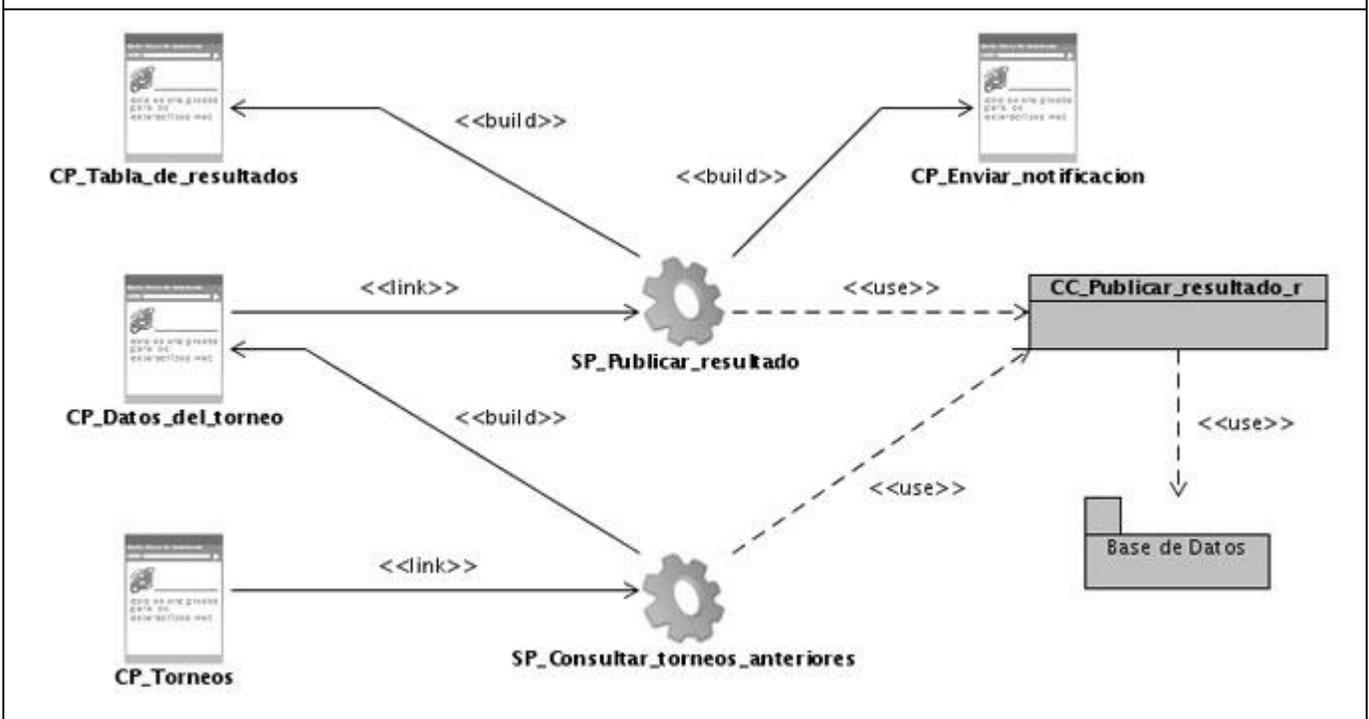
3.3.1.3- Caso de Uso: Invitar árbitros auxiliares



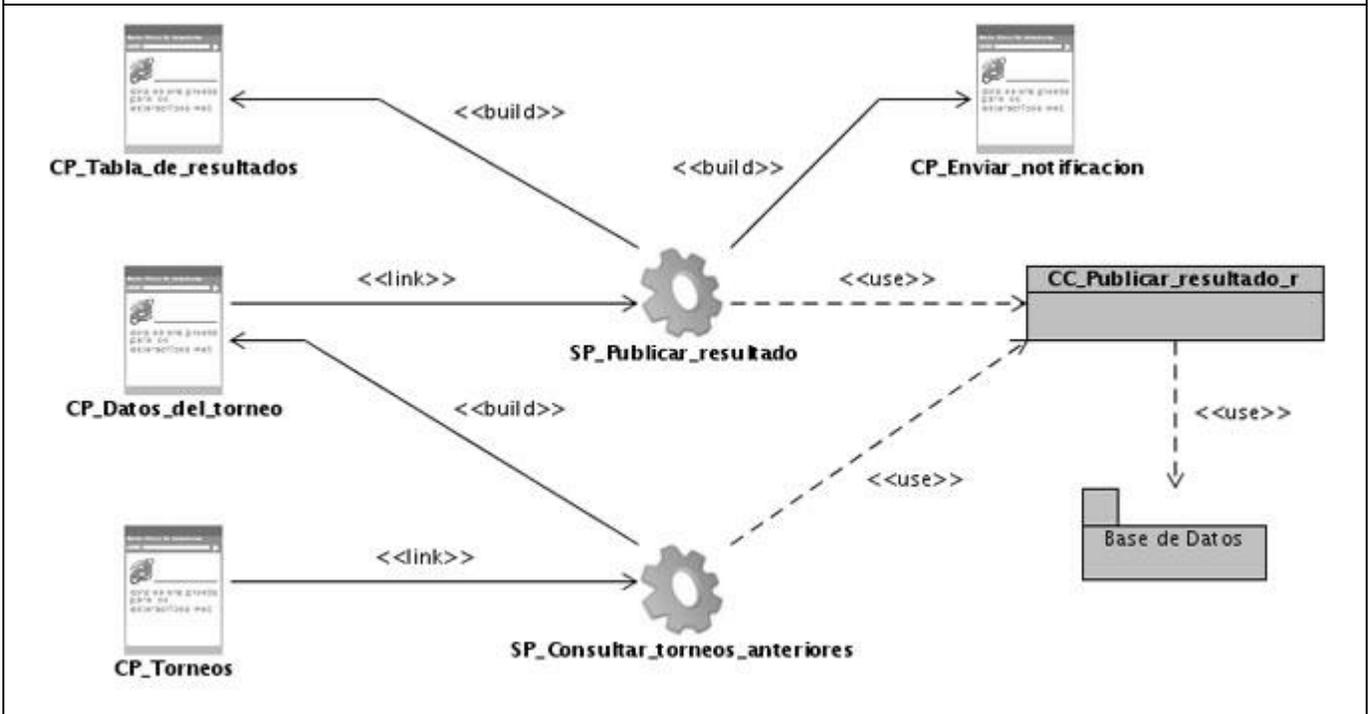
3.3.1.6- Caso de Uso: Realizar pareo de ronda



