

710-1237-00

UCI

Universidad de las Ciencias Informáticas

Facultad 8

Módulo para la Gestión de Información de Torneos de Ajedrez.

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

Autor(es)

Janys Yusleidys Torres González

Everaldo Cruz Suárez

Tutor

Ing. Sandra Gutiérrez Secades

Co-Tutor

MSc. Isbel Herrera del Sol

Ciudad de la Habana, Junio 2008

“Año 50 de la Revolución”

PENSAMIENTO



"El Ajedrez constituye un medio eficaz para la educación y formación del intelecto del hombre".

Ernesto Che Guevara

Agradecimientos

Janys Y. Torres González

Todos sabemos que las palabras no tienen la fuerza de un sentimiento ni la pasión del alma, que no son capaces de expresar con profundidad lo que se siente ni decir con claridad cuánto se ama.

Pero a pesar de todo quiero utilizarlas para decirle a mi mamá y a mi papá lo mucho que les agradezco el haberme brindado cada día de mi vida su cariño sincero, su apoyo incondicional, por hacerme ver que lo que se logra con sacrificio sabe mejor y es un sabor que dura para siempre, por nunca imponerme nada y permitirme hacer lo que me dicta el alma. Aprovecho esta oportunidad para decirles que sin su ayuda no hubiese sido capaz de recorrer ni la mitad del camino que he transitado hasta hoy.

Les agradezco a mis hermanos por existir en mi vida, y hacer de ella un pozo de felicidad, por su preocupación y apoyo.

Le agradezco también a mi abuela Nina, por tenerme siempre presente en su corazón y en su pensamiento y desear con toda el alma que mis sueños se hicieran realidad.

Le agradezco al resto de mi familia por darle a mi vida la armonía que necesito.

Agradecer a mi mejor amiga "Aliuska" por estar siempre en los momentos que mis fuerzas flaquean.

Le agradezco a mi tutora "Sandra Gutiérrez Secades" por toda su paciencia, preocupación y apoyo en todo este tiempo.

Les agradezco a Arisbel Hechavarría Rojas, a Yudisney Vázquez Ortiz (Yudita), a Rogert Santos García, a los muchachos del proyecto y a todos los que de una forma u otra me ayudaron.

Quiero agradecerle a mis compañeros de aula por permitirme conocer personas tan maravillosas, con sus defectos y virtudes pero geniales en general.

Hay alguien a quien debo agradecerle el haber hecho posible la realización, no solo de mi meta, sino la de todos, alguien que nos ha dado la oportunidad de crecer en la vida y de vivir con ilusiones, con amor, con paz: gracias Fidel.

Everaldo Cruz Suárez

A mi tutora Sandra Gutiérrez por su ayuda durante el desarrollo de nuestro trabajo.

A Arisbel, Duenytz y a todos los que de una forma u otra tuvieron que ver con nuestro resultado.

A toda mi familia que siempre ha estado ahí para ayudarme a salir adelante.

Dedicatoria

Janys Y. Torres González

A mi mamá y a mi papá, por significar lo más grande en mi vida y porque yo sé que mis sueños siempre han estado ligados a los suyos. A mis hermanos por darme otro gran motivo para vivir.

A todos las personas que me quieren y que de una forma u otra me apoyaron siempre.

A la Revolución por todo.

Everaldo Cruz Suárez

A mi abuelo Antonio Suárez muy especialmente, por ayudarme mucho en la vida y siempre estar apoyándome y compartiendo conmigo en todos los tiempos.

A mis padres Lourdes Suárez y Everaldo Cruz que esperan de mí lo mejor cada día, a quienes debo todo lo que hoy soy. Esos que siempre han estado ahí para tenderme la mano en buenas y malas, los que representan la mejor guía que he tenido en la vida.

A mi hermana Esther, mi primo David, mis abuelos Digna, Delia, Polo y muy especial a mi bisabuela Caruca, a quien quise mucho en vida y siempre llevo presente.

A mis tíos y tías Mary, Diana, Natacha, Tony y Giraldo por haberme ayudado siempre que los necesité.

A toda mi familia de forma general que son lo mejor que tengo en esta vida y a todos mis amigos y colegas que siempre han confiado en mi.



Resumen

Como parte del análisis de las necesidades expuestas por la Cátedra de Ajedrez Remberto A. Fernández González, de la Universidad de las Ciencias Informáticas; surge la idea de crear una aplicación web dinámica que permita dar cobertura a los torneos ajedrecísticos que se desarrollan en la Universidad, logrando automatizar la gestión de los mismos y proveer a los usuarios de determinadas funcionalidades de interés. Con el fin de obtener una solución con cierta cantidad de prestaciones, se realizó un estudio de las más prestigiosas herramientas para estos fines a los diferentes niveles, así como las principales tendencias en el uso de metodologías y tecnologías actuales utilizadas en el desarrollo de aplicaciones web. La documentación recoge la generación de los diferentes artefactos durante el proceso de Análisis, Diseño e Implementación del Módulo para la Gestión de Información de Torneos de Ajedrez. La aplicación se desarrollará sobre la plataforma Linux, y como lenguaje de programación del lado del servidor se utilizará el Pre-procesador de Hipertextos (PHP) en su versión 5. De los lenguajes del lado cliente se empleará JavaScript para validar algunas entradas de datos, el Lenguaje de Marcas Hipertextuales (HTML) para diseñar la estructura de los textos y presentarlos en forma de hipertexto y las Hojas de Estilo en Cascada (CSS) para definir la presentación del documento y separar esta de su estructura; se empleará además Asynchronous JavaScript + XML (AJAX) para hacer más eficiente la aplicación; entre otras tecnologías.



Índice

Introducción.....	1
Capítulo 1: Fundamentación Teórica.....	6
1.1 Introducción.....	6
1.2 Aplicaciones informáticas similares vinculadas al campo de acción de este trabajo	6
1.3 Tendencias y tecnologías actuales.....	8
1.4 Arquitectura.....	8
1.5 Lenguajes de Programación Web.....	10
1.5.1 Lenguajes al lado del servidor.....	10
1.5.1.1 PHP.....	10
1.5.1.2 Perl.....	13
1.5.1.3 Python.....	13
1.5.2 Lenguajes al lado del cliente.....	14
1.5.2.1 JavaScript.....	14
1.5.2.2 HTML.....	15
1.5.2.3 CSS.....	16
1.5.3 Combinación de tecnologías.....	17
1.5.3.1 AJAX.....	17
1.6 Servidor Web.....	18
1.6.1 Apache.....	19
1.6.2 Cherokee.....	20
1.7 Gestor de Base de Datos.....	20
1.7.1 MySQL.....	21
1.7.2 PostgreSQL.....	21
1.8 Servidor de Correo.....	22



1.8.1 SMTP Email Server	22
1.9 Proceso de Desarrollo de Software	22
1.10 Metodologías de desarrollo de software	23
1.10.1 El Proceso Unificado de Desarrollo (RUP)	23
1.10.2 Extreme Programing (XP)	25
1.10.3 Microsoft Solution Framework (MSF)	26
1.11 Lenguaje Unificado de Modelado (UML)	26
1.12 Herramientas CASE de Modelado con UML	28
1.12.1 Visual Paradigm	29
1.12.2 Umbrello	30
1.13 Entorno de desarrollo integrado. Zend Studio	30
1.14 Conclusiones	31
Capítulo 2: Características del Sistema	33
2.1 Introducción	33
2.2 Objeto de Estudio	33
2.3 Flujo actual de los procesos	33
2.4 Análisis crítico de la ejecución de los procesos	34
2.5 Procesos objeto de automatización	35
2.6 Propuesta de Sistema	35
2.7 Modelo del Negocio	36
2.7.1 Descripción del Negocio Actual	36
2.8 Reglas del Negocio a tener en cuenta	37
2.9 Casos de Uso del Negocio (CUN)	37
2.10 Actores de Negocio	37
2.11 Diagrama de casos de uso del negocio	37
2.12 Trabajadores del negocio	38



2.13 Entidades del negocio	39
2.14 Especificación de los Casos de Uso del Negocio	39
2.15 Diagrama de clases del modelo de objetos	44
2.16 Definición de los requisitos funcionales	44
2.17 Definición de los requisitos no funcionales	48
2.18 Actores del Sistema.....	51
2.19 Casos de Uso del Sistema.....	51
2.20 Diagrama de casos de uso del sistema	51
2.21 Descripción de los casos de uso del sistema.....	52
2.22 Conclusiones	59
Capítulo 3: Análisis y Diseño del Sistema.....	61
3.1 Introducción.....	61
3.2 Diagrama de clases del análisis	61
3.3 Diagrama de clases del Diseño	63
3.4 Diagrama de Secuencia del Diseño:.....	65
3.6 Descripción textual de las clases de diseño web	68
3.7 Modelo de clases persistentes.....	76
3.8 Principios de Diseño	77
3.9 Interfaz de Usuario	78
3.10 Concepción general de la ayuda.....	78
3.11 Tratamiento de errores.	79
3.12 Seguridad	79
3.13 Conclusiones	79
Capítulo 4: Implementación.....	80
4.1 Introducción.....	80
4.2 Modelo de despliegue.....	80



4.3 Modelo de Implementación.....	81
Capítulo 5: Estudio de Factibilidad.....	84
5.1 Introducción.....	84
5.2 Puntos de Casos de Uso.....	84
5.3 Cálculo de Puntos de Casos de Uso sin Ajustar.....	84
5.3.1 Factor de Peso de los Actores sin Ajustar (UAW).....	84
5.3.2 Factor de Peso de los Casos de Uso sin Ajustar (UUCW).....	85
5.4 Cálculo de Puntos de Caso de Uso Ajustados.....	86
5.4.1 Factor de Complejidad Técnica (TCF).....	86
5.4.2 Factor de Ambiente (EF).....	88
5.5 Calcular el Esfuerzo del Flujo de Trabajo Implementación.....	89
5.6 Costo.....	91
5.7 Beneficios tangibles e intangibles.....	91
5.7.1 Tangibles.....	91
5.7.2 Intangibles.....	91
5.8 Análisis costo-beneficio.....	92
5.9 Conclusiones.....	92
Conclusiones.....	93
Recomendaciones.....	94
Referencias Bibliográficas.....	95
Bibliografía.....	97
Glosario de Términos.....	100



Índice

Tabla

Tabla 1: Actores del negocio	37
Tabla 2: Trabajadores del Negocio.....	39
Tabla 3: Descripción Textual del CUN "Organizar Torneo"	42
Tabla 4: Descripción Textual del CUN "Solicitar Información sobre Torneos Anteriores"	43
Tabla 5: Actores del Sistema.....	51
Tabla 6: Descripción Textual del Caso de Uso del Sistema "Autenticar Usuario"	54
Tabla 7: Descripción Textual del Caso de Uso del Sistema "Gestionar Eventos"	56
Tabla 8: Descripción Textual del Caso de Uso del Sistema "Gestionar Torneo"	59
Tabla 9: Descripción de la clase entidad "Torneo"	69
Tabla 10: Descripción de la clase entidad "Ronda"	70
Tabla 11: Descripción de la clase entidad "Árbitro"	70
Tabla 12: Descripción de la clase controladora "Control_Torneo"	72
Tabla 13: Descripción de la clase controladora "Control_Ronda"	72
Tabla 14: Descripción de la clase controladora "Control_Árbitro"	73
Tabla 15: Descripción de la clase de acceso a datos "Acceso_Datos"	74
Tabla 16: Descripción de la clase de acceso a datos "AD_Torneo"	75
Tabla 17: Descripción de la clase de acceso a datos "AD_Ronda"	76
Tabla 18: Factor de Peso de los Actores sin Ajustar	85
Tabla 19: Factor de Peso de los Casos de Uso sin Ajustar	86
Tabla 20: Factor de Complejidad Técnica	87
Tabla 21: Factor de Ambiente	88
Tabla 22: Distribución del esfuerzo entre las diferentes actividades de un proyecto.....	90
Tabla 23: Estimación Final	92



