

**Universidad de las Ciencias Informáticas  
Facultad 8**



**Título: Análisis y Diseño del  
Módulo Ajedrez por Correspondencia del  
Proyecto Infodrez**

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

**Autor:** Carlos Alberto Cruz Brache

**Tutor(es):** Ing. Alison Muñoz Capote

**Co-tutor:** Lic. Tomas Orlando

Ciudad de la habana, Junio, 2008

Año 50 de la Revolución

**DECLARACIÓN DE AUTORÍA.**

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

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

**Carlos Alberto Cruz Brache**

**Alison Muñoz Capote**

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Tutor

## **AGRADECIMIENTOS.**

A mi mamá por haber depositado en mi toda su confianza y amor, por haber sabido guiarme por el buen camino, siempre dándome su apoyo incondicional. A la memoria de mi papá que aunque no se encuentre conmigo físicamente siempre fue mi mayor fuente de inspiración y fuerza desde que la vida decidió separarme de él, siempre supo guiarme como el mejor padre del mundo sembrando en mí los valores que me caracterizan, con su ejemplo de firmeza, audacia e inteligencia.

A mi familia por apoyarme siempre en todo. Especialmente a mi hermano, mi abuelo, mi tío Berutica, mi tía Cuca, mi tío Cheito, mi tío Alberto por su cariño insuperable y su apoyo. También a mis primos que son inigualables y a todos que aunque no los haya mencionados siempre estarán en mi corazón.

A todos mis tíos y tías en especial para mi tía Martica en su enorme cariño y apoyo a mi y mi mamá. A mi tío Arturo por seguir muy de cerca mi futuro ocupando así con su enorme apoyo el papel de un padre desde el mismo momento que me hicieron falta consejos paternos.

A la Universidad de las Ciencias Informáticas por la formación brindada durante estos cinco años de estudio.

A mis amigos y compañeros de todos los tiempos, a todos aquellos que se detuvieron al menos un instante para regalarme un gesto de amistad, una frase de apoyo...

A todas aquellas personas que de una forma u otra contribuyeron al desarrollo de este trabajo.

A la Revolución, por darme la posibilidad de hacer realidad este sueño.

## **DEDICATORIA.**

Dedico este trabajo a mi mamá, por haberme dado la vida, porque la mayor parte de lo que soy es por ella, por todo lo que ha pasado para darme lo mejor, por la enseñanza que me ha dado, su amor, paciencia, sacrificios, dedicación y consejos. Además por ser la persona más importante en mi vida.

En especial al mayor protagonista de esta dedicatoria a mi padre querido que aunque la vida no le dio la oportunidad de verme graduado siempre está presente en cada paso que doy en cada decisión de mi vida y siempre está y estará presente cada día de mi vida en mi corazón, a ti Carlos Cruz Fuentes muchas Gracias, porque este sueño más que mío es tuyo.

A mi Hermano que lo quiero con la vida, a mi abuelo Cheo y a todos mis primos.

A mis tíos y tías que han confiado en mí y me han dado consejos en momentos duros.

A mi tío Titi por ser como mi segundo padre y guiarme siempre por el buen camino, su enorme sacrificio para que saliera adelante y por confiar en mí en todo momento.

A todos los que quiero y siempre han querido lo mejor para mí los tengo en el corazón y el pensamiento.

## **RESUMEN.**

Nuestra universidad cuenta con una cátedra de ajedrez donde se imparten clases de ajedrez y se realizan torneos a nivel UCI, así como también a sido sede de torneos nacionales, para estos torneos es necesario asistir de forma presencial y esto requiere de tiempo, por esta situación algunos usuarios de este deporte no pueden asistir por causa de trabajo o estudio, otro problema que existe es que la modalidad del ajedrez, el ajedrez por correspondencia no se juega en la UCI. El presente trabajo, tiene como objetivo diseñar una aplicación informática para automatizar los procesos de gestión de un torneo de ajedrez por correspondencia. Corresponde al desarrollo del módulo Ajedrez por correspondencia, llevado a cabo por el proyecto Infodrez. Para el desarrollo del sistema, se siguieron los pasos que propone el Rational Unified Process (RUP), para los flujos de trabajo, Modelación del Negocio, Requisitos, Análisis y Diseño. Se utilizó el Lenguaje Unificado de Modelado (UML) como soporte a la metodología y la herramienta CASE Visual Paradigm para el modelado visual. Obteniéndose como resultado el diseño del software con la propuesta de su implementación. Con el diseño del sistema propuesto se pretende facilitar la gestión de las partidas de ajedrez por correspondencia en la UCI, donde los amantes de este juego de ajedrez por email cuenten con una aplicación Web donde puedan registrarse y ser usuarios de la misma y que desde sus ordenadores puedan acceder a la aplicación haciéndoles que sus partidas sean mas amenas y placenteras.

## **PALABRAS CLAVE.**

Ajedrez por correspondencia, aplicación web, Torneo.

**TABLA DE CONTENIDOS.**

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	3
1.1 Introducción.....	3
1.2 Estado del Arte.....	3
1.2.1 Ajedrez por Correspondencia.....	3
1.2.2 Ajedrez por Correspondencia en Cuba.....	4
1.2.3 Sistemas que gestionan el Ajedrez Por Correspondencia (Internacional).....	5
1.2.4 Sistemas que gestionan el Ajedrez por Correspondencia (Nacional).....	6
1.3 Solución a través de un sistema de gestión Web.....	6
1.4 Arquitectura Cliente-Servidor.....	7
1.4.1 Estilos de arquitecturas y patrones.....	7
1.4.2 Modelo Cliente Servidor de tres Capas ( <i>Three Tier</i> ).....	8
1.4.3 ¿Por qué MVC?.....	8
1.4.4 Tendencias, Tecnologías y Metodologías actuales.....	9
1.5 Técnicas y Tecnologías del lado del Cliente.....	9
1.5.1 Script del lado del cliente.....	9
1.6 Tecnologías del lado del servidor.....	12
1.6.1 Script del lado del servidor.....	12
1.7 Tecnologías de desarrollo web (Combinación de varias tecnologías).....	14
1.8 Gestor de Base de Datos.....	14
1.9 Servidor de aplicaciones Web Apache.....	16
1.10 Metodologías de desarrollo de software.....	16
1.11 Lenguaje de Modelado.....	19
1.12 Herramientas CASE.....	19
1.13 Herramientas de desarrollo.....	20
1.14 Conclusiones.....	21
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	22
2.1 Introducción.....	22
2.2 Situación Problemática.....	22
2.3 Propuesta del Sistema.....	22
2.4 Modelado del Negocio.....	23
2.5 Actores y Trabajadores del Negocio.....	24
2.6 Diagrama de Casos de Uso del Negocio.....	26

2.7 Descripciones Textuales. ....	26
2.8 Diagramas de Actividades.....	35
2.9 Diagramas de Clases del Modelo de Objetos.....	42
2.10 Especificación de los requisitos del sistema .....	47
2.10.1 Requerimientos funcionales: .....	47
2.10.2 Requisitos no Funcionales.....	49
2.11 Modelación del Sistema. ....	51
2.12 Actores del Sistema. ....	51
2.12 Diagrama de casos de uso del sistema. ....	52
2.12.1 Descripción de los casos de uso del sistema.....	53
2.13 Conclusiones.....	62
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....</b>	<b>64</b>
3.1 Introducción. ....	64
3.2 Modelo de análisis. ....	64
3.2.1 Diagrama de Clases de Análisis.....	64
3.3. Modelo de Diseño. ....	66
3.3.1. Diagramas de Secuencia y Colaboración.....	67
3.4 Diagramas de Clases del Diseño. ....	75
3.5 Diseño de la base de datos. ....	81
3.5.1 Diagrama de clases persistentes.....	82
3.5.2 Modelo entidad relación. ....	83
3.6 Conclusiones.....	83
<b>CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD. ....</b>	<b>84</b>
4.1 Introducción. ....	84
4.2 Técnicas de Estimación. ....	84
4.3 Planificación basada en casos de usos.....	84
4.3.1 Cálculo de Puntos de Casos de Uso sin Ajustar.....	85
4.4 Cálculo de Puntos de Caso de Uso Ajustados. ....	86
4.5 Calcular el Esfuerzo del Flujo de Trabajo Implementación. ....	89
4.6 Costo. ....	91
4.7 Beneficio tangible e intangible.....	91
4.7.1 Tangibles.....	91
4.7.2 Intangibles.....	91
4.8 Análisis costo-beneficio.....	92
4.9 Conclusiones.....	92

CONCLUSIONES.....	93
RECOMENDACIONES.....	94
REFERENCIAS BIBLIOGRÁFICAS.....	95
BIBLIOGRAFÍA.....	98
ANEXOS.....	100
GLOSARIO DE TÉRMINOS.....	113

### ÍNDICE DE TABLAS.

Tabla 1: Descripción de los actores del negocio.....	24
Tabla 2: Descripción de los trabajadores del negocio.....	25
Tabla 3: Descripción textual del caso de uso del negocio “Organizar Torneo”.....	26
Tabla 4: Descripción textual del caso de uso del negocio “Jugar Torneo”.....	28
Tabla 5: Descripción textual del caso de uso del negocio “Solicitar Torneo”.....	29
Tabla 6: Descripción textual del caso de uso del negocio “Reportar Resultados del Torneo”.....	30
Tabla 7: Descripción textual del caso de uso del negocio “Gestionar Rating de los Jugadores”.....	31
Tabla 8: Descripción textual del caso de uso del negocio “Solicitar Licencia”.....	32
Tabla 9: Descripción textual del caso de uso del negocio “Solicitar Licencia”.....	33
Tabla 10: Descripción textual del caso de uso del negocio “Jugar Partida”.....	34
Tabla 11: Descripción de los Actores del Sistema.....	51
Tabla 12: Descripción del caso de uso “Autenticarse”.....	53
Tabla 13: Descripción del caso de uso “Mostrar Torneos Disponibles”.....	54
Tabla 14: Descripción del caso de uso “Jugar Partida”.....	55
Tabla 15: Descripción del caso de uso “Gestionar Torneo”.....	56
Tabla 16: Descripción del caso de uso “Gestionar Partida”.....	59
Tabla 17: Descripción del caso de uso “Solicitar Torneo”.....	61
Tabla 18: Factor de Peso de los Actores sin Ajustar (UAW).....	85
Tabla 19: Factor de Peso de los Casos de Uso sin Ajustar (UUCW).....	86
Tabla 20: Factor de Complejidad Técnica (TCF).....	87
Tabla 21: Criterio de la distribución del esfuerzo.....	90
Tabla 22: Descripción del caso de uso “Permitir Unirse a Torneo”.....	100
Tabla 23: Descripción del caso de uso “Mostrar datos de la partida”.....	101
Tabla 24: Descripción del caso de uso “Realizar Pareo”.....	102
Tabla 25: Descripción del caso de uso “Gestionar Jugadores del torneo”.....	103

### ÍNDICE DE FIGURAS.

Figura 1: Diagrama de Casos de Uso del Negocio.....	26
Figura 2: Diagrama de actividades “Gestionar Rating de los Jugadores”.....	36



Figura 3: Diagrama de actividades “Organizar torneo” .....	37
Figura 4: Diagrama de actividades “Reportar resultados del Torneo” .....	38
Figura 5: Diagrama de actividades “Solicitar Licencia” .....	39
Figura 6: Diagrama de actividades “Solicitar Torneo” .....	40
Figura 7: Diagrama de actividades “Jugar Torneo” .....	41
Figura 8: Diagrama de actividades “Jugar Partida” .....	42
Figura 9: Modelo de Objeto del Negocio .....	43
Figura 10: Modelo de Objeto “Gestionar Rating de los Jugadores” .....	44
Figura 11: Modelo de Objeto “Jugar Partida” .....	44
Figura 12: Modelo de Objeto “Jugar Torneo” .....	45
Figura 13: Modelo de Objeto “Organizar Torneo” .....	45
Figura 14: Modelo de Objeto “Reportar Resultados del Torneo” .....	46
Figura 15: Modelo de Objeto “Solicitar Licencia” .....	46
Figura 16: Modelo de Objeto “Solicitar Torneo” .....	47
Figura 17: Diagrama de Casos de Uso del Sistema.....	52
Figura 18: Diagrama de clases de análisis del caso de uso “Permitir Unirse a Torneo” .....	65
Figura 19: Diagrama de clases de análisis del caso de uso “Autenticarse” .....	65
Figura 20: Diagrama de clases de análisis del caso de uso “Gestionar Partida” .....	65
Figura 21: Diagrama de clases de análisis del caso de uso “Solicitar Licencia” .....	66
Figura 22: Diagrama de clases de análisis del caso de uso “Jugar Partida” .....	66
Figura 23: Diagrama de clases de análisis del caso de uso “Gestionar Torneo” .....	66
Figura 24: Diagrama de Secuencia del Caso de Uso: “Autenticarse” .....	67
Figura 25: Diagrama de Secuencia del caso de uso “Gestionar Partida” .....	68
Figura 26: Diagrama de Secuencia del caso de uso “Jugar Partida” .....	68
Figura 27: Diagrama de Secuencia del Caso de Uso: “Permitir Unirse a Torneo” .....	69
Figura 28: Diagrama de Secuencia del caso de uso “Realizar Jugada” .....	69
Figura 29: Diagrama de Secuencia del Caso de Uso: “Solicitar Licencia” .....	69
Figura 30: Diagrama de Colaboración del Caso de Uso: “Gestionar Partida” .....	70
Figura 31: Diagrama de Colaboración del caso de uso “Jugar Partida” .....	71
Figura 32: Diagrama de Colaboración del Caso de Uso: “Autenticarse” .....	72
Figura 33: Diagrama de Colaboración del caso de uso “Permitir Unirse a Torneo” .....	73
Figura 34: Diagrama de Colaboración del Caso de Uso: “Realizar Jugada” .....	74
Figura 35: Diagrama de Colaboración del Caso de Uso: “Solicitar Licencia” .....	75
Figura 36: Diagramas de Clases del Diseño del caso de uso “Autenticarse” .....	76
Figura 37: Diagramas de Clases del Diseño del caso de uso “Gestionar Partida” .....	77
Figura 38: Diagramas de Clases del Diseño del caso de uso “Gestionar Torneo” .....	78
Figura 39: Diagramas de Clases del Diseño del caso de uso “Jugar Partida” .....	79
Figura 40: Diagramas de Clases del Diseño del caso de uso “Realizar Jugada” .....	80
Figura 41: Diagramas de Clases del Diseño del caso de uso “Solicitar Licencia” .....	81
Figura 42: Diagramas de Clases Persistentes .....	82
Figura 43: Modelo Entidad Relación .....	83
Figura 44: Diagrama de clases de análisis del caso de uso “Definir_Partidas_No_Cocluidas” .....	106
Figura 45: Diagrama de clases de análisis del caso de uso “Gestionar_Jugadores_del _Torneo” .....	106
Figura 46: Diagrama de clases de análisis del caso de uso “Imprimir Estado de la Partida” ...	106

---

Figura 47: Diagrama de clases de análisis del caso de uso “Mostrar Torneos Disponibles” ...	106
Figura 48: Diagrama de clases de análisis del caso de uso “Mostrar Datos de la Partida” .....	107
Figura 49: Diagrama de clases de análisis del caso de uso “Notificar Por Correo” .....	107
Figura 50: Diagrama de clases de análisis del caso de uso “Realizar Jugada” .....	107
Figura 51: Diagrama de clases de análisis del caso de uso “Realizar Pareo” .....	107
Figura 52: Diagrama de Secuencia del caso de uso “Gestionar Torneo” Escenario “Crear Torneo” .....	108
Figura 53: Diagrama de Secuencia del caso de uso “Gestionar Torneo” Escenario “Editar Torneo” .....	108
Figura 54: Diagrama de Secuencia del caso de uso “Gestionar Torneo” Escenario “Eliminar Torneo” .....	108
Figura 55: Diagrama de Secuencia del caso de uso “Mostrar Datos de Partida” .....	109
Figura 56: Diagrama de Secuencia del caso de uso “Mostrar Torneos Disponibles” .....	109
Figura 57: Diagrama de Secuencia del caso de uso “Notificar por Correo” .....	109
Figura 58: Diagrama de Secuencia del caso de uso “Realizar Pareo” .....	110
Figura 59: Diagramas de Clases del Diseño del caso de uso “Definir Partidas No Concluidas” .....	110
Figura 60: Diagramas de Clases del Diseño del caso de uso “Gestionar Jugadores del torneo” .....	111
Figura 61: Diagramas de Clases del Diseño del caso de uso “Mostrar Datos de partida” .....	111
Figura 62: Diagramas de Clases del Diseño del caso de uso “Realizar Pareo” .....	112

## INTRODUCCIÓN.

La Informatización de La Sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de La Información y las Comunicaciones en la vida cotidiana, para satisfacer las necesidades de todas las esferas de la sociedad, en su esfuerzo por lograr cada vez más eficacia en todos los procesos y por tanto un aumento en la calidad de vida de los ciudadanos.

Tras el triunfo revolucionario, Cuba se propuso un camino de desarrollo que pudiera satisfacer por igual las necesidades básicas de su población, sobre la base de una distribución más justa y equitativa de la riqueza. Hoy Cuba tiene un nuevo reto: la necesidad de dominar e introducir en la práctica social las Tecnologías de La Información y las Comunicaciones y lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría a nuestra sociedad acercarse más hacia el objetivo de un desarrollo sostenible.

Cuba ha trazado el Programa Nacional de Informatización de La Sociedad Cubana, que persigue agilizar el uso masivo de las Tecnologías de La Información a escala nacional, en las diferentes esferas y sectores buscando la optimización de recursos y el logro de mayor productividad en dichas esferas.

La Universidad de Ciencias Informáticas, expresión de los programas de la Batalla de Ideas que hoy se desarrollan en nuestro país y eficaz medio para lograr la informatización de la sociedad, hace su aporte a la industria cubana de software con la creación del proyecto Infodrez, ejemplo de la puesta en práctica y desarrollo de estas políticas antes mencionadas.

La cátedra de ajedrez de La UCI no cuenta actualmente con una aplicación Web que permita el desarrollo de una partida de la modalidad ajedrez por correspondencia, por lo que el **problema** a solucionar, consiste en: ¿Cómo dotar a La Cátedra de Ajedrez de La UCI, Remberto Fernández, del Diseño de una aplicación Web, que permita la gestión de partidas desarrolladas en la modalidad ajedrez por correspondencia?

Por tanto el **objeto de estudio** de este trabajo son los procesos que gestionen los contenidos relacionados con el ajedrez por correspondencia.

De esto se deriva que el **campo de acción** que abarca este trabajo, es la automatización de los procesos identificados en la gestión del ajedrez por correspondencia y el diseño de una aplicación Web que permita gestionar los mismos.

Como **Idea a defender** se parte que si se desarrolla el diseño de un sistema informático adecuado a los procedimientos relacionados con un torneo de ajedrez por correspondencia, entonces facilitará la realización del juego de ajedrez, en este caso el ajedrez por correspondencia.

El **objetivo general** de este trabajo será diseñar una aplicación informática del módulo ajedrez por correspondencia que posibilite la gestión de un sistema que permita la automatización de un torneo de la modalidad de ajedrez por correspondencia.

De acuerdo con esta propuesta se derivan los siguientes **objetivos específicos**:

- Realizar una investigación para caracterizar los sitios de ajedrez por correspondencia existentes.
- Realizar el análisis del dominio que abarcara el modulo de ajedrez por correspondencia.
- Elaborar el análisis y diseño del sistema e investigar la arquitectura adecuada para este sistema.
- Conformar la documentación del sistema para el desarrollo de posteriores versiones.
- Entrevistar especialistas y visitar instituciones de ajedrez para familiarizarse con el entorno de trabajo.

Para cumplir con estos objetivos y resolver la situación problemática planteada, se proponen las siguientes **tareas**:

- Investigar acerca de las herramientas, tendencias y tecnologías actuales relacionadas con el ajedrez por correspondencia.
- Realizar una investigación acerca de las metodologías existentes para seleccionar la que utilizaremos en nuestro sistema.
- Diseñar una aplicación que dé soporte a los procesos de Gestión para jugar el ajedrez por correspondencia.
- Generar la documentación de la información referente al análisis y diseño del sistema.
- Realizar un levantamiento de requisitos, para conocer las inquietudes del cliente y lograr una familiarización con estos.

### **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.**

#### **1.1 Introducción.**

Este capítulo está destinado a brindar una panorámica de los temas relacionados a las aplicaciones Web relacionadas con la modalidad de ajedrez por correspondencia. También se abordarán las diferentes teorías y fundamentos aplicados para el análisis y diseño del sistema, así como la descripción de los principales conceptos asociados al dominio del problema que son necesarios para entender el modelo de negocio y la propuesta de solución. En este Capítulo se detallan los procesos elementales de negocio, se realiza un estudio y análisis de las tecnologías apropiadas para desarrollos de aplicaciones Web, escogiendo la mejor opción para la construcción del sistema propuesto.

#### **1.2 Estado del Arte.**

El ajedrez es un juego de mesa para dos personas. Más precisamente, es un juego de guerra. 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.

##### **1.2.1 Ajedrez por Correspondencia.**

En el ajedrez por correspondencia los jugadores juegan separados físicamente y temporalmente. No están uno enfrente de otro, ni están "presentes" en el momento de hacerse las jugadas. Los movimientos se transmiten por los medios que se usan para comunicarse entre personas: en general "correspondencia". Que puede ser: correo postal, e-mail. El tiempo de reflexión se calcula en días, es una modalidad muy extendida que incluso posee revistas especializadas (por ejemplo Fernschach) y que da lugar a competiciones oficiales a todos los niveles (regionales, nacionales e internacionales; algunas sirven para determinar el Campeón Mundial de Ajedrez por Correspondencia). Una partida de ajedrez postal se desarrolla enviando cada jugada mediante una carta (o una carta - postal) en la que el movimiento se halla anotado. Cuando se participa en un torneo, todas las partidas se juegan a la vez, lo cual representa disponer de bastante tiempo. Desde el punto de vista histórico se tienen noticias, se tienen noticias de que los reyes Enrique I de Inglaterra (1100 - 1135) y Luis VI de Francia (1108 - 1137) disputaron una partida "a través del Canal de La Mnacha", en que las jugadas se enviaban por medio de correos especiales. A parte de este dato hay muy pocas noticias de la época sobre la especialidad, solo es destacable el comentario sobre el orientalista inglés T Hyde, quien en su libro "De ludis orientalibus", publicado en 1694, dice: " me han comunicado que algunos comerciantes venecianos y croatas, tienen la costumbre, también existente en otros países, de jugar al ajedrez por correspondencia. Cada movimiento es confiado en una carta y, por ello una partida cuesta muy cara a

los jugadores."Hay noticias, hacia 1740, que aseguran que el Rey de Prusia, Federico el Grande, y la Zarina Catalina de Rusia jugaron al ajedrez por correspondencia con Voltaire. Del mismo modo se sabe que Philidor había jugado por correspondencia en 1766.La primera partida postal que se conserva se celebró en 1804 entre unos oficiales del ejército holandés de La Haya y Breda. Pero más conocidos son los encuentros entre Edimburgo-Londres (1824-2828), Liverpool/Manchester (1825-1826), París-Londres (1834-1836) y Budapest-París (1843-1846).Entre los jugadores más destacados de ajedrez postal figuran: Alekhine, Nimzowich, Eliskases, H. Müller, Keres, Kelly, Kevitz, Berliner, Dyckhoff, Estrin, Nielsen, Ragozin, Purdy...Una variante de la modalidad son las partidas por telégrafo, por radio o mediante un periódico, pues, esencialmente, tiene las mismas características ajedrecísticas; solo cambia el soporte físico en que se transmite la información de las jugadas que se van efectuando. Ajedrez por correspondencia ha sido jugado en formas organizada a nivel internacional desde 1928. El actual ICCF (ver debajo), que fija normas y cuyas reglas son sancionadas por el FIDE, fue fundado en 1951. Hoy en día el ICCF también organiza torneos de ajedrez por vía el correo electrónico, pero el ajedrez por correspondencia tradicional subsistirá hasta que todos los jugadores de ajedrez en el mundo sean capaces de conectarse a Internet. Esta sección destaca sobre todo el ajedrez por correspondencia en torno al ICCF. Sin embargo, hay otras organizaciones que aplican las mismas reglas. Pero la mayor parte de ellas, así como los sitios Web, actúan como intermediarios u organizan partidos y torneos por correo electrónico, además del ajedrez por correspondencia. Están registrados en la categoría Sitios de juego. [5].

### **1.2.2 Ajedrez por Correspondencia en Cuba.**

La creación de una organización cubana dedicada al desarrollo de competencias oficiales de ajedrez postal ocurrió a partir de 1972, hace 25 años. Si nos ajustamos a las referencias dadas por Carlos A. Palacios, el más documentado historiador que ha tenido el ajedrez en nuestro país, la práctica del "ajedrez a distancia" se produce en Cuba, fundamentalmente, mediante partidas jugadas por telégrafo, teléfono, etc., asociadas a encuentros especiales, que nada tuvieron que ver con la práctica oficial, regular y organizada, de torneos típicamente postales. A partir de 1960 se inicia en Cuba un gran movimiento que conduce al desarrollo masivo del ajedrez, que alcanza su punto culminante con la celebración en La Habana de dos eventos trascendentales: el IV Torneo Internacional Capablanca In Memoriam - en el que participa el GM Rober Fischer "a distancia" - ; y la memorable XVII Olimpiada Mundial de Ajedrez. Un movimiento nacional tan poderoso habría conducido, más temprano que tarde, a que de alguna manera se comenzaran a organizar en Cuba torneos de ajedrez por correspondencia.

### **¿Qué es la FECAP?**

La Federación Cubana de Ajedrez Postal es el organismo rector del juego ciencia a distancia en el país y agrupa en su seno a los amantes de esta disciplina, que se asocian de forma voluntaria a la misma, sin tener que abonar cuota o pago alguno, de acuerdo con la política que para el deporte tiene el Estado cubano. Surge en abril de 1972, anexa a la Comisión Nacional de Ajedrez. Es miembro Pleno del Comité Olímpico Cubano (COC) desde octubre de 1989. Pertenece al Instituto Nacional de Deportes y Recreación (INDER). La FECAP es el miembro número 52 de la International Correspondence Chess Federation (ICCF), desde el Congreso celebrado en Suecia en 1977. Es además, miembro de la Confederación Americana de Ajedrez Postal (CADAP). Es la encargada de organizar y desarrollar las competencias entre sus miembros, instándoles a mantener buenas relaciones humanas y compatibilizar la rivalidad deportiva con indispensables muestras de confraternidad y amistad. Al hacerse miembro del mismo puede jugar torneos provinciales, inscribiéndose en su delegación provincial o participar en torneos nacionales, dirigiéndose a la Comisión Nacional de Eventos. [9].

### **1.2.3 Sistemas que gestionan el Ajedrez Por Correspondencia (Internacional).**

#### **¿Qué es la I.C.C.F?**

I.C.C.F. son las siglas, en inglés, de la Federación Internacional de Ajedrez por Correspondencia, organismo gestor y superior de las organizaciones que existen en los distintos países del mundo. La ICCF es una organización reconocida por la FIDE (Federación Internacional de Ajedrez). La ICCF organiza los torneos internacionales por correspondencia (Campeonato del Mundo, torneos Copa, Memoriales, etc.). Normaliza, homogeneiza y dirige las actividades desarrolladas por sus miembros, como La AEAC. Además organiza Congresos Mundiales Anuales donde acuden todos los representantes de los distintos países. Otro Sistema que gestiona actualmente el Ajedrez por Correspondencia es La FEDA (Federación Española de Ajedrez), este organismo que gestiona y dirige los torneos de ajedrez de España, licencias de los jugadores etc. De igual manera que la AEAC es una asociación nacional vinculada a una internacional, la ICCF, la FEDA depende de la FIDE (Federación Internacional de Ajedrez). [4].

#### **1.2.4 Sistemas que gestionan el Ajedrez por Correspondencia (Nacional).**

A nivel nacional se cuenta actualmente con el sitio de ajedrez en Cuba ([www.cuba.cu/ajedrez](http://www.cuba.cu/ajedrez)), en el que no se trata el tema del Ajedrez por Correspondencia y los jugadores de esta modalidad del Ajedrez tienen que recurrir a sitios internacionales.

#### **¿Por qué esta propuesta?**

La razón fundamental de esta propuesta radica en dotar a los Jugadores del Ajedrez por Correspondencia de un Sistema que les permita realizar jugadas por correspondencia, siendo esta herramienta software libre, y soportada sobre una plataforma Web integrada (Infodrez) que contará con otros temas relacionados con el ajedrez .

#### **1.3 Solución a través de un sistema de gestión Web.**

Las aplicaciones Web pueden considerarse como un sitio Web al que se accede de manera normal a través de un navegador, pero dotado de interactividad para la gestión, una base de datos y contraseñas de entrada, sin necesidad de instalar en cada equipo ningún otro programa específico.. Además estas son las encargadas de manejar el estado del negocio y de gestionar datos almacenados con ayuda de algún Sistema Gestor de Base de Datos (SGBD) para poder ser observadas por cierta cantidad de usuarios. Esta es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. Las aplicaciones Web generan dinámicamente una serie de páginas Web en un formato estándar (HTML, XHTML), razón por la cual pueden ser accedidas desde un navegador Web. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Al albergarse la aplicación en un servidor remoto, el funcionamiento y los valiosos datos que contiene son totalmente independientes del ordenador utilizado para la gestión. Así, la normal operación de la aplicación es inmune a una avería de hardware, virus informáticos, o cualquier otro problema local. Esta es accedida a través de un navegador, valiéndose del protocolo HTTP (Hyper Text Transfer Protocol). En términos más simples, una aplicación Web es un Sistema Web que permite a los usuarios ejecutar la lógica de negocio a través de un Navegador (Browser). [13].



## 1.4 Arquitectura Cliente-Servidor.

La tecnología denominada Cliente-Servidor se utiliza generalmente en las aplicaciones de Internet e Intranet.

Ante todo se debe recurrir a los siguientes conceptos:

**Cliente:** Aplicación informática que se utiliza para acceder a los servicios que ofrece un servidor, normalmente a través de una red de telecomunicaciones.

**Servidor:** Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. El ordenador en el que se ejecuta un programa que realiza alguna tarea en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de un ordenador y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor.

Este modelo consiste esencialmente en las peticiones o solicitudes que realiza un programa, los clientes informáticos a otro programa, o sea un servidor. Estos últimos ofrecen una respuesta. La capacidad de proceso en esta arquitectura está repartida entre usuarios y servidores aunque se hace necesario e importante las ventajas en cuanto al aspecto organizativo debido a la centralización de la gestión de la información y la separación de las responsabilidades. Ello posibilita, clarifica y facilita el diseño del sistema. [10].

### 1.4.1 Estilos de arquitecturas y patrones.

Los patrones son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. La dinámica incontenible de la producción de patrones en la práctica de la arquitectura de software, hace que todo el mundo acepte que existen diversas clases de patrones: de análisis, de arquitectura (*divididos en progresivamente estructurales, sistemas distribuidos, sistemas interactivos, sistemas adaptables*), de diseño (*conductuales, creacionales, estructurales*), de organización o proceso, de programación y los llamados idiomas, entre otros. Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución. Dentro de las soluciones estudiadas el patrón más recomendado por especialistas para aplicaciones como la propuesta en el trabajo es el Modelo Vista Controlador (MVC).

Elementos del patrón:

- **Modelo:** Contendrá una lista de usuarios autorizados junto con sus contraseñas, la base de datos sobre la que efectuar las consultas y los módulos (*clases o librerías de funciones*) que se

encarguen de realizar las operaciones (*comprobar la información del usuario, construir la consulta, efectuar la consulta*).

- **Vista:** Muestra la información del modelo al usuario y al necesitar poco o nada de código, puede ser desarrollada por un equipo de diseñadores independiente del de programadores.
- **Controlador:** Gestiona las entradas del usuario y serviría de "pegamento", recogería las órdenes y datos que el usuario enviase desde la vista, las traduciría en operaciones del modelo de negocio y, en base a los resultados, mostrará los resultados en uno u otro formulario de la vista. [11].

#### 1.4.2 Modelo Cliente Servidor de tres Capas (*Three Tier*).

Es una realidad que el objetivo de las aplicaciones modernas no es sólo procesar transacciones y generar reportes, sino el de que sean sistemas de información robustos y que puedan cambiar conforme cambien las necesidades de la empresa o institución. Mientras las aplicaciones se utilicen en el ámbito y edificio de una empresa o institución de regular tamaño, el modelo cliente servidor de dos capas donde se tiene un servidor de aplicación que contiene mezclada la lógica del negocio con la interfaz que le brinda al usuario y un servidor de bases de datos pudiera ser adecuado. Pero cuando la aplicación rebasa los límites físicos de un edificio y se vuelve estatal o mundial, con numerosos clientes distribuidos en cualquier parte, o cuando la aplicación es complicada, el modelo de dos capas se hace insuficiente. Cada uno de los componentes de la aplicación en una arquitectura de tres capas se separa en una sola entidad. Esto te permite implementar componentes de una manera más flexible, es decir, la aplicación tiene que estar preparada para los posibles cambios que el cliente pueda pedir sin tener que reescribir totalmente la aplicación. Este tipo de arquitectura es la más compleja. En esta arquitectura todas las peticiones de los clientes se controlan en la capa correspondiente a la lógica del negocio. Cuando el cliente necesita hacer una petición se la hace a la capa en la que se encuentra la lógica del negocio. Esto es bastante importante pues eso quiere decir que:

El cliente no tiene que tener drivers ODBC ni la problemática consiguiente de instalación de los drivers por tanto se reduce el costo de mantener las aplicaciones cliente.

El cliente y el Gestor de Reglas de negocio tienen que hablar el mismo lenguaje.

El Gestor de Reglas de Negocio y el Servidor de Datos tienen que hablar el mismo lenguaje (*ODBC*). [22].

#### 1.4.3 ¿Por qué MVC?

Un propósito común en numerosos sistemas es el de tomar datos de un almacenamiento y mostrarlos al usuario. Luego que el usuario introduce modificaciones, las mismas se reflejan en el almacenamiento. El flujo de información ocurrirá entre el almacenamiento y la interfaz, una tentación

común, un impulso espontáneo (*hoy se llamaría un anti-patrón*) es unir ambas piezas para reducir la cantidad de código y optimizar la performance. Sin embargo, esta idea es antagónica al hecho de que la interfaz suele cambiar, o acostumbra depender de distintas clases de dispositivos (*clientes ricos, browsers, PDAs*); la programación de interfaces de HTML, además, requiere habilidades muy distintas de la programación de lógica de negocios. Otro problema es que las aplicaciones tienden a incorporar lógica de negocios que van más allá de la transmisión de datos. Resumiendo lo anterior aplicando este patrón se logra separar el modelo de negocios de la presentación usada, así la información podrá ser vista.

#### **1.4.4 Tendencias, Tecnologías y Metodologías actuales.**

En la actualidad, para la realización de un software se tienen en cuenta una serie de tendencias y tecnologías informáticas, con el objetivo de buscar la más adecuada para el desarrollo del mismo. Las tecnologías de la Información no son más que aquellas herramientas computacionales e informáticas que procesan, almacenan, sintetizan, recuperan y presentan información representada de la más variada forma. Es un conjunto de herramientas, soportes y canales para el tratamiento y acceso a la información. Constituyen nuevos soportes y canales para dar forma, registrar, almacenar y difundir contenidos informacionales. Algunos ejemplos de estas tecnologías son la pizarra digital (ordenador personal +proyector multimedia) y La Web.

#### **1.5 Técnicas y Tecnologías del lado del Cliente.**

Un cliente Web (navegador o browser) es una aplicación de software que permite al usuario recuperar y visualizar documentos de hipertexto. La parte cliente de las aplicaciones Web está formada por el código HTML que forma la página Web, con opción a código ejecutable mediante los lenguajes de scripting de los navegadores (JavaScript, Vbscript, etc.) o mediante pequeños programas en Java (applets). La programación del lado del cliente tiene como principal ventaja que la ejecución de la aplicación se delega al cliente, con lo cual se evita recargar al servidor de trabajo. Entre los navegadores más populares están, el Netscape, Internet Explorer, Mozilla. Los lenguajes más comunes que se ejecutan en el cliente son:

*VBScript* y *JavaScript*. [23].

##### **1.5.1 Script del lado del cliente.**

Una práctica muy difundida en el mundo de la programación web es emplear lenguajes interpretados del lado del cliente para añadir más funcionalidad a las aplicaciones, especialmente para crear una experiencia interactiva que no requiera recargar la página cada vez.

### **JavaScript.**

JavaScript es un lenguaje de programación que ha permitido el gran desarrollo de la Web, ha sido el avance más significativo en el logro de páginas Web dinámicas y exactas en cuanto a posición y presentación de su contenido, es un lenguaje robusto y a la vez ligero, el cual a pesar de ser considerado por muchos como un lenguaje no orientado a objetos permite implementar varias de las características de este paradigma de programación, basado en prototipos, las nuevas clases se generan clonando las clases bases y extendiendo su funcionalidad, cualquier navegador puede interpretar el código JavaScript dentro de las páginas Web, permite la máxima interactividad entre el usuario y la página, además la verificación de los datos introducidos por el usuario antes de enviar el formulario al servidor, el manejo de applets y plugins dentro de múltiples marcos de HTML. Este lenguaje puede desarrollarse en el cliente, se aplica en el navegador del cliente (JavaScript en el cliente) y además en el servidor (JavaScript en el servidor) el que nos ocupa en este trabajo es JavaScript en el cliente.

Este lenguaje de programación tiene algunas limitaciones:

JavaScript por definición no es un lenguaje orientado a objetos y debido a la potencia de esta programación se ha conseguido un funcionamiento similar, intuitivo y potente.

- No se precompila.
- No es obligatorio declarar las variables.
- Verifica las referencias en tiempo de ejecución.
- No tiene protección del código, ya que se descarga en texto claro.