3.3.1.7- Caso de Uso: Publicar resultado de ronda



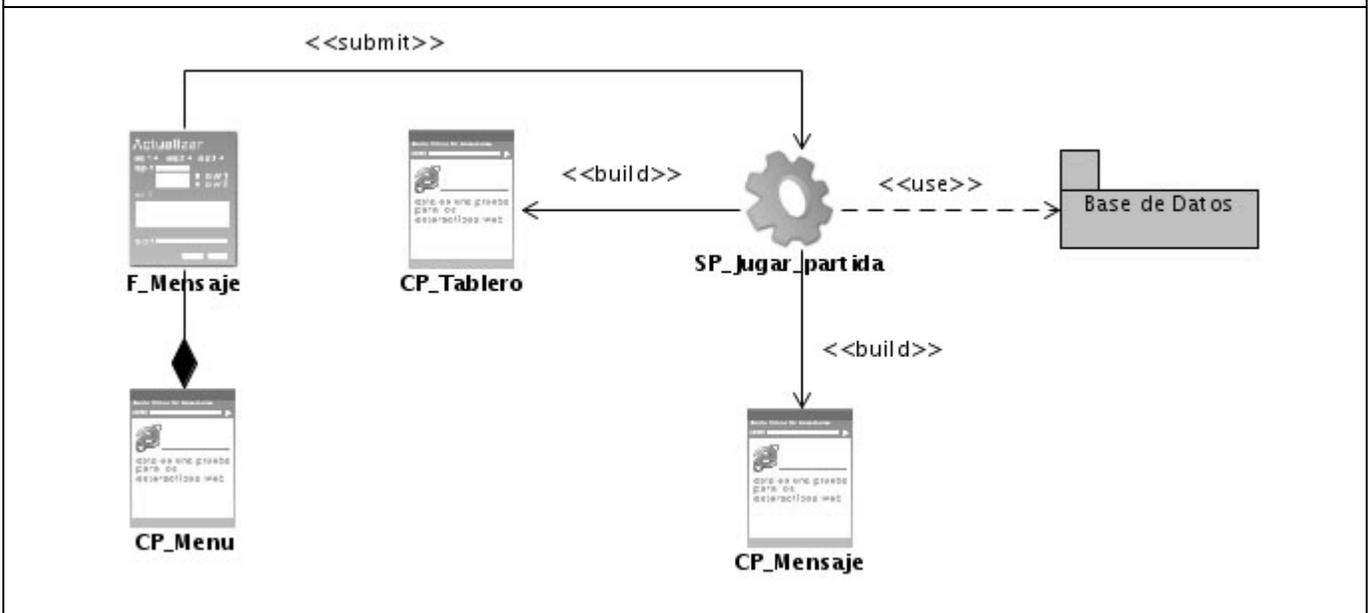
3.3.1.8- Caso de Uso: Publicar resultado final



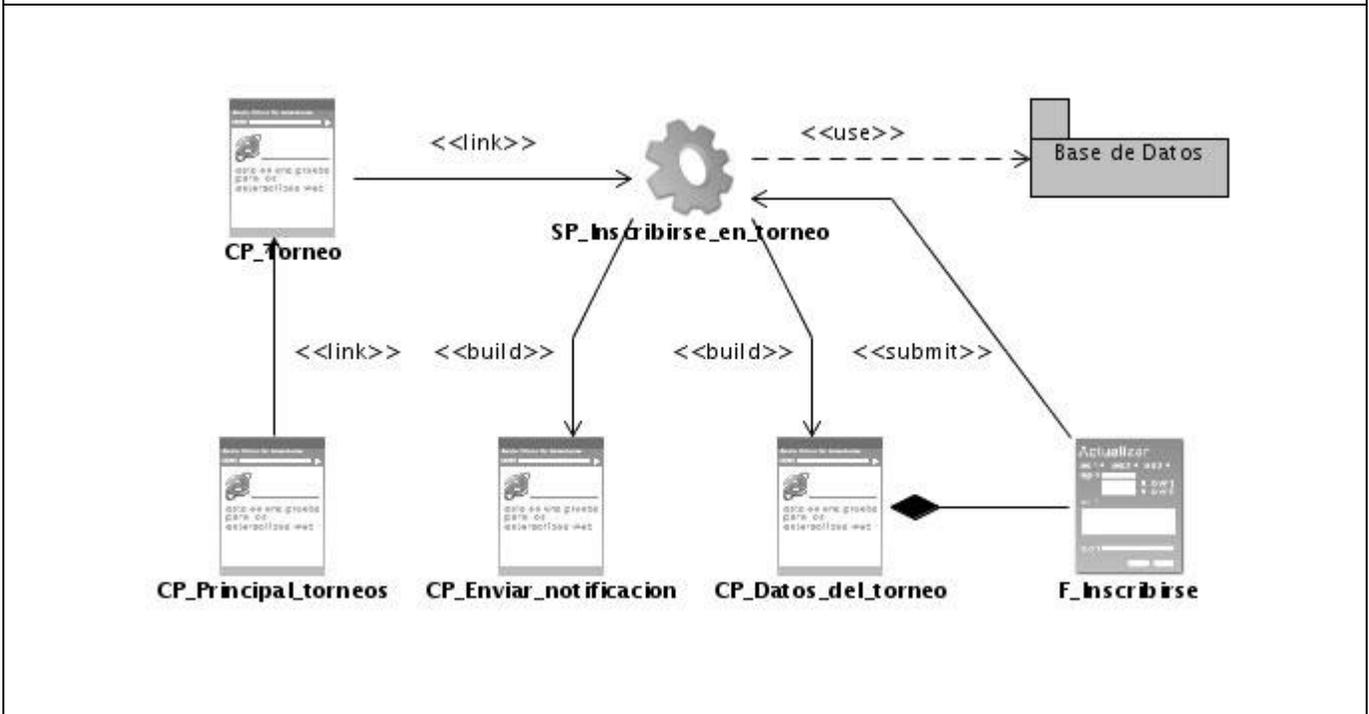
3.3.1.9- Caso de Uso: Enviar notificación