Índice

Figuras

Figura 1: Diagrama de Casos de Uso del Negocio	38
Figura 2: Diagrama de Actividades del CUN "Organizar Torneo"	42
Figura 3: Diagrama de Actividades del CUN Solicitar Información sobre Torneos Anteriores	43
Figura 4: Diagrama de Clases del Modelo de Objetos del Negocio	44
Figura 5: Diagrama de Casos de Uso del Sistema	52
Figura 6: Diagrama de clases de análisis del caso de uso "Autenticar Usuario"	62
Figura 7: Diagrama de clases de análisis del caso de uso "Gestionar Eventos"	62
Figura 8: Diagrama de clases de análisis del caso de uso "Gestionar Torneo"	62
Figura 9: Diagrama de clases de análisis del caso de uso "Gestionar Rondas"	63
Figura 10: Diagrama de clases Web para el caso de uso "Autenticar usuario"	64
Figura 11: Diagrama de clases Web para el caso de uso "Gestionar Eventos"	64
Figura 12: Diagrama de clases Web para el caso de uso "Gestionar Torneo"	65
Figura 13: Diagrama de clases Web para el caso de uso "Gestionar Ronda"	65
Figura 14: Diagrama de Secuencia del Diseño del Caso de Uso "Autenticar Usuario"	66
Figura 15: Diagrama de Secuencia del Diseño del Caso de Uso "Registrar Usuario"	66
Figura 16: Diagrama de Secuencia del Diseño del Caso de Uso "Gestionar Torneo"	67
Figura 17: Diagrama de Secuencia del Diseño del Caso de Uso "Gestionar Ronda"	67
Figura 18: Diagrama de clases del subsistema de "Acceso a Datos"	68
Figura 19: Diagrama de Clases Persistentes	77
Figura 20: Modelo de Despliegue	80
Figura 21: Diagrama de componentes del sistema	81
Figura 22: Diagrama de componentes del caso de uso "Autenticar Usuario"	82
Figura 23: Diagrama de componentes del caso de uso "Gestionar Evento"	82



Figura 24: Diagrama de componentes del caso de uso "Gestionar Torneo"	83
Figura 25: Diagrama de componentes del caso de uso "Gestionar Ronda"	83



Introducción

En el país el ajedrez es una actividad de gran tradición, estimada por sus valores intelectuales, culturales y éticos, que es practicado por cientos de miles de personas de todas las edades. Durante el año se desarrollan una serie de eventos de esta índole que tienen carácter internacional y nacional.

La Universidad de las Ciencias Informáticas (UCI) ha tenido desde sus inicios entre sus primeras prioridades la promoción del juego ciencia. Debido a ello en su segundo año se creó un concurso para presentar la aplicación web del ajedrez, en el cual quedó como ganadora: Ucidrez. La misma era soportada solamente por el Internet Explorer, perteneciente a tecnología propietaria, y tenía el juego online como la funcionalidad más utilizada por la comunidad de usuarios, esta aplicación aunque muy seria – y no visible en navegadores libres – tenía problemas de seguridad e inconvenientes de poca navegabilidad.

En marzo de 2006 la facultad 10 migra completamente hacia software libre; lo cual trajo consigo un problema para los ajedrecistas de la facultad, ya que la única aplicación existente para el ajedrez en la Universidad, era soportada solamente por el Internet Explorer.

Como se puede apreciar una parte de la comunidad universitaria adoptó la plataforma informática libre para el trabajo en todas sus tareas, por lo que la aplicación web de ajedrez, existente en ese momento: Ucidrez, perdió funcionalidad por su incompatibilidad con navegadores libres, surgiendo así la necesidad de desarrollar un software que se adaptara a la utilización de la nueva tecnología.

Con el objetivo de contribuir a la migración de la Universidad hacia software libre y lograr una aplicación web que diera la posibilidad de tener una plataforma de ajedrez visible en cualquier sistema operativo, se elaboró y conceptualizó la idea de **Infodrez**, como iniciativa del Proyecto Unicornios y bajo la supervisión de la Cátedra Honorífica de Ajedrez Remberto A. Fernández González. Dicha aplicación prevalece en la actualidad ofreciendo grandes funcionalidades.

Hasta aquí se ha expuesto la evolución que ha tenido informáticamente el ajedrez en la Universidad, el cual cuenta actualmente con una aplicación en la cual los ajedrecistas de la institución tienen la oportunidad de realizar partidas en tiempo real.

Además del evento mencionado anteriormente, en la Cátedra de Ajedrez de la UCI a lo largo de todo el curso se desarrollan una serie de manifestaciones del ajedrez entre las que se destacan principalmente los torneos, los cuales no son más que organizar a un número de



jugadores para que jueguen determinadas rondas (suma de varias partidas); de acuerdo a la cantidad de participantes se selecciona el Sistema de Juego a emplear, que puede ser: Todos contra Todos, Suizo, Scheveningen o Muerte Súbita.

Para la organización de un torneo se necesita llevar a cabo una serie de pasos previos al evento tales como: invitar a los participantes: jugadores y árbitros (almacenar una breve biografía de los mismos), organizar los locales donde se efectuará, difundir la convocatoria del torneo, requisitos para la realización del torneo, definir los objetivos del torneo, en fin, preparar todas las condiciones para que estos se lleven a cabo con la calidad requerida y los involucrados reciban la mejor atención. Una vez conceptualizado el torneo los responsables deben tener el registro de todos los que participarán en el mismo: jugadores (estándares de cada jugador), árbitros e invitados especiales. Durante el curso del torneo se necesita controlar en todo momento la información que se genera: cantidad de rondas, quiénes se enfrentan, cuáles son los resultados de cada ronda, entre otras.

Como se puede apreciar la realización de un torneo de ajedrez necesita una amplia documentación, cuyo manejo y difusión se hace engorroso por no contar con la herramienta adecuada para la gestión de la información generada, dado lo cual se identifican las siguientes dificultades en el proceso actual:

- ✓ Procesamiento de la información de forma manual lo que trae como consecuencia que la actualización de la misma demore y se pierda en muchos casos su orden lógico. Debido a esto, la comunidad ajedrecista no conoce con exactitud lo que sucede en el evento, ya que se dificulta el acceso y visualización de los datos de manera eficiente y confiable.
- ✓ La administración y organización de la información no se lleva a cabo con las herramientas indicadas ya que está almacenada en diferentes lugares como: papeles, documentos digitales en formato Word, Excel, lo que conlleva a que la probabilidad de pérdida y desorganización de la información sea elevada.
- ✓ Al estar la información distribuida en varios lugares se puede apreciar que la universidad no cuenta con una aplicación que la centralice y dé a los interesados la posibilidad de tener cierto nivel de acceso a estos datos. Es de vital importancia que se registre no solo lo que se está efectuando en el momento, sino también acontecimientos de ediciones anteriores, de manera que se almacenen históricamente.

El desarrollo de Aplicaciones Web para el Ajedrez, es una esfera de la informática de muchos años de experiencia, sin embargo en el país son escasas las aplicaciones web que informan de



las principales noticias de los torneos de ajedrez que se realizan a nivel internacional y nacional.

La Universidad tampoco cuenta con una aplicación web que sea específicamente para torneos de ajedrez, en la cual se registre toda la información que se genera durante los mismos.

Es evidente que la Cátedra de Ajedrez de la Universidad y el país requieren de una herramienta que permita dar cobertura a los torneos ajedrecísticos que se desarrollan: de categoría internacional, nacional y de la propia Universidad, con el objetivo de centralizar la información asociada a dichos eventos.

Como consecuencia de lo planteado anteriormente, surge la siguiente interrogante que constituye el **Problema Científico** de la presente investigación: ¿Cómo dar cobertura a los torneos de ajedrez que se realizan en la Universidad?

Para dar solución a la problemática planteada el **Objeto de Estudio** sobre el cual se actuará es el Proceso de Desarrollo de Aplicaciones Web para el Ajedrez. Donde el **Campo de Acción** se encontrará determinado por el Proceso de Desarrollo de Aplicaciones Web para la gestión de los contenidos referentes a los torneos de ajedrez.

El **Objetivo General** de la investigación es: Realizar el análisis, diseño e implementación de una Aplicación Web para dar soporte a los torneos ajedrecísticos que se realizan en la Universidad. Del mismo se derivan los siguientes **Objetivos Específicos**:

- ✓ Informatizar los procesos referentes a la gestión de torneos de ajedrez que se llevan a cabo en la Cátedra de Ajedrez de la Universidad.
- ✓ Elaborar una propuesta de una Aplicación Web que satisfaga las necesidades de la Cátedra de Ajedrez de la Universidad.
- ✓ Elaborar la documentación del sistema para el desarrollo de posteriores versiones.

Con el propósito de dar cumplimiento a los objetivos mencionados se propone el desarrollo de las siguientes **Tareas Investigativas**:

- ✓ Realizar un estudio del ajedrez, y de todo lo referente a los torneos y características específicas de los mismos.
- ✓ Realizar una investigación para caracterizar los sitios de los torneos de ajedrez de mayor prestigio.
- ✓ Realizar un estudio de las tendencias y tecnologías actuales para el desarrollo de aplicaciones web y seleccionar las más adecuadas para la realización del sistema.



- ✓ Proponer una solución que resuelva la problemática planteada.
- ✓ Realizar el análisis, diseño e implementación de la aplicación.
- ✓ Elaborar la documentación de todo el flujo de trabajo.

Por todo lo planteado anteriormente se determina como **Idea a Defender**: Si se realiza una aplicación web que facilite la gestión de los contenidos relacionados con los torneos de ajedrez, la información que se manipula referente a los mismos será confiable, centralizada y de rápido acceso.

La elaboración y puesta en práctica de los resultados de esta investigación brindará los siguientes **Aportes Prácticos**:

- ✓ Generación de una aplicación web para la gestión de la información relacionada con los torneos de ajedrez.

En la WEB:

Administradores:

- ✓ Crear y administrar su propio sitio web mediante una herramienta de fácil manejo.
- ✓ Organizar debidamente la información referente a los torneos de ajedrez que se realizan, de manera que puedan obtener información significativa de los mismos.
- ✓ Actualizar la información almacenada cada vez que se requiera.

Usuarios:

- ✓ Contarán con una aplicación web que los mantendrá informados y mediante la cual podrán seguir los acontecimientos de los torneos que se realicen en el mundo, en el país o en la Universidad.

El desarrollo de este trabajo se ha concebido en 5 capítulos, en los mismos se abordarán:

- ✓ **Capítulo 1. Fundamentación Teórica:** Se realiza un estudio del arte sobre las diferentes aplicaciones que incluyen en alguna medida la gestión de torneos ajedrecísticos, además de analizar algunas de las herramientas y tecnologías utilizadas para el desarrollo de aplicaciones web en software libre, así como las diferentes metodologías que pueden utilizarse para organizar todo este proceso. Se seleccionan las más adecuadas para el curso de la investigación y se presenta una propuesta de solución en una primera iteración.



- ✓ **Capítulo 2. Características del Sistema:** En este capítulo se comenta y describe el objeto de estudio del presente trabajo y se muestran cuáles son los procesos objetos de automatización. Se brinda una breve descripción del negocio correspondiente y se describen actores, trabajadores, entidades y reglas del negocio. Se elaboran el diagrama de casos de uso del negocio (en el cual se relacionan actores y procesos del negocio) y el modelo de objetos (donde se presenta la colaboración entre trabajadores del negocio, entidades del negocio y unidades de trabajo).

Igualmente se identifican los requisitos funcionales y no funcionales del sistema, solución a las necesidades existentes; evidenciando además, cuáles son las funcionalidades que ofrece el sistema, al tiempo que se describen los actores y casos de uso del mismo.

- ✓ **Capítulo 3. Análisis y Diseño del Sistema:** En el presente capítulo se expone lo referente al análisis y diseño de la aplicación en desarrollo. Se tendrá en cuenta que el análisis consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace el sistema, de manera que su interés se centra solamente en los requisitos funcionales. Por otro lado, el diseño representa un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en pocas palabras, el diseño se preocupa en CÓMO cumple el sistema sus objetivos y debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.
- ✓ **Capítulo 4. Implementación:** En este capítulo se describe cómo los elementos del modelo de diseño, como las clases; son implementados en términos de componentes. Los componentes representan algo físico que se crea, se modifica o se elimina en el proceso de implementación.
- ✓ **Capítulo 5. Estudio de Factibilidad:** En este capítulo se realiza el estudio de factibilidad del sistema realizado, por ser fundamental para la realización de un proyecto la estimación del esfuerzo humano, el tiempo de desarrollo que se requiere para la ejecución del mismo y también su costo. El método que se utilizará para el estudio, será el análisis de Puntos de Casos de Uso.