El JavaScript es un lenguaje interpretado, o sea, que el sistema lo lee y traduce al mismo tiempo que lo va ejecutando, no confundir con el Java. Con JavaScript se pueden lograr efectos realmente mágicos en una página, así como formularios, imágenes, textos, etc., tiene la ventaja de que la mayoría de los navegadores no tienen problemas para interpretarlo. JavaScript es una forma de trasladar algunas tareas simples al lado del cliente. Cuenta con un conjunto de nodos, que no es más que todo lo que está incluido dentro de una etiqueta de inicio y otra de final. Cada etiqueta se considera un nodo en sí misma. Se distinguen los nodos de texto y los nodos de elementos. El concepto de clases en JavaScript está dado por el prototipo, con la propiedad prototype de los objetos prototipos podemos añadir todas las propiedades que se quieran, no a la instancia sino al mismo

prototipo. Permite codificar de una manera flexible, definir un objeto prototipo y si se necesita añadir alguna propiedad(es) lo único que se tiene que hacer es asignar el `true`. Se puede añadir además propiedades a los objetos prototipo predefinidos por el lenguaje. JavaScript tiene una sintaxis que es en general parecida a la de C, excepto que no hay una función `main` sino que lo que no está dentro de una función se ejecutará mientras se cargue la página. Todos los métodos que JavaScript trae definidos son funciones desde el momento que se ejecutan. Se pueden crear funciones que ejecuten comandos que se le añadan. A partir de la versión 1.3 se puede anidar funciones que amplían notablemente las posibilidades y flexibilidad de la programación. En JavaScript las funciones pueden manipularse dentro del mismo lenguaje, esto es una gran ventaja, ya que además de ser creada puede utilizarse como dato, Por tanto puede asignarse a una variable, guardarse en los elementos de una matriz o en las propiedades de un objeto. Las funciones tienen argumentos que son tratados como objetos y forman una matriz. Esta matriz siempre tiene el nombre de `argument` (`()`). JavaScript recibe información a través de eventos y propiedades de objetos, y la entrega mediante propiedades de objetos y métodos. [24].

### **¿Por qué JavaScript?**

A causa de su gran compatibilidad con todos los navegadores modernos es sin dudas el script más utilizado del lado del cliente. Con este lenguaje a parte de definir interactividades con el usuario se pueden crear efectos especiales en las páginas los cuales son interpretados y ejecutados por el navegador de modo que el único recurso que se hace indispensable para ejecutar un código JavaScript es el navegador Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del Modelo de Objetos de Documento, frecuentemente abreviado DOM, es una interfaz de programación de aplicaciones (en lo adelante API) para documentos HTML (páginas Web) y XML. Define la estructura lógica de los documentos y el modo en que se accede y manipula un documento. En la especificación del DOM, el término "documento" se utiliza en un sentido amplio ya que es el documento el contenedor que soporta los demás elementos. A través del DOM los programadores pueden construir documentos, navegar por su estructura, añadir, modificar o eliminar elementos y contenido. Se puede acceder a cualquier elemento que se encuentre en un documento HTML o XML, y se puede modificar, eliminar o añadir usando DOM, salvo algunas excepciones.

### **Extensible Markup Language (XML).**

**XML**, sigla en inglés de *Extensible Markup Language* (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML. XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. [26].

### **HyperText Markup Language.**

#### **HTML (Acrónimo en inglés de HyperText Markup Language)**

Es un lenguaje de composición de documentos y especificación de ligas de hipertexto, que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. HTML también indica cómo hacer un documento interactivo a través de ligas especiales de hipertexto, las cuales conectan diferentes documentos, ya sea en su computadora o en otras, así como otros recursos de Internet, como FTP. [25].

### **1.6 Tecnologías del lado del servidor.**

Las tecnologías del lado del servidor son aquellos programas o servicios que corren en un servidor remoto y que brindan funcionalidades al sistema.

#### **1.6.1 Script del lado del servidor.**

Los lenguajes de scripts que se ejecutan en el servidor Web son interpretados por alguna aplicación alojada en dicho servidor y que se envían al cliente en un formato comprensible para él, son independientes del cliente por lo que son mucho más flexibles en relación al cambio de un

navegador a otro o con respecto a las versiones del mismo. El cliente solo verá el código HTML terminado. [14].

### **PHP.**

PHP permite la creación de aplicaciones Web muy robustas, al posibilitar la conexión a diferentes tipos de servidores de base de datos como: MySQL, Postgres, Oracle y otros. Tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos y puede interactuar con varios de los servidores Web más populares. Los principales usos de PHP son: la programación de páginas Web dinámicas, la programación en consola y la creación de aplicaciones gráficas independientes del navegador. La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Entre los lenguajes del tipo C se incluye al Java y Javascript, de hecho mucha de la funcionalidad del PHP se la debe al C. PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código. Tiene diferentes interfaces para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD35. [29].

### **¿Porque PHP?**

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

### **Processing Extraction Report Language (PERL).**

Es un lenguaje de programación interpretado, muy utilizado para construir aplicaciones CGI para la Web, se trata de un lenguaje de programación muy práctico para extraer información de archivos de

texto y generar informes a partir del contenido de los ficheros. Es un lenguaje de libre uso, es decir, gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. Permite efectuar búsquedas de secuencias de caracteres. Perl se utiliza con fuerza en la escritura de procesos CGI (Common Gateway Interface) instalados en un servidor Web o para el desarrollo de procesos de mantenimiento de las actividades de un servidor. No es muy recomendado para desarrollar procesos de puro cálculo científico o programas que necesitan una gran velocidad y precisión de cálculo o elaboraciones numéricas complejas pero se convierte en una necesidad para quien tenga que manejar un sitio Web que no esté compuesto sólo de texto e imagen. [28].

### **1.7 Tecnologías de desarrollo web (Combinación de varias tecnologías).**

#### **AJAX** (Asynchronous JavaScript and XML)

Sus siglas en español (JavaScript y XML asíncronos), es una técnica para el desarrollo web con la que se pueden crear aplicaciones interactivas, sin la necesidad de recargar la página web, para ello se utiliza la combinación de diversas tecnologías como son:

HTML (o XHTML1) y hojas de estilos en cascada (CSS), con las cuales se puede mostrar la página web a los usuarios.

Document Object Model (DOM) y JavaScript, para interactuar dinámicamente con los datos.

AJAX no es un lenguaje de programación en sí, sino, la asociación de varios que en conjunto nos definen una nueva tecnología para el desarrollo de aplicaciones web. [12].

### **1.8 Gestor de Base de Datos.**

Es un conjunto de herramientas que proporciona a administradores, analistas, programadores, usuarios los medios necesarios para describir, recuperar y manipular los datos almacenados en la BD, manteniendo la seguridad, integridad y confidencialidad de los mismos. [18].

Sus principales funciones son:

- Definir los datos.
- Manipular los datos.
- Garantizar la seguridad e integridad de los datos.
- La gestión de las transacciones y el acceso concurrente.
- Evitar la redundancia de los datos, eliminando así la inconsistencia de los mismos.
- Mejorar la eficacia de acceso a los datos, en especial en el caso de consultas inesperadas.



### MySQL.

Uno de los Sistemas Gestores de Base de Datos que se presentan es MySQL. El mismo cuenta con más de seis millones de instalaciones, es multihilo, multiusuario, rápido y muy robusto. Es software libre, publicado bajo la licencia GPL (GNU Public License) y mantenido por la compañía sueca MySQL AB. MySQL es muy utilizado en los sistemas Web. Tiene gran popularidad por estar grandemente ligado a PHP. Es capaz de funcionar sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows. [3].

Otras características importantes que posee MySQL son:

- Posee soporte completo para operadores y funciones de cláusulas select, where, group by y order by.
- Mediante la verificación basada en el host y tráfico de contraseñas encriptado ofrece un sistema de contraseñas y privilegios seguro, denotando cualidades de seguridad altamente confiables.
- Es capaz de soportar gran cantidad de datos.
- En general, de las diferentes arquitecturas que los SGBD la más extendida es la relacional, siendo una de las que más éxito ha logrado. De acuerdo a las exigencias del cliente se utiliza MySQL.

### PostgreSQL.

Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2. Soporta tanto la programación orientada a objetos como la funcional. Las funciones tienen la particularidad de poder ejecutarse con los privilegios del usuario ejecutor o con los privilegios de un usuario definido previamente lo que supone un alto nivel de seguridad. Estas funciones son referidas en otros SGBD como procedimientos almacenados (stored procedures). PostgreSQL tiene la extraordinaria potencialidad de permitir que mientras un proceso escribe en una tabla, otros accedan a la misma sin necesidad de bloqueos esto es posible gracias a un sistema denominado **Acceso Concurrente Multiversión (MVCC)**. [27].

Este gestor se identifica además por:

- Disparadores (triggers).
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.

### **¿Por qué se utilizará MySQL?**

Aquí conviene resaltar que la elección de uno u otro SGBD vendrá dada por las necesidades específicas del Web que se quiera implementar, y en muchos casos será una decisión subjetiva. A fecha de hoy, la versión libre de InterBase aún ofrece un rendimiento pobre en entornos con gran número de usuarios concurrentes por lo que las opciones principales son MySQL y PostgreSQL; la diferencia fundamental entre ambos SGBDs es la implementación de SQL y la velocidad, si se necesita rapidez y características sencillas la elección es MySQL. También es un sistema de administración relacional de bases de datos, donde la base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server.

### **1.9 Servidor de aplicaciones Web Apache.**

El servidor HTTP Apache es un software gratuito de código fuente abierto que funciona sobre una multitud de sistemas operativos como: Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementan el protocolo HTTP/1.1. Es un servidor altamente configurable de diseño modular, es muy sencillo ampliar sus capacidades. Actualmente existen muchos módulos para Apache que son adaptables a este. Apache trabaja con PHP y otros lenguajes de script. También trabaja con Java. Teniendo todo el soporte que se necesita para tener páginas dinámicas. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. [15].

### **1.10 Metodologías de desarrollo de software.**

Un proceso de desarrollo de software detallado y completo suele denominarse "Metodología". Sin embargo existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Vale destacar que todo desarrollo de software es riesgoso y difícil de controlar,

por lo que es necesario aplicar una metodología que permita obtener como resultado un producto de calidad. Una definición acertada, en correspondencia con el objetivo de la investigación es la abordada por Maddison donde define metodología como “Conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas de Información”. Las metodologías guían el proceso de desarrollo y la clave del éxito de un proyecto de software es la elección correcta de la metodología, que puede conducir al programador a desarrollar un buen sistema de software, logrando el crecimiento de la calidad y la productividad del producto. La elección de la metodología adecuada es más importante que utilizar las mejores y más potentes herramientas. [16].

### **eXtreme Programming (XP).**

Es la más destacada de los procesos ágiles de desarrollo de software. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.[16].

Las características fundamentales del método son:

- Desarrollo iterativo e incremental, pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas, las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que calidad del código escrito de esta manera es mayor (el código es revisado y discutido mientras se escribe) esto es más importante que la posible pérdida de productividad inmediata.
- Frecuente interacción del equipo de programación con el cliente o usuario, un representante del cliente trabaja junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Obliga a realizar entregas frecuentes.

### **Rational Unified Process (RUP).**

El Proceso Racional Unificado o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado. Sus principales características son:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

Inicio: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.

Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.

Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.

Transición: se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados. [8].

### 1.11 Lenguaje de Modelado.

El Lenguaje Unificado de Modelado (**UML**) prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas:

1. Diagramas de Casos de Uso para modelar los procesos.
2. Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
3. Diagramas de Colaboración para modelar interacciones entre objetos.
4. Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
5. Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
6. Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
7. Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
8. Diagramas de Componentes para modelar componentes.
9. Diagramas de Implementación para modelar la distribución del sistema. [17].

### 1.12 Herramientas CASE.

Las Herramientas **CASE** son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otros.

### **Visual Paradigm.**

Es una herramienta Case para UML, de fácil de uso y completa, con soporte multiplataforma, posibilita una alta interoperabilidad con otras aplicaciones. Apoya un gran número de idiomas en la generación de código y la ingeniería inversa en Java, PHP, entre otros. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al momento, creando de forma simple toda la documentación. Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso de la aproximación orientada a objeto. Además brinda la posibilidad de documentar todo el trabajo, en formato PDF o en forma de un sitio WEB, sin necesidad de utilizar herramientas externas. Una vez que los diagramas han sido confeccionados, es posible generar documentos que combinan texto e imágenes obtenidas a partir de los diagramas de manera muy sencilla, incluso Visual Paradigm genera toda la documentación de lo que se ha hecho hasta el momento, cumpliendo con estándares establecidos. Brinda también la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir del código. Entre los lenguajes conocidos por Visual Paradigm para la ingeniería inversa, se encuentran los siguientes:

- Java Source.
- C++ Source.
- PHP 5.0 Source.
- XML. [19].

### **1.13 Herramientas de desarrollo.**

#### **NuSphere PHPEd.**

Es un potente editor de PHP tanto para personas experimentadas como para principiantes, con soporte para múltiples formatos. Resalta los distintos tags con diferentes colores para hacer mucho más fácil la programación. Esta herramienta incluye un cliente de FTP (File Transfer Protocol) y un servidor Web integrados, que se configuran según las necesidades del trabajo que se vaya a realizar.

#### **Zend Studio (Zend Development Environment).**

Este IDE, acrónimo de Integrated Development Environment es una de las herramientas más potente de programación para el lenguaje PHP. Cuenta con analizadores de código, permite completamiento de código entre otras ventajas. 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. Lo más destacable es que contiene una ayuda contextual con todas las librerías

de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. El hecho de que esté desarrollado sobre Java da la ventaja de que puede ser utilizado en cualquier sistema operativo. [2].

### **1.14 Conclusiones.**

En este capítulo se analizan y fundamentan los conceptos, tecnologías y metodología que son utilizadas para el desarrollo del sistema propuesto. Después de este análisis y la fundamentación realizada se definió una arquitectura en tres capas según el modelo Cliente Servidor, implementando el patrón Modelo Vista Controlador, se define RUP (*Proceso de desarrollo unificado*) como metodología de desarrollo, el lenguaje de programación que se utilizará es PHP y JavaScript para el control de las diferentes funciones a realizar en el cliente, como gestor de base de datos se estableció el MySQL. Como herramienta case que se usará como entorno de creación de diagramas para UML el Visual Paradigm, y como lenguaje de modelado UML. Una vez conocidas las herramientas óptimas y los conceptos a utilizar se puede empezar a desarrollar la propuesta de sistema.

### **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.**

#### **2.1 Introducción.**

El presente capítulo brinda una descripción general del objeto de estudio, los principales problemas existentes en la Cátedra de ajedrez de la UCI al no contar con una aplicación que gestione el juego de ajedrez por correspondencia dando así una visión de lo que funcionalmente debe realizar el sistema teniendo en cuenta esto se realiza un análisis de ellos. Obteniendo el modelo de negocio, de ahí surgen los requisitos que deberá tener y cumplir el sistema para satisfacer las necesidades de la institución; a partir de estos, se obtiene el diagrama de casos de uso del sistema.

A continuación se explican los artefactos mencionados anteriormente como resultado de los flujos de trabajo modelamiento del negocio y captura de requisitos.

#### **2.2 Situación Problemática.**

Nuestra universidad cuenta con una cátedra de ajedrez donde se imparten clases de ajedrez y se realizan torneos a nivel UCI, así como también a sido sede de torneos nacionales, para estos torneos es necesario asistir de forma presencial y esto requiere de tiempo, por esta situación algunos usuarios de este deporte no pueden asistir por causa de trabajo o estudio, otro problema que existe es que la modalidad del ajedrez, el ajedrez por correspondencia no se juega en la UCI. Dada la situación anterior surge en la Universidad la necesidad de desarrollar un sistema para automatizar el juego de este atractivo y universal deporte y en especial la modalidad del ajedrez por correspondencia, con este objetivo surge el proyecto Infodrez. Para que Infodrez opere correctamente se hace imprescindible la realización del módulo Ajedrez por Correspondencia donde los amantes de este juego de ajedrez por email cuenten con una aplicación Web donde puedan registrarse y ser usuarios de la misma y que desde sus ordenadores puedan acceder a la aplicación haciéndoles que sus partidas sean mas amenas y placenteras, además que se sientan mas motivado por este deporte.

#### **2.3 Propuesta del Sistema.**

La aplicación que se propone facilitará la gestión de torneos de ajedrez por correspondencia. Mostrará una interfaz gráfica amigable y fácil de comprender a los usuarios, contendrá un menú que permite que el trabajo sea más organizado. El sistema requiere autenticación, y en caso de que no estén registrados dará la posibilidad de registrarse en la aplicación. Ya una vez autenticado los permisos dependen del rol asignado, contribuyendo a la seguridad. La aplicación permitirá al Organizador del Torneo crear, editar y eliminar Torneos mostrando una interfaz grafica con un formulario donde



muestre los datos del torneo, este en su rol tendrá los permisos necesarios para la gestión de los torneos del sistema. También contara con una interfaz para mostrar todos los torneos que estén disponibles en el sistema dándole la posibilidad al actor principal del sistema el jugador de poder inscribirse en el torneo deseado y comenzar a jugar esta modalidad del ajedrez mediante un visor PGN que la interfaz de jugar partida contara, así como cada jugada realizada por el jugador se mostrara en esa misma interfaz al lado del visor PGN y además esta jugada será enviada por Email al jugador oponente junto con las jugadas efectuadas anteriormente ,el sistema también le permitirá al jugador de Solicitar un torneo en especifico, ver todas sus partidas en el torneo, además contara con un tablero de análisis dándole la posibilidad al jugador de ejercitar posibles jugadas, contará con la opción de imprimir cualquiera da sus partidas, también esta aplicación tendrá una ayuda donde explique el funcionamiento general del sitio, otra de las posibilidades del jugador será la de solicitar licencia ,donde el jugador solicita la licencia al Director del torneo que sería el Arbitro principal del torneo el cual tendrá los permisos de gestionar los jugadores registrados en el sistema así como concederle la licencia a los jugadores y definir las partidas no concluidas en el Torneo. El Sistema también contara con la conexión a un servidor de correo ya que las jugadas realizadas se enviaran por correo así como también la solicitud de la licencia y las notificaciones que se realizan en la aplicación.

### **2.4 Modelado del Negocio.**

El modelo de negocio es un artefacto de la disciplina de ingeniería del software que tiene como objetivo conocer la estructura y dinámica de la organización en el cual se va a implantar el sistema, comprender los problemas actuales e identificar las posibles mejoras. Además asegurar que el equipo de proyecto, los usuarios finales, tengan un entendimiento común de la organización; nos da una visión de qué es lo necesario hacer para satisfacer las exigencias de los usuarios. Teniendo en cuenta que en el flujo de trabajo de modelamiento de negocio se generan artefactos, a continuación se hace referencia a los mismos.

## 2.5 Actores y Trabajadores del Negocio.

Es cualquier individuo, organización, grupo o máquina que interactúa con el negocio, son los que se benefician con los procesos del negocio. Considerando lo anterior se determinaron los siguientes actores el presente negocio.

**Tabla 1: Descripción de los actores del negocio.**

Actores del negocio	Justificación
Encargado de Reportar resultados del torneo	Es el encargado de reportar un informe con los resultados del torneo a la federación.
Federación	Es la organización rectora de todos los eventos relacionados con el ajedrez a la cual se le entregan todos los reportes oficiales, gestiona el rating de los jugadores y la encargada de tomar la decisión de Organizar un torneo.
Jugador	Son los protagonistas en una partida de ajedrez, en este caso en particular una partida por correspondencia, donde tiene el derecho de solicitar una licencia así como solicitar un torneo específico.

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio.

Tabla 2: Descripción de los trabajadores del negocio.

Trabajadores del negocio	Justificación
Director del Torneo (Arbitro Principal)	Es la persona encargada de dirigir y arbitrar el torneo, así como conceder licencias solicitadas por los jugadores, actualizar la tabla de resultado del torneo, confecciona listado de ganadores y notifica las posiciones del torneo. También decide partidas no concluidas.
Encargado del rating	Este se encarga de calcular y actualizar el rating del torneo, una vez estas actividades realizadas realiza un informe con el rating actualizado de cada jugador seguidamente de esto envía informe actualizado con el rating a los jugadores del torneo.
Organizador del Torneo	Es el encargado de organizar un torneo determinado creando las bases del mismo eligiendo al director del torneo y conformando listado de jugadores, además realiza el pareo del torneo.
Encargado de Publicar los Resultados del Torneo	Este recibe el informe con los resultados del torneo y publica los mismos en un lugar destinado por la federación.
Encargado de las solicitudes de torneos	Es el encargado de analizar los motivos de la licencia solicitada por el jugador y así tiene la autoridad de concedérsela o no.

### 2.6 Diagrama de Casos de Uso del Negocio.

Un diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del mismo.

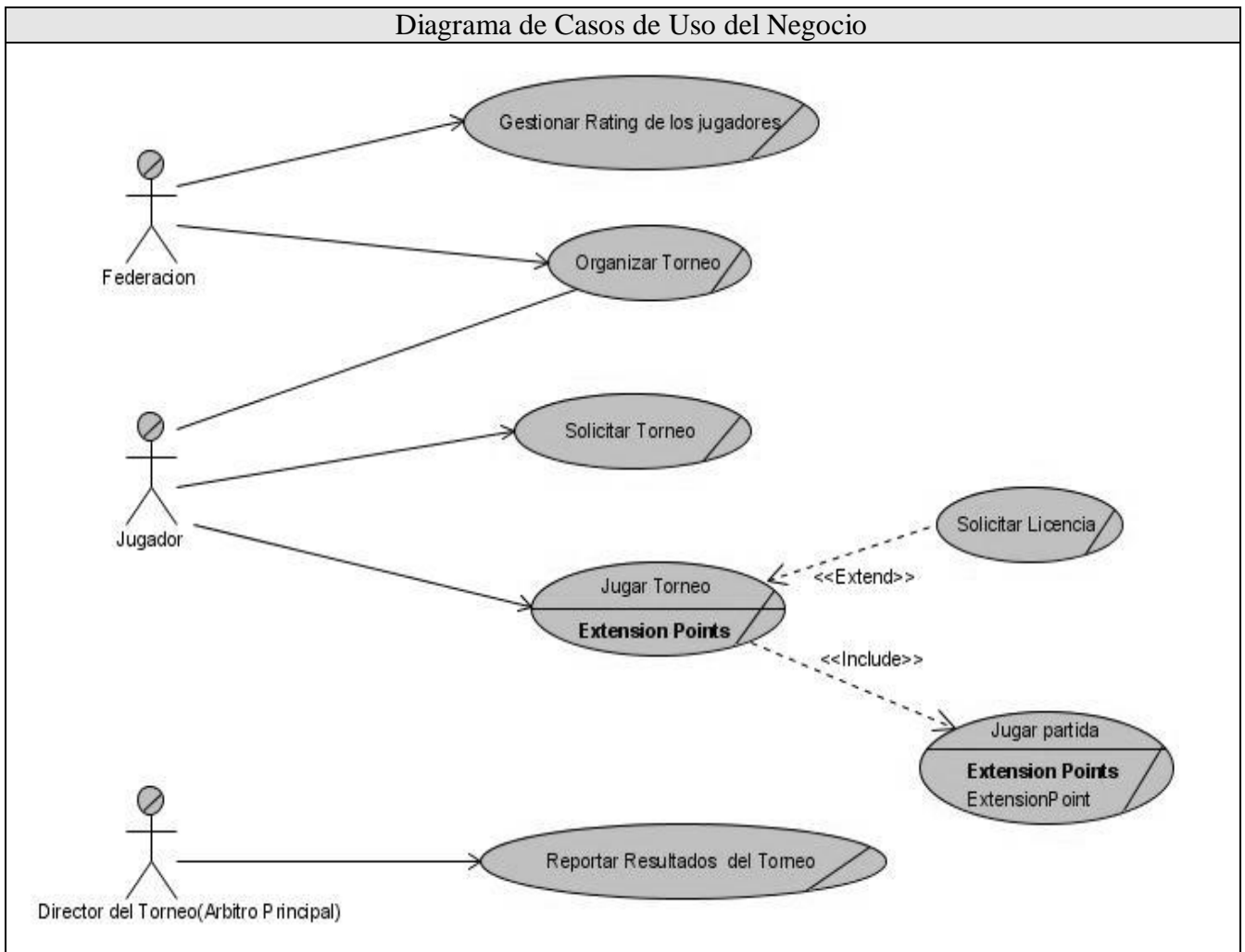


Figura 1: Diagrama de Casos de Uso del Negocio.

### 2.7 Descripciones Textuales.

Tabla 3: Descripción textual del caso de uso del negocio “Organizar Torneo”.

Caso de Uso	Organizar Torneo
<b>Actores</b>	Federación(inicia), Jugador
<b>Propósito</b>	Organizar un torneo de ajedrez por correspondencia con el con el mejor nivel técnico como organizativo posible.
<b>Resumen</b>	El CUN se inicia cuando la federación decide organizar un torneo y convoca a

	<p>un organizador para que organice el torneo, este a su vez crea las bases de dicho torneo así como todos los datos que requiere, elige al director del torneo que será a su vez el arbitro principal, confecciona la lista de jugadores que se encontraban en espera, lanza una convocatoria a los mismos, los jugadores que acepten esta convocatoria serán inscritos en el torneo y se realizara el listado de jugadores confirmados. Ya confeccionado el listado de jugadores confirmados el organizador del torneo es el encargado de realizar el pareo conformando así la tabla de pareo, auto seguido a esto envía tabla de pareo a la federación y esta se encarga de enviarle la tabla de pareo y todos los datos que requiere este torneo en especifico así como el calendario del torneo a los jugadores. Los jugadores reciben todos estos datos.</p>
<b>Curso normal de los eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>
1. La Federación decide organizar un torneo.	
.	2. Organizador del Torneo crea las bases del torneo.
	3. Organizador del Torneo elige al director del torneo (Arbitro Principal).
	4. Organizador del Torneo crea listado de jugadores.
	5. Organizador del Torneo convoca a los jugadores
6. El jugador recibe convocatoria.	
7. El jugador acepta convocatoria.	
8. El jugador envía confirmación.	
	9. Organizador del Torneo recibe confirmación
	10. Organizador del Torneo crea un listado con los Jugadores confirmados.

	11. Organizador del Torneo organiza Pareo.
	12. Envía Tabla de Pareo a la federación
13. Federación recibe Tabla de Pareo	
14. Federación envía Tabla de Pareo y calendario a los jugadores	
15. El jugador recibe Tabla de Pareo y calendario	
<b>Flujo Alternativo</b>	
7. El jugador no acepte convocatoria y termina el CUN.	

Tabla 4: Descripción textual del caso de uso del negocio “Jugar Torneo”.

Caso de Uso		Jugar Torneo
<b>Actores</b>	Jugador(inicia)	
<b>Propósito</b>	El jugador juegue todas las partidas del torneo, así habrá jugado satisfactoriamente el Torneo.	
<b>Resumen</b>	El CUN se inicia cuando el jugador juega una partidas del torneo y envía una notificación al director del torneo que en este caso será el arbitro principal del torneo, este a su vez comprueba si es la ultima partida del jugador, si es así actualiza la tabla de resultados y confecciona listado de ganadores, notifica y envía las posiciones del torneo a los jugadores. Los jugadores reciben la notificación de posiciones.	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>	
1. El jugador juega una Partida.		
2. El jugador envía Notificación al Director del Torneo.		

	3. Director del Torneo Recibe Notificación.
	4. Si es última partida el Director del Torneo actualiza Tabla de Resultados.
	5. Director del Torneo confecciona listado de ganadores.
	6. Director del Torneo notifica las posiciones del torneo
7.El jugador recibe la notificación de posiciones.	
<b>Flujo Alternativo</b>	
	4. Si no es la última partida el Director del Torneo actualiza Tabla de Resultados.
	6. Director del Torneo notifica las posiciones del torneo
7.EL jugador Recibe la notificación de las posiciones.	

**Tabla 5: Descripción textual del caso de uso del negocio “Solicitar Torneo”.**

Caso de Uso		Solicitar Torneo
<b>Actores</b>	Jugador(inicia)	
<b>Propósito</b>	El jugador tenga la posibilidad de solicitar un torneo específico de su preferencia.	
<b>Resumen</b>	El CUN se inicia cuando el jugador solicita a la federación un tipo de torneo específico en el que quiere participar, la federación asigna al encargado de las solicitudes de torneos a esta actividad, este recibe la solicitud si existe el torneo solicitado en la federación se le envía convocatoria al Jugador, el jugador recibe la convocatoria de la Federación y entra al torneo.	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>	

1. El jugador solicita a la federación un tipo de torneo específico en el que quiere participar.	
	2. Encargado de las solicitudes de torneos recibe la solicitud del jugador.
	3. Si existe el torneo solicitado el encargado de las solicitudes de torneos envía Convocatoria al Jugador.
4. El jugador recibe Convocatoria de la Federación.	
5. El jugador entra al torneo.	
<b>Flujo Alternativo</b>	
	3. Si no existe el torneo solicitado el encargado de las solicitudes de torneos coloca en el listado de jugadores en espera.
	4. Envía notificación de que no hay Torneo solicitado
5. Recibe notificación.	

Tabla 6: Descripción textual del caso de uso del negocio “Reportar Resultados del Torneo”.

Caso de Uso		Reportar Resultados del Torneo
<b>Actores</b>	Encargado de Reportar Resultados de Torneo (inicia).	
<b>Propósito</b>	Reportar los resultados de un torneo terminado.	
<b>Resumen</b>	El CUN se inicia cuando Encargado de Reportar Resultados de Torneo recibe los resultados del torneo y envía un informe con los resultados del torneo a la Federación, es aquí donde Encargado de Publicar los Resultados del Torneo de la federación recibe el informe con los resultados del torneo y publica los resultados del torneo.	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>	



1. Encargado de Reportar Resultados de Torneo recibe los resultados del torneo.	
2. Encargado de Reportar Resultados de Torneo envía un informe con los resultados del torneo a la Federación.	
	3. Encargado de Publicar los Resultados del Torneo recibe el informe con los resultados del torneo.
	4. Encargado de Publicar los Resultados del Torneo publica los resultados del torneo.

**Tabla 7: Descripción textual del caso de uso del negocio “Gestionar Rating de los Jugadores”**

<b>Caso de Uso</b>		<b>Gestionar Rating de los Jugadores</b>
<b>Actores</b>	Federación (inicia).	
<b>Propósito</b>	Gestionar el rating de los jugadores una vez concluido el torneo.	
<b>Resumen</b>	El CUN se inicia cuando la Federación recibe los resultados del torneo y le entrega resultados del torneo al encargado del rating de los jugadores, este calcula el rating de los jugadores del torneo ya calculado el rating actualiza el rating de los jugadores del torneo es aquí donde realiza un informe actualizado con el rating de los jugadores del torneo, también otras de las actividades que realiza el encargado del rating en la federación es enviar el rating actualizado correspondiente a cada jugador del torneo, además publica el rating actualizado de los jugadores del torneo.	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>	
1. Federación recibe los resultados del torneo.		
2. Federación entrega resultados del torneo al encargado del rating de los jugadores.		
	3. Encargado del rating calcula el rating de los jugadores del torneo.	

	4. Encargado del rating actualiza el rating de los jugadores del torneo.
	5. Encargado del rating realiza un informe actualizado con el rating de los jugadores del torneo.
	6. Encargado del rating envía el rating actualizado correspondiente a cada jugador del torneo.
	7. Encargado del rating Publica el rating actualizado de los jugadores del torneo.

**Tabla 8: Descripción textual del caso de uso del negocio “Solicitar Licencia”**

<b>Caso de Uso</b>		<b>Solicitar Licencia</b>
<b>Actores</b>	Jugador(inicia)	
<b>Propósito</b>	El jugador tenga la posibilidad de solicitar licencia de descanso en el torneo.	
<b>Resumen</b>	El CUN se inicia cuando el jugador envía una solicitud de licencia al director del torneo (Arbitro Principal), el arbitro principal analiza y decide si le concede la licencia al jugador, si el director del torneo decide concederle la licencia coloca el estado del Jugador en licencia y envía decisión de licencia al jugador, el jugador recibe la decisión del Arbitro principal y recesa de jugar el torneo por el tiempo de la licencia.	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>	
1. Jugador envía una solicitud de Licencia.		
	2. Director del torneo (Arbitro Principal) recibe la solicitud de licencia.	
	3. Director del torneo (Arbitro Principal) analiza y decide si le concede la licencia al jugador.	
	4. Director del torneo (Arbitro Principal) si decide concederle la licencia al jugador coloca el estado del jugador en licencia.	

	5. Director del torneo (Arbitro Principal) envía decisión de licencia al jugador.
6. Jugador recibe decisión del director del torneo.	
7. Jugador recesa de jugar el torneo por el tiempo de la licencia.	
<b>Flujo Alternativo</b>	
	4. Director del torneo (Arbitro Principal) si no decide concederle la licencia al jugador.
	5. Director del torneo (Arbitro Principal) envía Negación de Licencia al jugador.
6. Jugador recibe la negación y continúa jugando el torneo.	

Tabla 9: Descripción textual del caso de uso del negocio “Solicitar Licencia”

Caso de Uso		Solicitar Licencia
<b>Actores</b>	Jugador(inicia)	
<b>Propósito</b>	El jugador tenga la posibilidad de solicitar licencia de descanso en el torneo.	
<b>Resumen</b>	El CUN se inicia cuando el jugador envía una solicitud de licencia al director del torneo (Arbitro Principal), el arbitro principal analiza y decide si le concede la licencia al jugador, si el director del torneo decide concederle la licencia coloca el estado del Jugador en licencia y envía decisión de licencia al jugador, el jugador recibe la decisión del Arbitro principal y recesa de jugar el torneo por el tiempo de la licencia.	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>	
1. Jugador envía una solicitud de Licencia.		
	2. Director del torneo (Arbitro Principal) recibe la solicitud de licencia.	

	3. Director del torneo (Arbitro Principal) analiza y decide si le concede la licencia al jugador.
	4. Director del torneo (Arbitro Principal) si decide concederle la licencia al jugador coloca el estado del jugador en licencia.
	5. Director del torneo (Arbitro Principal) envía decisión de licencia al jugador.
6. Jugador recibe decisión del director del torneo.	
7. Jugador recesa de jugar el torneo por el tiempo de la licencia.	
<b>Flujo Alternativo</b>	
	4. Director del torneo (Arbitro Principal) si no decide concederle la licencia al jugador.
	5. Director del torneo (Arbitro Principal) envía Negación de Licencia al jugador.
6. Jugador recibe la negación y continúa jugando el torneo.	

Tabla 10: Descripción textual del caso de uso del negocio “Jugar Partida”

Caso de Uso		Jugar Partida
<b>Actores</b>	Jugador(inicia)	
<b>Propósito</b>	El jugador tenga la posibilidad de solicitar licencia de descanso en el torneo.	
<b>Resumen</b>	El CUN se inicia cuando el jugador envía una solicitud de licencia al director del torneo (Arbitro Principal), el arbitro principal analiza y decide si le concede la licencia al jugador, si el director del torneo decide concederle la licencia coloca el estado del Jugador en licencia y envía decisión de licencia al jugador, el jugador recibe la decisión del Arbitro principal y recesa de jugar el torneo por el tiempo de la licencia.	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>	

1. Jugador envía una solicitud de Licencia.	
	2. Director del torneo (Arbitro Principal) recibe la solicitud de licencia.
	3. Director del torneo (Arbitro Principal) analiza y decide si le concede la licencia al jugador.
	4. Director del torneo (Arbitro Principal) si decide concederle la licencia al jugador coloca el estado del jugador en licencia.
	5. Director del torneo (Arbitro Principal) envía decisión de licencia al jugador.
6. Jugador recibe decisión del director del torneo.	
7. Jugador recesa de jugar el torneo por el tiempo de la licencia.	
<b>Flujo Alternativo</b>	
	4. Director del torneo (Arbitro Principal) si no decide concederle la licencia al jugador.
	5. Director del torneo (Arbitro Principal) envía Negación de Licencia al jugador.
6. Jugador recibe la negación y continúa jugando el torneo.	

## 2.8 Diagramas de Actividades.

Los casos de uso del negocio tienen secuencias de actividades que en conjunto, producen algo para el actor del negocio. La estructura del flujo de actividades se describe gráficamente con la ayuda de un diagrama de actividad, este describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio. Como parte importante dentro de estos diagramas se definen las actividades a automatizar, identificadas con el color más claro.

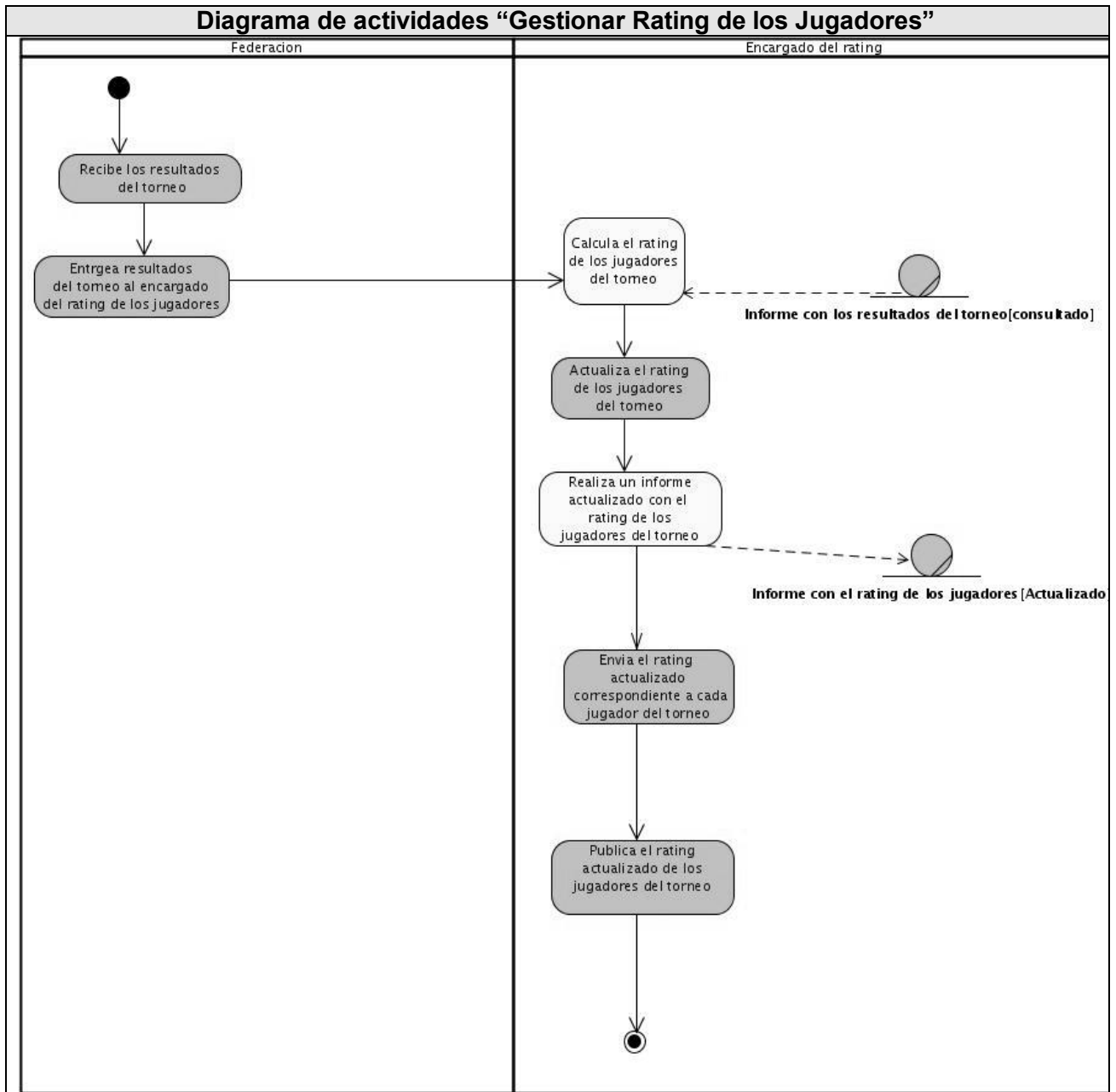


Figura 2: Diagrama de actividades “Gestionar Rating de los Jugadores”.