No genera diagrama

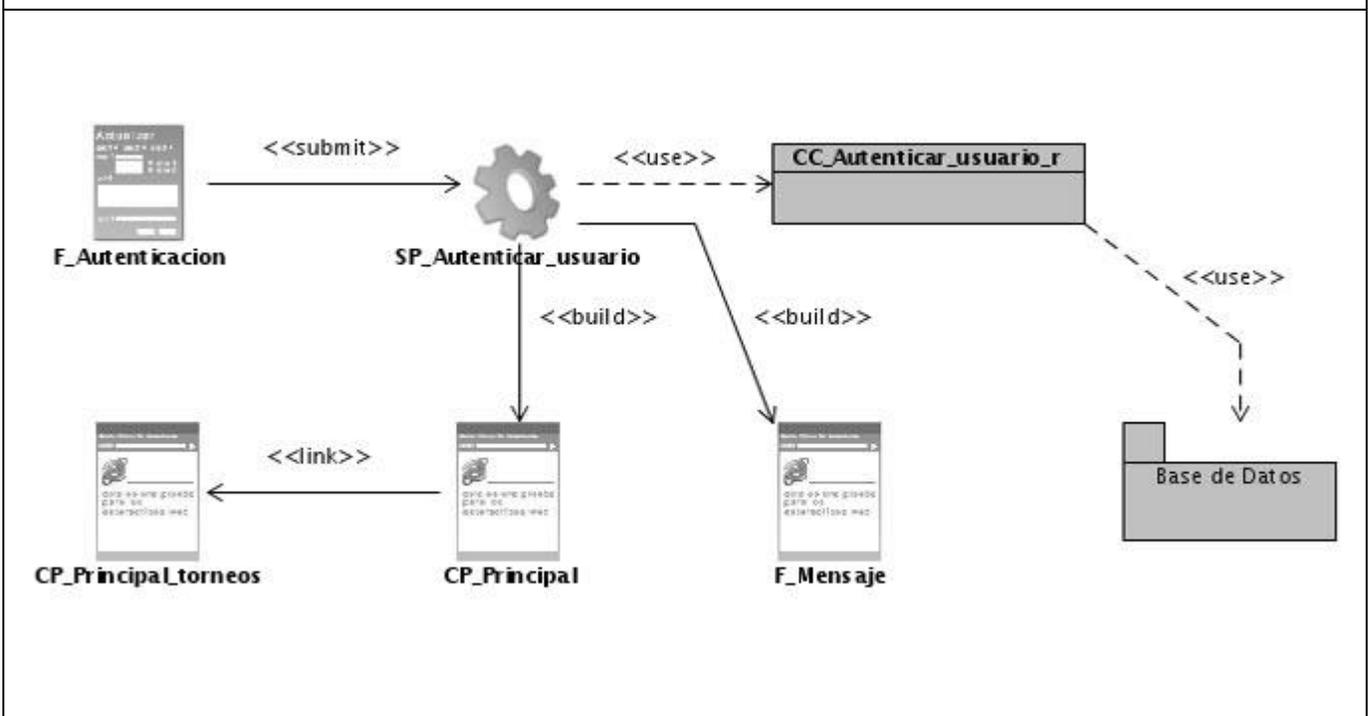
3.3.1.10- Caso de Uso: Jugar partida de ronda



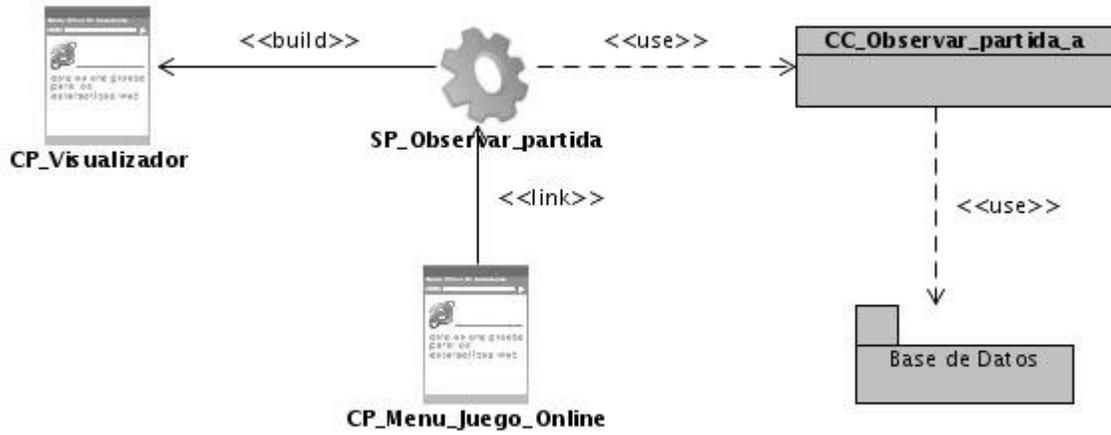
3.3.1.11- Caso de Uso: Inscribirse en torneo