Capítulo 1

Fundamentación Teórica

1.1 Introducción

En este capítulo se realiza un estudio de las aplicaciones similares en el mundo ajedrecístico así como un análisis de las herramientas y tecnologías actuales para el desarrollo de aplicaciones web en software libre; seleccionando las más adecuadas para la creación de la aplicación propuesta.

1.2 Aplicaciones informáticas similares vinculadas al campo de acción de este trabajo

El desarrollo de Aplicaciones Web para el Ajedrez es una esfera de la informática de muchos años de trabajo, lo que ha propiciado que en la actualidad existan en todo el mundo muchos sitios que aborden el tema.

Con respecto a los torneos de ajedrez actualmente existen innumerables sitios que brindan una amplia información de los mismos, tales como: las noticias más importantes del torneo que se está efectuando en el momento, participantes del torneo, descripción y caracterización de cada una de las rondas, cuáles fueron los resultados en las mismas, una breve biografía de los jugadores, galería de fotos, reportes completos de cada ronda, estadísticas, historia del torneo, así como la posibilidad de contar con un visor de las partidas realizadas, entre otras. También los usuarios pueden acceder en la misma medida a información de ediciones pasadas.

Los sitios más populares actualmente que gestionan información relacionada a los torneos de ajedrez a nivel internacional son: www.chessbase.com, www.coruschess.com, www.chesscenter.com, www.chessvibes.com y www.ajedrezmadrid.com, los cuales a pesar de toda la información que ofrecen, no satisfacen las necesidades planteadas, en primera instancia, porque son sitios a los cuales toda la población del país no tiene acceso y por otra parte, son sitios propietarios que no permiten adicionarles módulos creados para algunos aspectos específicos, como podría ser: ubicar las noticias netamente cubanas.

En el país el desarrollo de aplicaciones web que gestionen la información referente a las distintas modalidades del ajedrez es bastante ínfimo, solo existe alguna que otra página que trate alguna modalidad en específico y cuya información del tema es bastante pobre.



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

Con respecto a los torneos internacionales que se realizan en Cuba, solo existen sitios que dedican una de sus secciones a la publicación de las noticias más relevantes de los mismos, sitios como: *www.vanguardia.co.cu*, *www.juventudrebelde.cu* y *www.jit.cu*; además de algunos sitios de la prensa cubana que lo hacen de manera ocasional y solamente para los torneos más importantes.

El sitio Capablanca, expone en una sección información referente al torneo Capablanca específicamente; y así puede mencionarse, lo que ocurre con los torneos cubanos internacionales, ocurre con los nacionales; solo se poseen pequeñas publicaciones de la información que generan, la cual se puede encontrar en sitios como: *www.vanguardia.co.cu*.

Queda muy claro que en el país un sitio que abarque sola y exclusivamente la información referente a los torneos de ajedrez; donde se acople toda la información que genera un torneo y posibilite el rápido acceso a la misma manteniendo al tanto de todo lo que acontece en estos eventos a los que les pueda interesar; no existe.

La Universidad tampoco cuenta con una aplicación para la gestión de la información relacionada a los torneos de ajedrez. Es por ello que el proyecto Infodrez se ha trazado el objetivo de elaborar un Sistema Gestor de Contenidos (CMS) para el ajedrez y en este caso particular, la realización de una aplicación web que gestione toda la información relacionada a los torneos de ajedrez que se realizan en la institución.

Cabría preguntarse en este caso, *¿Por qué una Aplicación Web para la solución?* Es importante destacar que una aplicación web es un sistema donde la entrada del usuario afecta el estado del negocio y se eligió esta para darle solución a los problemas existentes por las siguientes razones:

- ✓ Compatibilidad multiplataforma.
- ✓ Actualización.
- ✓ Rápido acceso.
- ✓ Menos requerimientos de memoria.
- ✓ Múltiples usuarios concurrentes.
- ✓ Datos seguros, entre otros.[1]

Además de que instalar una aplicación web consiste en configurar los componentes del lado del servidor en la red y NO es necesaria una instalación o configuración en el lado cliente.



1.3 Tendencias y tecnologías actuales

Probablemente la informática sea la herramienta más poderosa que el hombre ha tenido jamás en sus manos y actualmente interviene de forma directa ó indirecta en, prácticamente, todas las actividades humanas. Permitir que esta herramienta sea controlada y restringida por entidades que sólo se interesan en su propio beneficio supone un perjuicio para las sociedades, irreparable. El Software Libre representa una oportunidad histórica de tomar el control del destino, es importante no dejarlo pasar.

El presente trabajo se desarrollará sobre el Sistema Operativo Linux, por políticas del proyecto y la Universidad, es por ello que se centra en el estudio de herramientas y tecnologías software libre.

Al optar por el modelo de desarrollo y distribución libre, la organización queda habilitada a utilizar la gran cantidad de herramientas libres disponibles en la actualidad. Esto no sólo implica únicamente la ejecución de dichas herramientas, sino también la modificación de las mismas para adaptarlas a casos particulares y la exploración de sus mecanismos de funcionamiento para luego reutilizarlos en futuros desarrollos. De esta forma se obtiene una ventaja significativa respecto a aquellas organizaciones que basan su negocio en el modelo propietario o cerrado, que no pueden utilizar esta base de herramientas y conocimiento.

En los epígrafes siguientes se identifican y describen algunas de las herramientas tecnológicas actuales, de estas se determinarán cuáles se utilizarán para el desarrollo de la aplicación con la cual se pretende dar respuesta a la problemática planteada anteriormente. Para la selección de las herramientas se tendrán en cuenta las características del entorno en el cual se aplicará el sistema y las necesidades planteadas.

1.4 Arquitectura

La arquitectura de un sistema constituye un amplio marco que describe su forma y su estructura, sus componentes y cómo estos interactúan.

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente dentro de la organización.

La arquitectura representa elementos significativos de cada modelo, es el esqueleto o base de una aplicación, da la posibilidad de analizar la aplicación desde varios puntos de vista.



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

Es posible decir que la Arquitectura del Software se centra en aquellos aspectos del diseño y desarrollo que no pueden tratarse de forma adecuada dentro de los módulos que forman el sistema.

Objetivos de la Arquitectura Software

- ✓ Comprender y manejar la estructura de las aplicaciones complejas.
- ✓ Reutilizar dicha estructura (o parte de ella).
- ✓ Planificar la evolución de la aplicación.
- ✓ Analizar la corrección de la aplicación y su grado de cumplimiento frente a los requisitos iniciales.
- ✓ Permitir el estudio de propiedades específicas.

Ventajas de la Arquitectura Software

- ✓ Resalta aquellos aspectos estructuralmente importantes, tanto funcionales como no funcionales.
- ✓ Elimina muchos riesgos y errores de diseño, desarrollo e implantación.
- ✓ Permite un desarrollo paralelo, aumentando la productividad.

En el Proceso Unificado de Desarrollo(RUP), la arquitectura se compone de 4+1 vistas fundamentalmente, se dice que 4 de estas vistas están regidas por 1 vista rectora, estas vistas son: Vista de Casos de Uso, Vista Lógica, Vista de Procesos, Vista de Implementación, Vista de Despliegue. Estas vistas son la parte esencial de lo que se obtuvo en las etapas que más tributan a la arquitectura.

Generalmente en los sistemas se utilizan una serie de estilos arquitectónicos tales como: Modelo Vista Controlador (MVC), Arquitectura Basada en Objetos, Arquitectura Basada en Servicios (SOA) y Arquitectura en n_ Capas, este último es el que se decidió utilizar, por ser uno de los estilos más usados, ya que su uso está basado en la división del sistema en capas, separando los niveles de abstracción en sub-partes que se comunican entre sí e intercambian entradas y salidas para su funcionamiento.

La arquitectura en capas es muy usada en la realización de aplicaciones web, pues trae consigo la ventaja de aislar definitivamente la lógica de negocio de todo lo que tenga que ver con el origen de datos, ya que desde el manejo de la conexión, hasta la ejecución de una



Módulo para la Gestión de Información de Torneos de Ajedrez *Fundamentación Teórica*

consulta, la manejará la capa de Acceso a Datos. De este modo, ante cualquier cambio que ocurra, solo se tocará la capa específica, si se han respetado las reglas básicas de diseño, entonces no se deberán afrontar grandes modificaciones. Dentro de la arquitectura en n-capas existen diferentes variantes, de las cuales se decidió utilizar para el desarrollo del sistema propuesto la variante 3 capas, definiendo la misma por capas lógicas, teniendo una capa de presentación, una capa de lógica de negocio y la capa de acceso a datos, por último se representa la Base de Datos como fuente de datos.

- ✓ **Capa de Presentación o Interfaz de Usuario:** es la capa que se encuentra formada por formularios y controles, con la cual interactúan los usuarios.
- ✓ **Capa de Lógica de Negocio:** es la capa donde se encuentra la lógica del negocio o de la aplicación, la cual está formada por las entidades a través de las cuales se representan los objetos que van a ser manejados por toda la aplicación.
- ✓ **Capa de Acceso a Datos:** Capa final en el flujo e intercambio entre todas las capas, permite conocer el formato de los datos que se intercambian a través de los componentes de acceso a datos y la capa lógica de negocio, permite el acceso a los datos a través de los mismos.

1.5 Lenguajes de Programación Web

1.5.1 Lenguajes al lado del servidor

Los lenguajes de lado servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él [2]. Son lenguajes independientes del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo.

En estos lenguajes los scripts son almacenados en el servidor quien los ejecuta y traduce a HTML por lo que permanecen ocultos para el cliente.

1.5.1.1 PHP

Es un lenguaje de programación web libre e interpretado usado normalmente para la creación de páginas web dinámicas. Su acrónimo recursivo significa "PHP Hypertext Pre-processor" (inicialmente PHP Tools, o, *Personal Home Page Tools*). El funcionamiento de este lenguaje fluye de la manera siguiente: su interpretación y ejecución se da en el servidor web, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente.

Este lenguaje se puede utilizar además para generar archivos PDF, Flash, así como imágenes en diferentes formatos.

PHP permite la conexión a diferentes tipos de servidores de base de datos tales como: MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite; lo cual facilita la creación de aplicaciones web muy robustas.

Es un lenguaje de programación que tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux), Windows y Mac, y puede interactuar con los servidores web más populares (Apache). Actualmente PHP se encuentra en su versión 5 cuyo principal objetivo ha sido mejorar los mecanismos de la Programación Orientada a Objetos (POO) para solucionar las carencias de las anteriores versiones. Un paso necesario para conseguir que PHP sea un lenguaje apto para todo tipo de aplicaciones y entornos, incluso los más exigentes [3].

Algunas mejoras de esta nueva versión son su rapidez gracias a que primero se compila y luego se ejecuta, mientras que antes se ejecutaba mientras se interpretaba el código, otra mejora es su mayor independencia del servidor Web creando versiones de PHP nativas para más plataformas y un API (Interfaz de Programación de Aplicaciones) más elaborado y con más funciones.

Los principales usos del **PHP** son los siguientes:

- ✓ Programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos MySQL.
- ✓ Programación en consola, al estilo de Perl o Shell scripting.
- ✓ Creación de aplicaciones gráficas independientes del navegador, lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado.

Ventajas:

- ✓ Es un lenguaje multiplataforma.



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- ✓ 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, cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por citar algún ejemplo.
- ✓ No requiere definición de tipos de variables.
- ✓ Muy fácil de aprender.
- ✓ Se caracteriza por ser un lenguaje muy rápido.
- ✓ Es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor.
- ✓ Es independiente de la plataforma, puesto que existe un módulo de PHP para casi cualquier servidor web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje.
- ✓ Su independencia de la plataforma permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.
- ✓ Es completamente expandible. [4]

Desventajas:

- ✓ Se necesita instalar un servidor web.
- ✓ Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- ✓ La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- ✓ La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.



1.5.1.2 Perl

Perl (Lenguaje Práctico de Extracción y Reporte) es un lenguaje de programación web utilizado por muchas personas para el desarrollo de interfaces gráficas de usuario (GUI), desarrollo de páginas web, administración de sistema, programación en red, entre otras aplicaciones.