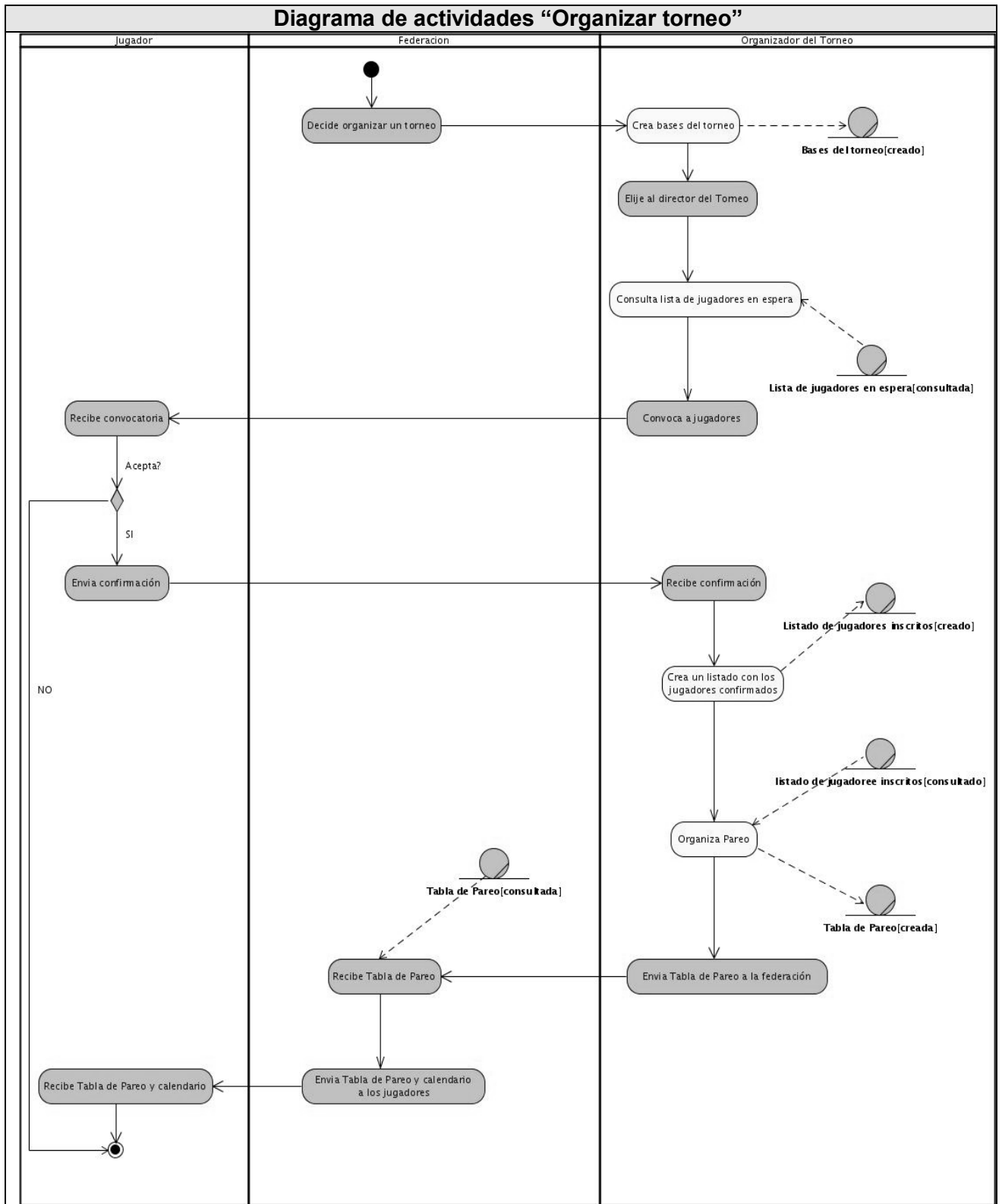


Figura 3: Diagrama de actividades “Organizar torneo”

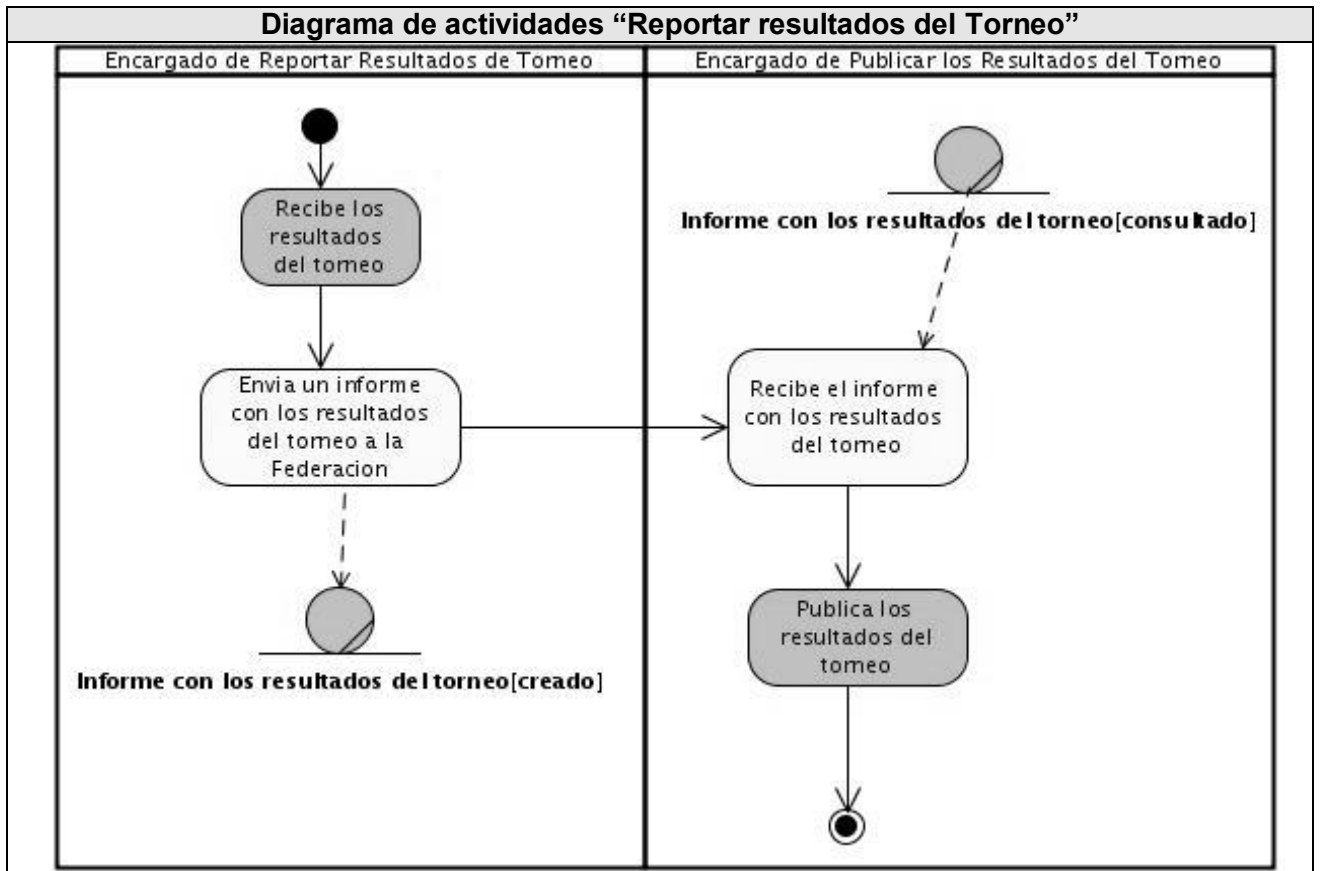


Figura 4: Diagrama de actividades “Reportar resultados del Torneo”



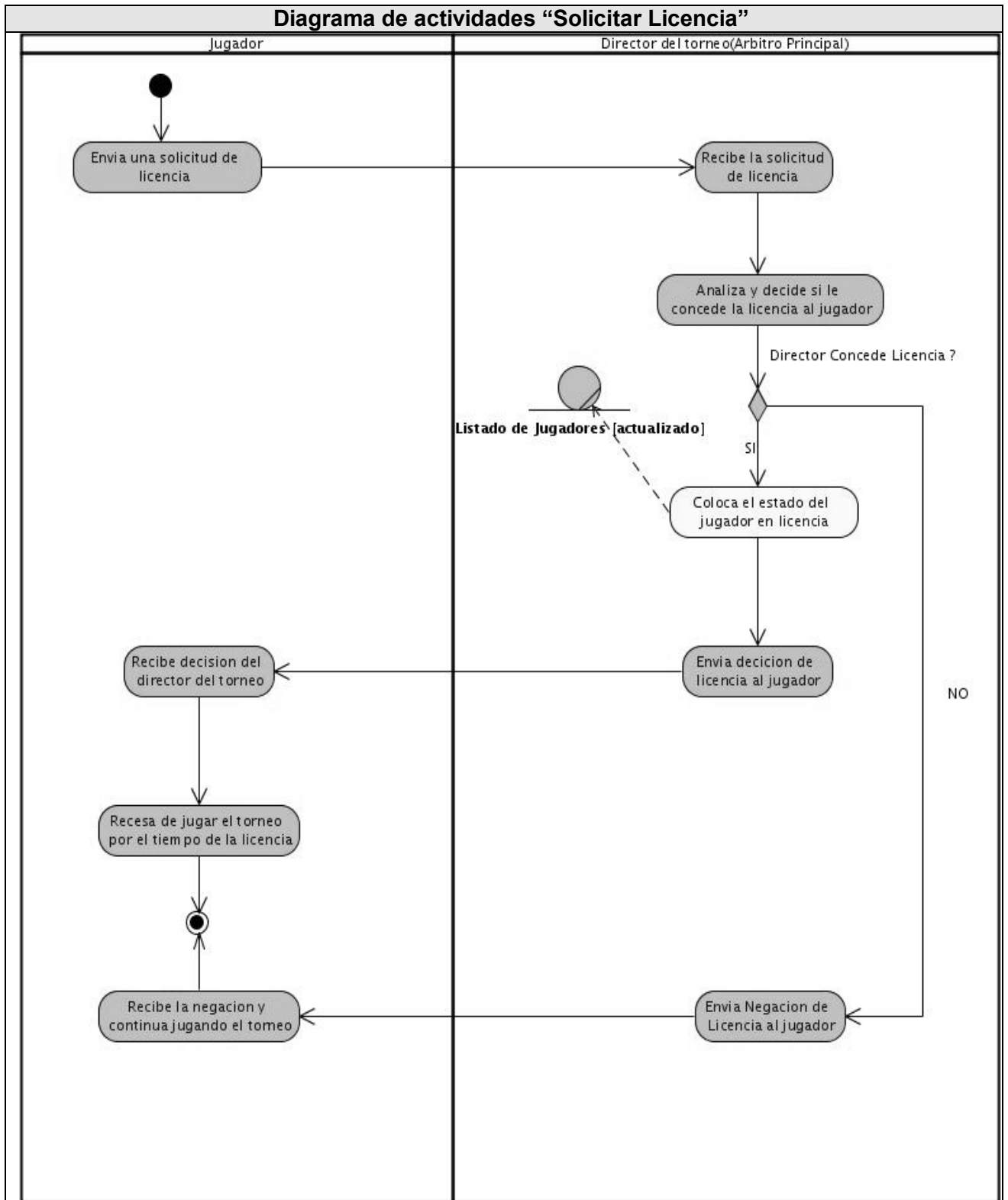


Figura 5: Diagrama de actividades "Solicitar Licencia"

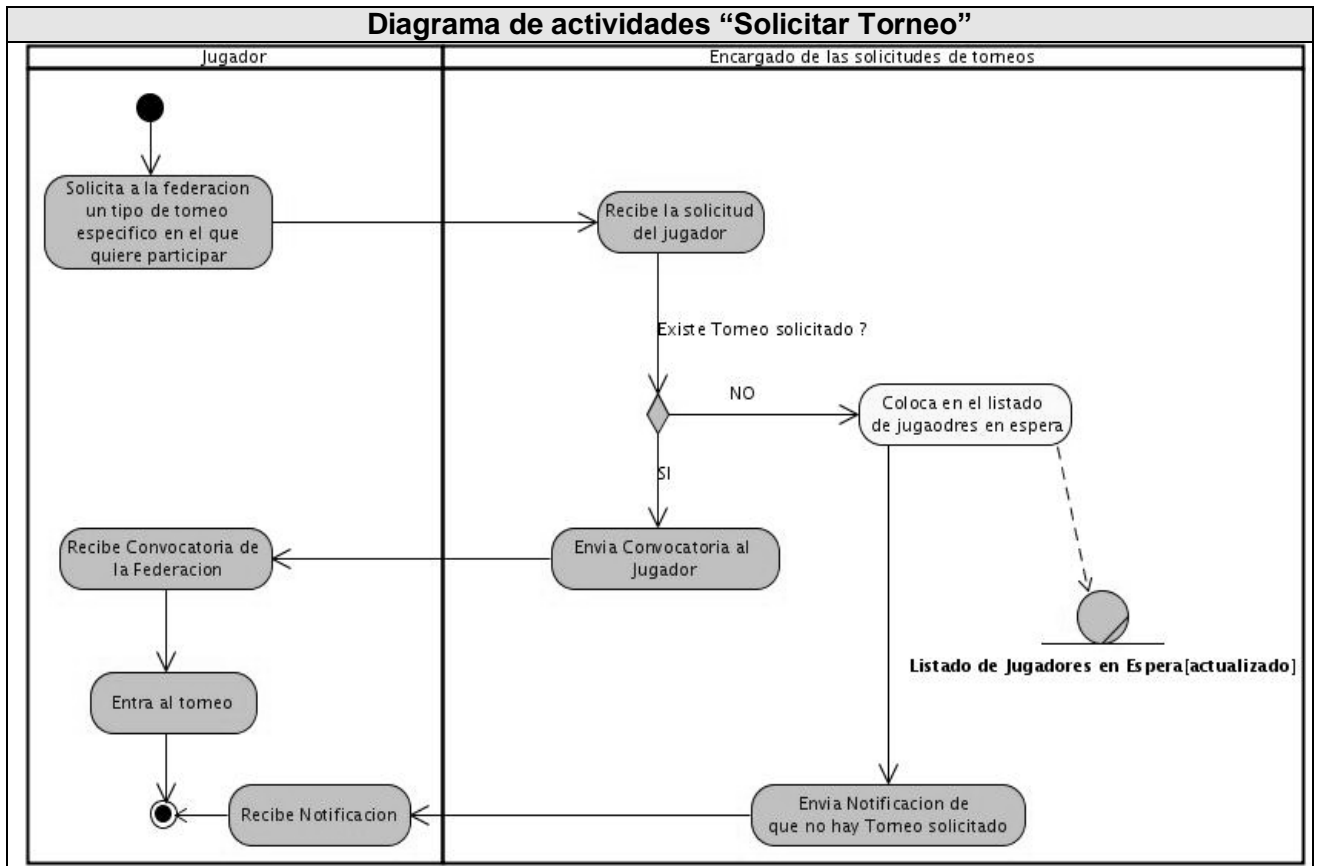


Figura 6: Diagrama de actividades "Solicitar Torneo"

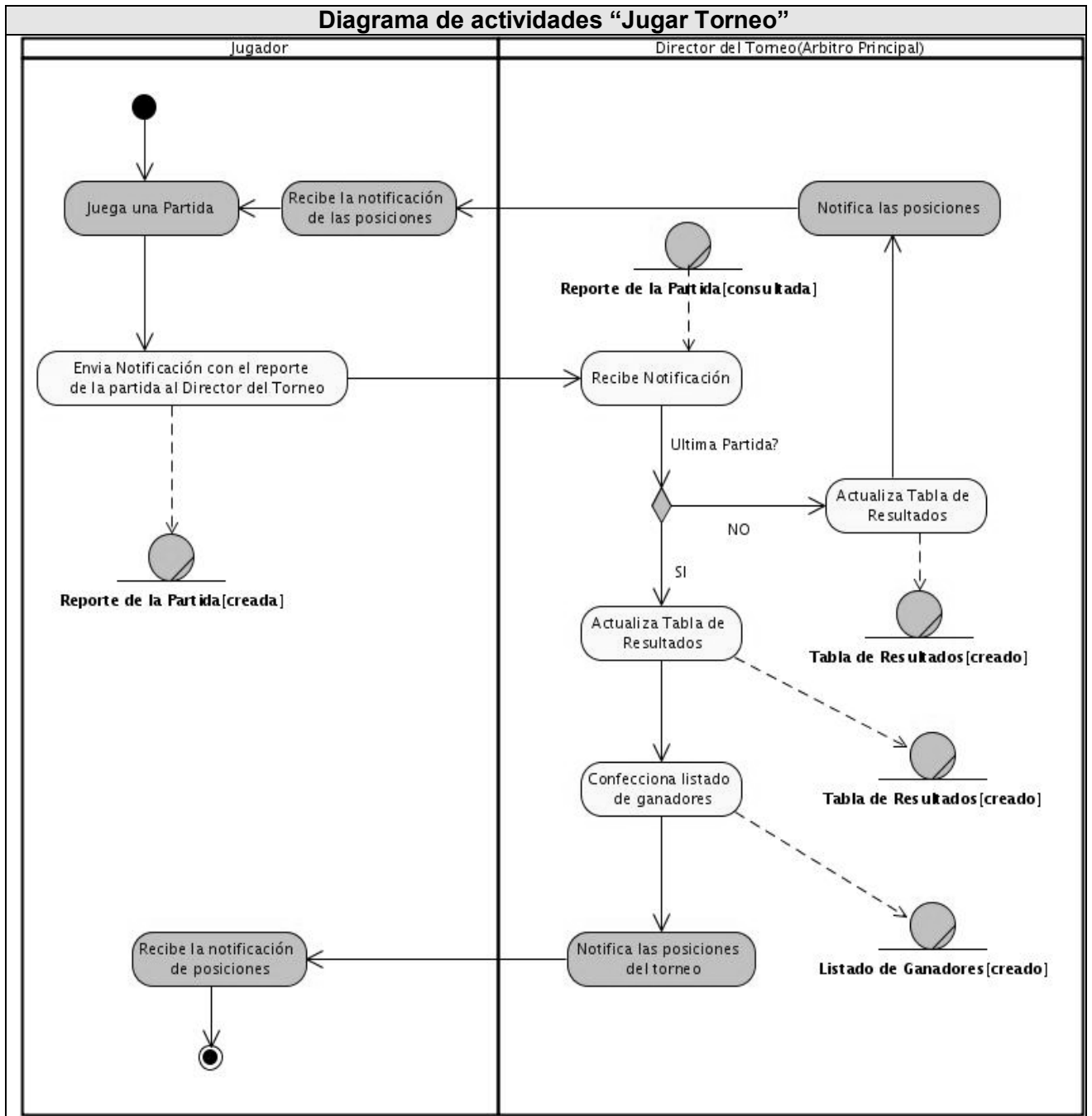


Figura 7: Diagrama de actividades "Jugar Torneo"

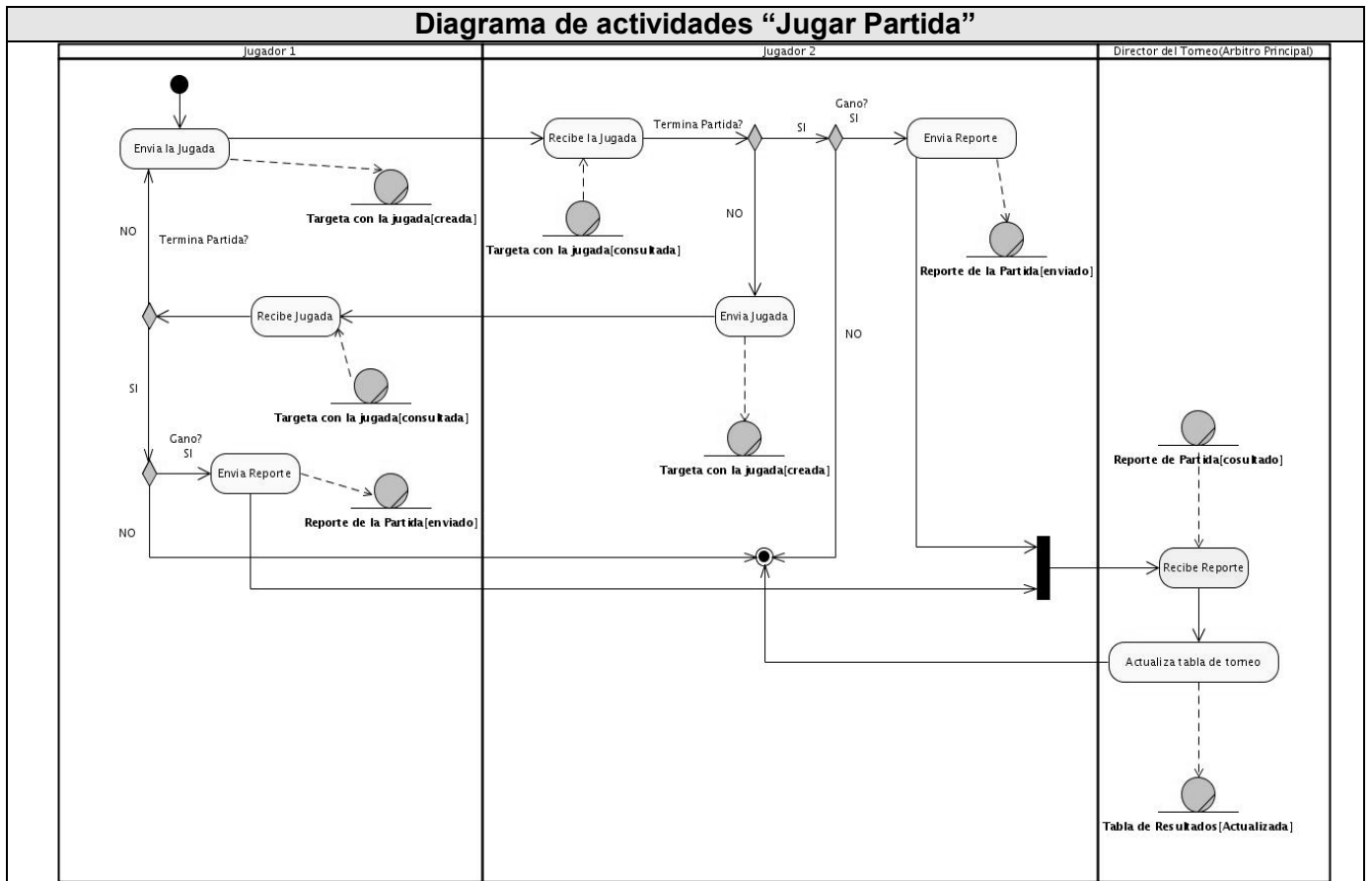


Figura 8: Diagrama de actividades "Jugar Partida"

### 2.9 Diagramas de Clases del Modelo de Objetos.

Este diagrama describe cómo colaboran los trabajadores y las entidades del negocio, y proporciona un acercamiento a la identificación de los futuros actores y entidades del sistema.

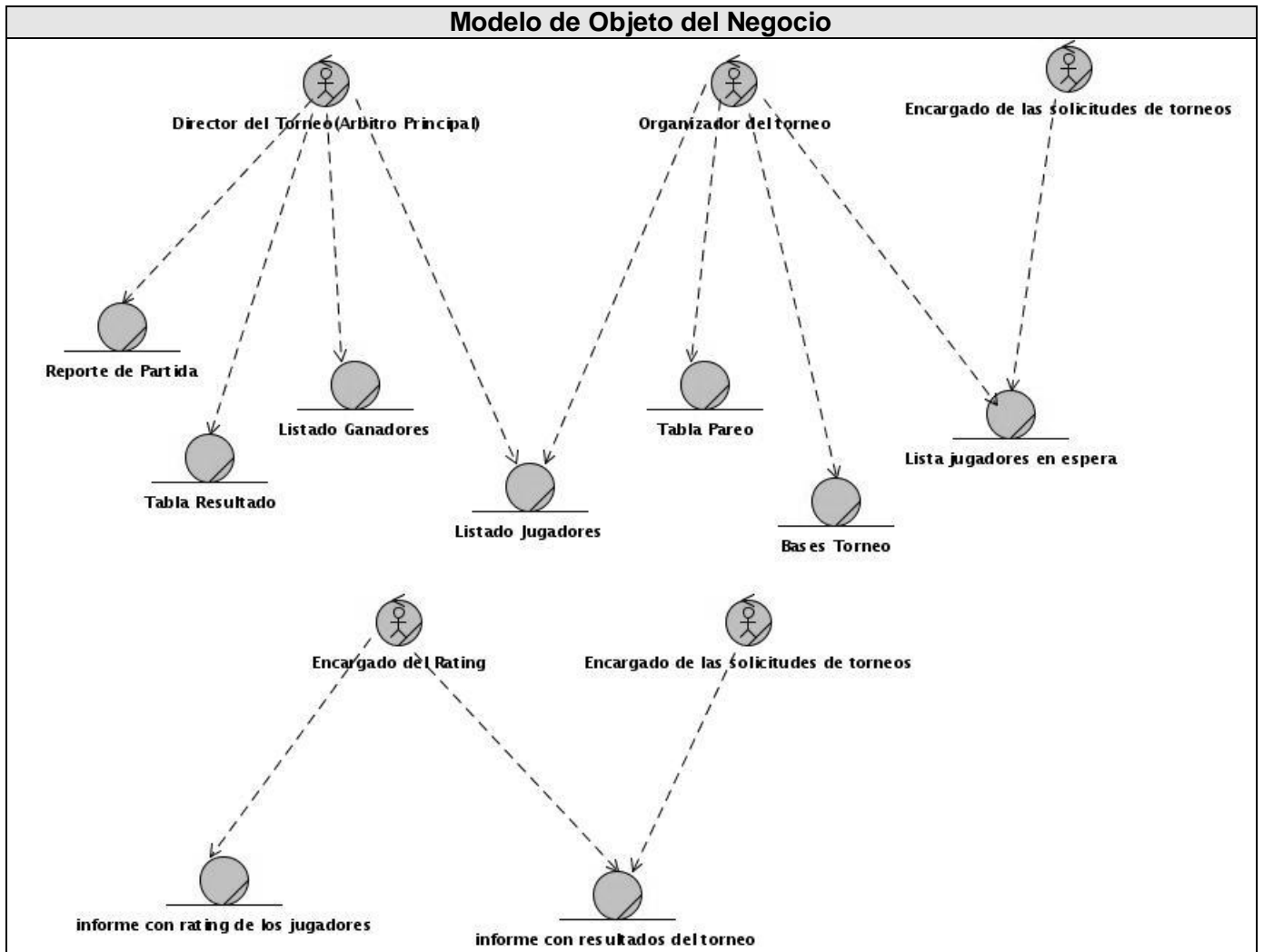


Figura 9: Modelo de Objeto del Negocio

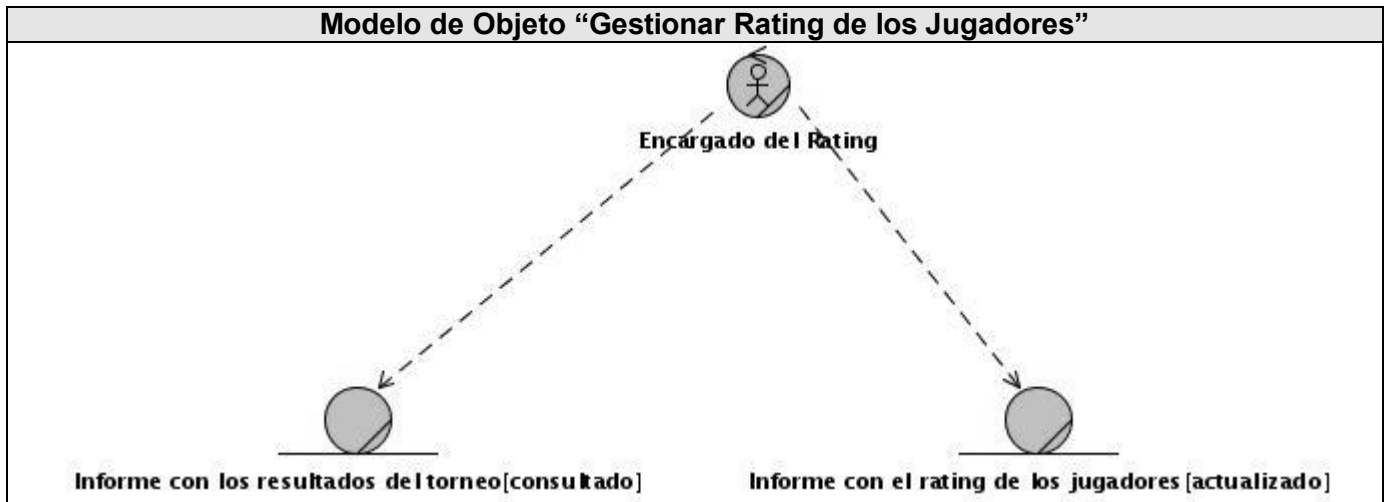


Figura 10: Modelo de Objeto "Gestionar Rating de los Jugadores"

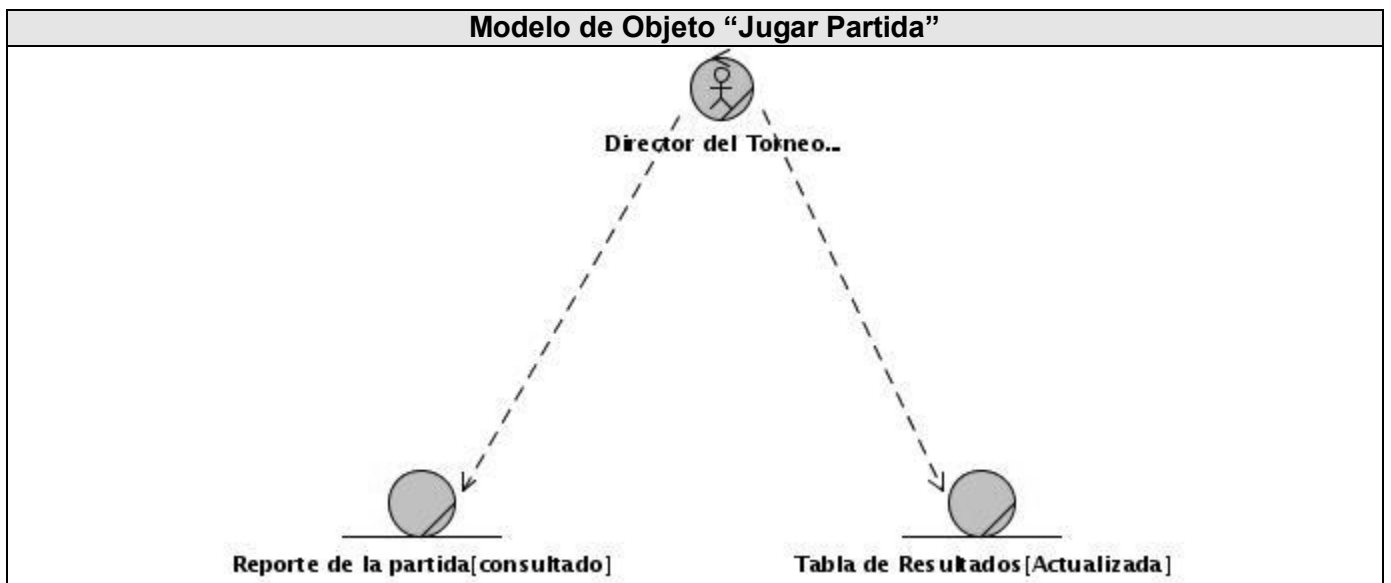


Figura 11: Modelo de Objeto "Jugar Partida"

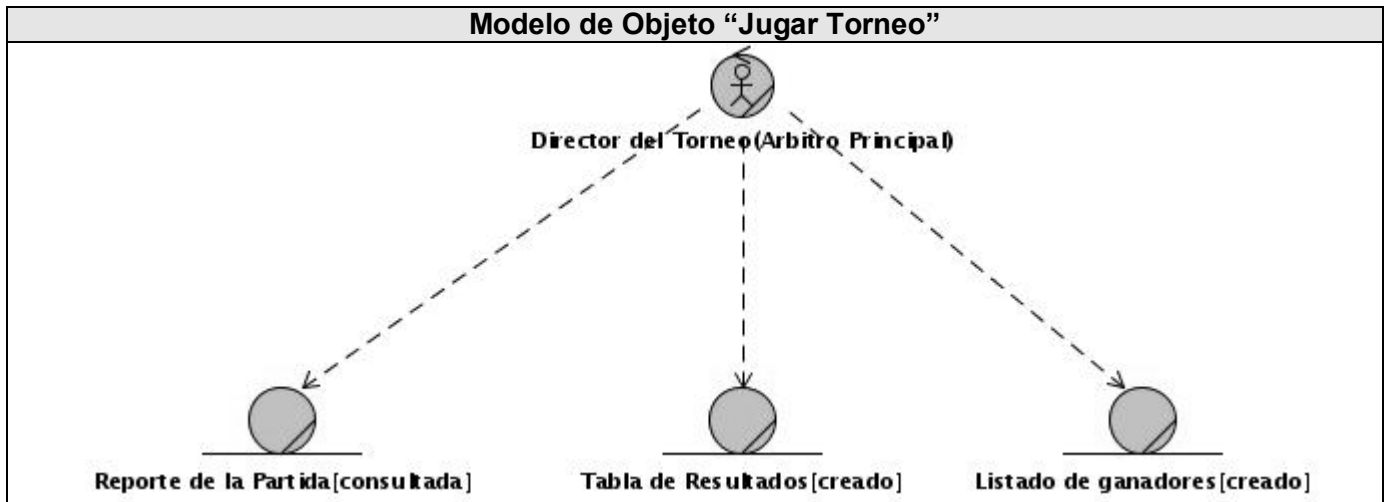


Figura 12: Modelo de Objeto "Jugar Torneo"

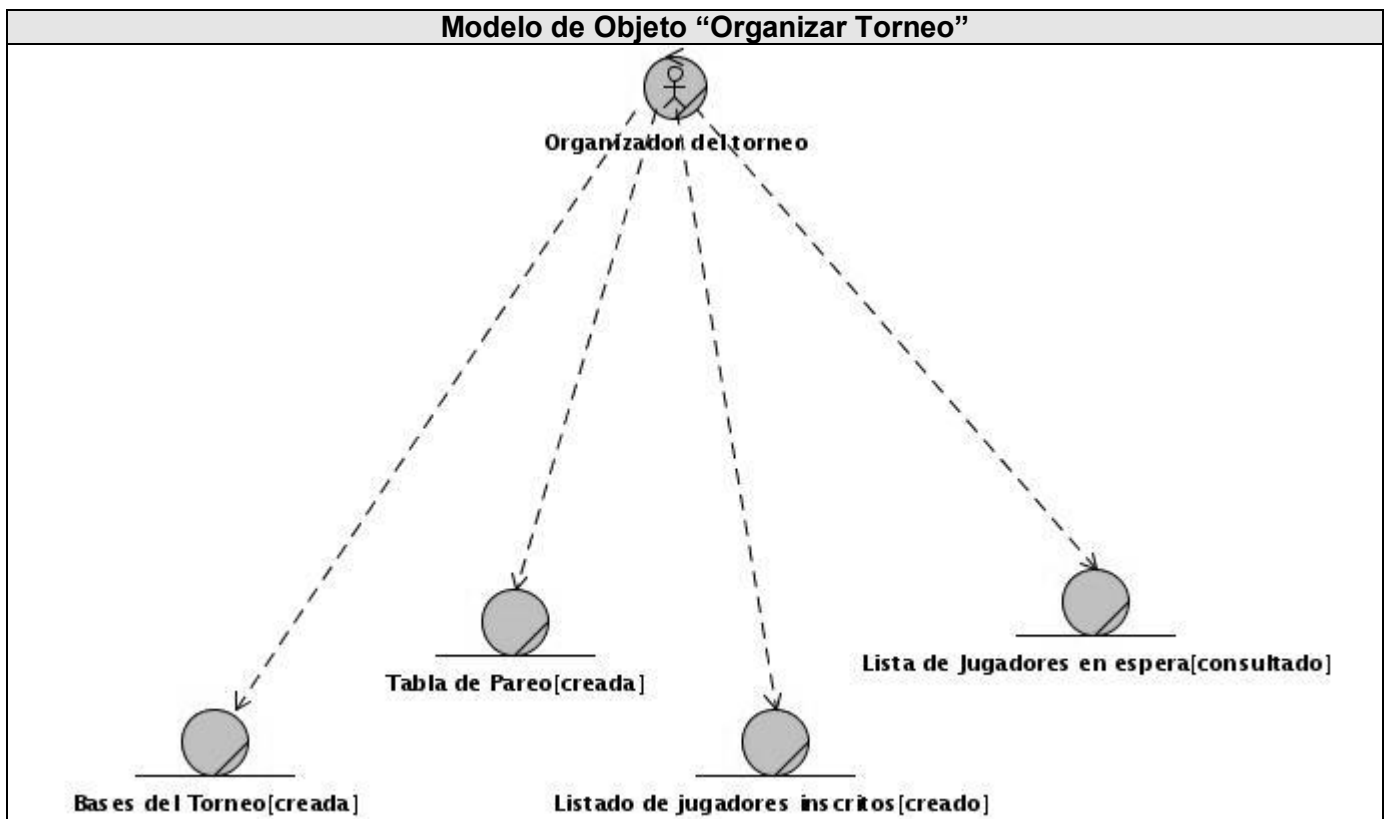


Figura 13: Modelo de Objeto "Organizar Torneo"