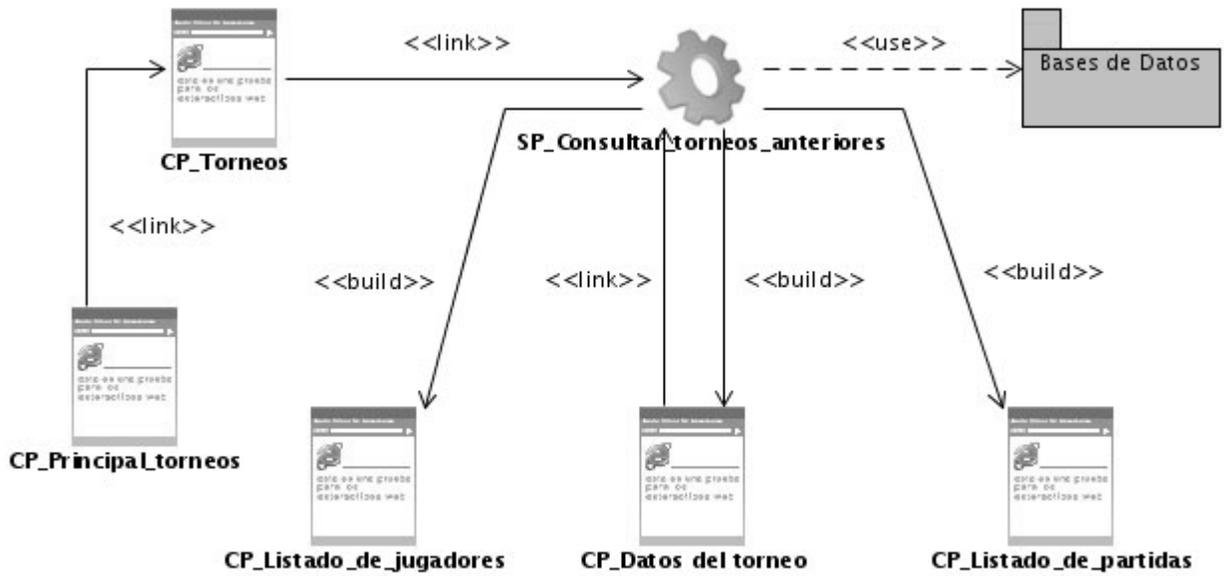
3.3.1.12- Caso de Uso: Autenticarse



3.3.1.13- Caso de Uso: Ver partidas online



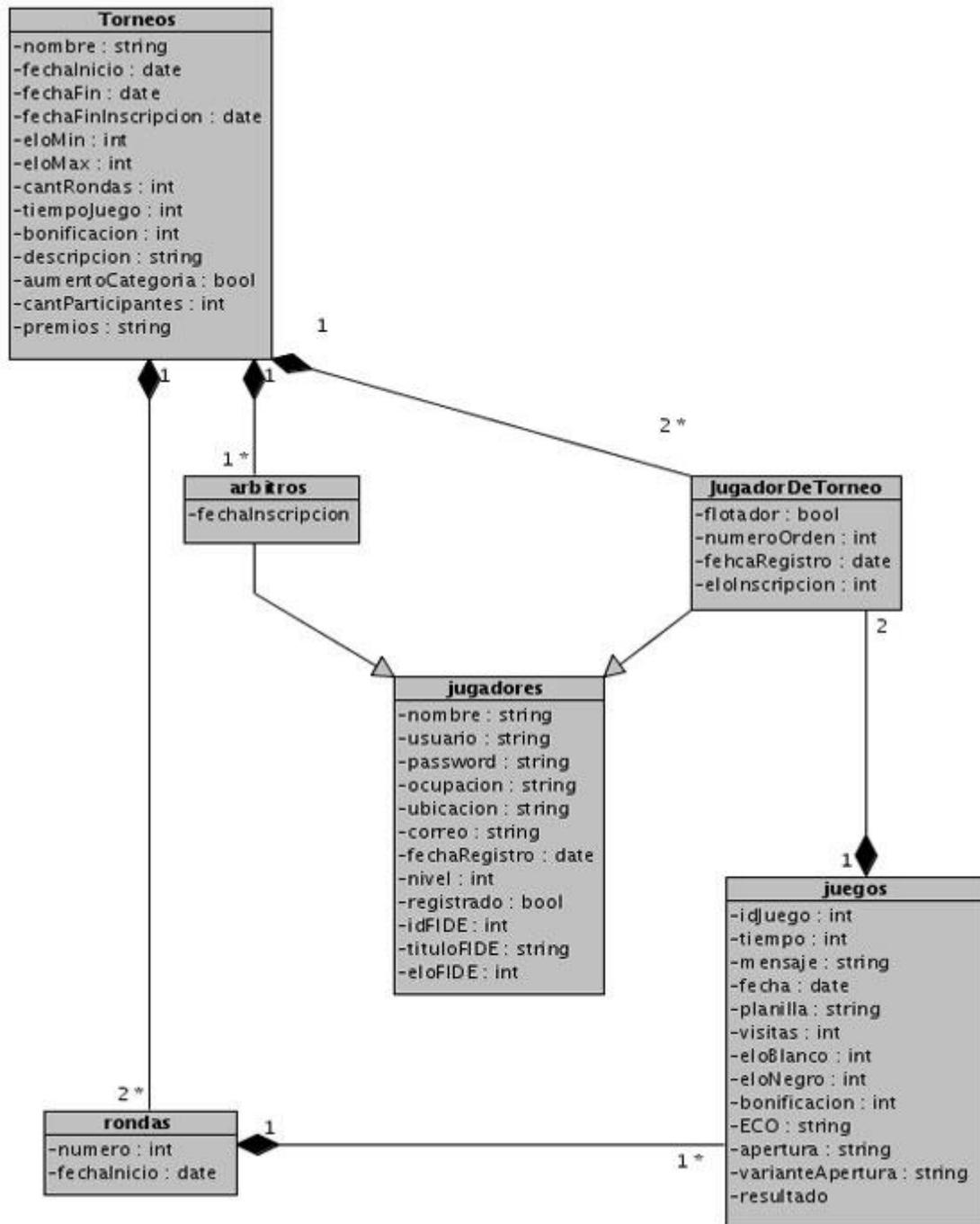
3.3.1.14- Caso de Uso: Consultar torneos anteriores