Es un lenguaje de programación web práctica, estable, flexible, poderosa, eficiente y multiplataforma, de código abierto con Licencia Publica General de GNU (GPL) y catalogado como el más popular debido a sus capacidades de manipulación de texto y el rápido ciclo de desarrollo.

Desventajas

- ✓ La principal desventaja de Perl se encuentra en el tiempo de ejecución de un programa, ya que un programa Perl es compilado cada vez que se ejecuta, por lo que puede resultar más lento que un programa similar escrito en otro lenguaje. Sin embargo, se han implementado técnicas para mejorar esta situación como guardar el compilado del programa en memoria y retrasar la compilación hasta que sea necesitada.
- ✓ PERL es un lenguaje rápido y potente que requiere obviamente un aprendizaje más largo y resulta más reservado para personas ya familiarizadas con la verdadera programación. [5]
- ✓ Utiliza muchos recursos de la máquina.

1.5.1.3 Python

Python es un lenguaje de scripting interpretados e independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web [6]. Es un lenguaje muy expresivo y considerado por muchos un lenguaje de programación de muy alto nivel, es muy legible, ofrece un entorno interactivo que facilita la realización de pruebas y ayuda a despejar dudas acerca de ciertas características del lenguaje. Su entorno de ejecución detecta muchos de los errores de programación que escapan al control de los compiladores y proporciona información muy rica para detectarlos y corregirlos.



Desventajas:

- ✓ Lentitud por ser un lenguaje interpretado [7].

1.5.2 Lenguajes al lado del cliente

Son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre-tratamiento [2].

Son totalmente independientes del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio.

1.5.2.1 JavaScript

Es un lenguaje interpretado por lo que no requiere compilación, utilizado principalmente en páginas web dinámicas que procesen la entrada del usuario y que sean capaces de gestionar datos persistentes usando objetos especiales, archivos y bases de datos relacionales. Su sintaxis es semejante a la del lenguaje java y el lenguaje C. Al contrario de java, JavaScript no es un lenguaje orientado a objetos propiamente dicho pues no dispone de herencia, es más bien un lenguaje basado en prototipos cuyo código interpretan todos los navegadores dentro de las páginas web.

Es un lenguaje muy fácil de aprender para quien ya conoce lenguajes similares como el C++ o Java. Tiene como característica principal el ser un lenguaje de scripting, pero, sobre todo, la de ser el lenguaje de scripting por excelencia y, sin lugar a dudas, el más usado [8], este se ejecuta en el agente usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Para explicar con pocas palabras la relación del mecanismo cliente-servidor, se puede decir que el servidor envía los datos al cliente y estos datos pueden llegar en dos formatos diversos: en formato texto (o ASCII) o en formato binario (o código máquina: El cliente sabe cómo comprender sólo el formato binario (es decir, la secuencia de 1 y 0), por lo que si los datos llegan en este formato son inmediatamente ejecutables (y, por desgracia, no dejan abierta la posibilidad de efectuar controles), mientras que si el formato es diverso tienen que ser interpretados y traducidos al formato binario y, por tanto, el cliente necesitará un filtro, o mejor dicho, un intérprete que sepa leer estos datos y los pueda traducir al binario. Los datos en formato texto son visibles al usuario como simples combinaciones de caracteres y de palabras y son, por tanto, fáciles de manipular, si bien requieren más tiempo para su interpretación a causa de los distintos pasos y de las transformaciones a las que deben



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

someterse para que el cliente pueda comprenderlos: Los datos en formato binario, sin embargo, son difíciles de comprender por el usuario, pero inmediatamente ejecutables por el cliente ya que no requieren fases intermedias.

Ventajas:

- ✓ Lenguaje de scripting seguro y fiable.
- ✓ Los script tienen capacidades limitadas, por razones de seguridad.
- ✓ Con JavaScript se pueden construir aplicaciones que varían desde la gestión de la información corporativa interna y su publicación en Intranets hasta la gestión masiva de transacciones de comercio electrónico.
- ✓ Se pueden crear páginas HTML dinámicas que procesen la entrada del usuario y que sean capaces de gestionar datos persistentes usando objetos especiales, archivos y bases de datos relacionales.
- ✓ Fue diseñado para ser utilizado dentro de "Browsers".
- ✓ El código JavaScript se ejecuta en el cliente por lo que el servidor no es solicitado más de lo debido; un script ejecutado en el servidor, sin embargo, sometería a éste a dura prueba y los servidores de capacidades más limitadas podrían resentir de una continua solicitud por un mayor número de usuarios.

Desventajas:

- ✓ A pesar de las ventajas que ofrece un "Browser" su funcionamiento es monolítico, lo que quiere decir que el ciclo de ejecución es realizado en una sola instancia o invocación, esto implica que no se pueden tener dos programas dinámicos simultáneamente en la misma ventana.
- ✓ Un problema importante es que el código es visible y puede ser leído por cualquiera, incluso si está protegido con las leyes del copyright.
- ✓ El código debe descargarse completamente.

1.5.2.2 HTML

HTML (Lenguaje de Marcas Hipertextuales) es un lenguaje de marcación o lenguaje estático diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Este puede ser creado y editado con cualquier editor de textos



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

básico, como puede ser Gedit, el Bloc de Notas de Windows, o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Wordpad, TextPad, Vim, entre otros. Existe además el famoso software de Macromedia llamado Dreamweaver, siendo uno de los más utilizados en el ámbito de diseño y programación Web.

Este lenguaje utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determinan la forma en la que debe aparecer en su navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador.

Desventajas:

- ✓ Lenguaje estático.
- ✓ La interpretación de cada navegador puede ser diferente.
- ✓ Guarda muchas etiquetas que pueden convertirse en "basura" y dificultan la corrección.
- ✓ El diseño es más lento.
- ✓ Las etiquetas son muy limitadas [7].

1.5.2.3 CSS

Las hojas de estilo en cascada (Hojas de Estilo en Cascada o CSS) es un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

Las ventajas de utilizar CSS son:

- ✓ Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- ✓ Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- ✓ Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

- ✓ El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

1.5.3 Combinación de tecnologías

1.5.3.1 AJAX

Es el acrónimo para Asynchronous JavaScript + XML, de la cual no se puede decir que es una tecnología web, sino que es el uso coordinado de distintas tecnologías en conjunto (XHTML, Document Object Model, El objeto XMLHttpRequest, XML) que permiten una mayor rapidez y eficiencia para las aplicaciones basadas en la web además de ser un término nuevo para describir dos capacidades de los navegadores: la posibilidad de hacer peticiones al servidor sin tener que volver a cargar la página y la posibilidad de analizar y trabajar con documentos XML. En concepto AJAX es: Cargar y renderizar una página, luego mantenerse en esa página mientras scripts y rutinas van al servidor buscando, en background, los datos que son usados para actualizar la página solo re-renderizando la página y mostrando u ocultando porciones de la misma[9].

Una aplicación AJAX elimina la naturaleza arrancar-frenar de la interacción en la Web introduciendo un intermediario -un motor AJAX- entre el usuario y el servidor.

AJAX incorpora:

- 1-Presentación basada en estándares usando XHTML y CSS.
- 2-Exhibición e interacción dinámicas usando el Document Object Model.
- 3-Intercambio y manipulación de datos usando XML and XSLT.
- 4-Recuperación de datos asincrónica usando XMLHttpRequest.
- 5-JavaScript poniendo todo junto [9].

Navegadores que permite:

El soporte de las aplicaciones AJAX dependerá de las características que el navegador permita.

- ✓ Navegadores basados en Gecko como Mozilla, Mozilla Firefox, SeaMonkey, Camino, Epiphany, NetScape versión 7.1 y superiores, y otros navegadores.
- ✓ Microsoft Internet Explorer para Windows versión 5.0 y superiores, y los navegadores basados en él.



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

- ✓ Navegadores con el API KHTML versión 3.2 y superiores implementado, incluyendo Konqueror versión 3.2 y superiores, Apple Safari versión 1.2 y superiores, y el Web Browser for S60 de Nokia tercera generación y posteriores.
- ✓ Opera versión 8.0 y superiores, incluyendo Opera Mobile Browser versión 8.0 y superiores.

Ventajas:

- ✓ Integra varias tecnologías existentes.
- ✓ Recuperación asíncrona de datos, el usuario no tienen que esperar después de una petición.
- ✓ Acercamiento de la metáfora de escritorio a la web.
- ✓ No requiere plugins.
- ✓ Se reduce el tamaño de la información intercambiada.
- ✓ La experiencia de usuario en la navegación es mucho más rica.
- ✓ El tráfico al servidor se reduce.

Desventajas:

- ✓ No dar señales inmediatas que los links se están cargando.
- ✓ Demasiado código Ajax hace lento al navegador.
- ✓ Uso excesivo de JavaScript: seguridad, compatibilidad, accesibilidad, complejidad.
- ✓ Pueden aumentar las llamadas al servidor.

1.6 Servidor Web

Un servidor web es un programa que implementa el protocolo HTTP (Protocolo de Transferencia de Hipertexto). Este protocolo está diseñado para transferir lo que se conoce como hipertextos, páginas web o páginas HTML (Lenguaje de Marcado de Hipertexto): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonido [10]. Los Servidores Web son usados para servir Páginas Web solicitadas por ordenadores clientes. Los clientes típicamente solicitan ver Páginas Web usando un Navegador.

Entre los servidores web más importantes pueden mencionarse: Apache y Cherokee.



1.6.1 Apache

El servidor HTTP Apache es un software (libre) de código abierto para plataformas Unix (BSD, GNU/Linux, entre otros), Windows, Macintosh y otras. Este servidor presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero no posee una interfaz gráfica que ayude en su configuración.

Apache es el servidor web más usado en el mundo, y es considerado el servidor web por excelencia, el cual tiene una gran aceptación puesto que actualmente casi el 70% de los servidores web visibles en Internet utilizan alguna versión de Apache [11]. Desde su nacimiento ha demostrado con creces su estabilidad, solidez y su mayor rendimiento ante sus competidores, sus principales metas de diseño son: velocidad, simplicidad, multiplataforma y facilidad del desarrollo distribuido.

Este servidor se encuentra estructurado en módulos, los cuales son una manera de agrupar y modularizar ciertos funcionamientos para el servidor, existen una gran cantidad de módulos para utilizarse con Apache, algunos son: "Virtual Hosting", "Mod_JK (Java)" y "Rewrite", entre otros [12]. Los módulos del Apache se pueden clasificar en tres categorías:

- ✓ Módulos Base: módulo con las funciones básicas del Apache [12].
- ✓ Módulos Multiproceso: son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones [12]. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.
- ✓ Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor [12]. De forma que no es necesario volver a instalar el software.

Una de las principales razones por las que se emplean módulos en Apache, es que no toda instalación requiere de las mismas funcionalidades, esto quiere decir que una instalación que utilice PHP probablemente no requiera de Tomcat (Java), o bien posiblemente no todas las instalaciones requieran de "Virtual Hosting"[13].

Ventajas:

- ✓ Apache es un servidor altamente configurable de diseño modular a través del cual se ha adaptado siempre a una gran variedad de entornos.



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

- ✓ Open Source: Código abierto, es el término con el que se conoce al software distribuido y desarrollado libremente.
- ✓ Multi-plataforma.
- ✓ Extensible.
- ✓ Popular.
- ✓ Gratuito.
- ✓ La utilización de apache ayuda en la mejora del posicionamiento. El servidor web Apache junto con el módulo mod_rewrite puede convertirse en una herramienta muy jugosa para crear páginas con enlaces amigables para los buscadores [14].
- ✓ Apache trabaja con gran cantidad de Perl, PHP y otros lenguajes de script.
- ✓ Apache permite la creación de ficheros de log a medida del administrador.
- ✓ Su coste gratuito y gran fiabilidad le convierten en una herramienta potente con propiedades como: ser capaz de presentar mensajes de error totalmente configurables, bases de datos de autenticación, entre otros.
- ✓ Fiabilidad

1.6.2 Cherokee

Cherokee HTTP Server es un servidor http gratuito y multiplataforma escrito enteramente en C., no depende más allá de un estándar de librerías C. Es un servidor muy rápido a la vez que implementa las funcionalidades más usadas por los servidores http.

Es extensible con bloqueos y soporta altas configuraciones para lectura de archivos o cadenas de caracteres, TLS/SSL (vía GNUTLS or Open SSL), virtual hosts, autenticaciones, rasgos de caches amigables, PHP, manejo de errores y otros.