Figura 14: Modelo de Objeto “Reportar Resultados del Torneo”

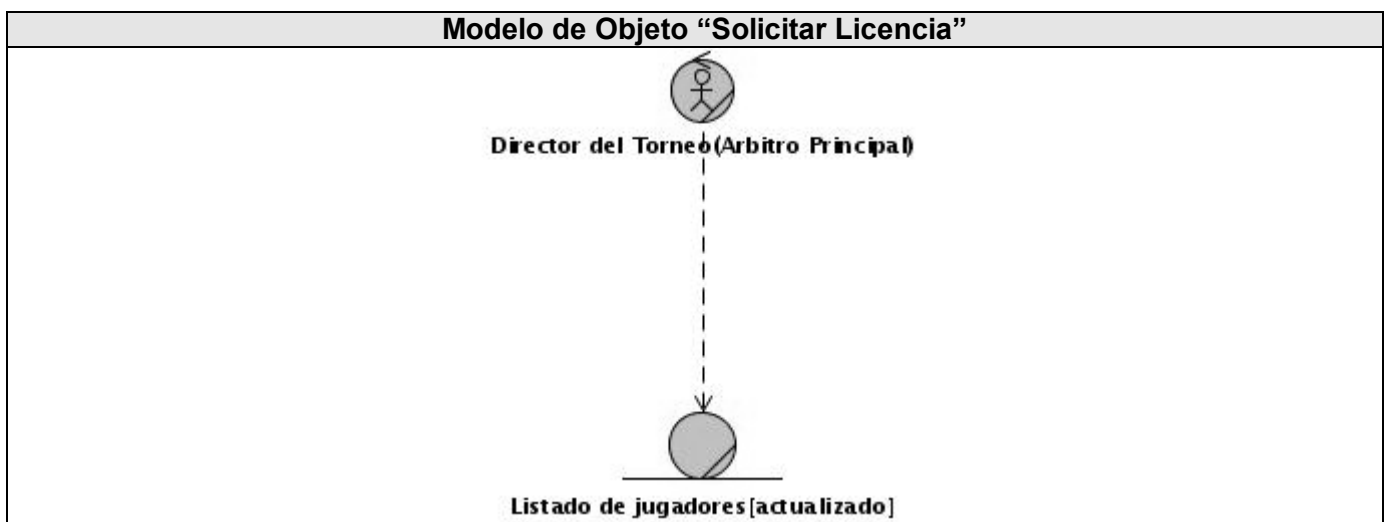


Figura 15: Modelo de Objeto “Solicitar Licencia”



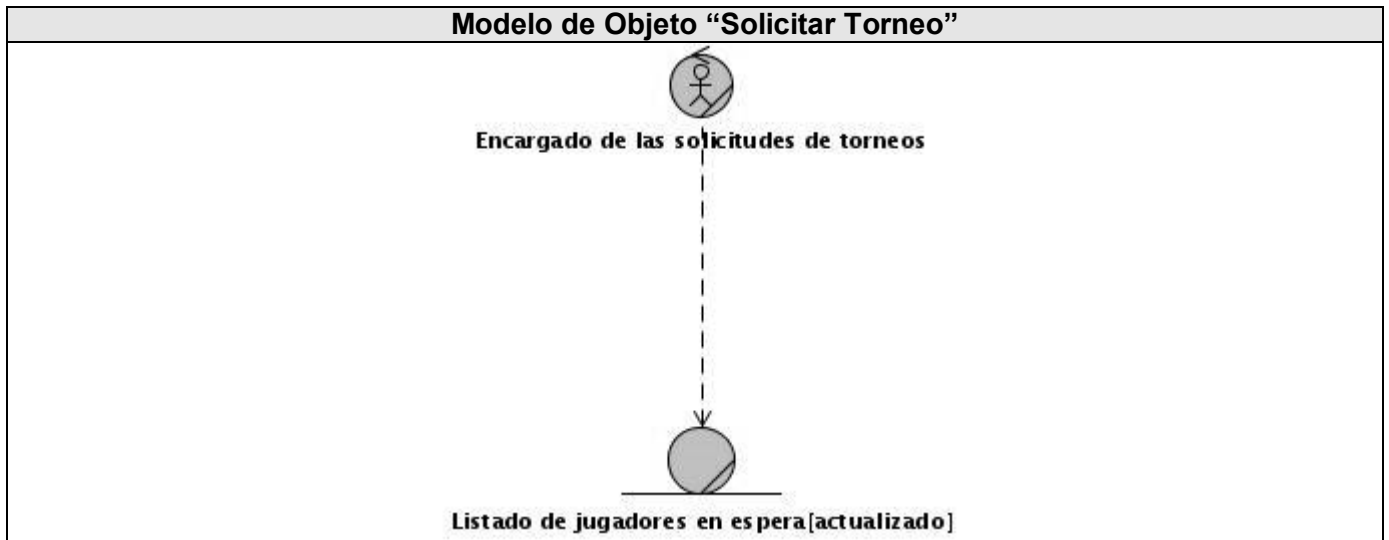


Figura 16: Modelo de Objeto "Solicitar Torneo"

## 2.10 Especificación de los requisitos del sistema

El flujo de trabajo de requisitos tiene como propósito fundamental guiar el desarrollo hacia el sistema correcto. Ello se puede obtener utilizando la descripción de los requisitos del sistema, o sea, condiciones y/o capacidades que el sistema debe cumplir. Las descripciones tienen que ser lo suficientemente buenas como para que se pueda llegar a acuerdos entre los clientes (entre ellos, los futuros usuarios) y los desarrolladores en cuanto a lo que debe y no debe hacer el sistema.

### 2.10.1 Requerimientos funcionales:

Los requerimientos funcionales son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del programa.

Para cumplir con los objetivos propuestos se prevé que el sistema tenga las siguientes funcionalidades:

#### **RF 1 Autenticarse**

RF 1.1 Autenticar usuarios.

*RF 1.2 Cerrar Sesión.*

#### **RF 2 Mostrar Torneos Disponibles**

*RF 2.1 Muestra los Torneos creados.*

**RF 3 Solicitar Torneo**

**RF 4 Permitir unirse a Torneo**

RF 4.1 Mostrar Torneos Disponibles

RF 4.2 Mostrar Datos de Inscripción

**RF5 Jugar Partida**

RF5.1 Actualizar el campo resultado de la tabla del jugador.

**RF6 Realizar Jugada**

RF6.1 Mostrar PNG del Tablero

RF6.2 Actualiza la planilla de la tabla jugada

**RF7 Notificar por Correo**

RF7.1 Mostrar Interfaz de Correo

**RF8 Gestionar Partida**

RF8.1 Crear Partida

RF8.2 Actualizar Partida

**RF9 Gestionar Torneo**

RF9.1 Crear Torneo

RF9.2 Editar Torneo

RF9.3 Eliminar Torneo

**RF10 Imprimir Estado de la Partida**

RF10.1 Obtener PGN de la Partida

**RF11 Gestionar Jugadores del torneo**

RF11.1 Buscar Jugador

RF11.2 Actualizar Jugador

RF11.3 Eliminar Jugador

**RF12 Mostrar Partida**

RF12.1 Mostrar partidas creadas

**RF13 Mostrar Datos de Partida**

RF13.1 Mostrar campos de la partida seleccionada.

**RF14 Mostrar Tablero de Análisis**

RF14.1 Mostrar PNG del Tablero Análisis.

**RF15 Realizar Pareo**

RF15.1 Actualizar tabla de jugador.

**RF16 Definir Partidas no Concluidas.**

RF16.1 Actualizar Partidas no concluidas.

**RF17 Conceder Licencia.**

RF17.1 Actualizar tabla de jugador.

**RF18 Solicitar Licencia.**

**RF19 Registrarse.**

RF19.1 Registrar usuario

**2.10.2 Requisitos no Funcionales.**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Existen múltiples categorías para clasificar a los requerimientos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta, aunque no limitan a la definición de otros.

**Apariencia o interfaz externa**

La interfaz debe ser sencilla, clara, intuitiva y amigable, manteniendo un formato estándar en todas las páginas. No contará con gran cantidad de imágenes, garantizando que el tiempo de descarga sea mínimo y así no demorar las respuestas al usuario. Teniendo en cuenta el fin para el que se desarrolla la aplicación debe ser serio, formal y tener una navegación sugerente.

**Interfaces Hardware.**

El Servidor debe tener como mínimo las siguientes características de hardware: Procesador Pentium II 450 MHz o superior, 128 Mb de memoria RAM (incluye la utilizada por el SO) y 2Gb de capacidad en disco duro.

**Interfaces Software.**

Este sistema correrá en los sistemas operativos siguientes: Linux y Windows NT/2000/XP. Además se utiliza la herramienta de modelado Visual Paradigm y el lenguaje de programación PHP.

**Usabilidad.**

La aplicación debe cumplir con los principales principios de usabilidad, debe brindarse comodidad a la hora de acceder a las diferentes funcionalidades que brinda la aplicación, así como será fácil de usar para cualquier usuario del sistema ya sea jugador, o árbitro de ajedrez con los conocimientos básicos

en informática y en el uso de aplicaciones web. La navegabilidad debe no debe ser muy compleja, todas las funcionalidades deben ser rápidamente accesibles por los usuarios del sistema.

### **Rendimiento.**

La disponibilidad de trabajo en red contra el servidor es constante. Teniendo como prioridad garantizar que el tiempo de respuesta a las solicitudes de los usuarios al sistema sea mínimo, o sea, en un período de tiempo breve (escasos segundos). De tal manera se evita la acumulación de trabajo por parte de los responsables. El sistema deberá ser los más estable y confiable posible.

### **Soporte.**

El sistema tiene el soporte garantizado ya que mismo será desarrollado por el proyecto Infodrez el cual se encargará en cualquier momento de brindar soporte a todas las aplicaciones desarrolladas por el mismo.

### **Portabilidad.**

El sistema funcionará sobre plataforma Linux, de acuerdo a los beneficios del software libre y las exigencias de los clientes de acuerdo a sus normas profesionales. En caso de cambio de plataforma por pedidos del cliente, el sistema se ajustará totalmente con la mayor brevedad gracias a la universalidad de las herramientas utilizadas que pueden ser soportadas tanto en software libre como software propietario. El producto está implantado sobre una plataforma Web, codificada en "HP 5.0" y sus sistemas de base de datos en MySQL.

### **Seguridad.**

Los usuarios deben autenticarse antes de entrar al sistema. La información debe transmitirse de manera segura, se debe garantizar la seguridad a todos los niveles (Interfaz, negocio y Acceso a datos) restringiendo las funcionalidades mediante roles de usuarios garantizando que la información sea accesible al usuario autorizado.

### **Legales.**

Reconocido y autorizado por instancias superiores tales como la directiva de la UCI.  
Documentación legal de uso como Declaración de Autoría.

### **Restricciones en el diseño y la implementación.**

Para la programación en PHP se recomienda el editor Zend Studio for Eclipse.  
Se usará la arquitectura MVC.

**Requisitos para la documentación de usuarios en línea y ayuda del sistema.**

Se debe brindar una interfaz amigable que explique las diferentes funcionalidades con que cuenta el sistema de manera rápida, además los manuales de usuario y toda la documentación actualizada de cada módulo de la aplicación.

**2.11 Modelación del Sistema.**

El modelado del sistema representa la funcionalidad completa de un sistema mostrando su interacción con los agentes externos. Esta representación se hace a través de las relaciones entre los actores y los casos de uso dentro del sistema. Los diagramas de casos de uso definen conjuntos de funcionalidades afines que el sistema debe cumplir para satisfacer todos los requerimientos que tiene a su cargo. Esos conjuntos de funcionalidades son representados por los diferentes diagramas que darán solución a la aplicación.

**2.12 Actores del Sistema.**

Estos no son más que trabajadores que participan directamente en el negocio y/o nuevos actores que surgen para ejecutar funcionalidades que aparecen debido a la presencia de requisitos especiales (Ver tabla 11).

**Tabla 11: Descripción de los Actores del Sistema**

Actores del sistema	Justificación
Organizador del Torneo	Es el encargado de Gestionar los Torneos ya sea crearlos, eliminar Torneos como Modificar los mismos, además de esto realiza el pareo del torneo creado.
Director del Torneo(Arbitro Principal)	Se encarga de de Gestionar las partidas en el torneo así como los jugadores del torneo, también posee la autoridad de definir las partidas no concluidas y conceder las licencias solicitadas por los jugadores.
Jugador	Es el autor principal en un torneo de ajedrez por Correspondencia.

2.12 Diagrama de casos de uso del sistema.

Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. Es decir, un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores.

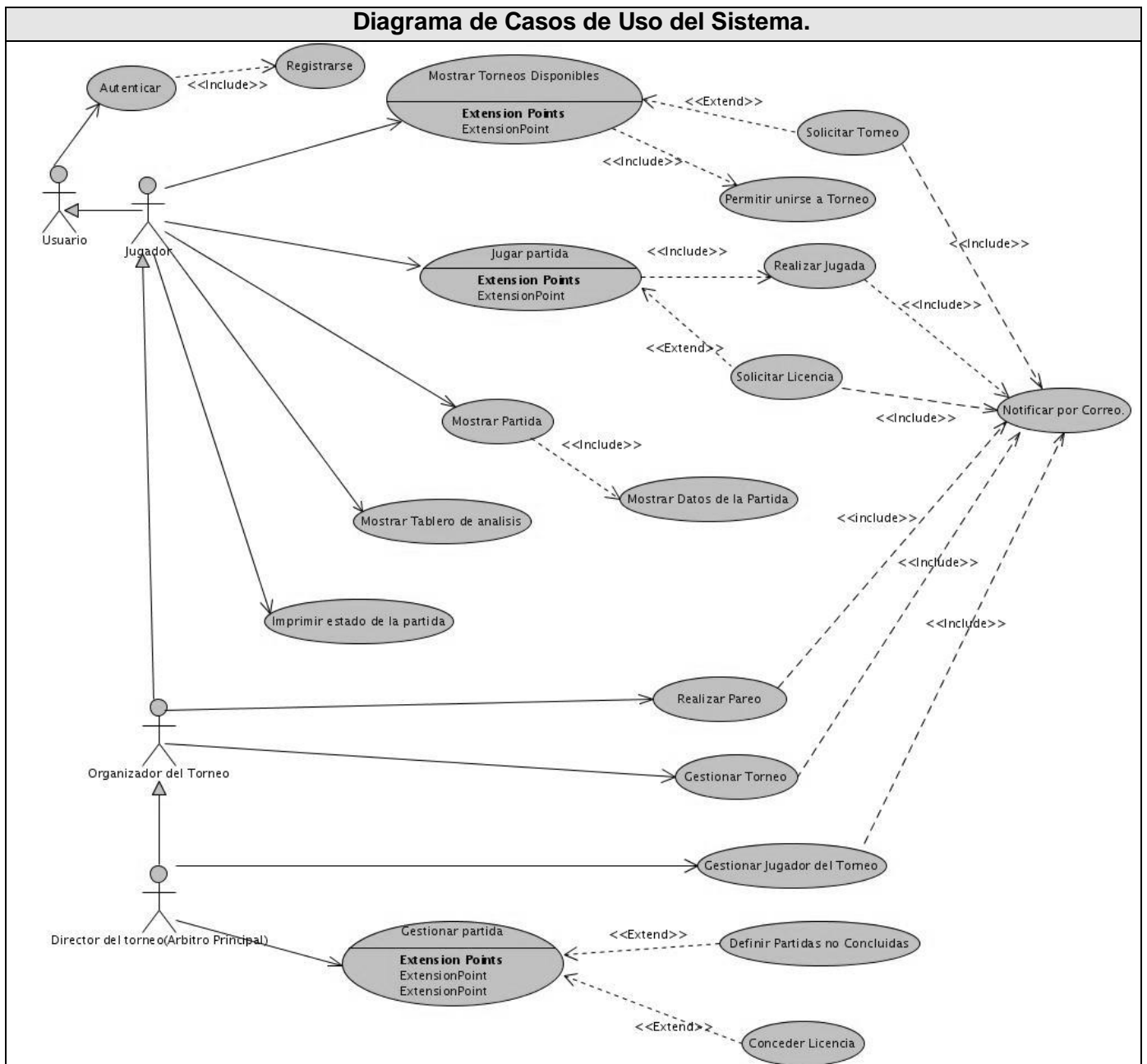


Figura 17: Diagrama de Casos de Uso del Sistema

2.12.1 Descripción de los casos de uso del sistema.

Tabla 12: Descripción del caso de uso “Autenticarse”

Caso de Uso		Autenticarse
<b>Actores</b>	Jugador (inicia), Organizador del torneo (inicia), Director del Torneo (Arbitro principal) (inicia).	
<b>Propósito</b>	Entrar al sistema para desempeñar el rol que le corresponde.	
<b>Resumen</b>	El CUS se inicia cuando el Jugador, Organizador del torneo o el director del torneo (Arbitro Principal) deciden autenticarse en el sistema, donde el sistema verifica si la autenticación es correcta y estos usuarios pasan a desempeñar su rol.	
<b>Referencias</b>	RF 1.1, RF1.2	
<b>Precondiciones</b>	Usuario sin autenticar.	
<b>Poscondiciones</b>	Usuario autenticado y dentro del sistema.	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1- Jugador, Organizador del Torneo, Director del Torneo: Accede al sistema vía Web.  2 Jugador, Organizador del Torneo, Director del Torneo: Introduce usuario y contraseña	3- Verifica si el usuario y la contraseña son correctos.  4- Si son correctos entran al sistema.  5- El caso de uso termina.	
<b>Curso alterno de los eventos</b>		

<p>5- Jugador, Director del Torneo o Organizador del Torneo: Corrige los datos incorrectos e intenta de nuevo entrar al sistema hasta introducir el usuario y la contraseña correctos.</p>	<p>4- Si los datos son erróneos le muestra un mensaje de error informándole al usuario que verifique si se autentico correctamente.</p> <p>6- Se ejecuta la actividad 4 del curso normal.</p>
<p><b>Prioridad</b></p>	<p>Crítico</p>

Tabla 13: Descripción del caso de uso “Mostrar Torneos Disponibles”

Caso de Uso	Mostrar Torneos Disponibles
<b>Actores</b>	Jugador (inicia).
<b>Propósito</b>	Mostrar un listado con los torneos disponibles en el Sistema hasta el momento.
<b>Resumen</b>	El CUS se inicia cuando el Jugador solicita al sistema de mostrar todos los torneos que se encuentran disponibles en el torneo, el sistema realiza una consulta a la base de datos en la tabla de torneos y muestra una lista con los torneos que hay disponibles hasta el momento.
<b>Referencias</b>	RF 2.1, RF4
<b>Precondiciones</b>	Debe existir al menos un torneo creado en el Sistema.
<b>Poscondiciones</b>	Muestra un listado con todos los torneos disponibles en el sistema y el jugador se inscribe en el de su preferencia.
<b>Curso normal de los eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>



1- Jugador: Solicita la opción de Mostrar Torneos Disponibles.	2- Realiza una consulta a la base de datos a la tabla de Torneos.  3-Muestra una lista con los torneos que hay disponibles hasta el momento.
	4- El caso de uso termina.
<b>Curso alternativo de los eventos</b>	
	3- Si no existen torneos disponibles se le envía un mensaje al jugador interesado informándole que no existen torneos disponibles, disculpe las molestias que esto pueda ocasionar.  4-El caso de uso finaliza.
<b>Prioridad</b>	Crítico

**Tabla 14: Descripción del caso de uso “Jugar Partida”**

<b>Caso de Uso</b>		<b>Jugar Partida</b>
<b>Actores</b>	Jugador (inicia).	
<b>Propósito</b>	Jugar una Partida de Ajedrez por Correspondencia mediante el Sistema.	
<b>Resumen</b>	El CUS se inicia cuando el Jugador solicita al sistema de jugar una partida, el sistema muestra al usuario una interfaz dándole la posibilidad de seleccionar la partida que desee jugar, es cuando el jugador selecciona la partida que desea jugar.	
<b>Referencias</b>	RF 5, RF 5.1, RF6	
<b>Precondiciones</b>	Deben existir las partidas creadas en el Sistema.	
<b>Poscondiciones</b>	Se realizan todas las jugadas correspondientes a dicha partida, además actualiza la tabla del jugador en el campo resultado.	
<b>Curso normal de los eventos</b>		

Acción del actor		Respuesta del sistema
1-Jugador: Solicita la opción de jugar partida en el sistema.		2- Muestra al usuario una interfaz dándole la posibilidad de seleccionar la partida que desee jugar.  4- Muestra al usuario una interfaz grafica dándole la posibilidad de empezar la partida.
3-Jugador: Selecciona la partida que desea jugar.		
5-Jugador: Juega la Partida completa del Torneo.		
		6- El caso de uso termina.
Curso alternativo de los eventos		
5-Jugador: No juega la partida completa por abandono y la partida queda sin concluir.		
6- El caso de uso finaliza.		
Prioridad	Crítico	

Tabla 15: Descripción del caso de uso “Gestionar Torneo”

Caso de Uso	Gestionar Torneo
<b>Actores</b>	Organizador del Torneo (inicia).
<b>Propósito</b>	Que el Organizador del Torneo tenga la posibilidad de Gestionar los Torneos en el Sistema ya sea Crear, Editar o Eliminar un Torneo.
<b>Resumen</b>	El CUS se inicia cuando el Organizador del Torneo selecciona la opción de Gestionar Torneo, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema realiza la acción seleccionada por el Organizador del Torneo y termina el CUS.
<b>Referencias</b>	RF9.1, RF9.2, RF9.3

<b>Precondiciones</b>	Existe una necesidad de gestionar un torneo.
<b>Poscondiciones</b>	Se Gestiona la acción seleccionada por el Organizador del Torneo
<b>Curso normal de los eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- Organizador del torneo: El caso de uso empieza cuando el organizador del torneo accede al vínculo Gestionar torneo.	2- El sistema muestra tres opciones crear torneo, Editar torneo, eliminar torneo.
3- Organizador del Torneo: Selecciona una de las tres opciones.	4- El sistema muestra según la opción solicitada Crear Torneo: Sección Crear Torneo, Editar Torneo: Sección Editar Torneo, Eliminar Torneo: Sección Eliminar Torneo.
	5- El caso de uso Termina.
<b>Sección : Crear Torneo</b>	
1-Organizador del torneo: Solicita la opción de Crear Torneo.	2-Muestra una interfaz donde le muestra al Organizador del torneo los campos que debe llenar para Crear un torneo.
3-Organizador del Torneo: Introduce los datos que requiere el formulario.	4-Envía los datos del torneo creado a la tabla de torneo en la base de datos.
	5- El caso de uso termina.
<b>Sección: Editar Torneo</b>	
1- Organizador del torneo: Solicita editar un Torneo.	2-Realiza una consulta a la base de datos, a la tabla de

<p>4- Organizador del Torneo: Selecciona el torneo que desea editar.</p> <p>6- Organizador del Torneo: Edita los datos que desee del Torneo.</p>	<p>torneos y solicita todos los torneos creados hasta el momento.</p> <p>3-Muestra una interfaz con todos los torneos.</p> <p>5-Muestra una interfaz con los datos del Torneo.</p> <p>7-Actualiza los campos editados en la base de datos.</p>
	<p>8- El caso de uso termina.</p>
<p><b>Curso alternativo de los eventos</b></p>	
	<p>3- Si no existen torneos creados se envía un mensaje informando que no existen torneos en el sistema.</p> <p>4-El caso de uso finaliza.</p>
<p><b>Sección : Eliminar Torneo</b></p>	
<p>1-Organizador del Torneo: Solicita la opción de eliminar un torneo del sistema.</p> <p>4- Organizador del Torneo: Selecciona el torneo que desea eliminar.</p>	<p>2-Realiza una consulta a la base de datos, a la tabla de torneos y solicita todos los torneos creados hasta el momento.</p> <p>3-Muestra una interfaz con todos los torneos existentes.</p> <p>5-Elimina el torneo seleccionado de la base de datos.</p>
	<p>6- El caso de uso termina.</p>
<p><b>Curso alternativo de los eventos</b></p>	
	<p>3- Si no existen torneos creados se envía un mensaje informando que no existen torneos en el sistema.</p> <p>4-El caso de uso finaliza.</p>
<p><b>Prioridad</b></p>	<p>Crítico</p>

Tabla 16: Descripción del caso de uso “Gestionar Partida”

Caso de Uso		Gestionar Partida
<b>Actores</b>	Director del Torneo (Arbitro Principal) (inicia).	
<b>Propósito</b>	Que el Director del Torneo tenga la posibilidad de Gestionar las partidas en el Sistema ya sea Crearlas o Actualizarlas.	
<b>Resumen</b>	El CUS se inicia cuando el Director del Torneo (Arbitro Principal) selecciona la opción de Gestionar Partida, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema realiza la acción seleccionada por el Director del Torneo y termina el CUS.	
<b>Referencias</b>	RF8.1, RF8.2,	
<b>Precondiciones</b>	Existe una necesidad de gestionar una partida.	
<b>Poscondiciones</b>	Se Gestiona la acción seleccionada por el Director del Torneo (Arbitro Principal)	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1- Director del Torneo: Se inicia el caso de uso cuando el director del torneo solicita Gestionar una Partida.	2- El sistema muestra dos opciones crear partida y actualizar partida.	
3- Director del Torneo: Selecciona una de las opciones que existen.	4- El sistema muestra según la opción solicitada Crear Partida: Sección Crear Partida, Actualizar Partida: Sección Actualizar Partida.	
	5- El caso de uso Termina.	
<b>Sección : Crear Partida</b>		
1- Director del Torneo: Solicita la opción de Crear Partida.	2-Realiza una consulta a la tabla de jugador en la base	

	<p>de datos y solicita los datos del jugador.</p> <p>3-Realiza el Pareo con los datos del jugador.</p> <p>4-Registra los datos de la partida según el pareo efectuado creándose la partida, y muestra el pareo.</p>
	5- El caso de uso termina.
<b>Sección: Actualizar Partida</b>	
<p>1- Director del Torneo: Solicita actualizar una Partida.</p> <p>4- Director del Torneo: Selecciona la partida que desea actualizar.</p> <p>6- Director del Torneo: Actualiza los datos que desee de la partida.</p>	<p>2-Realiza una consulta a la base de datos, a la tabla de partidas y solicita todos los partidas creados hasta el momento.</p> <p>3-Muestra una interfaz con todos las partidas.</p> <p>5-Muestra una interfaz con los datos de la Partida.</p> <p>7-Actualiza en la base de datos los campos seleccionados para cambiar.</p>
	8- El caso de uso termina.
<b>Curso alternativo de los eventos</b>	
	<p>3- Si no existen torneos creados se envía un mensaje informando que no existen torneos en el sistema.</p> <p>4-El caso de uso finaliza.</p>
<b>Sección : Eliminar Torneo</b>	
<p>1-Organizador del Torneo: Solicita la opción de eliminar un torneo del sistema.</p>	<p>2-Realiza una consulta a la base de datos, a la tabla de torneos y solicita todos los torneos creados hasta el momento.</p> <p>3-Muestra una interfaz con todos los torneos</p>

4- Organizador del Torneo: Selecciona el torneo que desea eliminar.	existentes. 5-Elimina el torneo seleccionado de la base de datos.
	6- El caso de uso termina.
<b>Curso alterno de los eventos</b>	
	3- Si no existen torneos creados se envía un mensaje informando que no existen torneos en el sistema. 4-El caso de uso finaliza.
<b>Prioridad</b>	Crítico

Tabla 17: Descripción del caso de uso “Solicitar Torneo”

Caso de Uso		Solicitar Torneo
<b>Actores</b>	Jugador (inicia).	
<b>Propósito</b>	El Jugador tenga la posibilidad de solicitar en el sistema un torneo de su preferencia.	
<b>Resumen</b>	El CUS se inicia cuando el Jugador solicita el torneo de su preferencia al sistema, este muestra una interfaz grafica con un formulario donde el jugador llena los campos con el torneo que desea, el sistema realiza una consulta a la base de datos y verifica si existe el torneo solicitado por el jugado enviando un mensaje ya sea de existencia o de no existencia.	
<b>Referencias</b>	RF 5, RF 5.1, RF6	
<b>Precondiciones</b>	Deben existir las partidas creadas en el Sistema.	
<b>Poscondiciones</b>	Se realizan todas las jugadas correspondientes a dicha partida, además actualiza la tabla del jugador en el campo resultado.	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	

<p>1-Jugador: Solicita la opción de jugar partida en el sistema.</p> <p>3-Jugador: Selecciona la partida que desea jugar.</p> <p>5-Jugador: Juega la Partida completa del Torneo.</p>	<p>2- El sistema muestra un formulario con los datos que conforma un torneo.</p> <p>4-Realiza una consulta a la base de datos a la tabla de torneo y verifica si hay algún torneos con las características solicitada.</p> <p>5-Si existe algún torneo con las características solicitadas se le envía un mensaje al jugador informándole que si se encuentra el torneo de su preferencia.</p>
	<p>6- El caso de uso termina.</p>
<b>Curso alterno de los eventos</b>	
<p>5- Si no existe ningún torneo con las características que desea el jugador en la base de datos se le envía un mensaje informándole que no existen torneos con las características que desea.</p> <p>6- El caso de uso finaliza.</p>	
<b>Prioridad</b>	Secundario

### 2.13 Conclusiones.

En este capítulo se ha abordado la situación problemática y las características del sistema. Se ha realizado el estudio de los objetivos y los procesos del negocio que los soportan, así como un análisis crítico de todos los procesos vinculados en el campo de acción. Se identificaron quiénes son los trabajadores y los actores y que intervienen en el mismo y con cuáles actividades y entidades interactúan estos. Mediante el modelado del negocio se entendieron mejor los métodos de trabajo con los diagramas de actividades por casos de uso del negocio y con el levantamiento de requerimientos



funcionales se definieron los casos de uso críticos a automatizar. Se muestran además, los requerimientos no funcionales. Se describieron los actores que interactúan con el sistema, los casos de uso a automatizar y se presentan los diagramas de casos de uso del sistema y el diagrama de clases del modelo de objetos.

## **CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.**

### **3.1 Introducción.**

En el presente capítulo se expone el análisis y diseño propuesto para la solución de la aplicación, modelándose los artefactos necesarios que contribuyen a la implementación del sistema. También se presentarán los diagramas de interacción, los diagramas de clases del diseño del sistema, el diagrama de clases persistente y el modelo de datos con que contara el sistema.

### **3.2 Modelo de análisis.**

El modelo de análisis es la primera representación técnica del sistema, por lo que puede considerarse como una primera aproximación al modelo de diseño, y es por tanto una entrada fundamental para las actividades de diseño e implementación subsiguientes. A partir de su realización se refinan y estructuran los requisitos obtenidos con anterioridad proporcionando una visión general del sistema.

El modelo de análisis debe lograr tres objetivos primarios:

- Describir lo que requiere el cliente.
- Establecer una base para la creación de un diseño de software.
- Definir un conjunto de requisitos que se pueda validar una vez que se construye el software.

#### **3.2.1 Diagrama de Clases de Análisis.**

El diagrama de clases de análisis es un artefacto en el que se representa los conceptos fundamentales en un dominio del problema. Los diagramas de clases de análisis, representan las definiciones y relaciones entre las clases. Las clases del análisis se clasifican en Interfaz, de Control o Entidad.

- Clase de Interfaz: Modelan la interacción entre el sistema y sus actores.
- Clase Entidad: Modelan información que posee una larga vida.
- Clase de Control: Representan coordinación, secuencia, transacciones, y control de otros objetos y a menudo encapsula a un caso de uso en concreto.