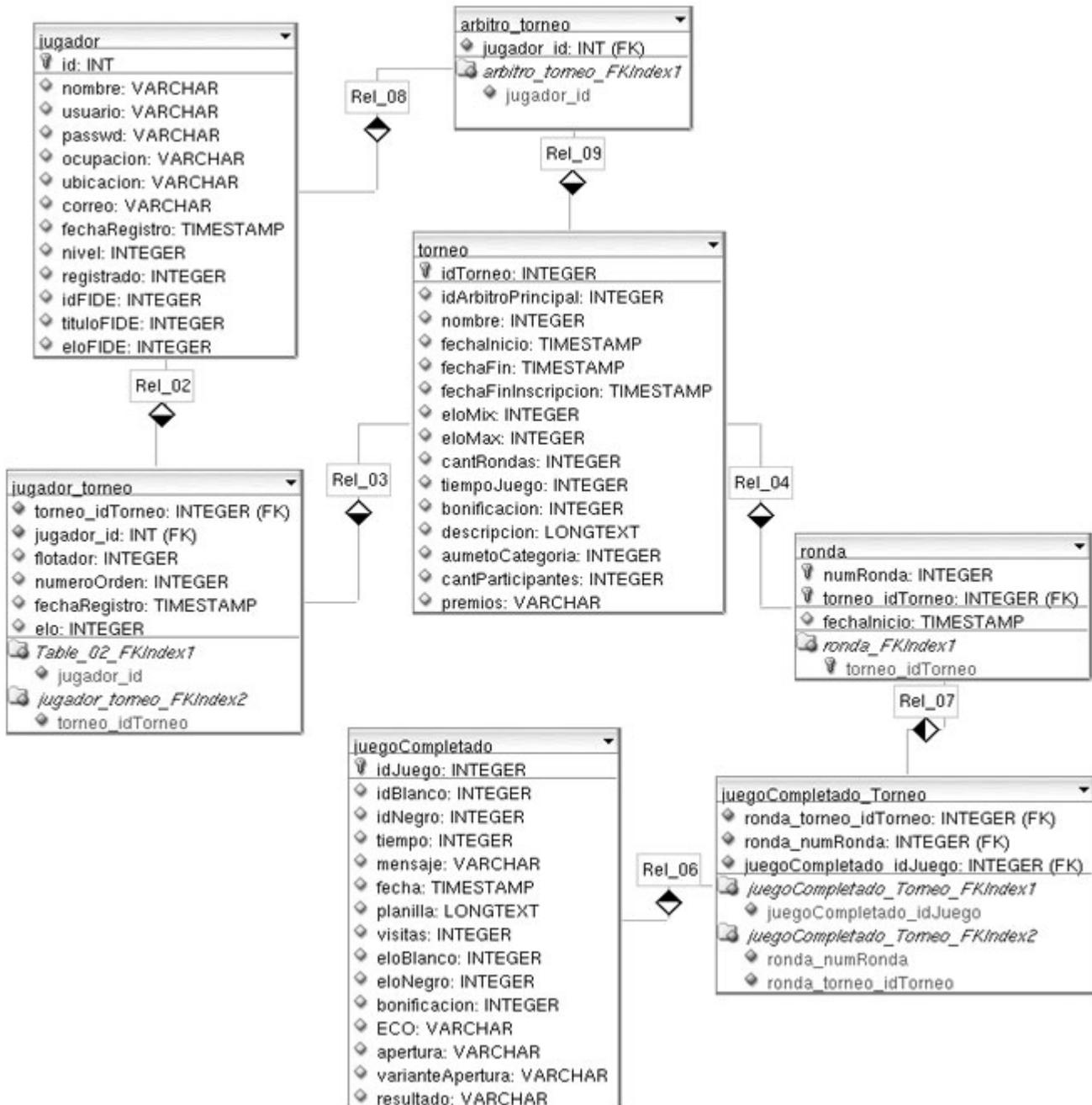
3.4- Diseño de la Base de Datos.

Aquí se presente al modelos de datos reflejado a través de un diagrama de clases persistentes. Cada una de estas clases representa todo los datos que se almacenan en todo el proceso de desarrollo del producto y las relaciones existentes entre ellas. A partir de este diagrama se presenta el modelo de datos y las descripciones de cada una de las tablas de Base de Datos.

3.4.1- Diagrama de clases persistentes.



3.4.2- Modelo de la Base de Datos.



3.4.3- Descripción de las tablas de la Base de Datos.

Nombre: jugador		
Descripción: Contiene los datos de los jugadores registrados en el juego online.		
Atributo	Tipo	Descripción
id	int	Identificador de jugador.
nombre	varchar	Nombre completo del jugador
usuario	varchar	Usuario del dominio
passwd	varchar	Contraseña de acceso
ocupacion	varchar	Si estudiante, profesor o trabajador.
ubicacion	varchar	Área a la que pertenece o facultad.
correo	varchar	Correo electrónico UCI.
fechaRegistro	timestamp	Fecha en que se registró en el juego online.
nivel	integer	Rol del usuario.
registrado	integer	Tipo de cuanta de usuario, si es LDAP, Directorio activo o registro por la web.
idFIDE	integer	Identificados que otorga la FIDE al jugador que ha alcanzado un nivel dentro de ella.
tituloFIDE	integer	Titulo FIDE en caso de ser jugador con título.
eloFIDE	integer	Elo oficial que le da la FIDE.