1.7 Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de Datos, asegurando su integridad, confidencialidad y seguridad, y debe permitir definir, construir y manipular la base de datos.



1.7.1 MySQL

MySQL es un gestor de bases de datos SQL (Structured Query Language) que constituye una implementación Cliente-Servidor que consta de un servidor y diferentes clientes (programas/librerías). Se puede agregar, acceder, y procesar datos grabados en una base de datos [15].

Sencillamente, MySQL, es un servidor de base de datos que funciona de forma separada del servidor Web, es gratuito, relacional, multihilo, multiplataforma y multiusuario con más de seis millones de instalaciones y constituye una versión ligera del SQL de Microsoft. Este servidor se emplea para acceder a una base de datos a través del navegador para lo cual se debe tener instalado PHP (u otro) en el Apache (u otro) para comunicarse con las bases de datos; siendo Apache, PHP y MySQL una combinación muy potente y muy usada. Actualmente, MySQL, es el servidor de base de datos Open Source más popular de Internet. Su conectividad, velocidad, rapidez, confiabilidad, robustez y facilidad de uso tanto para volúmenes de datos grandes como pequeños y además su seguridad hacen de MySQL un servidor bastante apropiado para acceder a bases de datos en Internet; y el hecho de ser una base de datos muy rápida en la lectura de datos es ideal para aplicaciones web ya que las mismas poseen baja concurrencia en la modificación de datos.

MySQL tiene la capacidad de trabajar servidores con varios procesadores. Posee gran velocidad en la utilización de joins y procesos de optimización. Soporta muchos tipos de columnas para las tablas: FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM y OpenGIS (Modelo Geométrico).

MySQL es más rápida que PostgreSQL a la hora de resolver consultas, tiene mejor documentación y se ha orientado más a facilitarle la vida al desarrollador proporcionando mejores herramientas de administración.

1.7.2 PostgreSQL

Es el gestor de bases de datos de código abierto más avanzado en la actualidad, ofrece control de concurrencia multi-versión, y soporta casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), cuenta además con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y python)[16].



Desventajas:

- ✓ Consume más recursos y carga más el sistema.
- ✓ Límite del tamaño de cada fila de las tablas a 8k (se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento).
- ✓ Es de 2 a 3 veces más lento que MySQL.
- ✓ Menos funciones en PHP.
- ✓ Consume gran cantidad de memoria.
- ✓ Responde a una única consulta

1.8 Servidor de Correo

Un servidor de correo es una aplicación informática que permite enviar mensajes de correo de unos usuarios a otros, independientemente de la red que los usuarios implicados en el servicio estén utilizando.

Para lograr la conexión se definen una serie de protocolos, cada uno con una finalidad definida y específica:

- ✓ Protocolo de Transferencia de Correo Simple (SMTP): Es el protocolo que se utiliza para que dos servidores de correo intercambien mensajes.
- ✓ Protocolo de Oficina de Correo (POP): Se utiliza para obtener los mensajes guardados en el servidor y pasárselos al usuario.
- ✓ Protocolo de Acceso a Mensaje de Internet (IMAP): Su finalidad es la misma que la de POP, pero el funcionamiento y las funcionalidades que ofrecen son diferentes.

1.8.1 SMTP EMail Server

Un servidor de correo SMTP es una aplicación que gestiona y entrega correos electrónicos.

Cada vez que se envíe un mensaje de correo electrónico, el cliente de correo interactúa con un servidor SMTP, que a su vez, se conecta a un servidor SMTP remoto para completar la entrega [17].

1.9 Proceso de Desarrollo de Software

Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los



pasos necesarios para terminar el proyecto. La existencia de este proceso, el cual se encarga de la integración de todas las fases del desarrollo de software, es sumamente importante pues en la actualidad la producción de software se ha incrementado en gran medida en cuanto a calidad, cantidad y complejidad.

Las piedras angulares del proceso de desarrollo del software son: el proyecto, las personas y el producto.

1.10 Metodologías de desarrollo de software

En el momento de llevar a cabo el desarrollo de un software siempre viene a la mente una pregunta: ¿Qué metodología usar?

Todo desarrollo de software es arriesgado y difícil de controlar, pero si no se lleva una metodología de por medio, lo que se obtiene son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Es por ello que antes que todo es necesario realizar un estudio de las metodologías existentes para así poder definir cuál es la más adecuada para el desarrollo de la aplicación.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software, la misma define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. En pocas palabras: una metodología es un proceso.

Algunas de las metodologías más utilizadas son: Proceso Unificado de Desarrollo (RUP), Extreme Programming (XP) y Microsoft Solution Framework (MSF).

1.10.1 El Proceso Unificado de Desarrollo (RUP)

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, es el resultado de varios años de trabajo, en el que se han unificado técnicas de desarrollo, a través de UML, y la utilización de muchas metodologías por parte de los clientes.

RUP divide su proceso de desarrollo de software en cuatro fases y es una metodología en la cual las actividades se han agrupado en grupos lógicos definiéndose nueve flujos de trabajo principales. Anexo 1

RUP es una metodología que define como sus principales elementos:

- ✓ **Trabajadores (quién):** Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

- ✓ **Actividades (cómo):** Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- ✓ **Artefactos (qué):** Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- ✓ **Flujo de actividades (cuándo):** Secuencia de actividades realizadas por trabajadores y que producen un resultado de valor observable.

Características de RUP

- ✓ Unifica los mejores elementos de metodologías anteriores.
- ✓ Preparado para desarrollar grandes y complejos proyectos.
- ✓ Orientado a Objetos.
- ✓ Utiliza el UML como lenguaje de representación visual.

El ciclo de vida de RUP se caracteriza por:

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



centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos.

1.10.2 Extreme Programing (XP)

XP o Programación Extrema, se considera la más popular de las denominadas metodologías ágiles [18], es flexible en cuanto a requerimientos cambiantes y no produce una potente documentación del sistema. Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer. Esta consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Fases de la Metodología XP

1ª Fase: Planificación del proyecto.

2ª Fase: Diseño.

3ª Fase: Codificación.

4ª Fase: Pruebas [18].

XP define cuatro variables para proyectos de software: **coste, tiempo, calidad y ámbito** [18].

La metodología XP se basa en:

- ✓ **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se puedan hacer pruebas de las fallas que pudieran ocurrir.
- ✓ **Re-fabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- ✓ **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, esta consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está realizando en ese momento.



1.10.3 Microsoft Solution Framework (MSF)

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

MSF tiene las siguientes características:

- ✓ **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- ✓ **Escalable:** puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas o más.
- ✓ **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- ✓ **Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos responsables de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

1.11 Lenguaje Unificado de Modelado (UML)

En la medida en que se hacen más complejos los problemas, se hace más beneficioso tener una proyección o abstracción de las soluciones en etapas tempranas del desarrollo de las mismas, a esta "representación en pequeña escala; Arquetipo o punto de referencia para imitar o reproducir" se denomina modelos, que sirven para la toma de decisiones, en el diseño y la implementación. Los modelos brindan una vista a menor y menos costosa escala de la solución final, y resulta más fácil efectuar cambios a los mismos, una vez detectados los problemas, que tener que efectuar cambios a la solución ya implementada.

Con el objetivo de hacer el trabajo en el desarrollo de software más eficiente surgen las denominadas herramientas CASE (por sus siglas en inglés Computer Aided Software Engineering), las mismas pueden llegar a brindar un soporte muy alto para las actividades del modelado y todo lo que de estas puede derivarse, automatizan tareas repetitivas que suelen ser tediosas y que no requieren de intelecto para su realización.



El modelado resulta beneficioso en al menos cuatro sentidos:

- ✓ **Ayuda a encontrar la solución.** Esta constituye una de las características más provechosa del modelado. El especialista construyendo un modelo, genera a partir de un proceso creativo, la solución a un nivel determinado de abstracción. A nadie se le ocurriría comenzar a crear tablas en el Sistema Gestor de Bases de Datos y relacionarlas sin haber efectuado un diseño previo, y haber confrontado todos los elementos, evaluando las posibles implicaciones, o sea, encontrar una solución posible, que luego es convertida a un modelo más concreto (la Base de Datos).
- ✓ **Es menos costoso.** Claramente es mucho más simple evaluar y cambiar un modelo, que cambiar el código fuente, cuesta menos construir un prototipo y efectuar sobre este cualquier cambio después de evaluarlo que implementar para luego rehacer. Es mucho más simple determinar en la solución a un alto nivel un error y extender su corrección en los momentos tempranos. Un modelo podría servir para evaluar la factibilidad de una solución; cuán simple puede resultar desechar un modelo, y cuán doloroso desechar una solución ya construida.
- ✓ **Facilita la generación automática del resultado final.** En muchas otras disciplinas, el modelado finaliza con la construcción de una maqueta (Modelo), y luego es utilizado solamente como referencia en la construcción real. Afortunadamente en el software, es diferente, los modelos que se construyen pueden ser empleados por las herramientas CASE para obtener el resultado final, el código fuente, las bases de datos y ficheros derivados que constituyen el producto de software.
- ✓ **Facilitan la comunicación y documentación.** Si para la construcción de los modelos se emplea un lenguaje estándar, estos pueden ser entendidos por especialistas a lo largo de todo el mundo. Adicionalmente las herramientas CASE brindan también soporte para efectuar reportes automáticamente a los modelos que permiten a la distancia de un clic, obtener la documentación total o parcial del modelo en cuestión.

Para modelar la Aplicación Web se utilizará el Lenguaje Unificado de Modelado, el cual es el más utilizado y conocido en la actualidad para la modelación de sistemas de software. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como: procesos de negocios y funciones del sistema, y aspectos



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

1.12 Herramientas CASE de Modelado con UML

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son varias aplicaciones informáticas que automatizan una parte del ciclo de desarrollo de software y cuyo principal destino es incrementar la productividad en el desarrollo de software logrando una disminución de sus costes en concepto de tiempo y de dinero.

Son una aplicación de métodos y técnicas a través de las cuales se hacen útiles a las personas comprender las capacidades de las computadoras, por medio de programas, de procedimientos y su respectiva documentación.

Las herramientas CASE permiten al administrador de un proyecto informático, llevar adelante el proyecto de forma eficaz y eficiente.

Estas herramientas son utilizadas para mostrar los elementos claves del proceso, de manera que sea posible entenderlo mejor.

Con la aplicación de estas herramientas se puede:

- ✓ Mejorar la productividad en el desarrollo y mantenimiento del software.
- ✓ Aumentar la calidad del software.
- ✓ Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- ✓ Mejorar la planificación de un proyecto.
- ✓ Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- ✓ Ayudar a la reutilización del software, portabilidad y estandarización de la documentación.
- ✓ Gestionar todas las fases de desarrollo de software con una misma herramienta.
- ✓ Facilitar el uso de las distintas metodologías propias de la ingeniería del software.



La clasificación más habitual basada en las fases del ciclo de desarrollo que cubren son:

- ✓ *Upper CASE (U-CASE)*, herramientas que ayudan en las fases de planificación, análisis de requisitos y estrategia del desarrollo, usando, entre otros diagramas UML.
- ✓ *Middle CASE (M-CASE)*, herramientas para automatizar tareas en el análisis y diseño de la aplicación.
- ✓ *Lower CASE (L-CASE)*, herramientas que semi-automatizan la generación de código, crean programas de detección de errores, soportan la depuración de programas y pruebas. Además automatizan la documentación completa de la aplicación. Aquí pueden incluirse las herramientas de desarrollo rápido de aplicaciones.

1.12.1 Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Algunas características de Visual Paradigm:

- ✓ Soporte de UML versión 2.1.
- ✓ Diagramas de Procesos de Negocio, Decisión, Actor de negocio, Documento.
- ✓ Modelado colaborativo con CVS y Subversión.
- ✓ Ingeniería de ida y vuelta.
- ✓ Ingeniería inversa - Código a modelo, código a diagrama.
- ✓ Generación de código - Modelo a código, diagrama a código.
- ✓ Editor de Detalles de Casos de Uso para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- ✓ Diagramas de flujo de datos.
- ✓ Ingeniería inversa de bases de datos [19].



1.12.2 Umbrello

Umbrello UML Modeller ayuda en el proceso de desarrollo de software usando el estándar 'Unified Modelling Language' (UML) que le permitirá crear diagramas para diseñar y documentar sus sistemas ya que es una herramienta de diagramas que ayuda en el proceso del desarrollo de software, la cual facilitará la creación de un producto de alta calidad, especialmente durante fases de análisis y diseño del proyecto.