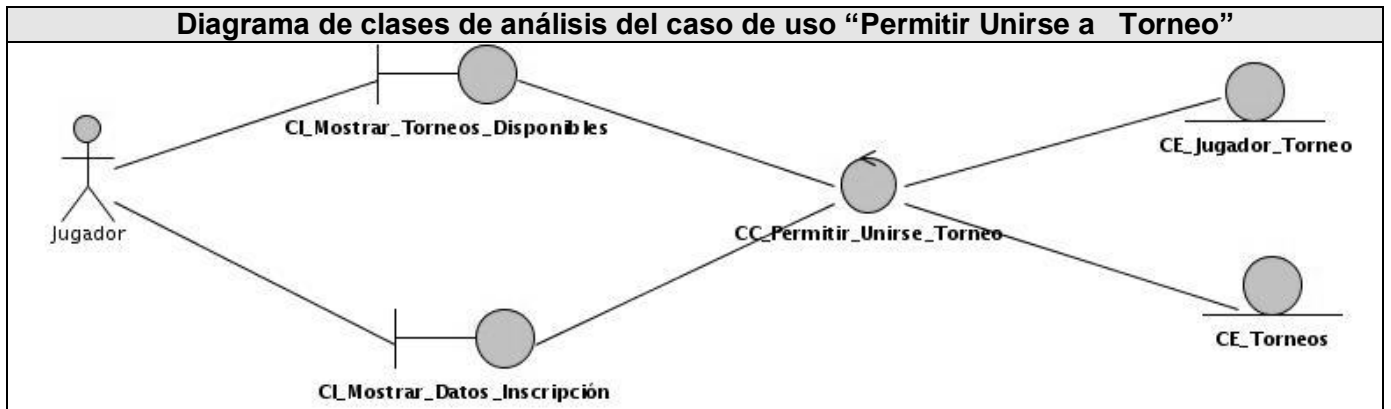


Figura 18: Diagrama de clases de análisis del caso de uso "Permitir Unirse a Torneo"

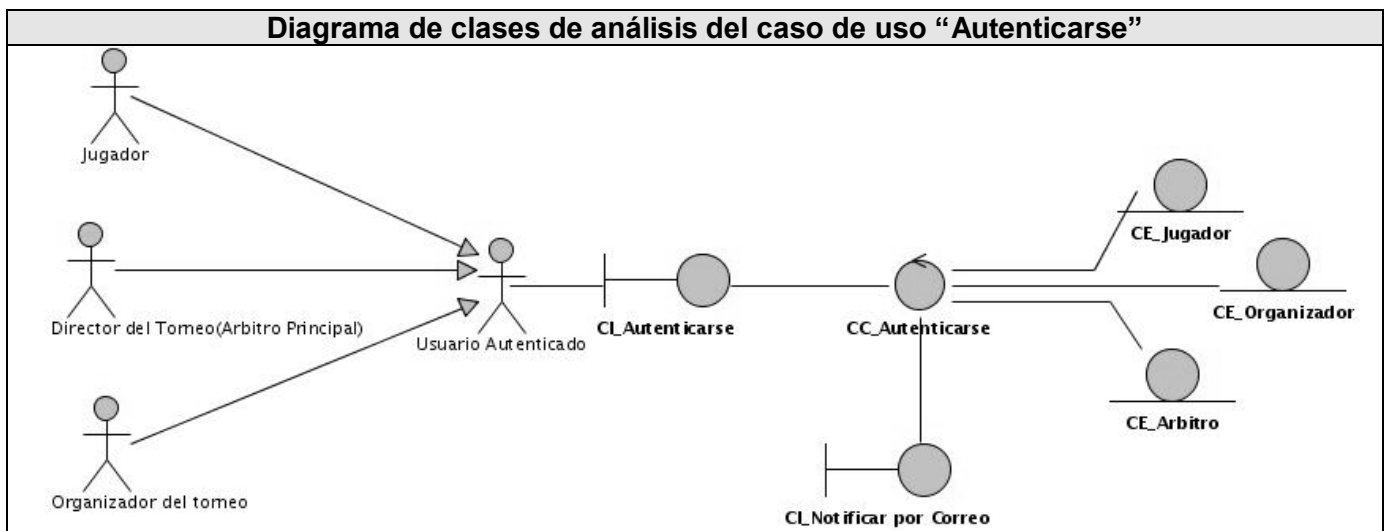


Figura 19: Diagrama de clases de análisis del caso de uso "Autenticarse"

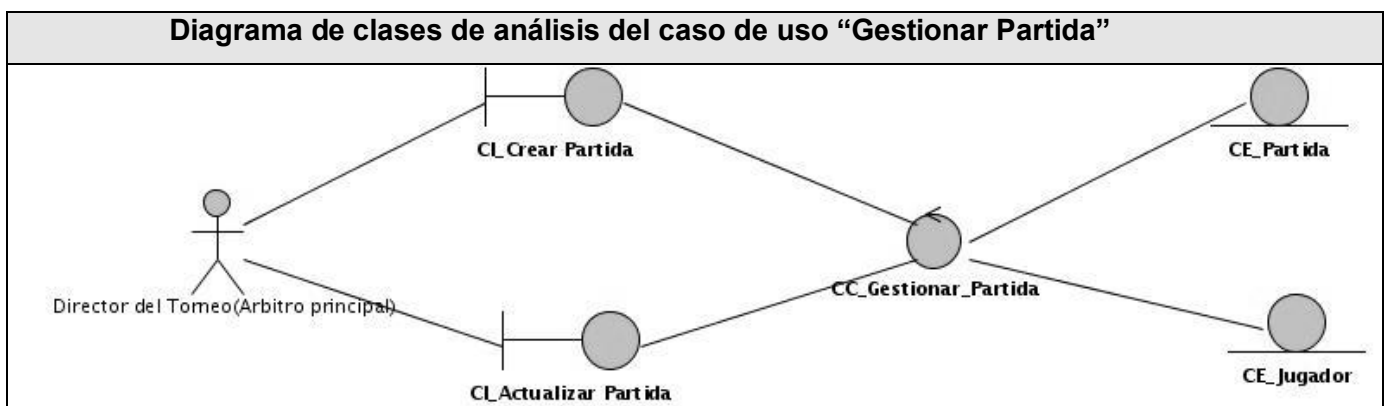


Figura 20: Diagrama de clases de análisis del caso de uso "Gestionar Partida"

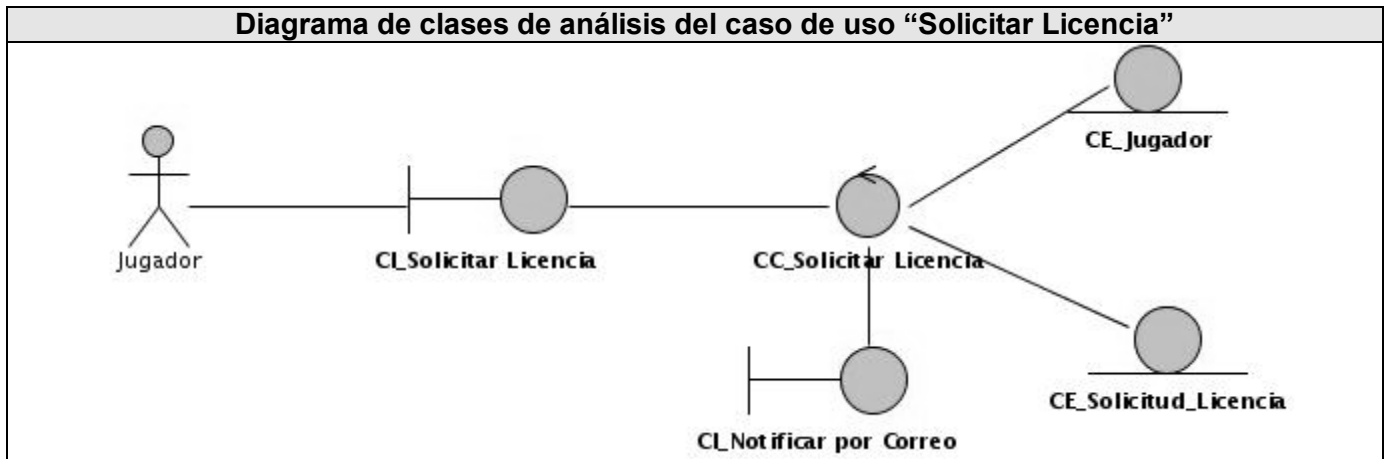


Figura 21: Diagrama de clases de análisis del caso de uso "Solicitar Licencia"

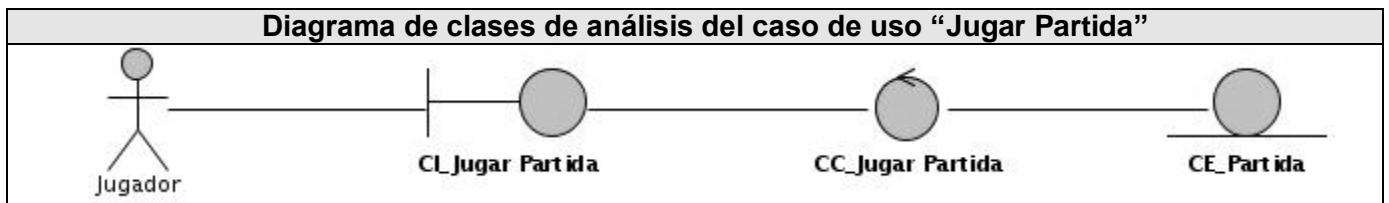


Figura 22: Diagrama de clases de análisis del caso de uso "Jugar Partida"

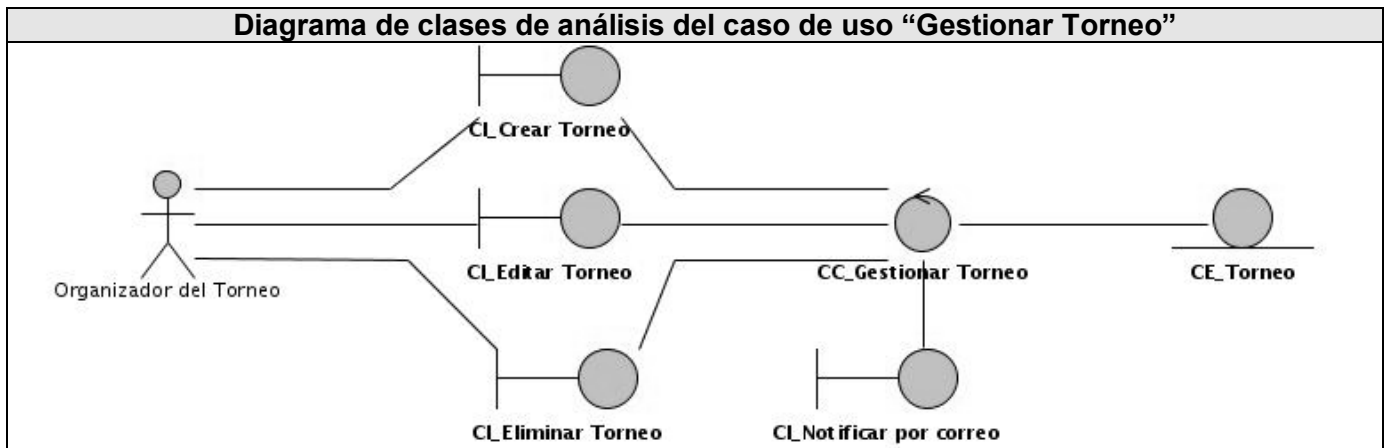


Figura 23: Diagrama de clases de análisis del caso de uso "Gestionar Torneo"

### 3.3. Modelo de Diseño.

El diseño es el centro de atención final de la fase de elaboración y el comienzo de las iteraciones de construcción, modela el sistema y se le da forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Además impone una

estructura que se debe conservar lo más fielmente posible cuando se le da forma al sistema. Mas tarde durante la fase de construcción, cuando la arquitectura es estable y los requisitos están bien entendidos, el centro de atención se desplaza a la implementación.

### 3.3.1. Diagramas de Secuencia y Colaboración.

Para observar la perspectiva cronológica más adecuada es necesario el Diagrama de Secuencia, pues muestra la secuencia explícita de los mensajes y realiza especificaciones de tiempo real para escenarios completo. El Diagrama de Colaboración ofrece una mejor visión espacial exponiendo los enlaces de comunicación entre objetos, muestra las relaciones entre objetos y son mejores para comprender todos los efectos que tiene un objeto y para el diseño de procedimientos. El diagrama de Colaboración puede obtenerse automáticamente a partir del correspondiente diagrama de Secuencia (o viceversa). En el presente trabajo se realizarán los diagramas de Secuencia, además de los diagramas de clases para cada caso de uso.

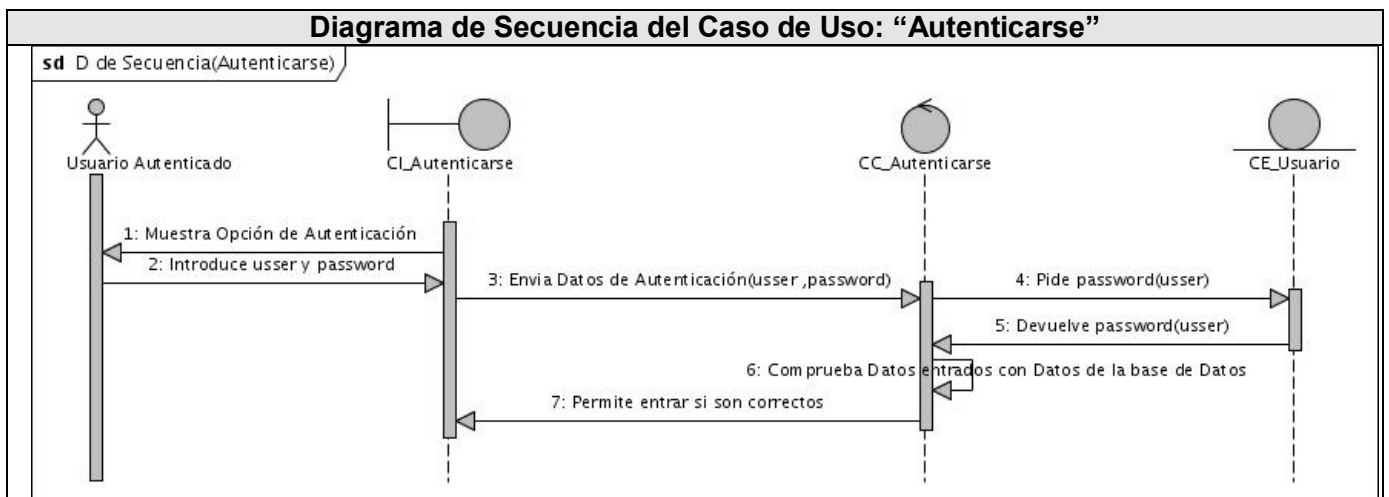


Figura 24: Diagrama de Secuencia del Caso de Uso: "Autenticarse"

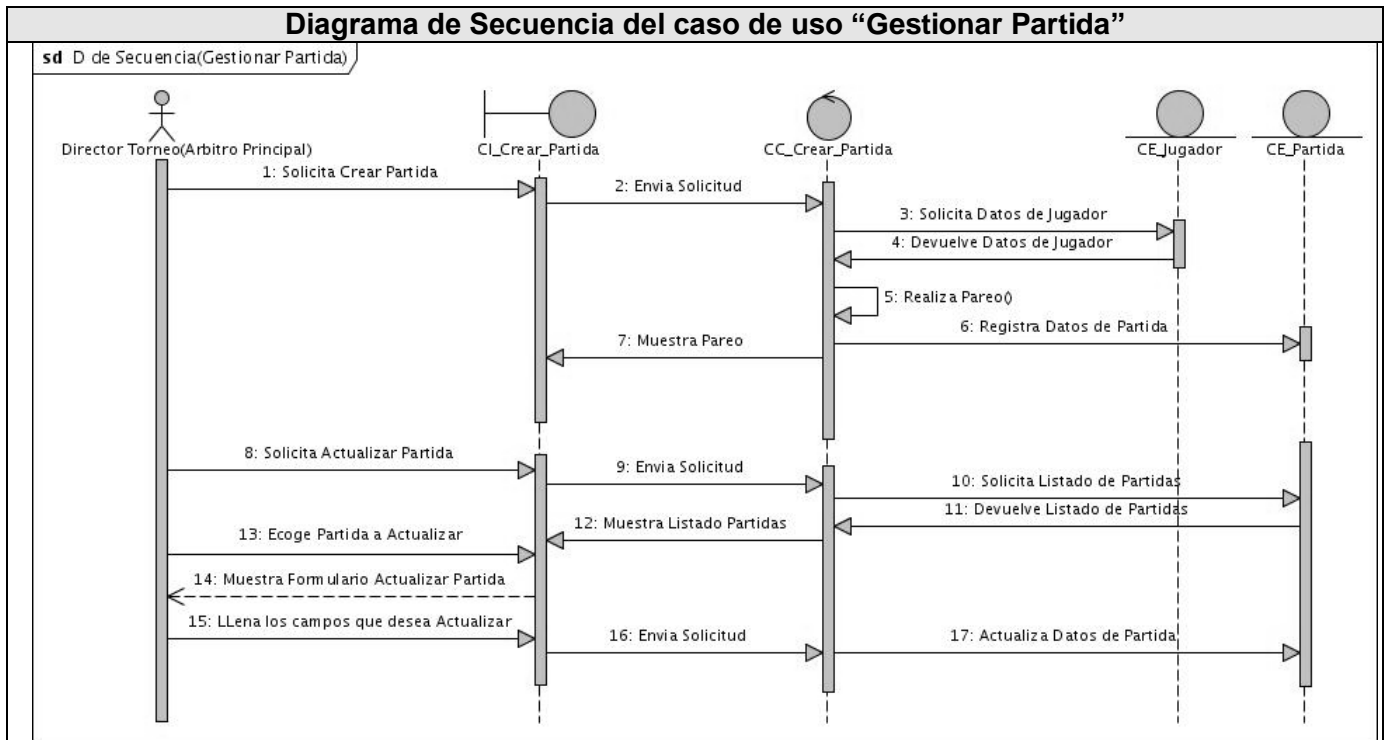


Figura 25: Diagrama de Secuencia del caso de uso "Gestionar Partida"

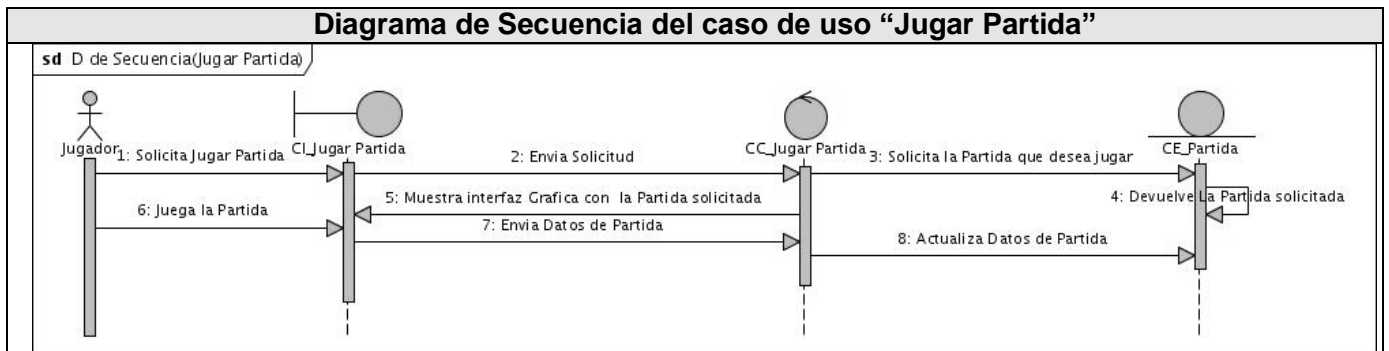


Figura 26: Diagrama de Secuencia del caso de uso "Jugar Partida"

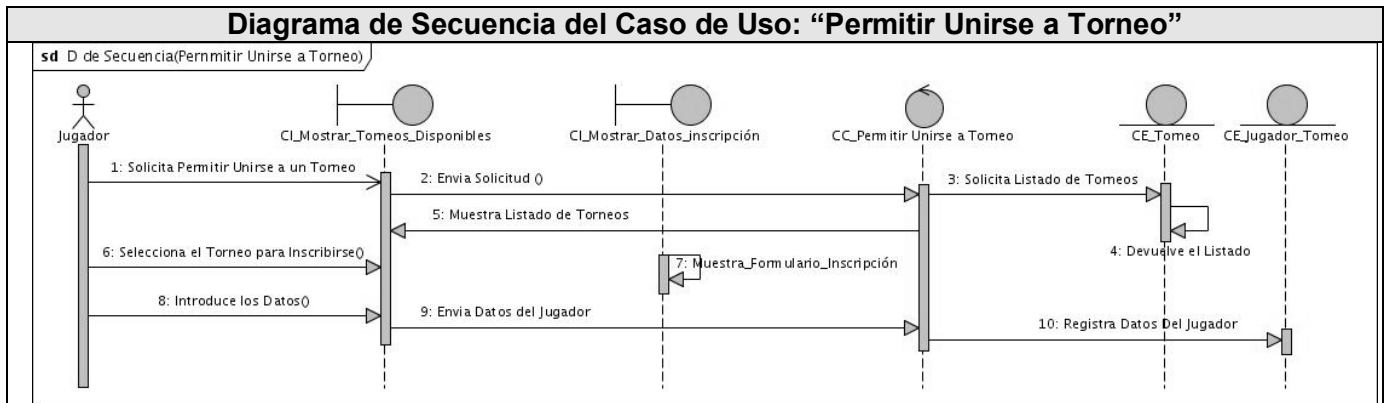


Figura 27: Diagrama de Secuencia del Caso de Uso: "Permitir Unirse a Torneo"

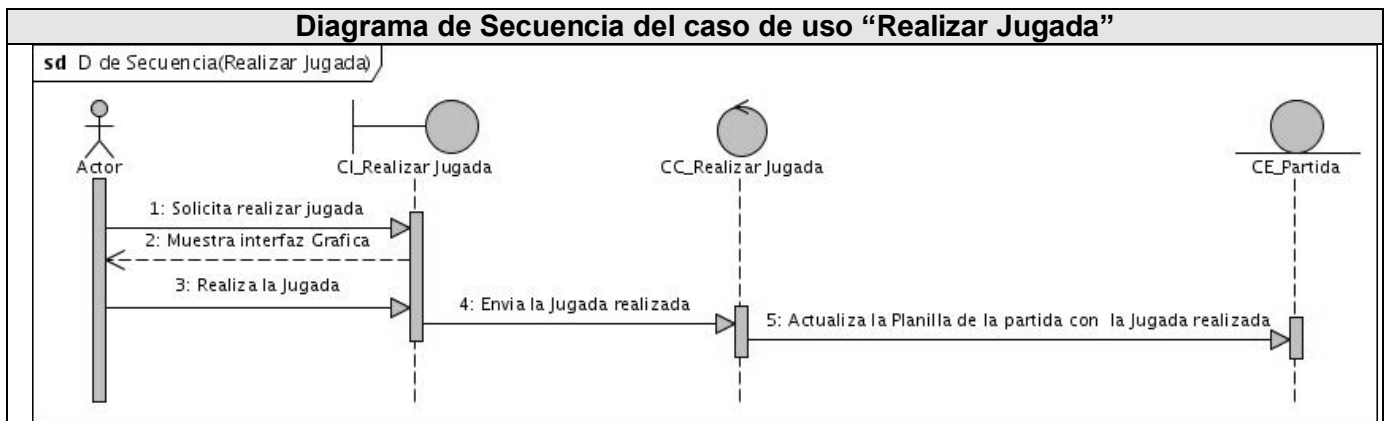


Figura 28: Diagrama de Secuencia del caso de uso "Realizar Jugada"

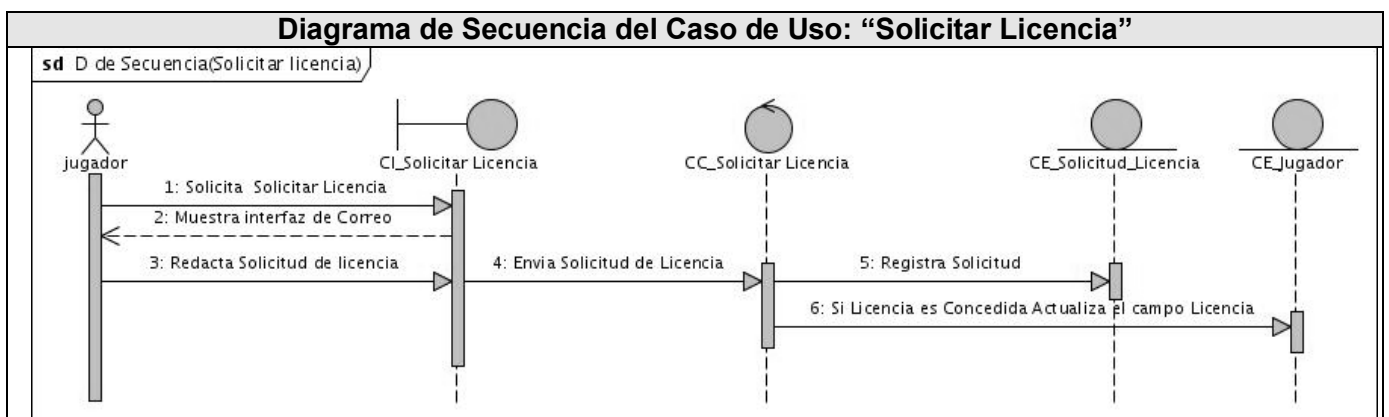


Figura 29: Diagrama de Secuencia del Caso de Uso: "Solicitar Licencia"

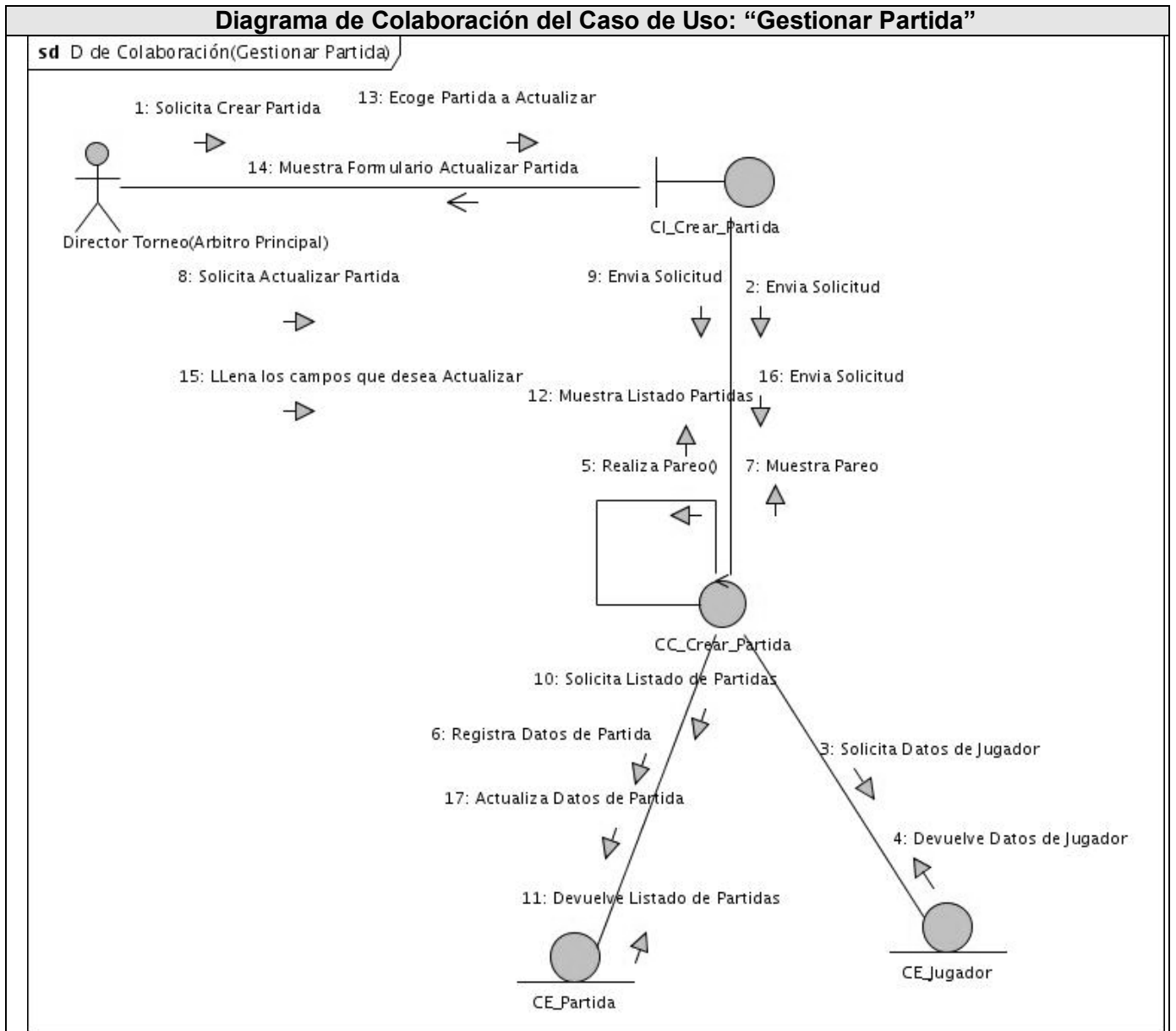


Figura 30: Diagrama de Colaboración del Caso de Uso: "Gestionar Partida"



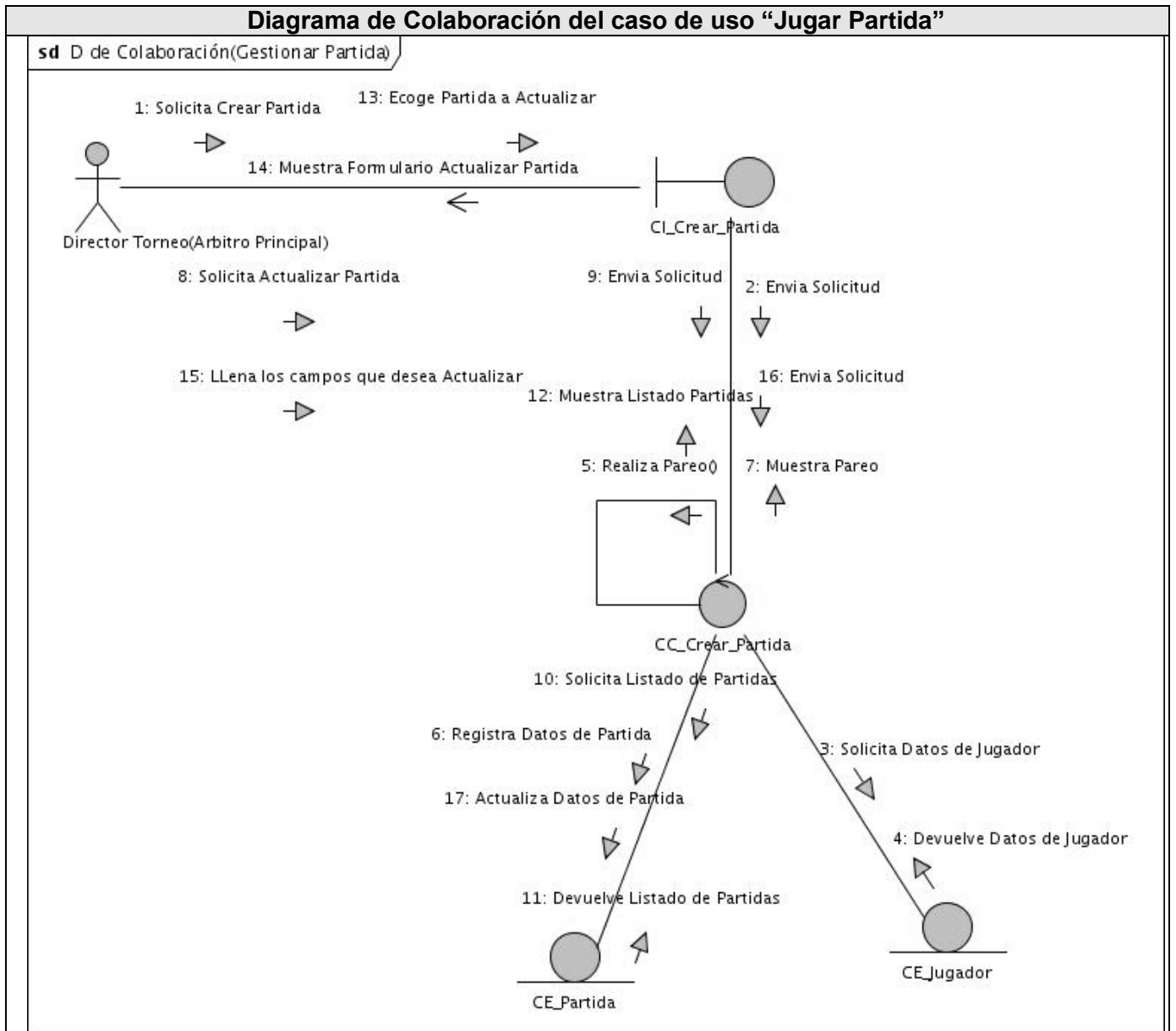
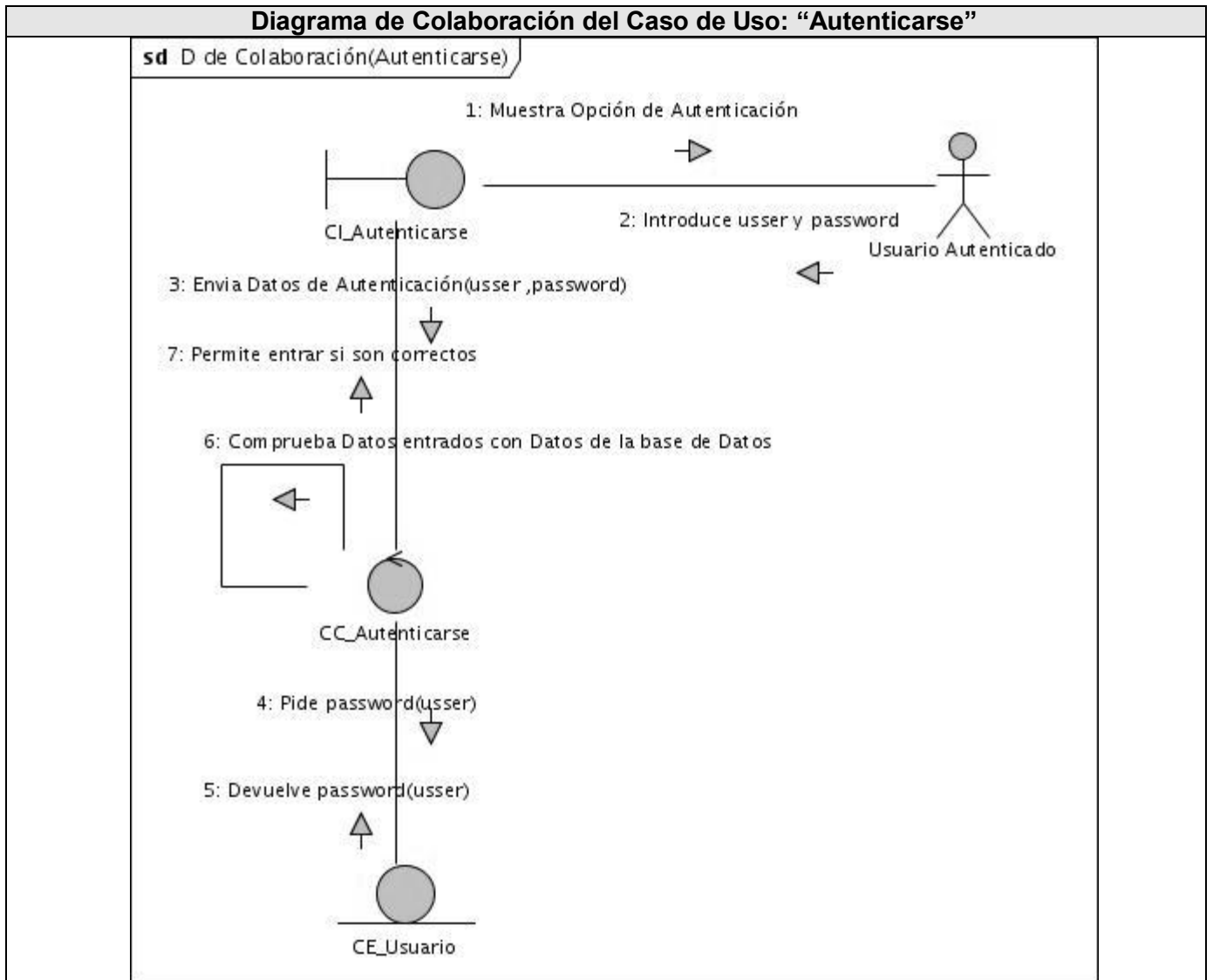


Figura 31: Diagrama de Colaboración del caso de uso “Jugar Partida”



**Figura 32: Diagrama de Colaboración del Caso de Uso: "Autenticarse"**

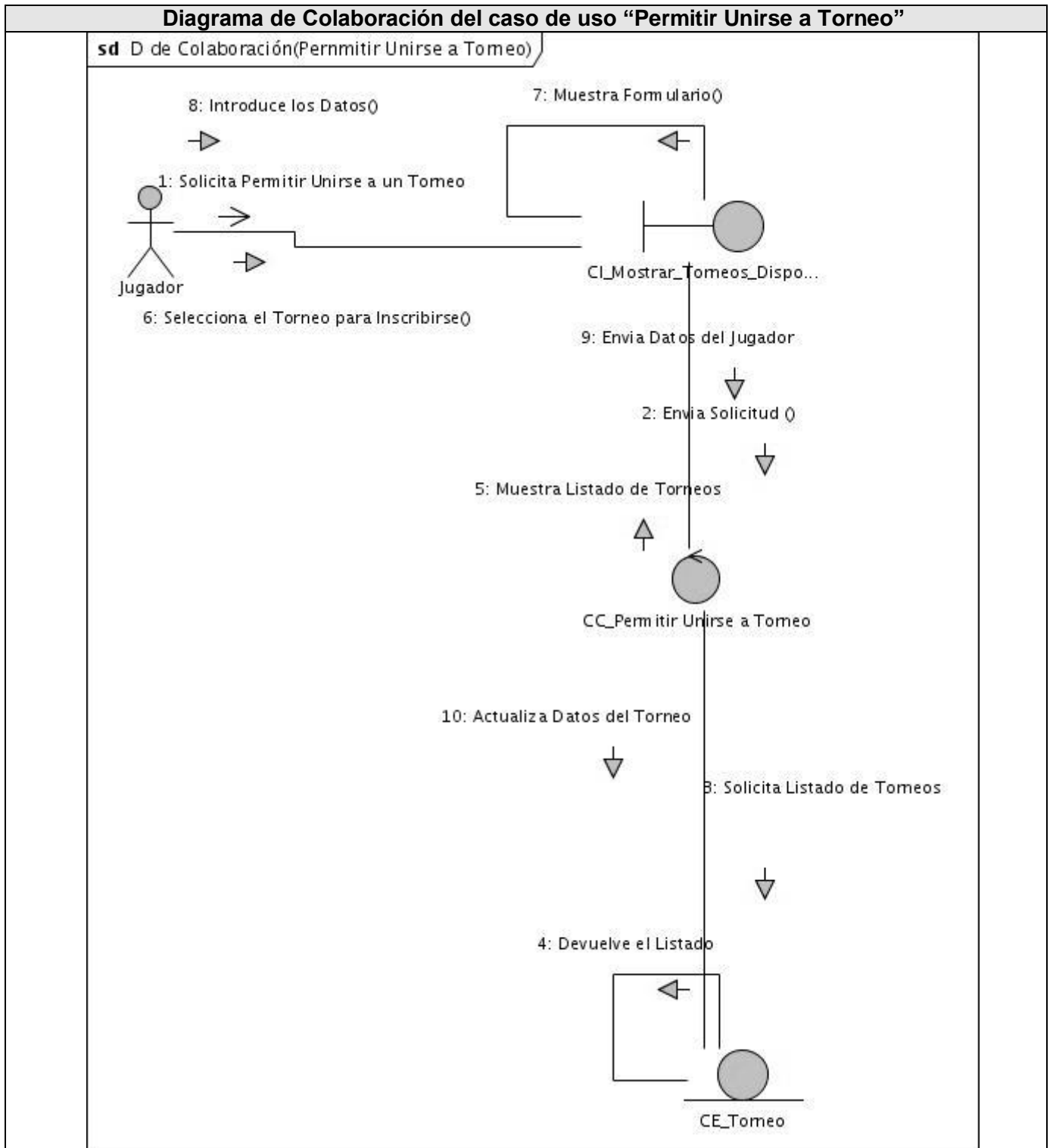


Figura 33: Diagrama de Colaboración del caso de uso "Permitir Unirse a Torneo"

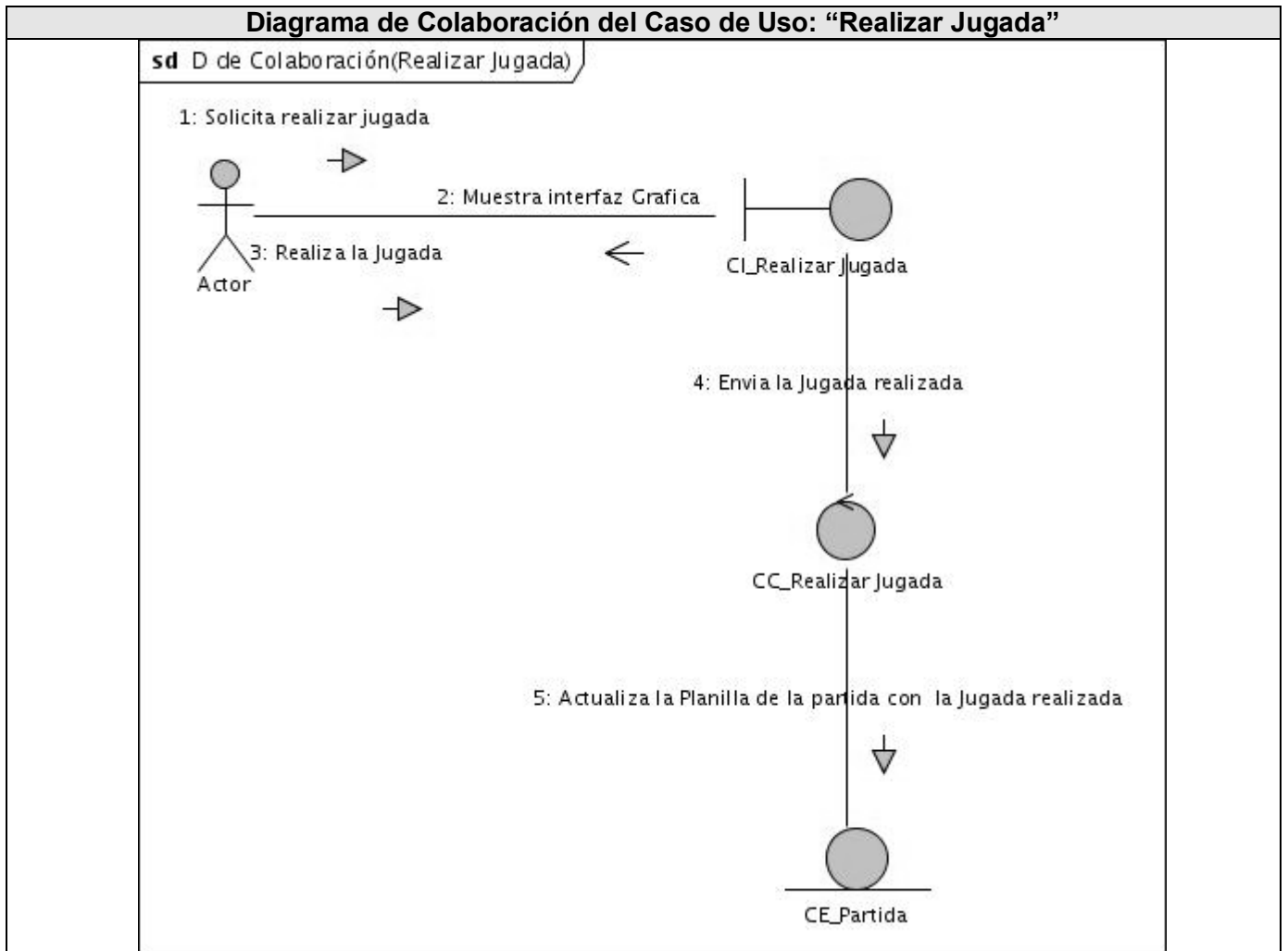
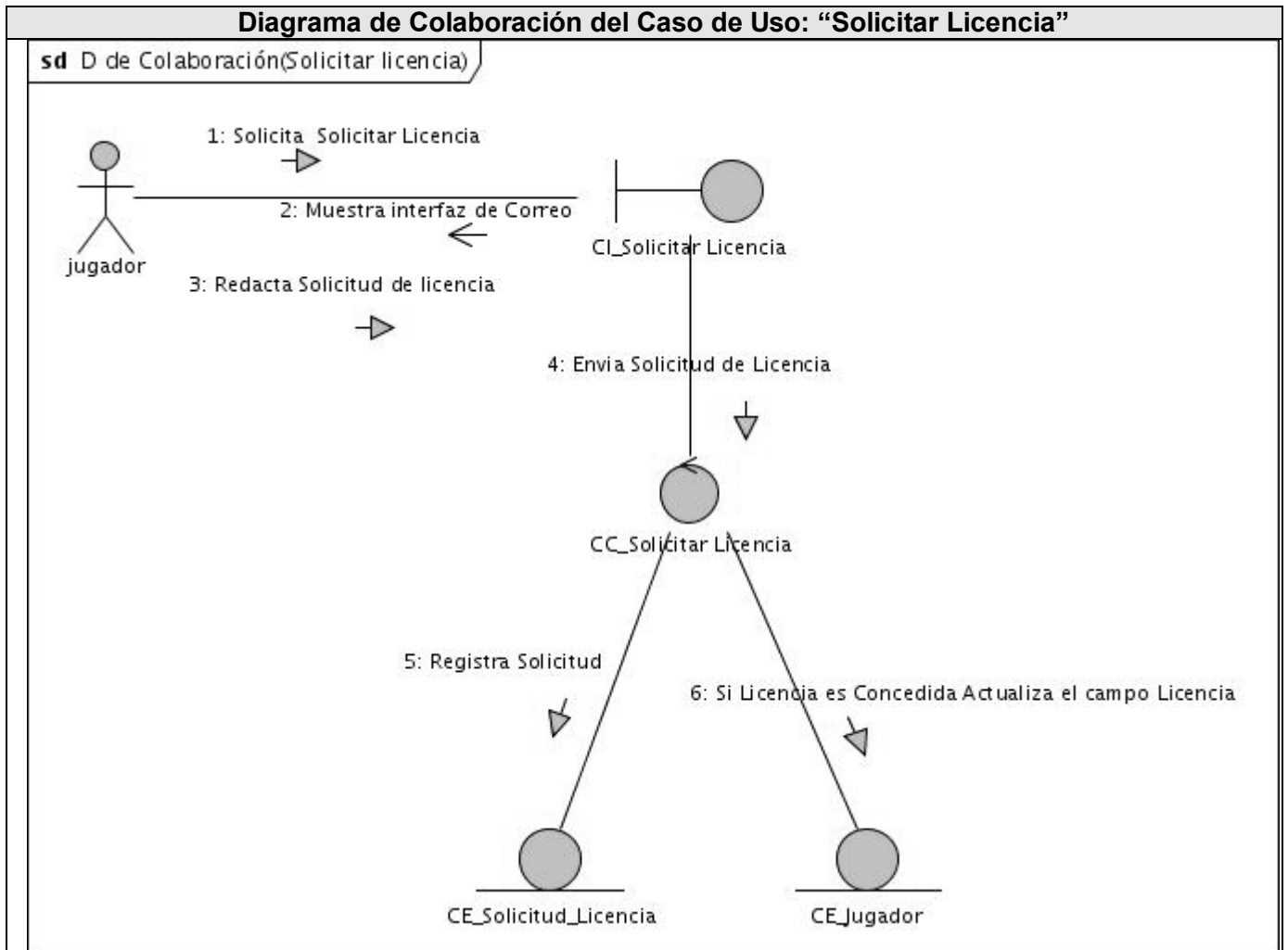


Figura 34: Diagrama de Colaboración del Caso de Uso: "Realizar Jugada"



**Figura 35: Diagrama de Colaboración del Caso de Uso: "Solicitar Licencia"**

### 3.4 Diagramas de Clases del Diseño.

Una clase de diseño es una construcción similar en la implementación del sistema. Los diagramas de clases de diseño exponen un conjunto de interfaces, colaboraciones y sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema. Los diagramas de clases de diseño del sistema está separado en casos de usos, es decir, cada caso de uso cuenta con un diagrama de clases, para hacer más fácil la comprensión.