Nombre: jugador_torneo		
Descripción: Contiene los datos de los jugadores del torneo.		
Atributo	Tipo	Descripción
torneo_idTorneo	integer	Lava de la tabla jugador.
jugador_id	int	Identificador de jugador.
flotador	integer	Jugador flotador del la ronda.
numeroOrden	integer	Numero inicial del jugador en el pareo.
fechaRegistro	timestamp	Fecha en que registró en el torneo.
elo	integer	Elo del jugador.

Nombre: torneo		
Descripción: Contiene los datos de los torneos online que se realizan.		
Atributo	Tipo	Descripción
idTorneo	integer	Identificador del torneo.
idArbitroPrincipal	integer	Identificador del árbitro.
nombre	integer	Nombre completo de árbitro.
fechaInicio	timestamp	Fecha de inicio del torneo.
fechaFin	timestamp	Fecha de finalización del torneo.
fechaFinInscripcion	timestamp	Fecha de finalización de la inscripción.
eloMix	integer	Elo mínimo de los jugadores del torneo.
eloMax	integer	Elo máximo de los jugadores del torneo.
cantRondas	integer	Cantidad de rondas.
tiempoJuego	integer	Tiempo a que se jugarán las partidas.
bonificacion	integer	Bonificación de tiempo.
descripcion	longtext	Datos generales del torneo.
aumentoCategoria	integer	Aumento de categoría.
cantParticipantes	integer	Cantidad de participantes del torneo.
premios	varchar	Premios del torneo.

Nombre: arbitro_torneo		
Descripción: Contiene los datos de los torneos online que se realizan.		
Atributo	Tipo	Descripción
jugador_id	int	Identificador del árbitro.

Nombre: ronda		
Descripción: Datos las rondas del torneo.		
Atributo	Tipo	Descripción
numRonda	integer	Número de ronda del torneo.
torneo_idTorneo	integer	Identificador de la ronda.

Nombre: juegoCompletado		
Descripción: Contiene los datos de los juegos completados o terminados.		
Atributo	Tipo	Descripción
idJuego	integer	Identificador del juego.
idBlanco	integer	Identificador del jugador con piezas blancas.
idNegro	integer	Identificador del jugador con piezas negras.
tiempo	integer	Tiempo de la partida.
mensaje	varchar	Si ganan las blancas o las negras.
fecha	timestamp	Fecha en que se realizó la partida.
planilla	longtext	Planilla donde se muestra el PGN.
visitas	integer	Cantidad de personas que observaron la partida una vez registrada.
eloBlanco	integer	Elo del jugador con piezas blancas.
eloNegro	integer	Elo del jugador con piezas negras.
bonificacion	integer	Bonificación de la partida.
ECO	varchar	Código ECO con se clasificó.
apertura	varchar	Apertura realizada en la partida.
varianteApertura	varchar	Variante de la apertura.
resultado	varchar	Resultado de la partida.

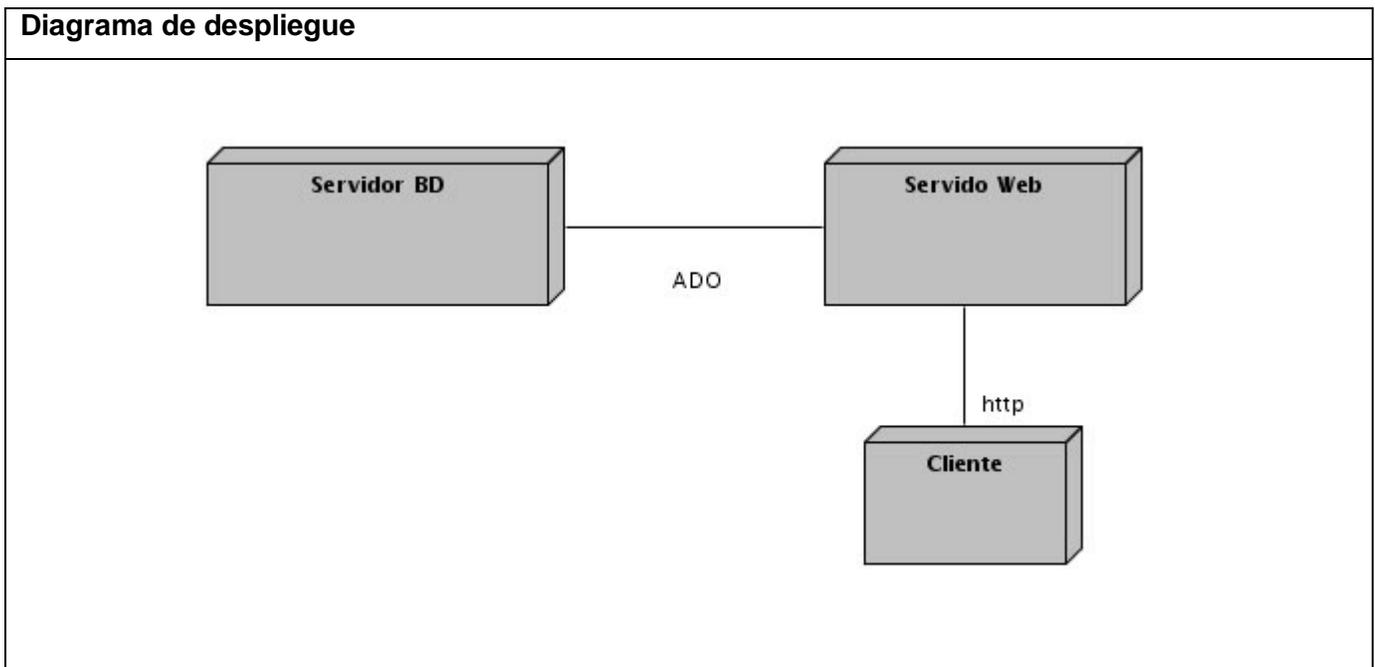
Nombre: juegoCompletado_Torneo		
Descripción: Datos las rondas del torneo.		
Atributo	Tipo	Descripción
ronda_torneo_idTorneo	integer	Identificador.
ronda_numRonda	integer	Número de ronda a la que pertenece.
juegoCompletado_idJuego	integer	Identificador del juego del torneo completado.