Desventajas:

- ✓ La pantalla de diseño del diagrama está poco automatizada. Se basa en Drag & Drop, pero es demasiado simple.
- ✓ Las imágenes que genera son demasiado sencillas.

1.13 Entorno de Desarrollo Integrado. Zend Studio

Zend Studio, 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.

El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario.

Está compuesto de dos partes en las que se dividen las funcionalidades: de parte del cliente y de parte del servidor. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración.

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. Esta ayuda funciona no solo para las funciones definidas en el lenguaje sino que también reporta ayuda con las funciones que se vayan creando por los usuarios.

La Gestión de Proyecto

Gestionar los proyectos es muy útil para mejorar la productividad en la programación. Los proyectos permiten guardar mucha más información al programa sobre los archivos, discos, servidores, entre otros, que se gestionen en las aplicaciones PHP.



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

Zend Studio implementa además unas interesantes opciones para trabajar en grupo, al integrar el sistema de trabajo conocido como CVS.

La Herramienta de Depuración

Zend Studio dispone de una herramienta muy interesante de depuración. Gracias a la cual se pueden ejecutar páginas y conocer en todo momento el contenido de las variables de la aplicación y las del entorno, como las cookies, las recibidas por formulario o en la sesión. Es posible colocar puntos de parada de los scripts y realizar las acciones típicas de depuración.

Para el desarrollo de la aplicación propuesta se utilizará Zend Studio 5, el cual es el soporte en desarrollos y pruebas de PHP con el set más completo de herramientas para la creación de aplicaciones altamente fiables como la que se quiere obtener. Asegura el desarrollo de software mediante la combinación del IDE líder para PHP con un entorno de prueba que agiliza la seguridad de la calidad, integración y las etapas de los procesos.

1.14 Conclusiones

Enfocando este trabajo a las necesidades que posee la Cátedra de Ajedrez de la Universidad, a partir de la evidente existencia de un problema de centralización y visualización de la información referente a los torneos de ajedrez que se llevan a cabo en la misma y con un previo estudio de las principales herramientas y tecnologías más utilizadas en la realización de aplicaciones web en software libre, se seleccionaron, para la posterior construcción y desarrollo del sistema que se propuso: como lenguaje de programación web para la implementación de la aplicación, PHP en su versión 5.0. Este lenguaje propicia facilidades para su aprendizaje, es multiplataforma, con amplia capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, y es uno de los lenguajes de programación web más manejados por la comunidad UCI. El hecho de que este trabajo será implementado sobre el Sistema Operativo Linux y que el lenguaje de programación web que se utilizará será PHP, es obvio que el servidor web que se empleará será Apache, en su versión 2.0, el más comúnmente utilizado en sistemas GNU/Linux. Como Sistema Gestor de Base de Datos se utilizará MySQL por ser Apache, PHP y MySQL una combinación muy potente y muy usada, además de que la popularidad de MySQL está muy ligada a PHP, el cual cuenta con una serie de funciones exclusivas para la interacción permitiendo así la obtención de los datos. En cuanto a consideraciones de estabilidad del servidor entre MySQL y PostgreSQL, cada comparativa da datos contradictorios, pero en general se puede decir que el más estable es MySQL, y que PostgreSQL tiende a



Módulo para la Gestión de Información de Torneos de Ajedrez Fundamentación Teórica

desperdiciar memoria y sobrecargar bastante el sistema. Como editor de páginas PHP solo se cuenta con Zend Studio y se utilizará su versión 5.0.

Como metodología que guiará cada uno de los procesos que se llevarán a cabo a lo largo del desarrollo de la aplicación web, se escogió la metodología RUP por ofrecer un proceso de desarrollo sumamente organizado, por tener la característica de permitir obtener en cada iteración un producto perfeccionado y una amplia documentación del sistema. Dada la metodología escogida el lenguaje de modelado más recomendado para la visualización de una abstracción de la solución, es UML; utilizando para el modelado del mismo la herramienta CASE Visual Paradigm, la cual permite elaborar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Se aplicará la arquitectura n-capas por tener grandes ventajas en el momento de realizar cambios o correcciones, y por razones expuestas con anterioridad.



Capítulo 2

Características del Sistema

2.1 Introducción

En este capítulo se comenta y describe el objeto de estudio del presente trabajo y se muestran cuáles son los procesos que previamente serán automatizados. Se da una breve descripción del negocio correspondiente y se describen actores, trabajadores, entidades y reglas del negocio. Se muestra además el diagrama de casos de uso del negocio en el cual se relacionan actores y procesos del negocio, y el modelo de objeto donde colaboran trabajadores del negocio, entidades del negocio y unidades de trabajo.

Por otra parte se identifican los requisitos funcionales y no funcionales del sistema, que darán solución a las necesidades existentes, cuáles son las funcionalidades que brinda el sistema y los actores y su interacción con el mismo

2.2 Objeto de Estudio

Para la realización de una aplicación web es necesario tener un basto conocimiento de todo lo que conlleva a su desarrollo, es por ello que el objeto de estudio del presente trabajo estará enfocado en el Proceso de Desarrollo de Aplicaciones Web para el ajedrez.

2.3 Flujo actual de los procesos

Si se analiza el flujo actual de los procesos se puede conocer en realidad cómo funciona el negocio que se está tratando, pudiéndose obtener además los posibles resultados, los cuales pueden ser: obtener una determinada información, un determinado servicio o simplemente un producto.

Cuando se realiza el estudio de cómo se ejecutan estos procesos es posible identificar los posibles problemas que entorpecen el desarrollo de las actividades que se realizan en la entidad donde están presentes. Dichos problemas pueden ser: tardanza en la entrega de una información, búsqueda de información sumamente engorrosa, pérdida o deterioro de la información, problemas en la visualización de la información, entre otros.

Los procesos que se llevan a cabo en la Cátedra de Ajedrez de la Universidad para la organización y actualización de los datos de un torneo de ajedrez son los siguientes:



Módulo para la Gestión de Información de Torneos de Ajedrez. Características del Sistema

Organizador del Torneo: Es el responsable de la organización de los diferentes torneos que se realizan en la Universidad a lo largo de todo el curso. Para ello elabora una propuesta de solicitud del torneo que incluye los objetivos del mismo la cual es entregada a las entidades superiores para la aprobación del torneo; además es el encargado de la elaboración del listados de los jugadores que desea invitar al torneo, cuyo listado incluye todos los datos de los participantes, además de solicitar los árbitros para el mismo. La invitación a los jugadores es hecha a través de los medios posible, teléfono, correo tradicional o personalmente. Por otra parte registra toda esta información generada en el torneo con el objetivo de poner parte de ella a disposición de los interesados.

2.4 Análisis crítico de la ejecución de los procesos

Los procesos que se llevan a cabo actualmente en la Cátedra de Ajedrez de la Universidad con respecto a la gestión de los torneos de ajedrez no se realizan con la eficiencia requerida producto de factores tales como:

- ✓ No centralización de la información generada en el torneo.
- ✓ Deterioro de la información
- ✓ Pérdida de la información
- ✓ Falta de organización en la información requerida para la organización del torneo.
- ✓ Manejo manual de la información
- ✓ Almacenamiento incorrecto de los datos.

Dichos factores traen consigo que en el momento de llevar a cabo la organización de un torneo el Organizador del Torneo necesite consultar una serie de documentos para obtener todos los aspectos que incluyen el desarrollo de dicho proceso. Antes de planificar la realización de un torneo el Organizador debe informarse de las actividades más relevantes que se llevan a cabo en la Universidad para evitar que las mismas coincidan con el torneo y obstaculicen el efectivo desarrollo del mismo, esta información es obtenida, en varias ocasiones, de diferentes puntos. Para confeccionar el listado de los jugadores con los que desea contar en el torneo necesita realizar una engorrosa búsqueda en el listado de jugadores de la Federación Internacional de Ajedrez (FIDE), confecciona un listado inicial con todos los datos de los jugadores, realiza la invitación de los mismos por las vías que le sean posibles, teléfono, correo tradicional, de forma personal, entre otras y va actualizando el listado inicial manualmente hasta obtener el listado oficial.



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

En la Cátedra de Ajedrez se lleva registro de todos los torneos que se realizan, lo cual va generando un cúmulo de información que hace engorrosa la búsqueda de datos determinados en determinado momento.

2.5 Procesos objeto de automatización

Con el objetivo de resolver los problemas existentes en la Cátedra de Ajedrez de la Universidad relacionados con el proceso de organización de los torneos de ajedrez, se pretende llevar a cabo la automatización de aquellos procesos que contribuyan a darle solución a los mismos.

Procesos a automatizar:

- ✓ Control del listado de los jugadores y árbitros del torneo.
- ✓ Invitación de los jugadores y árbitros.
- ✓ Control de las actividades docentes y extra-docentes de la Universidad.
- ✓ Registro de las partidas realizadas en cada ronda del torneo.
- ✓ Control de los resultados del torneo.
- ✓ Comprobar si es posible la realización del torneo.
- ✓ Almacenamiento de la información.

2.6 Propuesta de Sistema

Primeramente se debe puntualizar que el sistema trata de traducir y organizar las necesidades existentes.

Dada la anterior aclaración se propone realizar un sistema, con el cual se pretende optimizar los procesos que se llevan a cabo en la Cátedra de Ajedrez de la Universidad relacionados con la organización de torneos de ajedrez, de tal modo que se pueda obtener y gestionar de forma rápida cualquier información requerida, esto sería de gran utilidad para todos aquellos que estén relacionados de una forma u otra con la información que se maneja.

Se desea que el sistema le permita al organizador de la Cátedra de Ajedrez crear y administrar un sitio web para gestionar toda la información relacionada con los torneos de ajedrez, con el objetivo de que toda la información que se genere se encuentre centralizada y se pueda acceder a ella de forma rápida y eficiente.



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

Se pretende que el sistema proporcione todo lo necesario para la realización de cualquier torneo. Cuáles serían los aspectos que se deben tener en cuenta en el momento de organizar un torneo: cantidad mínima y máxima de jugadores que permite el sistema de juego, definir claramente los objetivos del torneo, buscar jugadores y árbitros, enviar la invitación al torneo, entre otras. El sistema permitirá además llevar registro y control de los resultados de cada ronda, y dará la posibilidad de agregarle a las mismas noticias y partidas. En conclusión, el sistema permitirá gestionar y consultar de forma rápida cualquier resultado o información del torneo.

2.7 Modelo del Negocio

Describir cómo funciona el negocio actualmente le da al desarrollador la posibilidad de comprender la estructura y dinámica de la organización, así como la posibilidad de identificar las posibles mejoras.

2.7.1 Descripción del Negocio Actual

En la Cátedra de Ajedrez de la UCI durante el curso se efectúan una serie de eventos entre los que se destacan la realización de torneos de ajedrez. Para la concepción de estos torneos el **Organizador del Torneo** elabora una propuesta del torneo que desea efectuar donde incluye los objetivos del mismo. Esta propuesta se la entrega al **Director de Deporte** quien da su aprobación y entrega el documento a la **Vice-rectoría de Extensión Universitaria**, si esta aprueba la propuesta la transfiere a la **Dirección de la Universidad**, la cual da la respuesta definitiva a la propuesta enviada.

Una vez aprobado el torneo el **Organizador del Torneo** prosigue a la gestión de los listados de los posibles jugadores y árbitros que participarán en el torneo y los contacta, ya que una de las características principales de los torneos que se realizan en la Universidad es que son torneos por invitación. El **Organizador del Torneo** registra la respuesta de los participantes y conforma el listado oficial de los que asistirán al torneo, terminando así la gestión del mismo.



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

2.8 Reglas del Negocio a tener en cuenta

Reglas de Acción

Restricción de Operaciones

- ✓ No se pueden planificar torneos simultáneos.
- ✓ Para la realización de un torneo de modalidad Todos contra Todos la cantidad de participantes debe ser como mínimo de 3 ó 4 y no debe exceder de 20.

2.9 Casos de Uso del Negocio (CUN)

Un caso de uso del negocio describe un conjunto de actividades lógicamente relacionadas que se ejecutan en una determinada secuencia en el negocio, de manera que producen un resultado de valor observable a un actor individual en el negocio. Los casos de uso del negocio representan un proceso del negocio.