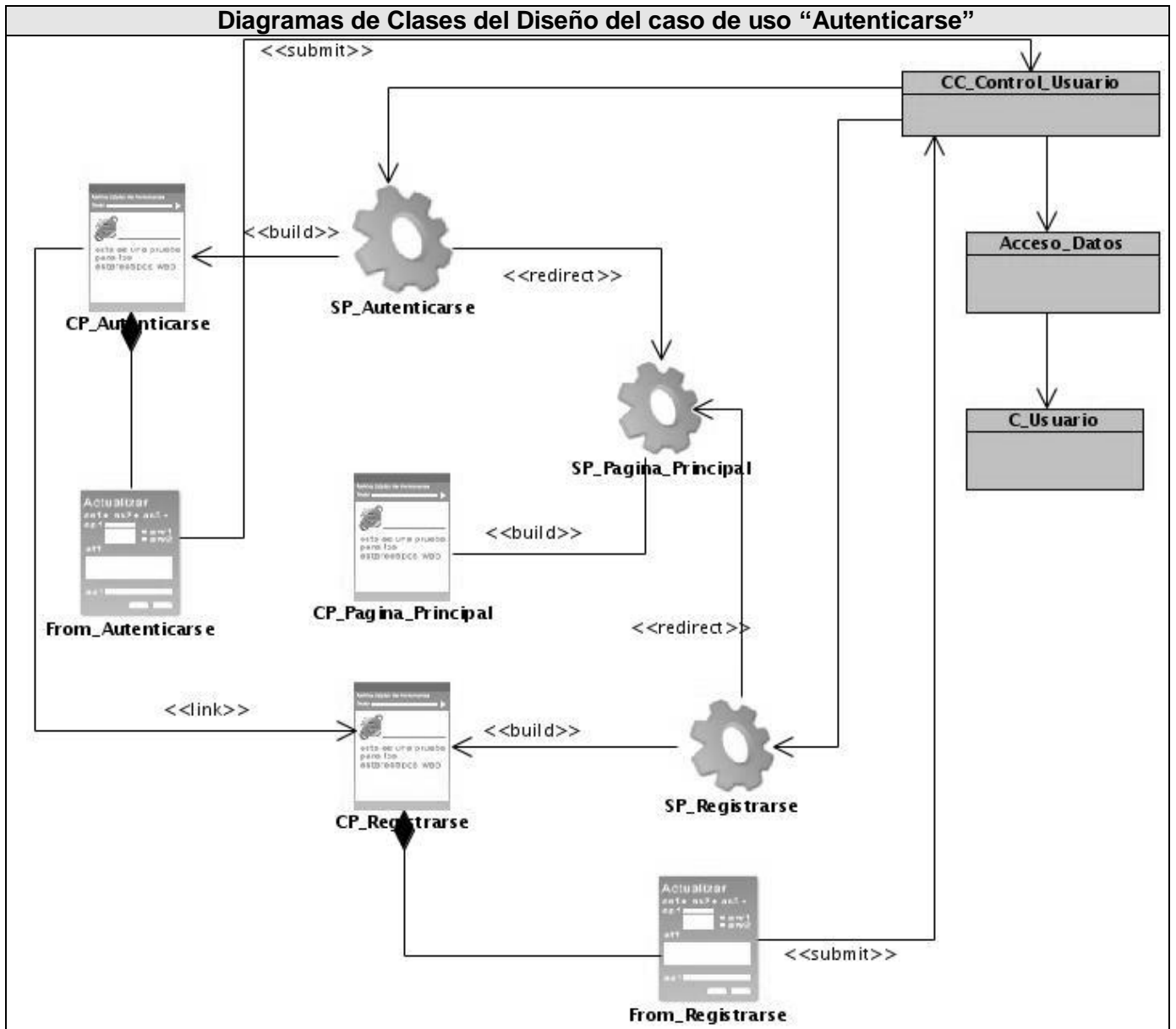


Figura 36: Diagramas de Clases del Diseño del caso de uso "Autenticarse"

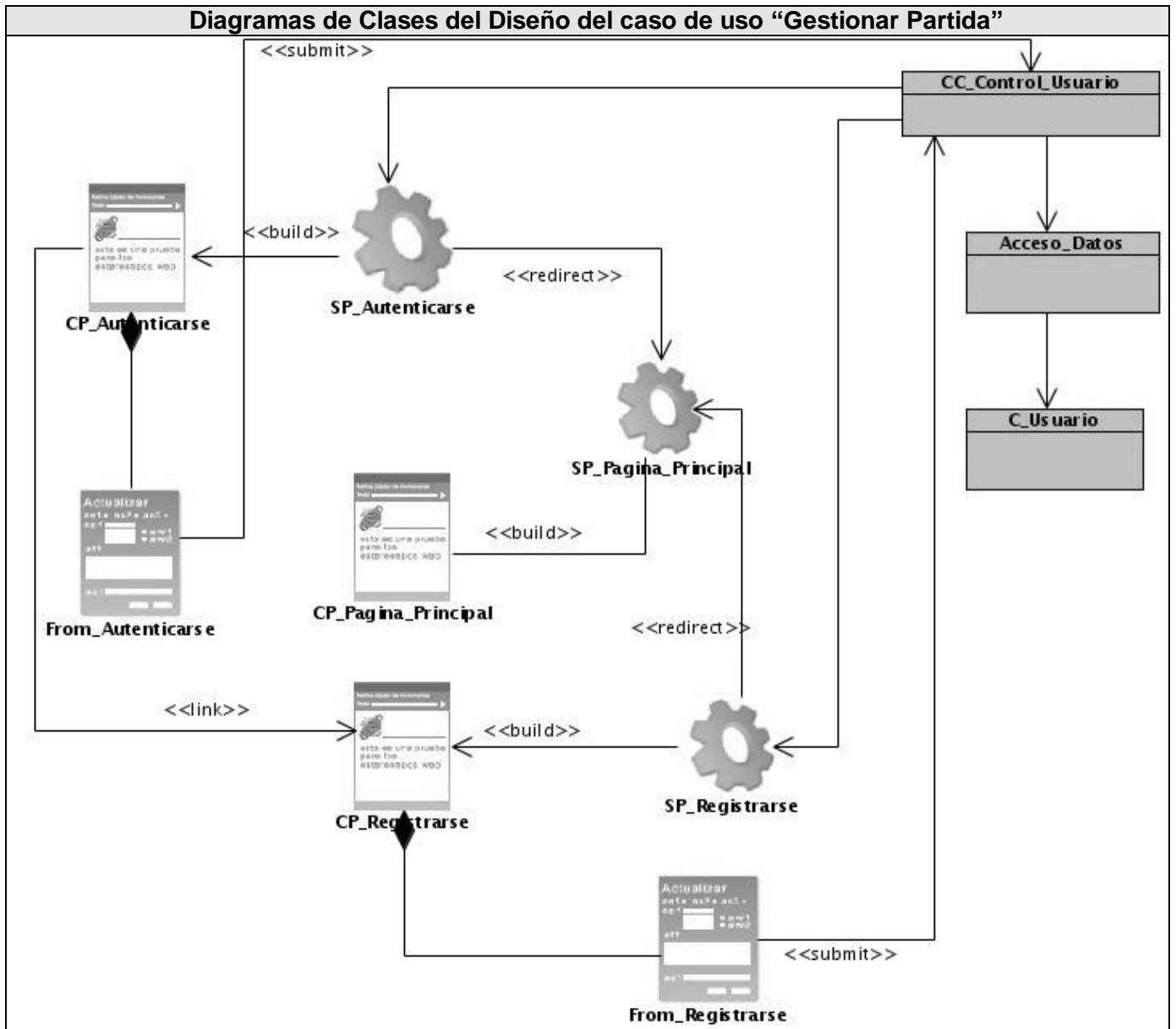


Figura 37: Diagramas de Clases del Diseño del caso de uso “Gestionar Partida”

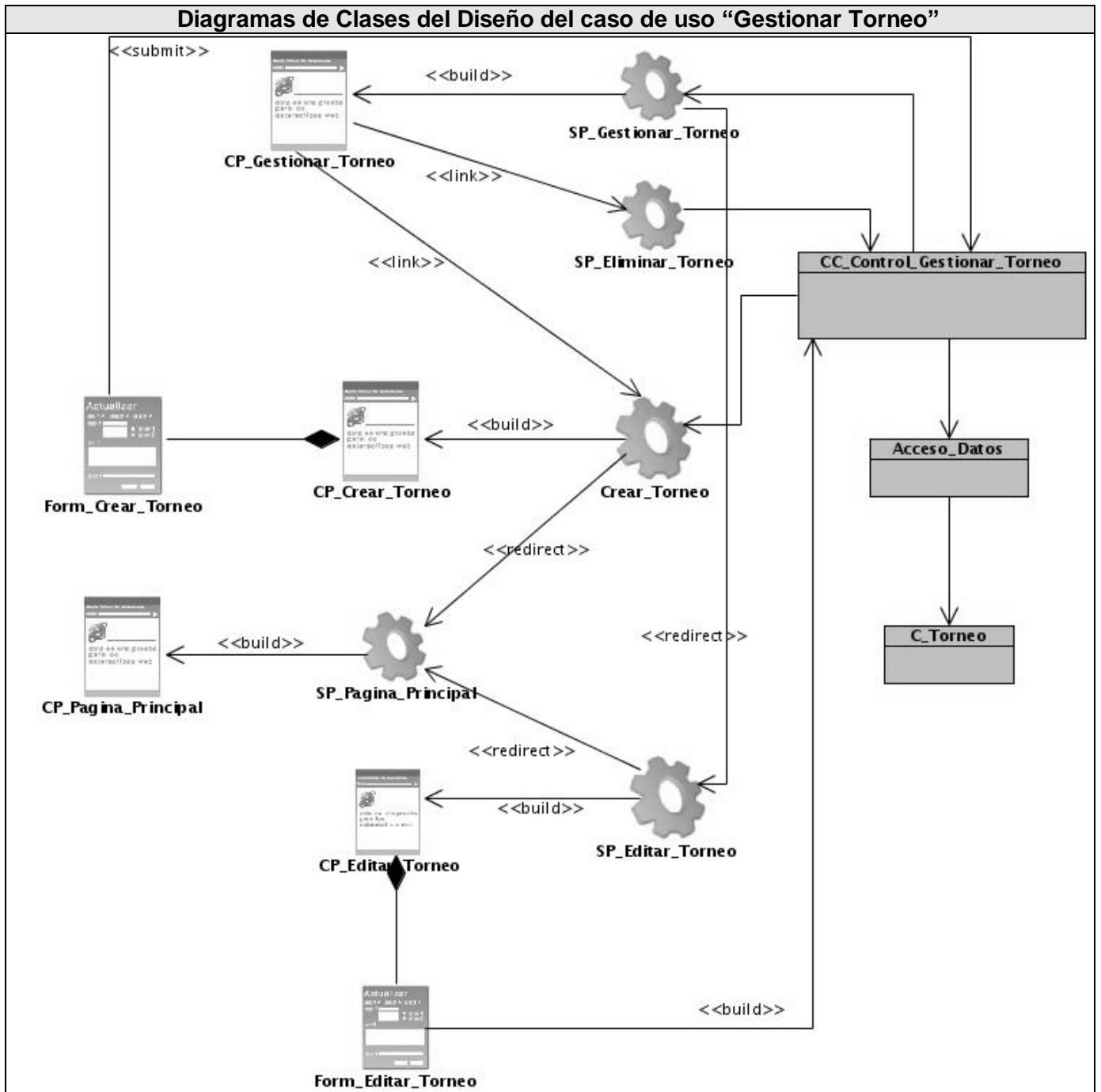


Figura 38: Diagramas de Clases del Diseño del caso de uso "Gestionar Torneo"



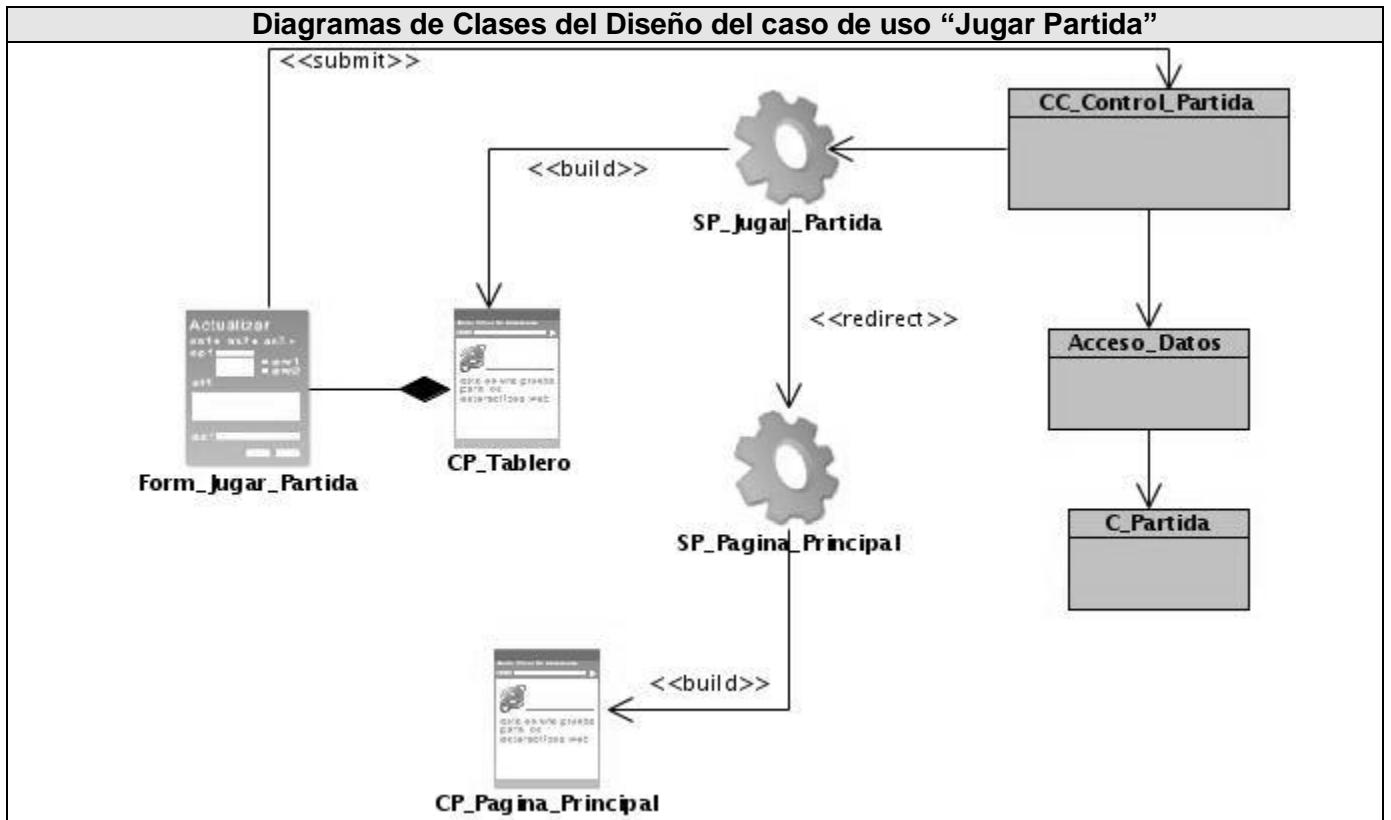


Figura 39: Diagramas de Clases del Diseño del caso de uso “Jugar Partida”

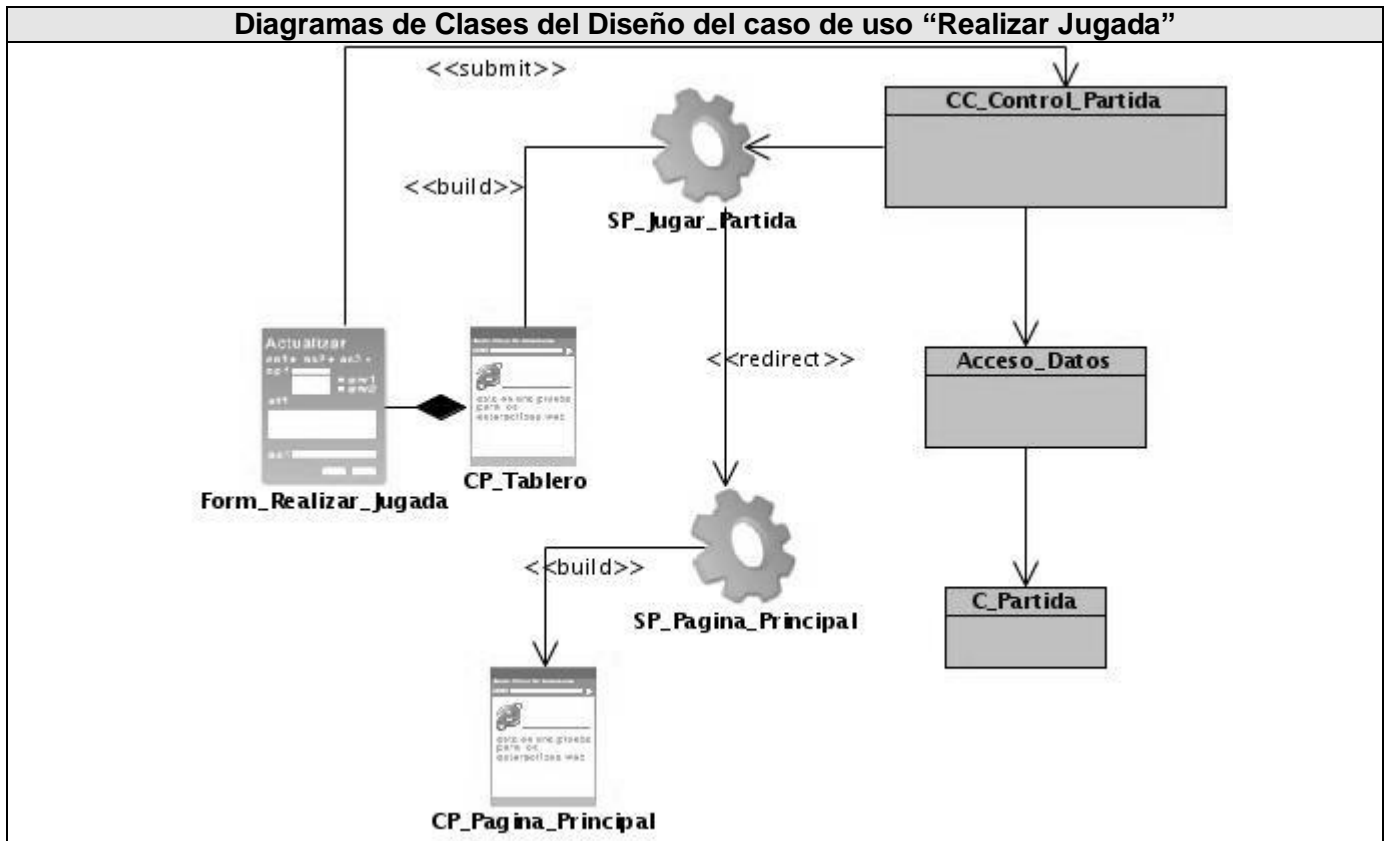


Figura 40: Diagramas de Clases del Diseño del caso de uso "Realizar Jugada"

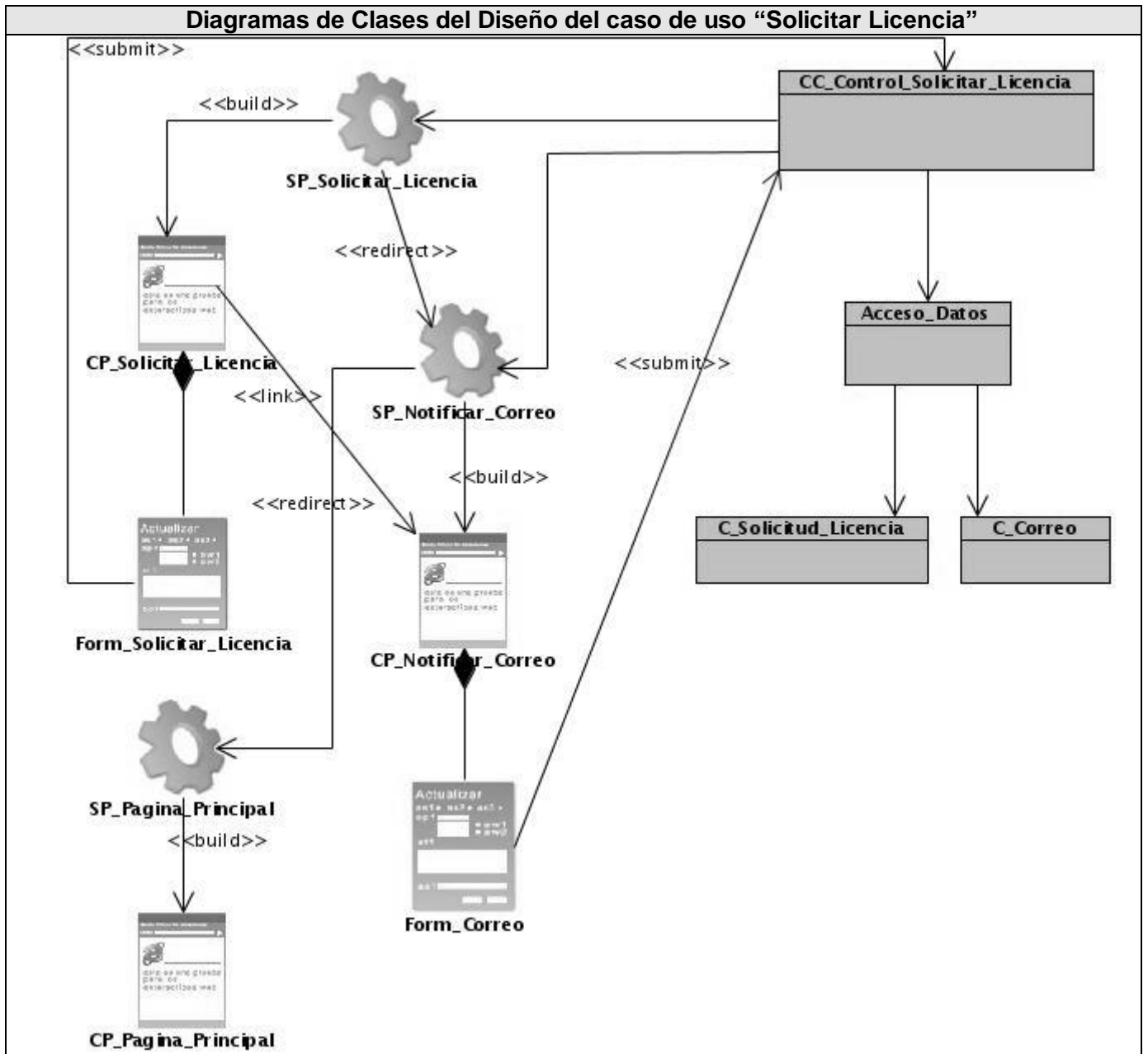


Figura 41: Diagramas de Clases del Diseño del caso de uso "Solicitar Licencia"

### 3.5 Diseño de la base de datos.

La base de datos es el sistema utilizado para el almacenamiento de datos y acceso controlado a los datos almacenados. En este epígrafe se muestra el diseño de la base de datos del sistema propuesto a través del diagrama de clases persistentes y el esquema de la base de datos generados a partir de este, el modelo entidad relación.

### 3.5.1 Diagrama de clases persistentes.

Las clases persistentes son las clases que necesitan ser capaz de guardar su estado en un medio permanente, la necesidad de guardar su estado esta dado por el almacenamiento físico permanente de la información de la clase, para la copia de seguridad en caso del fracaso del sistema, o para el intercambio de información. A continuación se muestra el diagrama de clases persistentes.

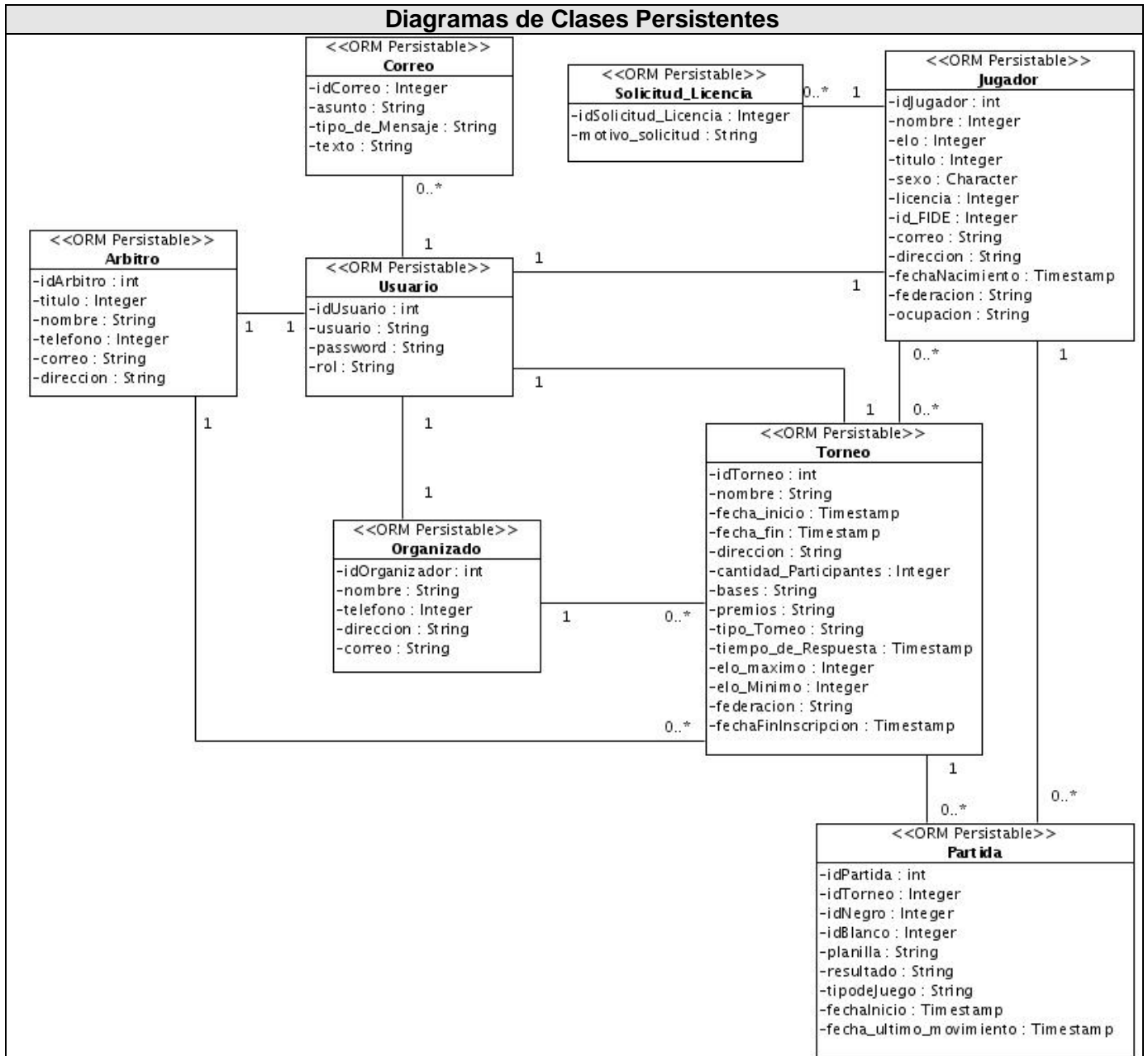


Figura 42: Diagramas de Clases Persistentes

### 3.5.2 Modelo entidad relación.

El modelo entidad relación describe la representación lógica y física de datos persistentes en el sistema. A continuación se muestra el modelo entidad relación.

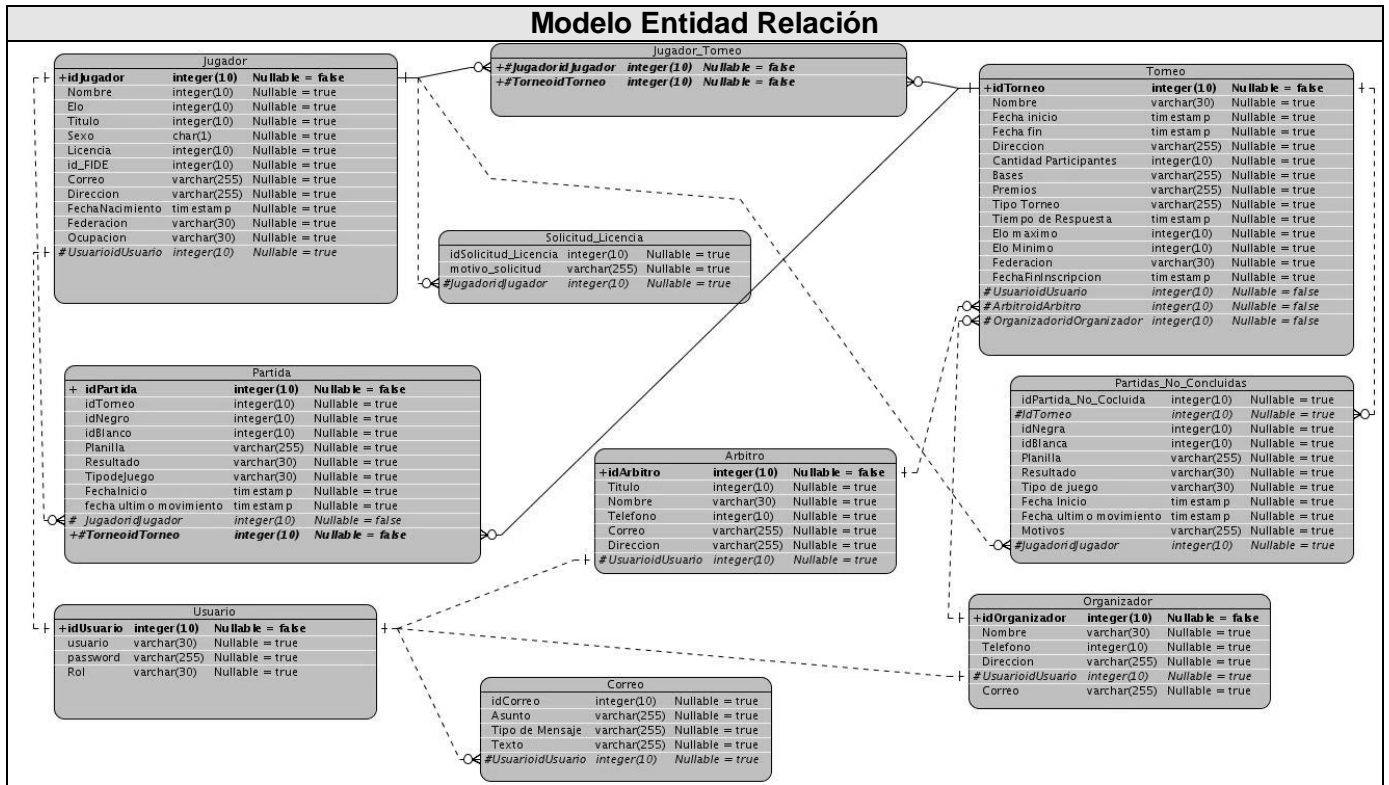


Figura 43: Modelo Entidad Relación

### 3.6 Conclusiones.

En este capítulo se mostró como a través del análisis y diseño, se transformó el modelo de análisis en un modelo de diseño, es decir, en una estructura de clasificadores y realizaciones de casos de uso. Además se construyeron varios artefactos para llevar a cabo el proceso de implementación del sistema. Se utilizaron diagramas de clases Web para explicar la lógica del negocio del sistema. También fue presentado el diagrama de clases persistentes de la base de datos, que contiene la información física que se utilizó para la construcción de la aplicación, así como el modelo de entidad relación que describe la representación lógica y física de datos persistentes en el sistema. En este momento ya se tiene confeccionada completamente la propuesta que trae este trabajo.

## **CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD.**

### **4.1 Introducción.**

Ante todo proyecto a desarrollar es preciso estimar y hacer el estudio de factibilidad para tener una medida del costo, cantidad de personas, esfuerzo y tiempo necesario para llevar a cabo dicho proyecto. En este capítulo se ofrece una descripción de la estimación del proyecto, así como los costos asociados al mismo, los beneficios tangibles e intangibles que reportaría su elaboración y el análisis entre los costos y los beneficios, haciendo una evaluación al final de la factibilidad del desarrollo del sistema propuesto.