Capítulo 4: Implementación

En este epígrafe se muestran los diagramas de despliegue y de componentes para un mejor entendimiento de la fase de implementación. En el primero se muestra como está el sistema ya desplegado sobre sistemas automatizados y en el segundo se muestran en sentido general cada uno de los paquetes implementados dentro de la aplicación.

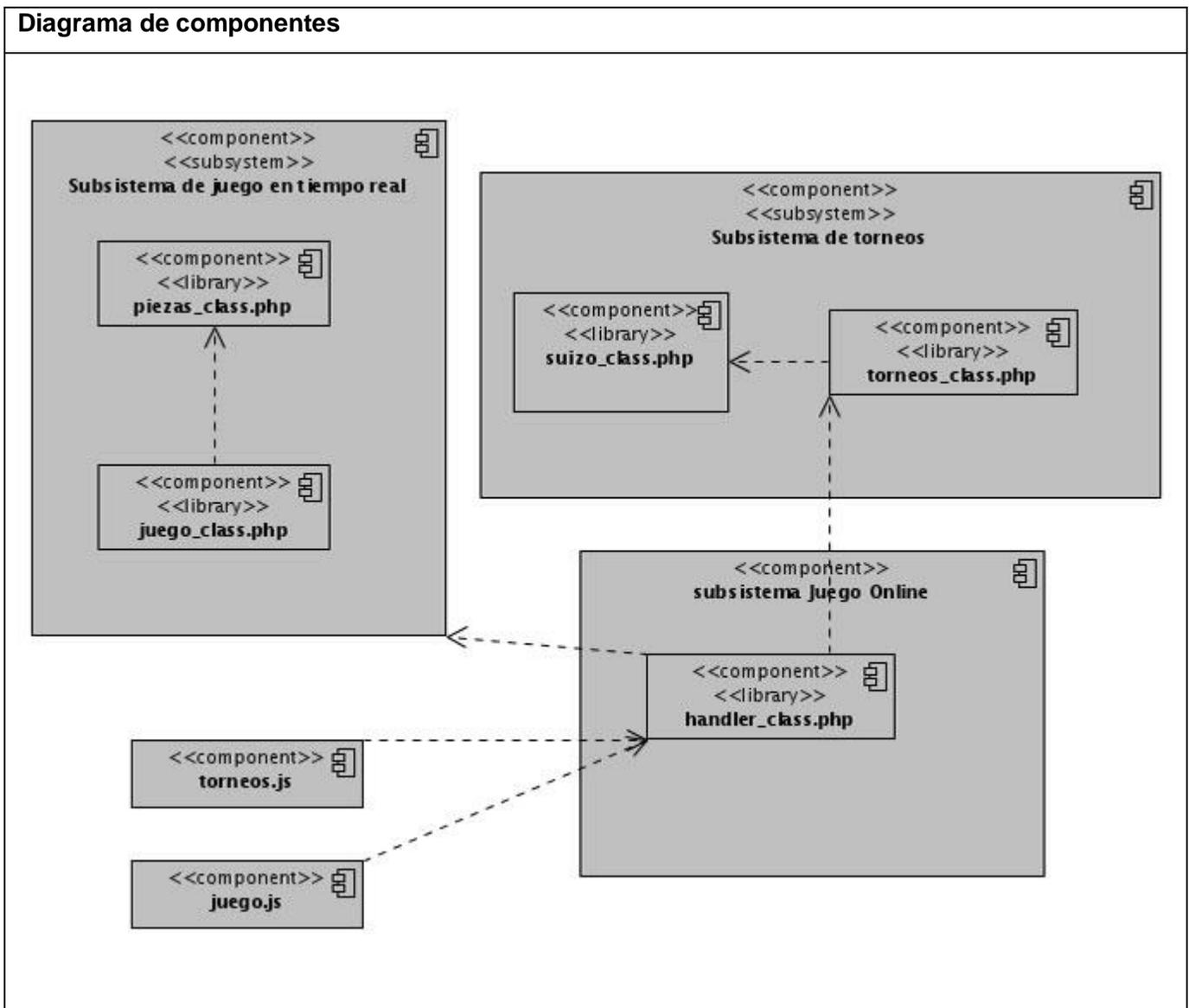
4.1- Diagrama de despliegue

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes *software*, objetos o procesos. En general un nodo será una unidad de computación de algún tipo, desde un sensor a un *mainframe*. Las instancias de componentes *software* pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz).



4.2- Diagrama de componentes.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando.



Conclusiones

El desarrollo de aplicaciones Web para jugar ajedrez online en Cuba es un mundo totalmente nuevo para la industria informática. Se conocen de proyectos realizados en otras universidades pero ninguno con resultados relevantes que se conozcan a nivel nacional. En estos momentos el Instituto Superior Latinoamericano de Ajedrez (ISLA) trabaja en base a desarrollar un evento donde todas las personas que desarrollan software para el ajedrez en nuestro país participen, propondrán además como sede nuestra Universidad, que mejor lugar para tan grandiosa actividad. Es una idea que se está manejando desde hace algún tiempo pero no hay nada materializado al respecto.