2.10 Actores del Negocio

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. Lo que se modela como actor es el rol que se juega cuando algo o alguien interactúan con el ambiente del negocio para beneficiarse de sus resultados.

A continuación se muestran los actores del negocio y una descripción de los mismos.

Actores del Negocio	Descripción
Organizador del Torneo	-Es la persona encargada de solicitar la aprobación de la convocatoria del torneo, de llevar la logística del evento, de invitar a los jugadores y llevar el registro de la información generada.
Usuario	-Persona espectadora que se interesa por el desarrollo del torneo, o por los datos almacenados de los mismos.

Tabla 1: Actores del negocio

2.11 Diagrama de casos de uso del negocio

Un diagrama constituye la representación gráfica de una serie de elementos, y visualiza el sistema desde diferentes perspectivas.



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

Un diagrama de casos de uso del negocio es la representación gráfica de los procesos del negocio y los actores que intervienen en el mismo, así como la relación que se establece entre ambos.

Seguidamente se muestra el diagrama de casos de uso correspondiente al negocio que se pretende automatizar:

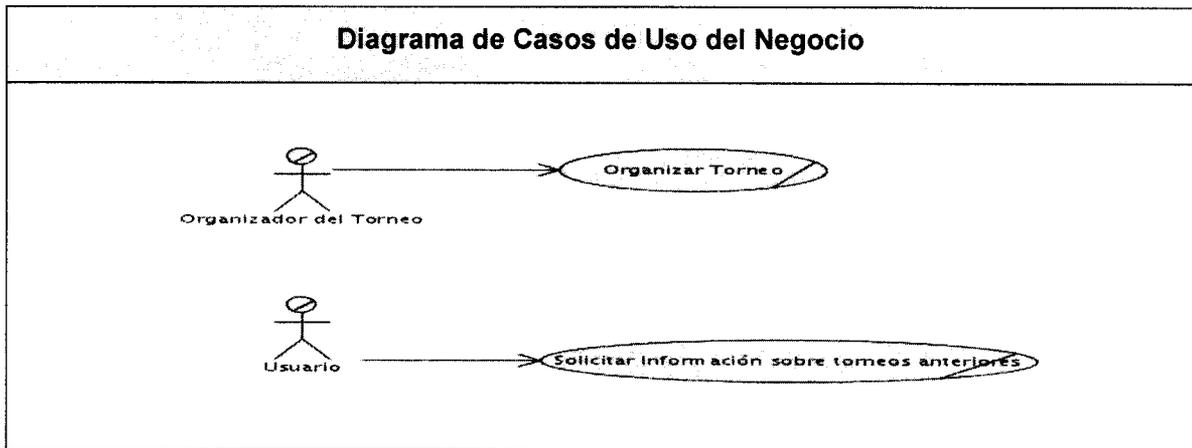


Figura 1: Diagrama de Casos de Uso del Negocio.

2.12 Trabajadores del Negocio

Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Son los encargados de realizar las actividades y son propietarios de elementos.

A continuación se muestran los trabajadores del negocio y la descripción correspondiente de los mismos:



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

Trabajador	Descripción
Director de Deporte	-Es la persona que dirige el deporte en la UCI, es el enlace entre la Universidad y la Cátedra de Ajedrez. -Es la persona que se encarga de gestionar las solicitudes que le hace el organizador para la organización del torneo.
Presidente de Reglas y Arbitraje	-Es la persona a quien se le solicitan los árbitros para el torneo.
Jugador	-Persona que se invita al torneo para que juegue en el mismo.
Vice-rectoría de Extensión Universitaria	-Entidad responsable de las actividades extra-docentes que se realizan en la Universidad, dígase, actividades culturales y deportivas.
Dirección de la Universidad	-Es la máxima entidad dentro de la UCI.
Organizador del Torneo	-Es la persona encargada llevar el registro de la información generada y ponerla a disposición de los usuarios.

Tabla 2: Trabajadores del Negocio

2.13 Entidades del Negocio

Representan los objetos que los trabajadores del negocio utilizan o generan durante la realización de los casos de uso del negocio. Son un contenedor de información que se utiliza en los procesos del negocio para obtener o actualizar los datos.

2.14 Especificación de los Casos de Uso del Negocio

Las especificaciones de los casos de uso del negocio se pueden hacer por dos vías: mediante la descripción textual o mediante la realización de diagramas de actividades.

Cuando se realiza mediante la descripción textual se describe con palabras cómo funciona el flujo de actividades para la realización de un caso de uso del negocio, o sea, qué desea obtener el actor del negocio y la secuencia de pasos que se darán en el negocio para satisfacer la solicitud del actor del negocio.

De manera similar ocurre cuando se utilizan los diagramas de actividades, solo que el flujo de las actividades correspondientes a la realización del caso de uso del negocio se lleva a



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

cabo de forma gráfica, esto ayuda a identificar qué funciones deberá asumir el producto software y quiénes serán los actores del futuro sistema.

A continuación se muestran las descripciones textuales y los diagramas de actividades de los distintos casos de uso del negocio.

2.14.1 Descripción textual y diagrama de actividades del Caso de Uso del Negocio

“Organizar Torneo”

Caso de Uso	Organizar Torneo
Actores:	Organizador del Torneo
Trabajadores:	Director de Deporte, Vice-rectoría de Extensión Universitaria, Dirección de la Universidad, Presidente de Reglas y Arbitraje y Jugador.
Resumen:	El CUN se inicia cuando el organizador del torneo solicita la aprobación para la realización del torneo, la solicitud es confirmada, se realizan las acciones correspondientes y termina el CUN.
Precondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1-El Organizador del Torneo solicita la aprobación del Torneo mediante la entrega de una propuesta del torneo donde incluye los objetivos del mismo.	1.1-El Director de Deporte recibe la propuesta.
	1.2-Confirma su aprobación envía la propuesta a la Vice-rectoría de Extensión Universitaria.
	1.3-La Vice-rectoría de Extensión Universitaria recibe la propuesta.
	1.4-Da su aprobación la envía a la Dirección de la Universidad.
	1.5-La Dirección de la Universidad da la respuesta definitiva sobre la realización del torneo.
	1.6-Envía la respuesta de aprobación al Organizador del Torneo.
2-El Organizador del Torneo consulta el	



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

listado de los jugadores.	
3-El Organizador del Torneo elabora el listado de jugadores.	
4- El Organizador del Torneo contacta a los jugadores.	4.1-Los jugadores dan su respuesta a la invitación hecha por el Organizador del Torneo.
5- El Organizador del Torneo registra la respuesta de los jugadores y conforma el listado oficial de los jugadores.	
6-El Organizador del Torneo solicita los árbitros para el torneo.	6.1-El Presidente de Reglas y Arbitraje se encarga de la gestión de los árbitros.
	6.2-El Presidente de Reglas y Arbitraje elabora el listado de árbitros y lo envía
7-El Organizador del Torneo recibe el listado.	
8-El Organizador del Torneo elabora el listado final de participantes.	
9-El Organizador del Torneo gestiona todo lo relacionado con la logística del torneo y termina el CUN.	
Flujo Alternativo	
Acción 1	1.2.1-Si el Director de Deporte no aprueba la propuesta se lo hace saber al Organizador y termina el CUN.
Acción 1.2	1.3.1-Si la Vice-rectoría de Extensión Universitaria no aprueba la propuesta se lo hace saber al Organizador y termina el CUN.
Acción 1.4	1.6.1-Si la Dirección de la Universidad no aprueba la propuesta se lo hace saber al Organizador del Torneo y el CUN termina.
Poscondiciones	Torneo organizado
Mejoras	Se puede automatizar el proceso de consultar el listado de jugadores FIDE, elaborar listado de jugadores, elaborar listado final de participantes, contactar los jugadores, enviar la respuesta de invitación al torneo y registrar la respuesta de los jugadores.
Prioridad	Crítica



Tabla 3: Descripción Textual del CUN “Organizar Torneo”

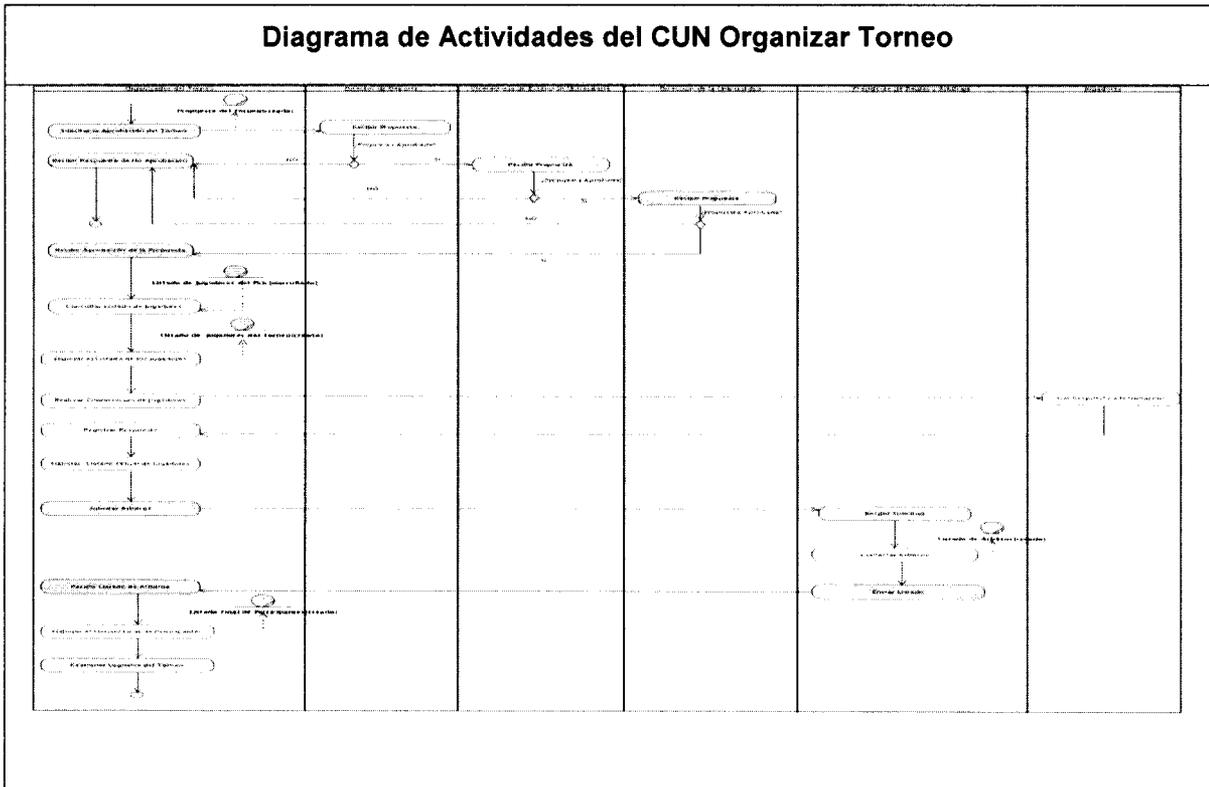


Figura 2: Diagrama de Actividades del CUN “Organizar Torneo”

2.14.2 Descripción textual y diagrama de actividades del Caso de Uso del Negocio

“Solicitar Información sobre Torneos Anteriores”

Caso de Uso	Solicitar Información sobre Torneos Anteriores
Actores:	Usuario
Trabajadores:	Organizador del Torneo
Resumen:	El CUN se inicia cuando el usuario se presenta en la cátedra de ajedrez para solicitar alguna información de torneos anteriores, el organizador del torneo o responsable de la cátedra le facilita la información y termina el CUN.
Precondicione s:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

1-El Usuario se presenta en la cátedra de ajedrez y solicita información de torneos anteriores.	1.1-El Organizador del Torneo solicita el torneo del cual el usuario desea obtener la información
2-El Usuario informa la edición y nombre del torneo del que desea obtener información.	2.1-El Organizador del Torneo solicita el tipo de información que desea obtener del torneo seleccionado.
3-El Usuario informa qué información desea obtener del torneo.	3.1-El Organizador del Torneo busca en la documentación que tiene archivada del torneo especificado por el usuario la información requerida.
	3.2- Si la información existe el Organizador del Torneo entrega la información solicitada por el usuario.
4-El Usuario obtiene la información que solicitó, se retira y termina el CUN.	
Acción 3.1	3.2-Si la información no está registrada, el Organizador del Torneo le informa al usuario que no existe esa información.
4-El Usuario se retira y termina el CUN.	
Mejoras	Se puede automatizar las consultas a la documentación de torneo.