### **4.2 Técnicas de Estimación.**

Las Técnicas de Estimación surgen tras la necesidad de planificar y controlar el esfuerzo, costo y tiempo de un proyecto de software. Las variables a estimar son:

- Esfuerzo: tiempo que necesita una persona para trabajar en el desarrollo del proyecto (hombres/mes, hombres/días, hombres/horas).
- Tiempo, Cantidad de Personas: recursos necesarios para desarrollar el software.
- Costo: relacionado con el software.
- Tamaño: la magnitud del software expresada en miles de líneas de código fuente (MLCF).

Existen varias alternativas para estimar dichas variables. Entre ellas están por “Puntos de Función ajustados” y por “Puntos de Casos de Uso”, ambas basadas en las funciones y también se puede contar con el Método Constructive Cost Model (COCOMO), alternativa basada en el tamaño.

### **4.3 Planificación basada en casos de usos.**

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

#### 4.3.1 Cálculo de Puntos de Casos de Uso sin Ajustar.

**UUCP = UAW + UUCW:** ecuación para el cálculo de Puntos de Casos de Uso sin ajustar.

Donde:

- **UUCP:** Puntos de Casos de Uso sin ajustar.
- **UAW:** Factor de Peso de los Actores sin ajustar.
- **UUCW:** Factor de Peso de los Casos de Uso sin ajustar.

#### Factor de Peso de los Actores sin Ajustar (UAW).

El cálculo de este valor se realiza mediante un análisis de la cantidad de Actores existentes en el sistema y la complejidad de cada uno de ellos. Para establecer la complejidad de los Actores se tiene en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

**Tabla 18: Factor de Peso de los Actores sin Ajustar (UAW)**

Tipo de Actor	Descripción	Factor de Peso	Cantidad * Factor de Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1	0x1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	1x2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	3x3

El factor de peso de los actores sin ajustar sería:

$$UAW=0x1 + 1x2 + 3x3=11.$$

#### Factor de Peso de los Casos de Uso sin Ajustar (UUCW).

El cálculo de este valor se realiza analizando la cantidad de Casos de Uso que existen en el sistema y la complejidad de cada uno de ellos. Para establecer la complejidad de los Casos de Uso se tiene en

cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción está representada por uno o más pasos del flujo normal de eventos del Caso de Uso donde se efectúa la secuencia de actividades completa o no se efectúa ninguna. En este caso se cuenta con 19 casos de usos.

**Tabla 19: Factor de Peso de los Casos de Uso sin Ajustar (UUCW)**

Tipo de Caso de Uso	Descripción	Factor de Peso	Cantidad * Factor de Peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5	17x5
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10	2x10
Complejo	El Caso de Uso contiene más de 8 transacciones.	15	0x15

El factor de peso de los Casos de Usos sin ajustar sería:  
 $UUCW = 17 \times 5 + 2 \times 10 + 0 \times 15 = 105$

El resultado de los Puntos de Casos de Uso sin ajustar serían:  
 $UUCP = UAW + UUCW = 11 + 105 = 116$ .

#### 4.4 Cálculo de Puntos de Caso de Uso Ajustados.

Cuando se tiene el valor de los Puntos de Casos de Usos sin Ajustar se debe ajustar el mismo mediante la ecuación:  $UCP = UUCP \times TCF \times EF$ .

Donde:

- **UCP:** Puntos de Casos de Uso ajustados.
- **UUCP:** Puntos de Casos de Uso sin ajustar.
- **TCF:** Factor de complejidad técnica.
- **EF:** Factor ambiente.



**Factor de Complejidad Técnica (TCF).**

El cálculo de este coeficiente se realiza mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema, los cuales se cuantifican con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

**Tabla 20: Factor de Complejidad Técnica (TCF)**

<b>Factor</b>	<b>Descripción</b>	<b>Peso</b>	<b>Valor asignado</b>	<b>Comentario</b>
T1	Sistema distribuido.	2	5	Es sistema es distribuido
T2	Tiempo de respuesta	1	3	En ocasiones es necesario esperar por entradas externas de los usuarios
T3	Eficiencia del usuario final	1	5	Escasas restricciones de eficiencia
T4	Procesamiento interno complejo	1	0	No se efectúan cálculos complejos
T5	El código debe ser reutilizable	1	4	Se requiere que el código sea reutilizable
T6	Facilidad de instalación	0.5	1	Escasos requerimientos de facilidad de instalación
T7	Facilidad de uso	0.5	5	El sistema es fácil de usar
T8	Portabilidad	2	5	Se requiere que le sistema sea portable
T9	Facilidad de cambio	1	4	Si se producen cambios en el sistema se incide en varios gastos
T10	Concurrencia	1	5	Utilización del sistema por

				varios usuarios
T11	Incluye objetivos especiales de seguridad	1	5	Requiere autenticación del usuario
T12	Provee acceso directo a terceras partes	1	2	Debe poder acceder a los datos de los demás módulos
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	0	El sistema es fácil de usar

El Factor de complejidad técnica sería:

$$TCF = 0.6 + 0.01 \times \Sigma (\text{Peso} \times \text{Valor asignado})$$

$$TCF = 0.6 + 0.01 \times \Sigma (2 \times 5 + 1 \times 3 + 1 \times 5 + 1 \times 0 + 1 \times 4 + 0.5 \times 1 + 0.5 \times 5 + 2 \times 5 + 1 \times 4 + 1 \times 5 + 1 \times 5 + 1 \times 2 + 1 \times 0)$$

$$TCF = 0.6 + 0.01 \times 51$$

$$TCF = 1.11.$$

#### Factor de ambiente (EF).

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo del sistema son los factores que se contemplan en el cálculo del Factor de ambiente. Se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Tabla 3.4 Factor de ambiente (EF)

Factor	Descripción	Peso	Valor asignado	Comentario
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	4	Se está bastante familiarizado con el modelo
E2	Experiencia en la aplicación.	0.5	0	El tiempo trabajado en la aplicación ha sido bastante
E2	Experiencia en orientación a objetos	1	4	La programación es

				orientada a objeto
E4	Capacidad del analista líder.	0.5	4	Se tiene capacidad media
E5	Motivación.	1	5	Se esta completamente motivado
E6	Estabilidad de los requerimientos.	2	4	Estabilidad media se esperan varios cambios
E7	Personal part-time.	-1	0	Personal full-time
E8	Dificultad del lenguaje de programación.	-1	4	El lenguaje utilizado es PHP5

El Factor de ambiente seria:

$$EF = 1.4 - 0.03 \times \sum_{i=0}^n (Peso_i \times Valor\ asignado_i)$$

$$EF = 1.4 - 0.03 \times (1.5 \times 4 + 0.5 \times 0 + 1 \times 4 + 0.5 \times 4 + 1 \times 5 + 2 \times 4 + (-1 \times 0) + (-1 \times 4))$$

$$EF = 1.4 - 0.03 \times 21$$

$$EF = 0.77$$

Como resultado final del cálculo de los Puntos de Casos de Uso Ajustados tenemos:

$$UCP = UUCP \times TCF \times EF$$

$$UCP = UUCP \times 1.11 \times 0.71$$

$$UCP = 137 \times 1.13 \times 0.77 = 100.9.$$

#### 4.5 Calcular el Esfuerzo del Flujo de Trabajo Implementación.

$$E = UCP \times CP$$

Donde:

**E:** Esfuerzo estimado en horas-hombre.

**UCP:** Puntos de Casos de Uso ajustados.

**CF:** Factor de conversión.

Para calcular el factor de convención necesario para hallar el esfuerzo se emplearon los siguientes criterios:

- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6. Resultado: 0
- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8. Resultado: 1
- Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.
- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

El Factor de Convención sería: 20 horas-hombre.

Calculando el esfuerzo necesario para el desarrollo de los casos de uso como:

$$E = UCP \times CP = 100.9 \times 20 = 2018 \text{ Horas/Hombre}$$

Es importante tener en cuenta que éste método proporciona una estimación del esfuerzo en horas\_hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso. Finalmente, para obtener una estimación más completa de cuánto durará el proyecto en total, se debe añadir a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Para ello se puede tener en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación:

**Tabla 21: Criterio de la distribución del esfuerzo**

Actividad	Porcentaje	Horas/Hombre
Análisis	20%	879.2
Diseño	30%	1318.8
Programación	50%	2198
Total	100%	4396

**Convirtiéndola hombres-mes.**

Teniendo en cuenta que la jornada de trabajo diaria es de 8 horas y que en el mes se trabaja 24 días aproximadamente, se llega a la conclusión que una persona trabaja en 1 mes 192 horas, por lo tanto:

$$Et = E \text{ (Horas-Hombres)} / 192 \text{ horas-mes}$$

$$\text{Quedando, } Et = 4396 \text{ Horas-Hombres} / 192 \text{ Horas-Mes} = 22 \text{ Mes-Hombres}$$

En el proyecto trabajaran dos hombres por lo tanto el tiempo de desarrollo es:

$$\text{Tiempo de desarrollo} = Et / \text{cantidad de hombres}$$

$$\text{Tiempo de desarrollo} = 22 / 2 = 11 = 11 \text{ meses}$$

**Salario.**

Para determinar el salario mensual se tiene en cuenta la posibilidad de que los desarrolladores del sistema pueden ser ingenieros recién graduados, por lo que se toma como salario mensual promedio:

**\$225.**

**4.6 Costo.**

Teniendo en cuenta el salario promedio de un ingeniero, el cual se definió anteriormente, y que el proyecto se desarrollara por dos personas, hallar el costo total, sería:

$$Ct = \text{Salario Mensual} * \text{Cantidad de Hombres} * \text{Tiempo de Desarrollo}$$

$$Ct = \$225 \times 2 \times 11 = \$4950$$

**4.7 Beneficio tangible e intangible.****4.7.1 Tangibles.**

El producto Análisis y diseño del modulo ajedrez por correspondencia del proyecto infodrez y ya con su respectiva implementación que se elaborara auto seguido de este trabajo tiene como objetivo poner a disposición de todos los jugadores de ajedrez por correspondencia una aplicación que le facilitara realizar una partida de esta modalidad del ajedrez desde sus ordenadores y con una mayor motivación, así como también les permitirá el control y gestión de torneos y jugadores al organizador del torneo y al director del torneo respectivamente que también serán usuarios de este Sistema. Se puede apreciar que se está en presencia de un producto de distribución gratuita, por lo que no se considera mencionar beneficios económicos.

Se puede decir que el costo por desarrollar el producto es \$4950(MN) moneda nacional y \$198(CUC) convertible.

**4.7.2 Intangibles.**

En este aspecto hay que destacar beneficios significativos, tales como:

- Control y registro del resultado de cada jugada, cada partida y cada torneo.
- Almacenamiento de información de los jugadores del torneo.
- Mayor orden a la hora de la gestión de un torneo por correspondencia.
- Acceso casi inmediato a la información almacenada
- Visualización de los torneos existentes.
- Mayor motivación e interés en el desarrollo de un torneo de ajedrez por correspondencia.

#### **4.8 Análisis costo-beneficio.**

Debe analizarse el costo de desarrollar la aplicación contra los beneficios y concluir si la aplicación es factible o no.

El principal aporte del Análisis y diseño del modulo ajedrez por correspondencia que con su implementación se convertirá en el modulo gestor de Información para el Ajedrez por Correspondencia del proyecto Infodrez está en el beneficio social que representa para la UCI, como para el país; ya que puede de esta forma contar con un avanzado sistema automatizado que lleva el control de una gran fisión de este juego universal que agrupa en su seno, a todos los amantes de esta modalidad del ajedrez que deseen participar en un torneo de ajedrez por correspondencia. De esta forma también pudiera representar un ahorro de capital monetario ya que no se hace necesaria que los usuarios de este juego tengan que pagar para jugar en los servidores internacionales. Luego de un análisis de los costos se llega a la conclusión que la realización de este proyecto requiere de pocos recursos y de un presupuesto pequeño comparado con lo que significaría la compra de alguna otra herramienta. Sumado a esto, son considerables los beneficios que se reportan con el uso de la aplicación que se presenta. Por tanto se considera que es factible desarrollar el proyecto.

#### **4.9 Conclusiones.**

En el presente capítulo se demostró que la implementación del sistema es factible estimándose un costo de \$4950 MN y \$198 CUC para su desarrollo por un tiempo de 11 meses, trabajando dos personas y un esfuerzo de 22 Mes-Hombres y un salario mínimo de \$225. Además Se expusieron los beneficios tangibles e intangibles que traen consigo el desarrollo y puesta en marcha del sistema propuesto.

## CONCLUSIONES.

Después de finalizado la propuesta para un sistema de gestión de un torneo de ajedrez por correspondencia se han llegado a las siguientes conclusiones.

- Se realizó un estudio de los principales sistemas que gestionan las partidas por correspondencia existentes en Cuba el Mundo.
- El sistema modelado soluciona los principales problemas que presenta la Cátedra de ajedrez de la universidad con el juego del ajedrez por correspondencia y su funcionamiento reporta beneficios considerables para los procesos de una partida de la modalidad del ajedrez por correspondencia.
- El resultado de este trabajo proporciona una visión más ampliada y detallada de los requerimientos del sistema para el juego del ajedrez por correspondencia.

Se puede concluir que se le ha dado cumplimiento al objetivo propuesto ya que se diseñó una aplicación informática para automatizar los procesos de gestión de un torneo de ajedrez por correspondencia, que facilita un mejor entendimiento para los diseñadores de base de datos e implementadores.

**RECOMENDACIONES.**

Los objetivos de este trabajo han sido cumplidos, pero durante el desarrollo del mismo surgieron ideas que se recomiendan implementarse en el futuro, de forma que se logre una mayor eficiencia en la aplicación. A continuación se citan algunas de ellas:

- Implementar del sistema propuesto siguiendo el patrón de diseño sugerido.
- Reutilizar la metodología, tecnologías y herramientas empleadas en la construcción de la aplicación, para el desarrollo de futuras iteraciones de la misma.
- Incorporar el uso del módulo de seguridad para establecer los niveles de acceso a los usuarios.
- Presentar a los especialistas de la FECAP el prototipo elaborado para recibir su visto bueno y en caso de ser necesario realizar el estudio correspondiente para asimilar los cambios que se pudieran proponer.
- En posteriores versiones del proyecto migrar el SGBD a PostgreSQL.
- Luego de realizarle un estudio más detallado y se hayan reforzado funcionalidades más abarcadoras el Sistema podrá Gestionar todo lo referente a un torneo de la modalidad ajedrez por correspondencia y no solo será una aplicación que se utilice en la Cátedra de Ajedrez de la UCI, también tendrá su utilidad a nivel de país.



**REFERENCIAS BIBLIOGRÁFICAS.**

1. Web, D. "Lenguajes de lado servidor." Mayo y Junio, 2007, [Disponible en: <http://www.desarrolloweb.com/articulos/243.php>].
2. Zend. "Zend.com." Mayo, 2007, [Disponible en: [http://www.zend.com/store/software/zend\\_studio/lp?engine=google&cmpg=zend\\_studio&k\\_id=zend\\_studio](http://www.zend.com/store/software/zend_studio/lp?engine=google&cmpg=zend_studio&k_id=zend_studio)]
3. MySQL AB. sitio Web de MySQL. Mayo 14, 2007. [Disponible en: <http://dev.mysql.com/doc/refman/5.0/es/what-is-mysql-ab.html>].
4. Federación Internacional de Ajedrez por Correspondencia (ICCF). [Disponible en: <http://es.wikipedia.org/wiki/ICCF>]
5. Ajedrez postal WIKIPEDIA. 2008. [Disponible en: [http://es.wikipedia.org/wiki/Ajedrez\\_postal](http://es.wikipedia.org/wiki/Ajedrez_postal)]
6. Yuniol Álvarez Betancourt, Sistema de gestión de información de jugadores y torneos de la Federación Cubana de Ajedrez Postal, Instituto Superior Politécnico "José Antonio Echeverría", Ciudad de La Habana, Cuba Junio, 2007, 132.
7. Francisco Acosta Ruiz ,Sitio del Ajedrez en Cuba. Historia, [Disponible en: <http://www.cuba.cu/ajedrez/>]
8. Jacobson, Ivar, Booch, Grady and Rumbaugh, James. El Proceso Unificado de Desarrollo de software. Addison-Wesley : s.n., 2000.
9. Editada por la Federación Cubana de Ajedrez y el Instituto Latinoamericano de Ajedrez (ISLA), Federación Cubana de Ajedrez Publicación quincenal . [Disponible en: <http://www.cuba.cu/ajedrez/boletin/postal27.html>]
10. Mugutu ,Arquitectura Cliente/Servidor , Lunes 19 de septiembre de 2005 [Disponible en:[http://www.mygnet.net/articulos/vb/introduccion\\_a\\_la\\_arquitectura\\_clienteservidor.82](http://www.mygnet.net/articulos/vb/introduccion_a_la_arquitectura_clienteservidor.82)]
11. Trygve Reenskaug Modelo Vista Controlador – Wikipedia 1979[Disponible en: [http://www.es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://www.es.wikipedia.org/wiki/Modelo_Vista_Controlador)]
12. Tecnología AJAX y sus diferentes técnicas de programación Feb 2008[Disponible en: <http://www.telepieza.com/wordpress/2008/02/08/historia-de-ajax-y-sus-diferentes-tecnicas-de-programacion/>]

13. Aplicación web – Wikipedia 2008 [Disponible en:  
[http://www.es.wikipedia.org/wiki/Aplicación\\_web](http://www.es.wikipedia.org/wiki/Aplicación_web)]
14. Script del lado del servidor Jul 2004 [Disponible en:  
[http://www.programacionweb.net/directorio/pagina/?url=es.wikipedia.org%2Fwiki%2FScript\\_del\\_lado\\_del\\_servidor](http://www.programacionweb.net/directorio/pagina/?url=es.wikipedia.org%2Fwiki%2FScript_del_lado_del_servidor)]
15. Servidor Web Enero 2005 [Disponible en:  
<http://www.casidiablo.net/%C2%BFque-es-un-servidor-web/>]
16. Metodologías de desarrollo de software Mayo del 2008 [Disponible en:  
[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)]
17. Lenguaje de Modelado [Disponible en:  
<http://www.monografias.com/trabajos16/lenguaje-modelado-unificado/lenguaje-modelado-unificado.shtml>]
18. El lenguaje de Gestión de Base de Datos. El lenguaje de Gestión de Base de Datos. Enero de 2007 [Disponible en:  
<http://www.utem.cl/web/lengsql.htm.>]
19. Visual Paradigm for UML. Visual Paradigm for UML. [En línea] Visual Paradigm. Febrero de 2007 [Disponible en:  
<http://www.visual-paradigm.com/product/vpuml>]
20. Larman, Craig. UML y Patrones Introducción al análisis y diseño orientado a objetos. UML y Patrones Introducción al análisis y diseño orientado a objetos. Junio 1999. [Disponible en:  
[http://alarcos.infcr.uclm.es/per/fgarcia/isoftware/doc/LabTr5\\_Multicapa.pdf.](http://alarcos.infcr.uclm.es/per/fgarcia/isoftware/doc/LabTr5_Multicapa.pdf.)]
21. Hernandis, José Alberto. Visual Paradigm for UML. Visual Paradigm for UML. Julio 04, 2005 [Disponible en:  
<http://www.versionzero.com/noticia/210/visual-paradigm-for-uml.>]
22. Jorge Ercoli, Arquitectura en 3 capas. Febrero 02, 2008 [Disponible en:  
<http://www.metodologiasdesistemas.blogspot.com/2008/02/desarrollar-en-3-capas-con.html>]
23. Luis Alberto Véliz , Técnica lado del cliente, [Disponible en:  
[http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ\\_2847/default.aspx](http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_2847/default.aspx)]
24. JavaScript. [Disponible en:  
<http://www.es.wikipedia.org/wiki/JavaScript>]

25. HyperText Markup Language[Disponible en:  
[http://www.es.wikipedia.org/wiki/Código\\_HTML](http://www.es.wikipedia.org/wiki/Código_HTML)]
26. Tutorial en castellano sobre XML, {Disponible en:  
<http://www.w3c.es/divulgacion/guiasbreves/tecnologiasXML>]
27. TiendaLinux.com, PostgreSQL.. Enero,20, 2007.[Disponible en:  
[http://soporte.tiendalinux.com/portal/Portfolio/postgresql\\_ventajas\\_html.](http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html.)]
28. HIETANIEMI, J. Perl, 2007d.[ Disponible en:  
<http://www.es.wikipedia.org/wiki/Perl>]
29. VÁZQUEZ, J. Desarrollo Web con PHP y MYSQL, 2007.Ediciones Anaya Multimedia.  
[Disponible en:  
<http://www.gnuinos.com/drupal/?q=node/84>]

**BIBLIOGRAFÍA.**

Aplicación web – Wikipedia 2008[Disponible en:  
[http://www.es.wikipedia.org/wiki/Aplicación\\_web](http://www.es.wikipedia.org/wiki/Aplicación_web)]

Ajedrez postal WIKIPEDIA. 2008. [Disponible en:  
[http://es.wikipedia.org/wiki/Ajedrez\\_postal](http://es.wikipedia.org/wiki/Ajedrez_postal)]

Editada por la Federación Cubana de Ajedrez y el Instituto Latinoamericano de Ajedrez (ISLA),  
Federación Cubana de Ajedrez Publicación quincenal . [Disponible en:  
<http://www.cuba.cu/ajedrez/boletin/postal27.html>]

El lenguaje de Gestión de Base de Datos. El lenguaje de Gestión de Base de Datos. Enero de  
2007[Disponible en:  
<http://www.utem.cl/web/lengsql.htm>.]

Francisco Acosta Ruiz ,Sitio del Ajedrez en Cuba. Historia, [Disponible en:  
<http://www.cuba.cu/ajedrez/>]

Francisco Acosta Ruiz ,Sitio del Ajedrez en Cuba. Historia, [Disponible en:  
<http://www.cuba.cu/ajedrez/>]

HIETANIEMI, J. Perl, 2007d.[ Disponible en:  
<http://www.es.wikipedia.org/wiki/Perl>]

Hernandis, José Alberto. Visual Paradigm for UML. Visual Paradigm for UML. Julio 04, 2005  
[Disponible en:  
<http://www.versionero.com/noticia/210/visual-paradigm-for-uml>.]

JavaScript. [Disponible en:  
<http://www.es.wikipedia.org/wiki/JavaScrip>]

Jacobson, Ivar, Booch, Grady and Rumbaugh, James. El Proceso Unificado de Desarrollo de software.  
Addison-Wesley : s.n., 2000.

La comunidad de usuarios de MySQL- junio 2007 [Disponible en:  
<http://www.mysql-hispano.org>]

Luis Alberto Véliz ,Técnica lado del cliente, [Disponible en:  
[http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ\\_2847/default.aspx](http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_2847/default.aspx)]

MySQL AB. sitio Web de MySQL. Mayo 14, 2007. [Disponible en: <http://dev.mysql.com/doc/refman/5.0/es/what-is-mysql-ab.html>].

Mesa, A, Rendón. A. "Herramientas CASE", 2005 [Disponible en: <http://www.monografias.com/trabajos24/herramientas-case/herramientas-case.shtml>]

Metodologías de desarrollo de software Mayo del 2008[Disponible en: [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)]

Mugutu ,Arquitectura Cliente/Servidor , Lunes 19 de septiembre de 2005 [Disponible en: [http://www.mygnet.net/articulos/vb/introduccion\\_a\\_la\\_arquitectura\\_clienteservidor.82](http://www.mygnet.net/articulos/vb/introduccion_a_la_arquitectura_clienteservidor.82)]

Tecnología AJAX y sus diferentes técnicas de programación Feb 2008[Disponible en: <http://www.telepieza.com/wordpress/2008/02/08/historia-de-ajax-y-sus-diferentes-tecnicas-de-programacion/>]

Trygve Reenskaug Modelo Vista Controlador – Wikipedia 1979[Disponible en: [http://www.es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://www.es.wikipedia.org/wiki/Modelo_Vista_Controlador)]

VÁZQUEZ, J. Desarrollo Web con PHP y MYSQL, 2007. Ediciones Anaya Multimedia. [Disponible en: <http://www.gnuinos.com/drupal/?q=node/84>]

Web, D. "Lenguajes de lado servidor." Mayo y Junio, 2007, [Disponible en: <http://www.desarrolloweb.com/articulos/243.php>].

Yuniol Álvarez Betancourt, Sistema de gestión de información de jugadores y torneos de la Federación Cubana de Ajedrez Postal, Instituto Superior Politécnico "José Antonio Echeverría", Ciudad de La Habana, Cuba Junio, 2007, 132.

## ANEXOS.

Tabla 22: Descripción del caso de uso “Permitir Unirse a Torneo”

Caso de Uso		Permitir Unirse a Torneo
<b>Actores</b>	Jugador (inicia).	
<b>Propósito</b>	El Jugador pueda unirse a cualquier torneo del sistema de su preferencia mediante una inscripción.	
<b>Resumen</b>	El CUS se inicia cuando el Jugador solicita unirse a un torneo de los existentes en la aplicación, mostrándole esta los torneos existentes y dando la posibilidad de que el jugador se inscriba en el de su preferencia.	
<b>Referencias</b>	RF 4.1, RF4.2	
<b>Precondiciones</b>	Deben existir torneos creados en el Sistema.	
<b>Poscondiciones</b>	El jugador queda inscrito en el torneo que desea listo para empezar a jugarlo.	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1- Jugador: Solicita unirse a un torneo del sistema.	2- El sistema realiza una consulta a la base de datos a la tabla de torneo y muestra los torneos que hay disponibles.	
3- Jugador: Selecciona el torneo al que quiere unirse.	4-El sistema muestra un formulario con los datos de inscripción de un torneo.	
5-Jugador: Llena el formulario mostrado por el sistema inscribiéndose así en el torneo.	6- El sistema registra los datos del jugador.	
	7- El caso de uso termina.	
<b>Prioridad</b>	Crítico	

Tabla 23: Descripción del caso de uso “Mostrar datos de la partida”

Caso de Uso		Mostrar datos de la Partida
<b>Actores</b>	Jugador (inicia).	
<b>Propósito</b>	El Jugador cuente con la posibilidad de observar los datos de cualquiera de sus partidas en el sistema.	
<b>Resumen</b>	El CUS se inicia cuando el Jugador solicita observar los datos de una de sus partidas, selecciona la partida que desea que el sistema muestre, este a su vez le muestra un formulario para que escoja los datos que desea que sean mostrados. El jugador selecciona los campos que desea que el sistema muestre.	
<b>Referencias</b>	RF 13.1	
<b>Precondiciones</b>	Deben existir partidas creadas en el sistema.	
<b>Poscondiciones</b>	El jugador observa de una de sus partidas los campos seleccionados.	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1- Jugador: Solicita ver datos de una partida.	2- El sistema muestra una interfaz con los datos de la partida.	
3- Jugador: Selecciona los datos que desea observar.	4-El sistema muestra los datos seleccionados por el jugador.	
	5- El caso de uso termina.	
<b>Curso alternativo de los eventos</b>		
<b>Prioridad</b>	Secundario	

Tabla 24: Descripción del caso de uso "Realizar Pareo"

Caso de Uso		Realizar Pareo
<b>Actores</b>	Organizador del Torneo (inicia).	
<b>Propósito</b>	El organizador de torneo realiza el pareo de un torneo con los jugadores del mismo porque así quedan conformadas las partidas del torneo.	
<b>Resumen</b>	El CUS se inicia cuando el Jugador solicita realizar el pareo de un torneo teniendo en cuenta los datos del jugador que le hacen falta para que así queden conformadas las partidas de dicho torneo.	
<b>Referencias</b>	RF 15.1	
<b>Precondiciones</b>	Deben estar inscritos todos los jugadores según la capacidad del torneo.	
<b>Poscondiciones</b>	Queda realizado el pareo actualizándose las tablas de los jugadores y conformadas las partidas	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1- Jugador: Solicita realizar pareo.	2- El sistema solicita a la base de datos en la tabla del jugador los datos necesarios para realizar el pareo. 3- El sistema efectúa el pareo 4- El sistema registra datos del pareo en la base de datos. 5- El sistema muestra el pareo realizado en el sistema. 6- El caso de uso termina.	
<b>Curso alterno de los eventos</b>		
	2- Si no se encuentran jugadores inscritos que requiere la capacidad del torneo el sistema enviara un mensaje informando que no se puede realizar el pareo porque el torneo todavía esta disponible para mas inscripciones. 3- El caso de uso finaliza.	
<b>Prioridad</b>	Secundario	



Tabla 25: Descripción del caso de uso “Gestionar Jugadores del torneo”

Caso de Uso		Gestionar Jugadores del Torneo
<b>Actores</b>	Director del Torneo (Arbitro principal) (inicia).	
<b>Propósito</b>	Que el Director del Torneo tenga la posibilidad de Gestionar los jugadores del torneos en el Sistema ya sea Actualizar, buscar o Eliminar un jugador.	
<b>Resumen</b>	El CUS se inicia cuando el Arbitro Principal del Torneo selecciona la opción de Gestionar Jugador, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema realiza la acción seleccionada por el Director del Torneo y termina el CUS.	
<b>Referencias</b>	RF11.1, RF11.2, RF11.3	
<b>Precondiciones</b>	Existe una necesidad de gestionar un jugador del torneo.	
<b>Poscondiciones</b>	Se Gestiona la acción seleccionada por el arbitro principal.	
<b>Curso normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1- Director del torneo: El caso de uso empieza cuando el organizador del torneo accede al vínculo Gestionar torneo.	2- El sistema muestra tres opciones actualizar jugador, buscar jugador, eliminar jugador.	
3- Director del Torneo: Selecciona una de las tres opciones.	4- El sistema muestra según la opción solicitada Actualizar Jugador: Sección Actualizar Jugador, Buscar Jugador: Sección Buscar Jugador, Eliminar Jugador: Sección Eliminar Jugador.	
	5- El caso de uso Termina.	

<b>Sección : Actualizar Jugador</b>	
1-Director del torneo: Solicita la opción de Actualizar Jugador.	
3-Director del Torneo: Selecciona los campos que desea actualizar del jugador.	2- El sistema muestra una interfaz donde le muestra al Director del torneo los campos que contiene un jugador.  4- El sistema envía los datos actualizados del Jugador a la base de datos.
	5- El caso de uso termina.
<b>Sección: Buscar Jugador</b>	
1- Director del torneo: Solicita buscar un jugador en el sistema.	
3-Director del Torneo: Llena el formulario de búsqueda según los campos deseados	2- El sistema muestra una interfaz con datos de selección de búsqueda.  4- El sistema realiza una consulta a la base de datos Solicitando al jugador según campos de selección.  5- El sistema muestra los datos del Jugador buscado.
	6- El caso de uso termina.
<b>Curso alternativo de los eventos</b>	
	4- Si no existe el jugador buscado en la base de datos del sistema se envía un mensaje informando que no existe el jugador con esas características. 5-El caso de uso finaliza.
<b>Sección : Eliminar Jugador</b>	

<p>1-Director del Torneo: Solicita la opción de eliminar un jugador del Torneo.</p> <p>4- Director del Torneo: Selecciona el Jugador que desea eliminar.</p>	<p>2- El sistema realiza una consulta a la base de datos, a la tabla de jugadores y solicita todos los jugadores inscritos hasta el momento.</p> <p>3- El sistema muestra una interfaz con el listado de jugadores existentes.</p> <p>5- El sistema elimina el jugador seleccionado de la base de datos.</p>
	6- El caso de uso termina.
<b>Curso alternativo de los eventos</b>	
	<p>3- Si no existen jugadores inscritos se envía un mensaje informando que no existen aun jugadores inscritos en los torneos.</p> <p>4-El caso de uso finaliza.</p>
<b>Prioridad</b>	Crítico

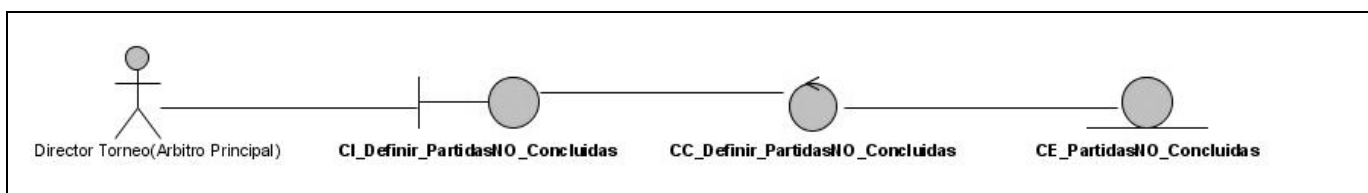


Figura 44: Diagrama de clases de análisis del caso de uso "Definir\_Partidas\_No\_Cocluidas"

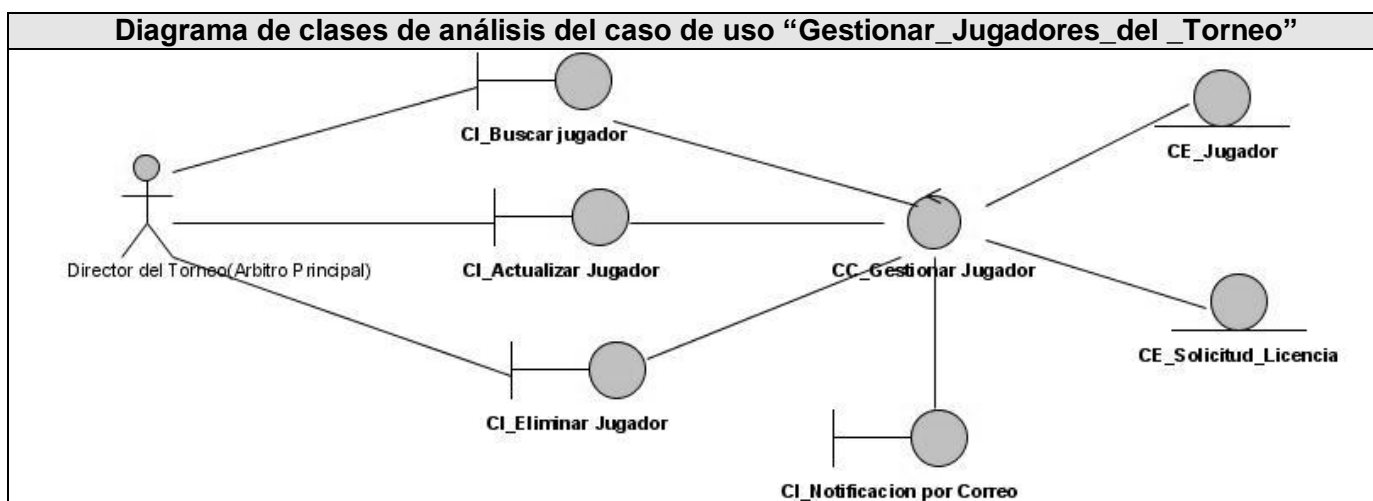


Figura 45: Diagrama de clases de análisis del caso de uso "Gestionar\_Jugadores\_del\_Torneo"

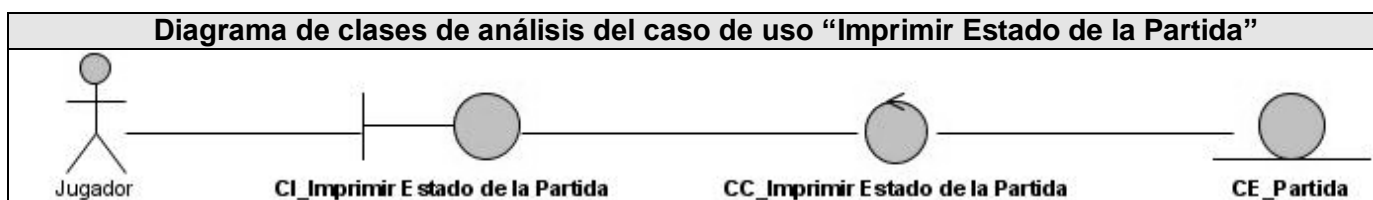


Figura 46: Diagrama de clases de análisis del caso de uso "Imprimir Estado de la Partida"

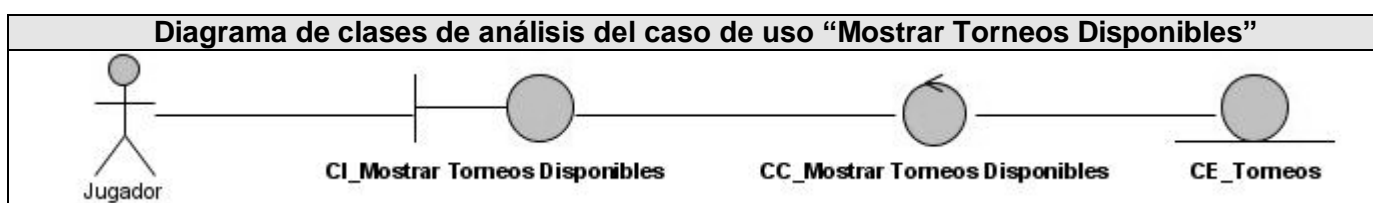


Figura 47: Diagrama de clases de análisis del caso de uso "Mostrar Torneos Disponibles"

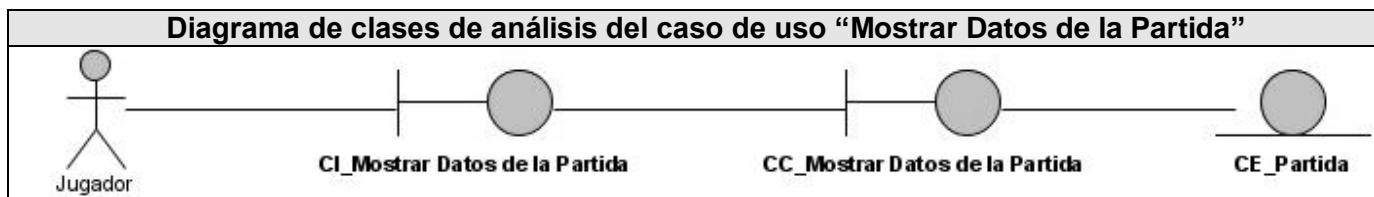


Figura 48: Diagrama de clases de análisis del caso de uso "Mostrar Datos de la Partida"