Con este nuevo Módulo de Torneos implementado y la plataforma Infodrez en conjunto, se desea de marcar un punto de partida en el desarrollo de software para el ajedrez en Cuba, ya que está en proyecto que se publique en la red los Joven Club de Computación y Electrónica, desde la cual se tendrá acceso a nivel nacional.

Hasta el momento este módulo tiene implementadas las modalidades de torneo “Suizo” y “Round Robin”, se desea en un futuro no muy lejano la implementación de los demás sistemas de torneos. Es necesario decir además que el número de visitas desde que se puso en funcionamiento este módulo ha aumentado en cifras asombrosas, parece ser una de las mejores opciones para los usuarios. Hablando de datos promedios, se ha demostrado que de 1100 visitas diarias que tenía el sitio ha llegando a 1540, de alrededor de 1300 partidas diarias en el juego online ahora se juegan 2700 y de 2800 usuarios que tenía el juego online hace un mes atrás ahora tiene más de 3750, casi 100 usuarios en un mes (Datos del 08-05-2008).

Esto es una muestra de todo lo que se puede hacer en aras de mejorar el desarrollo y la alfabetización del juego ciencia en Cuba.

Recomendaciones

- Implementar las demás modalidades de torneos de ajedrez.
- Asesorar los miembros del proyecto que seguirán desarrollando la plataforma Infodrez en base a transmitir la experiencia de trabajo.
- Optimizar la implementación del módulo de torneos.
- Implementar los torneos por grupos, o sea que cada torneo tenga más de un grupo.
- Implementar también los torneos por equipo.

Bibliografía

Blanco, Jairo. 2007. Blog de Jairo Blanco. *Tipos de Plataformas Web*. [En línea] Blog de Jairo Blanco, 16 de julio de 2007. [Citado el: 11 de abril de 2008.] http://www.jairoblanco.com/net/plataformas-web_90/.

Ciberaula. 2007. Ciberaula. *Ventajas de Linux*. [En línea] RedHat, 20 de abril de 2007. [Citado el: 15 de abril de 2008.] http://linux.ciberaula.com/articulo/ventajas_inconvenientes_linux/.

Comunidad de Fedora. 2007. Fedora-es. *MySQL vs Postgre*. [En línea] Comunidad de Fedora en Castellano, 16 de julio de 2007. [Citado el: 16 de abril de 2008.] <http://www.fedora-es.com/node/189>.

Daccach, José Camilo. 2008. Artículos Delta. *LAMP*. [En línea] Artículos Delta, 13 de enero de 2008. [Citado el: 10 de abril de 2008.] <http://www.deltaasesores.com/prof/PRO152.html>.

Filehungry. 2007. Filehungry. *Referencia de Visual Paradigm*. [En línea] Filehungry, diciembre de 2007. [Citado el: 17 de abril de 2008.] http://textver.filehungry.com/spanish/product/windows_software/programming/specialized_tools/visual_paradigm_for_uml_standard_edition__no_install_support.

Herrera del Sol, Isabel. 2007. *Sistema Suizo*. Ciudad Habana : s.n., 2007.

Jacobson, J. 2004. *El Lenguaje Unificado de Modelado*. 2004.

Maestrosdelweb. 2006. Maestrosdelweb. *Evaluando Zend Studio*. [En línea] Maestrosdelweb, 11 de marzo de 2006. [Citado el: 18 de abril de 2008.] <http://www.maestrosdelweb.com/editorial/zendstudio/>.

Wikipedia. 2008. Wikipedia. [En línea] 20 de 05 de 2008. [Citado el: 28 de 04 de 2008.] www.wikipedia.org. 1.

—. **2008.** Wikipedia. *Ajedrez*. [En línea] Wikipedia, 22 de mayo de 2008. [Citado el: 15 de febrero de 2008.] <http://es.wikipedia.org/wiki/Ajedrez>.

—. **2008.** Wikipedia. *Torneos*. [En línea] Wikipedia, 06 de marzo de 2008. [Citado el: 16 de febrero de 2008.] <http://es.wikipedia.org/wiki/Torneos>.

—. **2008.** Wikipedia. *Proceso Unificado de Rational*. [En línea] Wikipedia, 20 de mayo de 2008. [Citado el: 20 de marzo de 2008.] http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational.

—. **2008.** Wikipedia. *PHP*. [En línea] Wikipedia, 20 de mayo de 2008. [Citado el: 10 de abril de 2008.] <http://es.wikipedia.org/wiki/PHP#Ventajas>.

—. **2008.** Wikipedia. *Servidor Web Apache*. [En línea] Wikipedia, 12 de mayo de 2008. [Citado el: 16 de abril de 2008.] http://es.wikipedia.org/wiki/Servidor_HTTP_Apache .

—. **2008.** Wikipedia. *MySQL*. [En línea] Wikipedia, 12 de mayo de 2008. [Citado el: 17 de abril de 2008.] <http://es.wikipedia.org/wiki/MySQL>.

—. **2008.** Wikipedia. *Quanta Plus*. [En línea] Wikipedia, 24 de abril de 2008. [Citado el: 17 de abril de

2008.] http://es.wikipedia.org/wiki/Quanta_Plus .

—. **2008**. Wikipedia. *WYS/WYG*. [En línea] Wikipedia, 17 de marzo de 2008. [Citado el: 18 de abril de 2008.] <http://es.wikipedia.org/wiki/WYSIWYG>.

—. **2008**. Wikipedia. *Elo*. [En línea] Wikipedia, 12 de marzo de 2008. [Citado el: 01 de mayo de 2008.] <http://es.wikipedia.org/wiki/Elo> .

Glosario de Términos

Elo: El sistema de puntuación Elo (o, incorrectamente, sistema de puntuación ELO) es un método para calcular la fuerza relativa de los jugadores de juegos como el ajedrez y el go. Fue inventado para mejorar el sistema de clasificación vigente de los jugadores de ajedrez (Wikipedia, 2008).