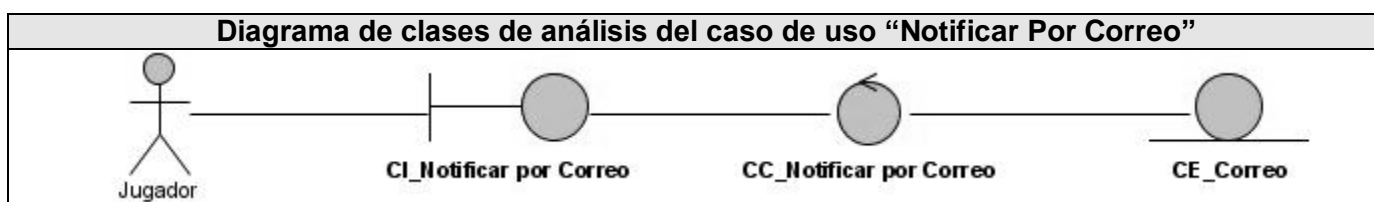


Figura 49: Diagrama de clases de análisis del caso de uso "Notificar Por Correo"

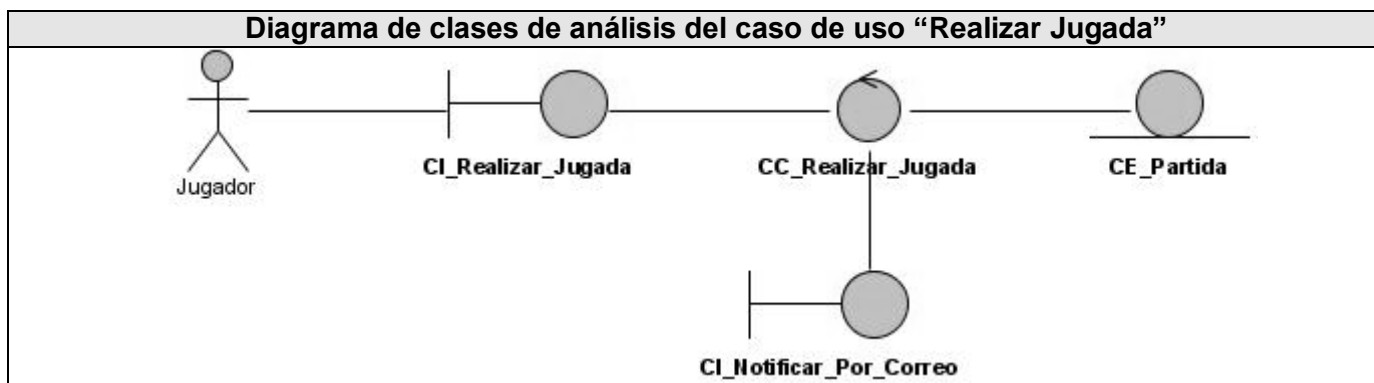


Figura 50: Diagrama de clases de análisis del caso de uso "Realizar Jugada"

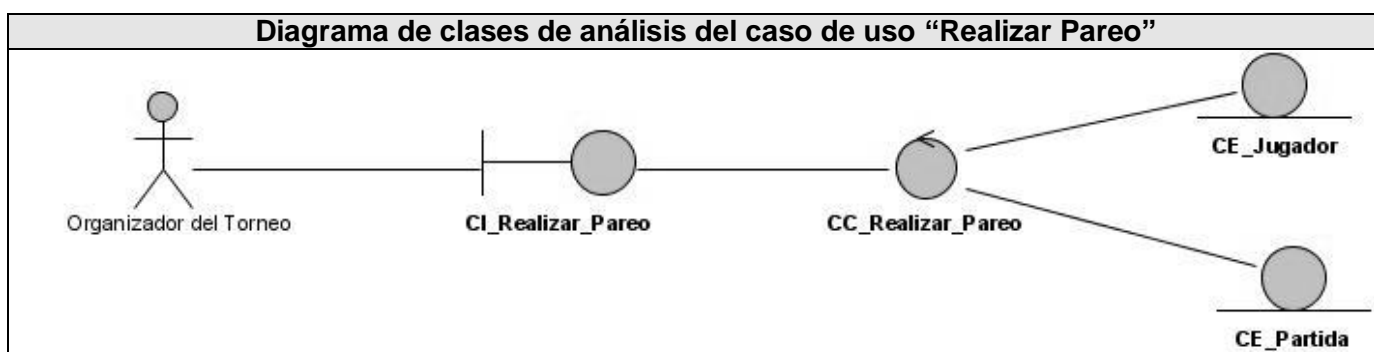


Figura 51: Diagrama de clases de análisis del caso de uso "Realizar Pareo"

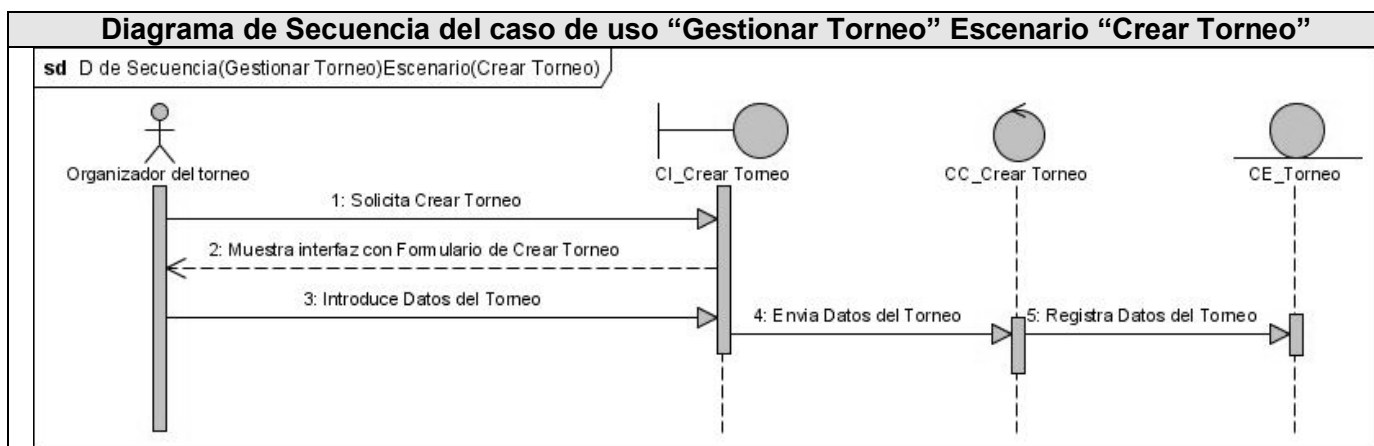


Figura 52: Diagrama de Secuencia del caso de uso "Gestionar Torneo" Escenario "Crear Torneo"

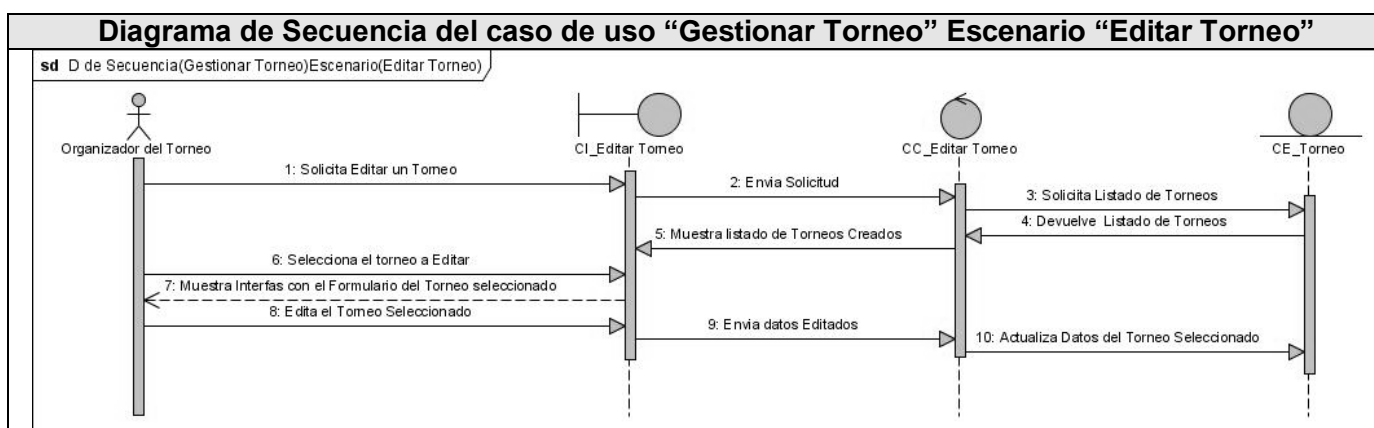


Figura 53: Diagrama de Secuencia del caso de uso "Gestionar Torneo" Escenario "Editar Torneo"

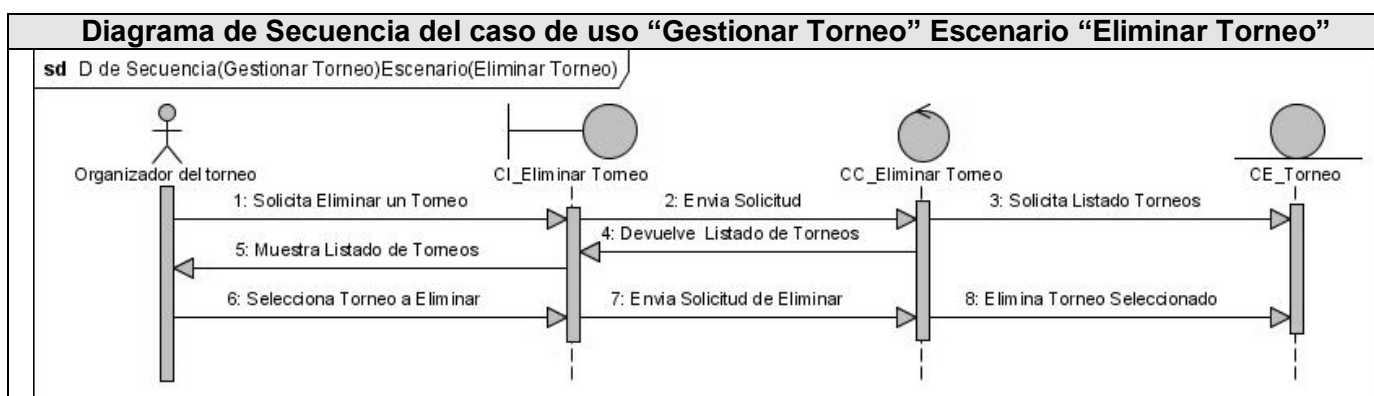


Figura 54: Diagrama de Secuencia del caso de uso "Gestionar Torneo" Escenario "Eliminar Torneo"

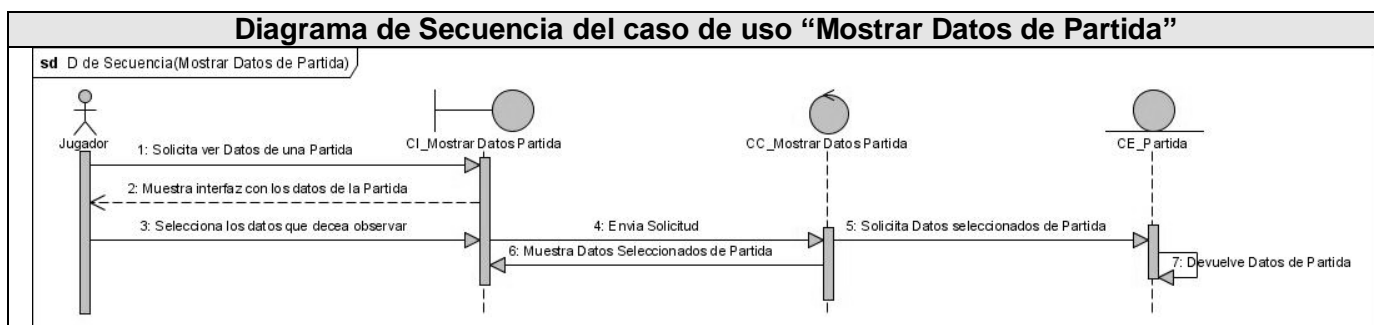


Figura 55: Diagrama de Secuencia del caso de uso "Mostrar Datos de Partida"

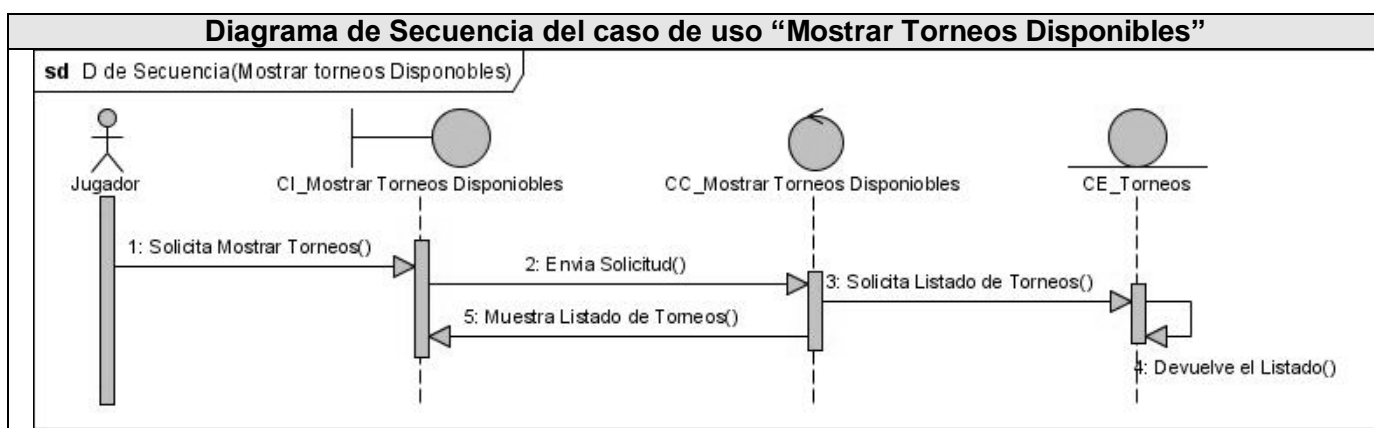


Figura 56: Diagrama de Secuencia del caso de uso "Mostrar Torneos Disponibles"

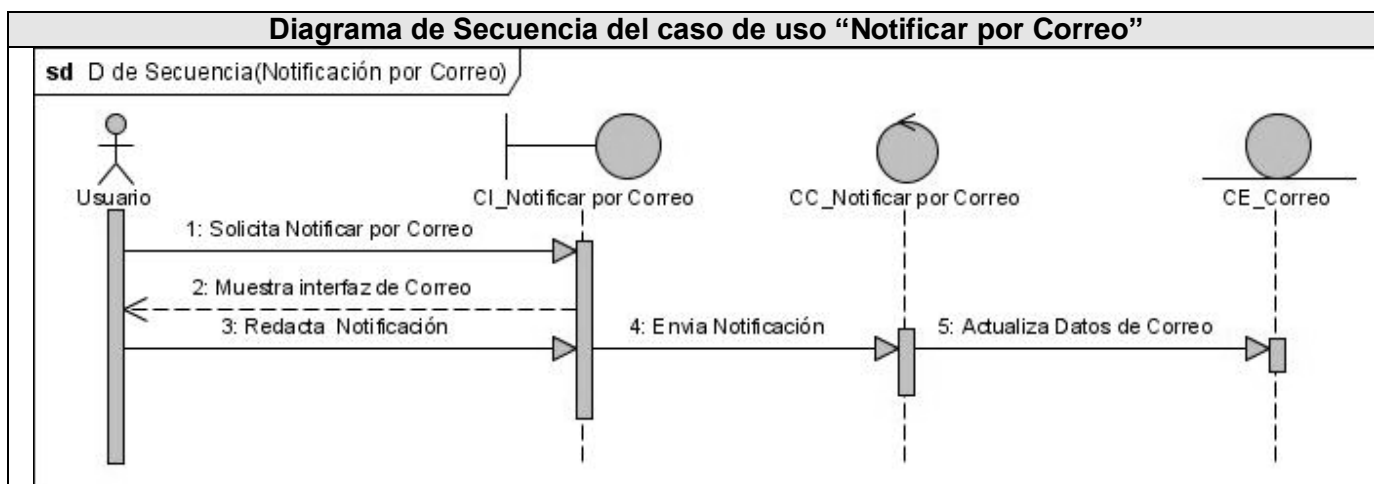


Figura 57: Diagrama de Secuencia del caso de uso "Notificar por Correo"

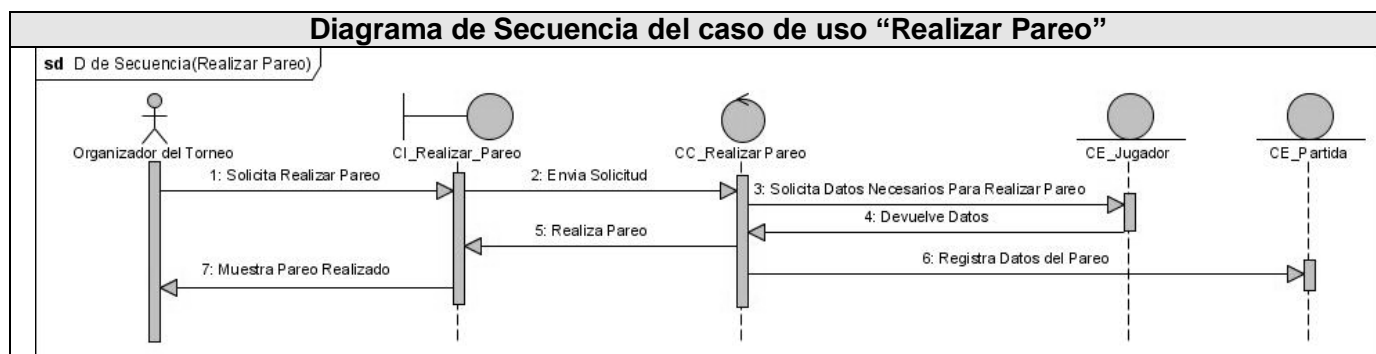


Figura 58: Diagrama de Secuencia del caso de uso "Realizar Pareo"

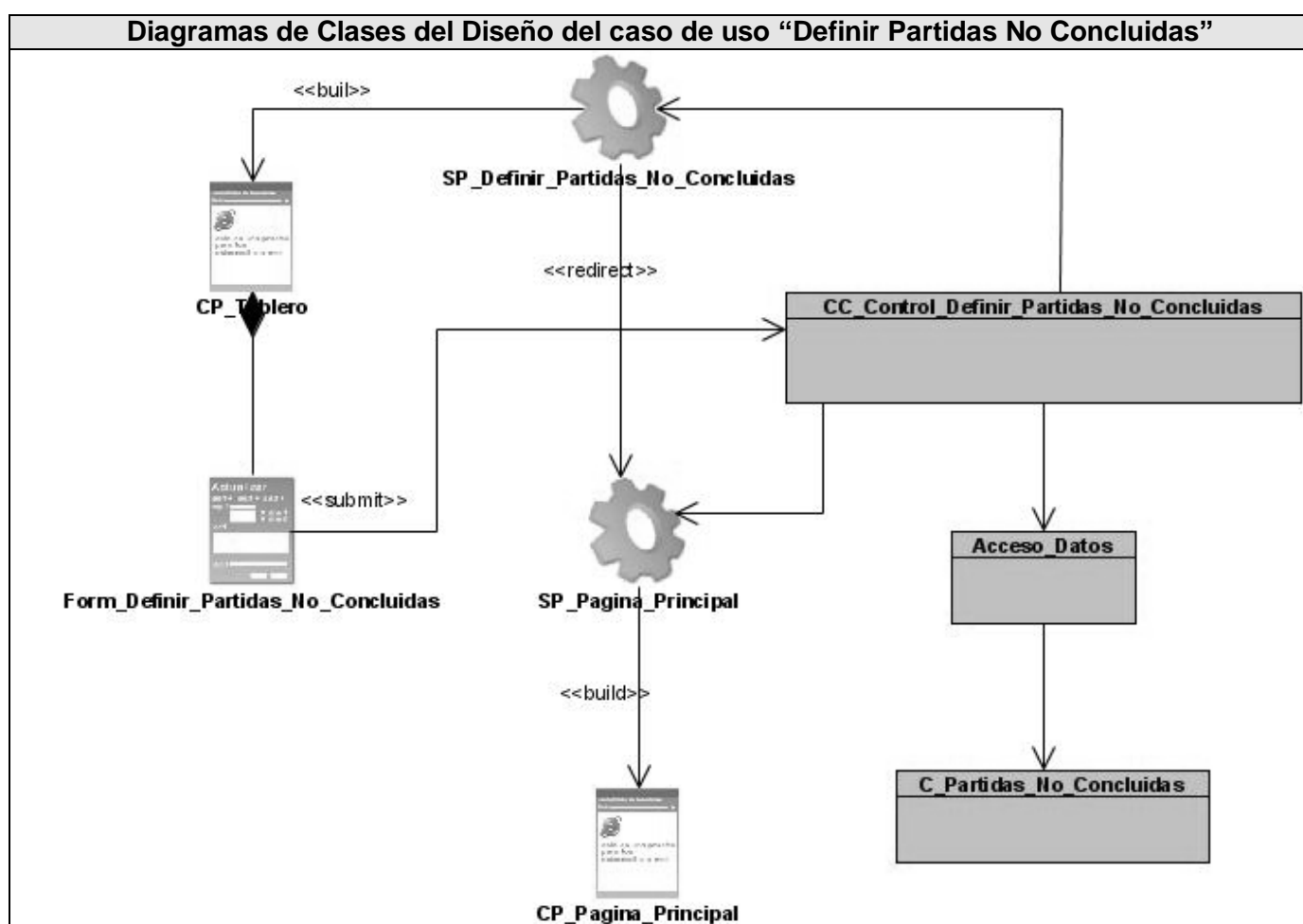


Figura 59: Diagramas de Clases del Diseño del caso de uso "Definir Partidas No Concluidas"



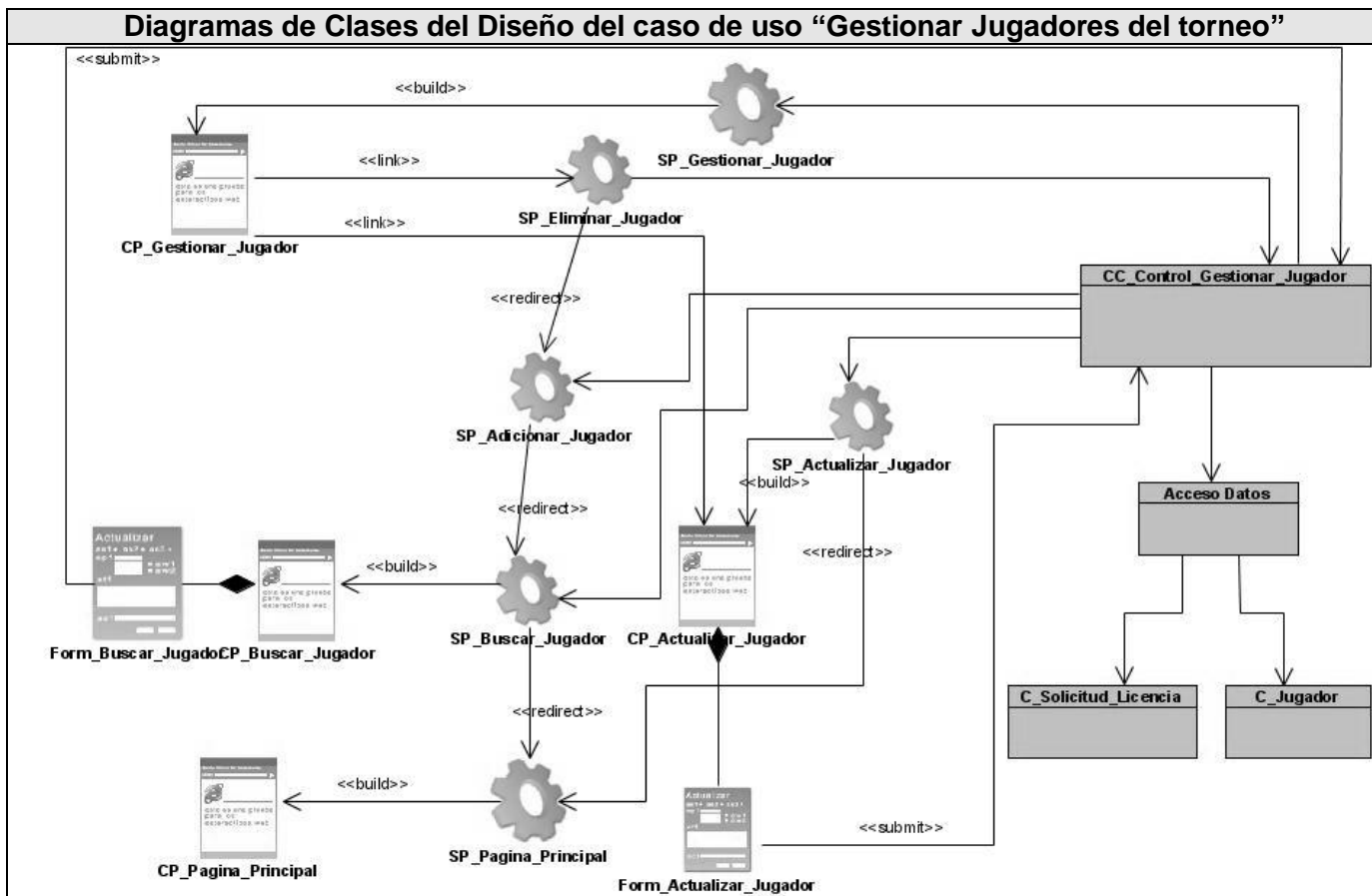


Figura 60: Diagramas de Clases del Diseño del caso de uso “Gestionar Jugadores del torneo”

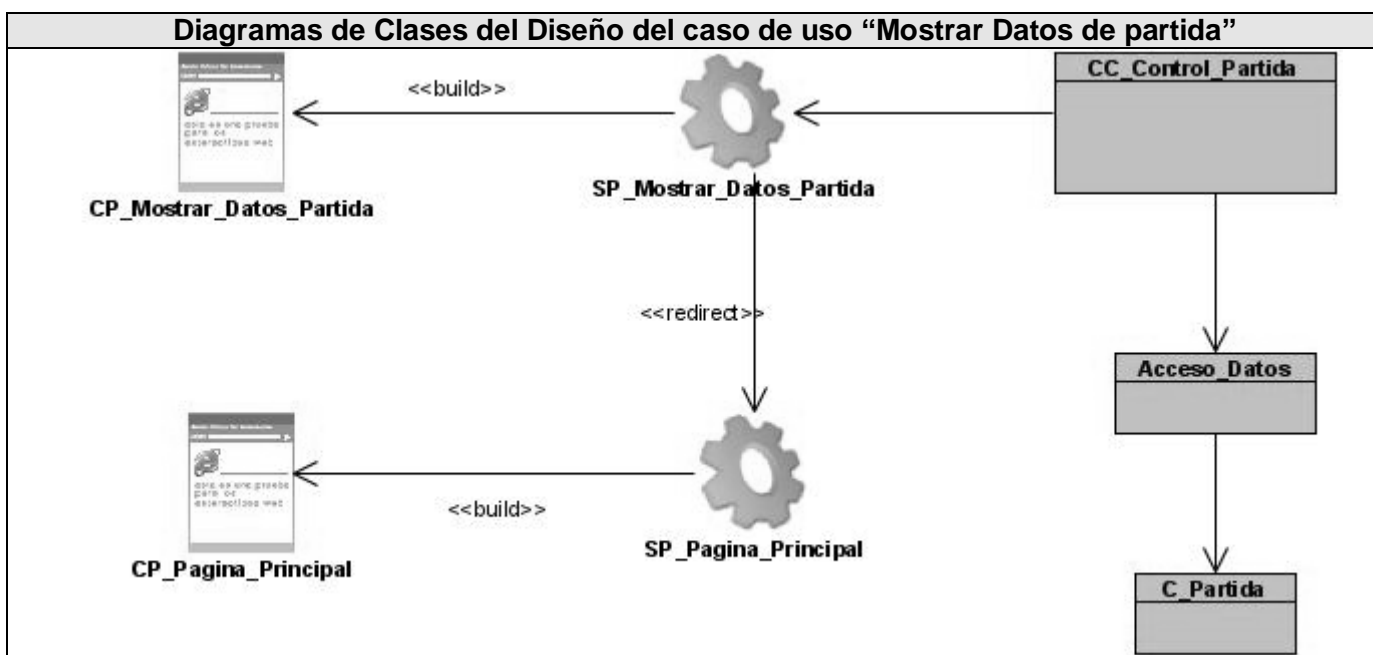


Figura 61: Diagramas de Clases del Diseño del caso de uso “Mostrar Datos de partida”

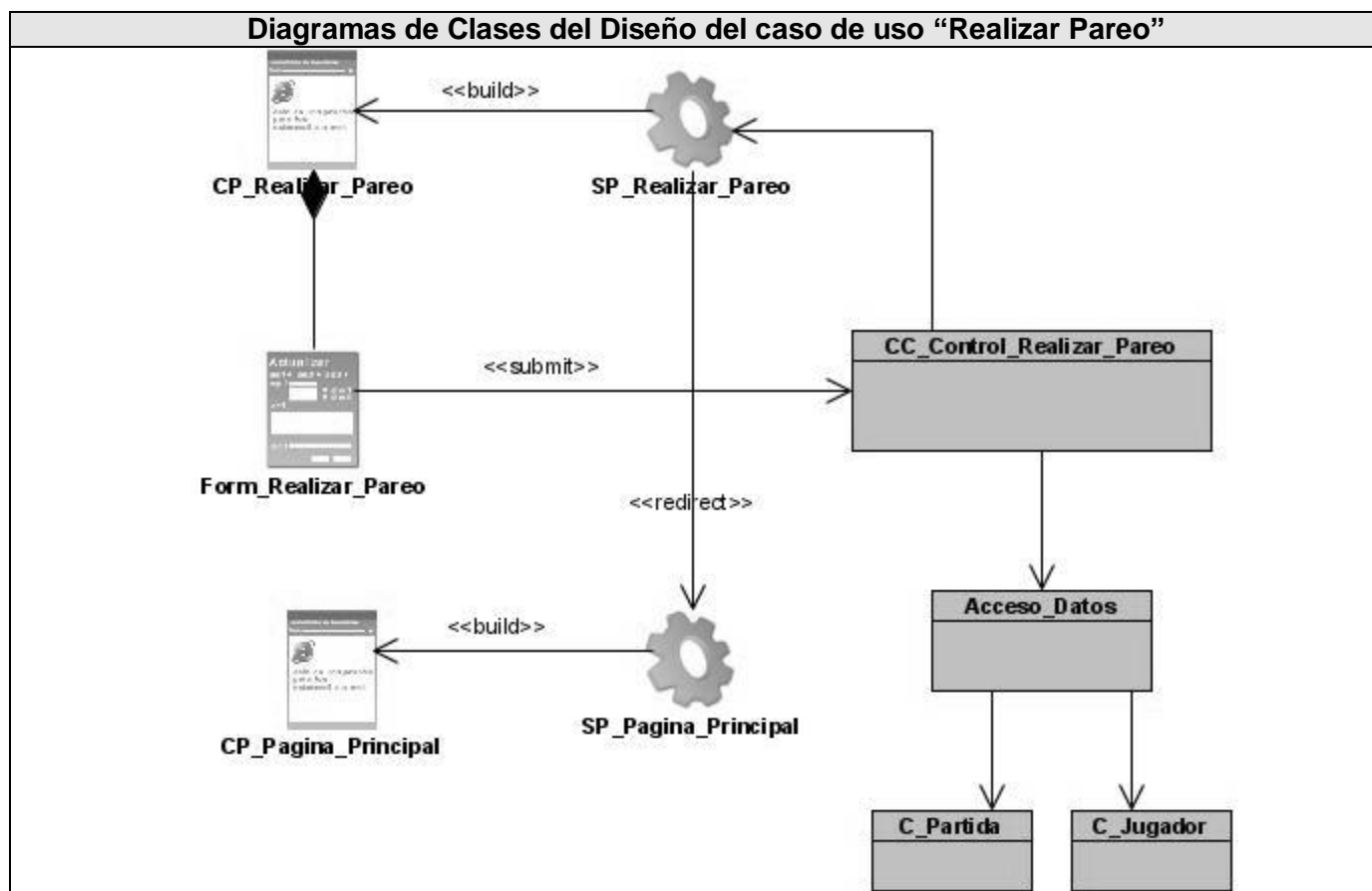


Figura 62: Diagramas de Clases del Diseño del caso de uso "Realizar Pareo"

## GLOSARIO DE TÉRMINOS

- 1- **Apache:** Servidor web de distribución libre. Fue desarrollado en 1995 y ha llegado a ser el más usado de Internet. Soporta los lenguajes PERL y PHP.
- 2- **Ajedrecista:** Un jugador que juega o practica el Ajedrez.
- 3- **Ajedrez:** Juego de mesa jugado entre dos personas sobre un tablero de 64 casillas alternadas de color claro y oscuro (blancas y negras). Cada jugador dispone de 16 piezas de Ajedrez blancas y negras respectivamente y la finalidad del juego es de dar Jaque Mate al Rey del jugador contrario.
- 4- **Aplicación:** Programas que permiten trabajar con el ordenador. Son aplicaciones los procesadores de texto, hojas de cálculo, bases de datos, etc.
- 5- **Adversario:** *Jugador del bando opuesto, llamado también oponente.*
- 6- **Ajax:** Acrónimo de *Asynchronous JavaScript And XML* (JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.
- 7- **CASE:** Acrónimo inglés de Computer Aided Software Engineering, que viene a significar Ingeniería de Software Asistida por Ordenador.
- 8- **CSS:** (Cascade Style Sheet) Hoja de Estilo en Cascada. Conjunto de instrucciones escritas en HTML que definen las apariencias de una página web con el objetivo de que sus estilos se parezcan.
- 9- **Diagrama:** Dibujo o gráfica que representa el tablero con todas sus casillas y la posición de todas, algunas o ninguna de las piezas.
- 10- **Duración de una partida:** Se mide por el número de jugadas y por el tiempo empleado. Una partida normal de 40 o 45 movimientos dura comúnmente alrededor de cinco horas. La duración es muy variable.
- 11- **DOM:** El DOM o Document Object Model (Modelo de Objetos de Documento) es una forma de representar documentos estructurados (tales como una página web HTML o un documento XML) que es independiente de cualquier lenguaje orientado a objetos.
- 12- **Elo, sistema:** Un sistema de valoración de jugadores de Ajedrez que fue desarrollado por el Profesor Arpad Elo (1903 - 1993) de Milwaukee. A un jugador se le asigna una puntuación inicial que aumenta o disminuye de acuerdo a los resultados obtenidos en partidas oficiales.

- 13- FIDE:** Fédération Internationale des Échecs, máximo organismo del Ajedrez mundial creado en el año de 1924 en París. Entre las facultades de la FIDE están todo lo relacionado a los títulos mundiales.
- 14- Final de partida:** Una partida de Ajedrez puede finalizar por alguna de estas razones: dar Jaquemate a uno de los Reyes; abandono de uno de los oponentes; tablas acordadas entre los dos jugadores; eliminación total de de piezas y Peones quedando solo ambos Reyes; insuficiencia de material para realizar un mate; ejecutar cincuenta movidas consecutivas sin mover Peones ni efectuar captura alguna (la partida se declara tablas).
- 15- FECAP:** Federación Cubana de Ajedrez Postal, organismo rector del juego ciencia a distancia en el país y agrupa en su seno a los amantes de esta disciplina.
- 16- Jugada:** Cada movimiento alterno efectuado por los dos contendientes en una partida de Ajedrez.
- 17- Jugador:** Cada uno de los dos contendientes en una partida.
- 18- HTML:** Acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Este lenguaje se basa en tags (instrucciones que le dicen al texto como deben mostrarse) y atributos (parámetros que dan valor al tag). Es el estándar usado en el World Wide Web.
- 19- HTTP (HyperText Transfer Protocol):** Protocolo de transferencia de hipertexto. Es un protocolo del nivel de aplicación usado para la transferencia de información entre sistemas, de forma clara y rápida. Ha sido usado por el World-Wide Web desde 1990.
- 20- ICCF:** International Correspondance Chess Federation (Federación internacional de Ajedrez por Correspondencia.)
- 21- JavaScript (JavaScript):** Lenguaje desarrollado por *Netscape*. Aunque es parecido a *Java* se diferencia de él en que los programas están incorporados en el fichero *HTML*.
- 22- MySQL:** Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.
- 23- Partida de Ajedrez:** Serie de jugadas o movimientos efectuados por los dos jugadores de Ajedrez con la finalidad de dar jaquemate al Rey del oponente.
- 24- PGN:** Portable Game Notation (Notación Portable de Partidas), un sistema de notación de Ajedrez estandarizado que comúnmente se usa en programas de visores para partidas de Ajedrez.

- 25- **Perl**: *Practical Extraction and Report Language*. Es un lenguaje de programación desarrollado por Larry Wall inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell.
- 26- **PHP**: Lenguaje procesador de Scripts servidor. Libre y de código abierto.
- 27- **PostgreSQL**: Es un servidor de base de datos relacional libre, liberado bajo la licencia BSD.
- 28- **Ranking**: Sistema de clasificación oficial para designar a los jugadores de Ajedrez basado en su fuerza Ajedrecística y que se logra en resultados individuales de partidas en Torneos de la Federación.
- 29- **RUP**: El Proceso Racional Unificado o RUP (Rational Unified Process), es un proceso de desarrollo de software.
- 30- **Scripts**: Un conjunto de comandos escritos en un lenguaje interpretado.
- 31- **Tiempo**: En el Juego del Ajedrez actual, la partida está limitada en el tiempo.
- 32- **Torneo**: Encuentro de Ajedrez en la que intervienen varios jugadores.
- 33- **UML**: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por la OMG (Grupo dedicado a la promoción de la tecnología orientada a objetos y su estandarización).
- 34- **XHTML**: Acrónimo inglés de eXtensible Hyper Text Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML.
- 35- **XML**: Acrónimo de "EXtensible Markup Language". Es decir, lenguaje de marcas extensible, es de reciente creación (febrero de 1998). XML es un metalenguaje, o sea, sirve para crear lenguajes. Fue diseñado para permitir la descripción de información contenida en el WWW a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes), puedan buscar, comparar y compartir información en la red.