Tabla 4: Descripción Textual del CUN "Solicitar Información sobre Torneos Anteriores"

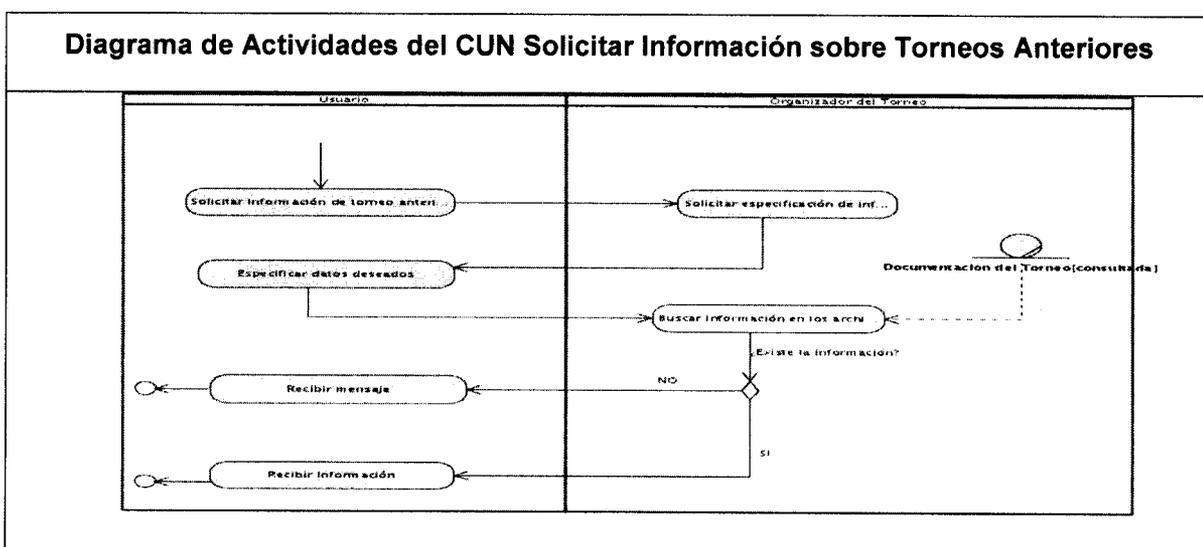


Figura 3: Diagrama de Actividades del CUN Solicitar Información sobre Torneos Anteriores



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

2.15 Diagrama de clases del modelo de objetos

Un modelo de objetos describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso del negocio. En sí el modelo de objeto describe cómo los trabajadores del negocio utilizan las entidades del negocio y las unidades de trabajo.

Seguidamente se muestra el modelo de objeto del negocio correspondiente:

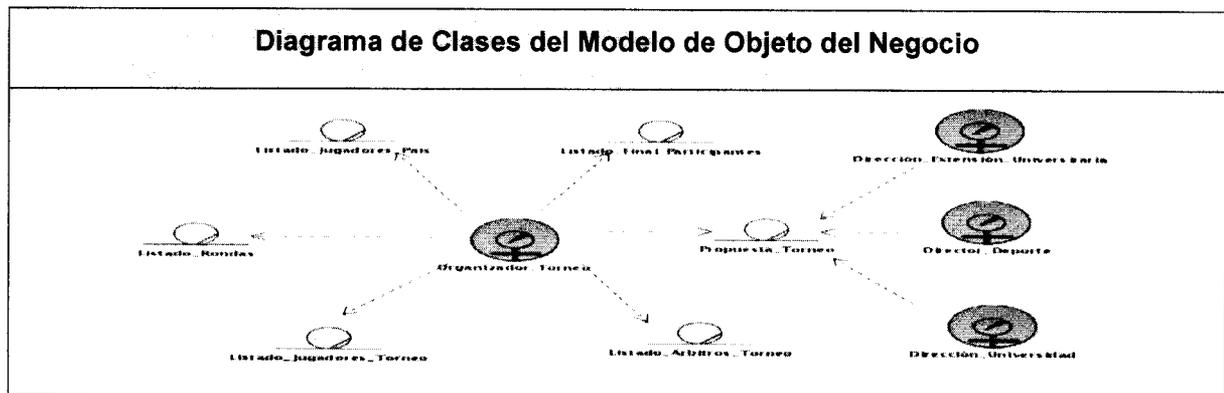


Figura 4: Diagrama de Clases del Modelo de Objetos del Negocio

2.16 Definición de los requisitos funcionales

Los requerimientos o requisitos funcionales, son capacidades o condiciones que el sistema debe cumplir, o sea, son objetivos o metas que deben estar presentes en el sistema para que el usuario o cliente se sienta satisfecho, y además representan un punto de partida para identificar qué debe hacer el sistema.

A continuación se presentan las condiciones que debe cumplir el sistema propuesto:

RF 1 Autenticar Usuario.

1.1 Autenticar Usuario.

1.2 Registrar Usuario.

1.3 Cerrar Sección.

RF 2 Gestionar Torneo.

2.1 Adicionar Torneo.

2.2.1 Entrar Datos del Torneo.

2.2.2 Verificar si se puede o no organizar el torneo.

2.2.3 Confeccionar los Objetivos del Torneo.



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

- 2.2.4 Entrar las Bases del Torneo.
- 2.2.5 Entrar la Historia del Torneo.
- 2.2.6 Entrar la Biografía de la persona que da nombre al torneo.
- 2.2.7 Entrar los premios que ofrecerá el torneo.
- 2.2.8 Verificar la cantidad de jugadores invitados.

2.3 Modificar Torneo.

- 2.3.1 Actualizar los datos del torneo.

2.4 Eliminar Torneo.

- 2.4.1 Eliminar la creación del torneo.

2.5 Enviar correo de invitación a jugadores y árbitros.

2.6 Enviar correo de cancelación de invitación a jugadores y árbitros.

RF 3 Gestionar Noticias.

3.1 Adicionar Noticias.

- 3.1.1 Agregar noticias a las rondas.
- 3.1.2 Mostrar el listado de las noticias adicionadas.

3.2 Modificar Noticia.

- 3.2.1 Modificar la noticia de las rondas.

3.3 Eliminar Noticia.

- 3.3.1 Eliminar la noticia de una ronda.

RF 4 Gestionar Ronda.

4.1 Adicionar Ronda.

4.2 Modificar Ronda.

4.3 Eliminar Ronda.

4.4 Listar las rondas existentes en el torneo.

RF 5 Gestionar Partidas.

5.1 Adicionar Partidas.



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

5.1.1 Llenar Plantilla de las partidas.

5.2 Modificar Partidas.

5.2.1 Cambiar datos de las partidas.

5.3 Listar las partidas de cada ronda.

5.3.1 Listar las partidas del torneo.

RF 6 Gestionar Evento.

6.1 Adicionar Eventos.

6.1.1 Permitir crear un cronograma de eventos.

6.2 Modificar Eventos.

6.2.1 Cambiar datos de los eventos del cronograma.

6.3 Eliminar Eventos.

6.4 Consultar Listado de eventos.

6.5 Listar los eventos existentes.

RF 7 Consultar opiniones.

7.1 Ver las opiniones de los participantes del torneo.

7.2 Ver opiniones de torneos pasados.

7.3 Eliminar opiniones.

RF 8 Gestionar Listado de Jugadores del Torneo.

8.1 Crear listado de jugadores del torneo.

8.2 Buscar jugadores por distintos criterios de búsqueda.

8.2.1 Buscar por nombre.

8.2.2 Buscar por elo.

8.2.3 Buscar por provincia.

8.3 Invitar jugadores al torneo.

8.4 Eliminar jugadores del listado del torneo.

8.5 Listar jugadores del torneo.



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

8.6 Listar jugadores del torneo con estado pendiente.

8.7 Listar jugadores del torneo con estado rechazado.

8.8 Listar jugadores del torneo con estado aceptado.

RF 9 Gestionar Listado de Árbitros del Torneo.

9.1 Crear listado de árbitros del torneo.

9.2 Buscar árbitros por distintos criterios de búsqueda.

9.2.1 Buscar por título.

9.2.2 Buscar por nombre.

9.2.3 Buscar por provincia.

9.3 Invitar árbitros al torneo.

9.4 Eliminar árbitros del listado del torneo.

9.5 Listar árbitros del torneo.

9.6 Listar árbitros del torneo con estado pendiente.

9.7 Listar árbitros del torneo con estado rechazado.

9.8 Listar árbitros del torneo con estado aceptado.

RF 10 Registrar Participantes.

10.1 Aceptar la invitación al torneo.

10.2 Rechazar la invitación al torneo.

10.3 Cambiar el estado de los jugadores y árbitros.

RF 11 Configurar Mensajes de Correo

11.1 Modificar los mensajes para el envío de correo a los invitados al torneo.

11.2 Mostrar los mensajes existentes.

11.3 Ofrecer tipos de mensajes.

RF 12 Consultar Torneo.

12.1 Consultar Listado de Jugadores del Torneo.

12.2 Consultar Listado de Árbitros del Torneo.



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

12.3 Consultar Resultados de las Ronda.

12.4 Consultar Bases del Torneo.

12.5 Consultar Objetivos del Torneo.

12.6 Consultar Historia del Torneo

12.7 Consultar Partidas.

RF 13 Ver Datos de los Participantes.

RF 14 Buscar Torneos Anteriores.

14.1 Buscar torneos por criterios.

14.1.1 Buscar por nombre.

14.1.2 Buscar año.

14.1.3 Buscar por provincia.

RF 15 Ver datos de los Torneos Anteriores.

RF 16 Opinar sobre el Torneo.

16.1 Entrar su opinión sobre el torneo desarrollado.

RF 17 Confirmar Participación.

17.1 Aceptar la Invitación al Torneo.

17.2 Rechazar la Invitación al Torneo.

2.17 Definición de los requisitos no funcionales

Los requerimientos o requisitos no funcionales son propiedades o cualidades que el producto debe tener, las cuales hacen que un producto sea fácil de usar, seguro, o interactivo. Estos requisitos se clasifican en categorías:

Software:

Para la instalación del servidor:

- ✓ Servidor Apache versión 2 o superior.
- ✓ PHP 4 o superior.
- ✓ Servidor de bases de datos MySQL versión 4 o superior.
- ✓ Sistemas Operativos GNU/Linux (recomendado) o Microsoft Windows.



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

Para la interpretación por clientes (Navegadores Web):

- ✓ Internet Explorer v6, Internet Explorer v7 (recomendado) o superior.
- ✓ Mozilla Firefox 2.* (recomendado) o superior (recomendado).
- ✓ Cualquier navegador libre (excepto Opera).

Hardware:

- ✓ Mínimo Requerido: RAM 128 Mb, Disco duro 512 MB.
- ✓ Mínimo Recomendado: RAM 256 Mb, Disco duro 1GB.
- ✓ Conexión de red por MODEM o fibra óptica.

Apariencia o Interfaz Externa:

- ✓ El diseño de la interfaz debe ser lo más sencillo y claro posible y contar con aspectos que definan claramente cada una de sus funcionalidades.
- ✓ La navegabilidad debe ser fácil y sencilla.

Usabilidad:

- ✓ El sistema podrá ser utilizado por cualquier usuario que acceda al mismo.
- ✓ Proporcionará rápido acceso de búsqueda de la información, en tiempos cortos.
- ✓ Contará con zonas de acceso limitado, a la cual solo accederán los usuarios autorizados con roles determinados.
- ✓ Fácil gestión de la información.

Rendimiento:

- ✓ El sistema deberá ser capaz de gestionar toda la información y dar respuestas rápidas, casi inmediatas, a las solicitudes realizadas.
- ✓ Debe ser eficiente en la gestión de las solicitudes que se realicen, para que el usuario obtenga, sin mucha navegación, el resultado deseado.
- ✓ Debe estar disponible las 24 horas del día.

Diseño e implementación:

- ✓ El sistema es recomendado que funcione sobre una plataforma de sistema operativo GNU/Linux a modo texto. Los lenguajes necesarios para trabajar son HTML, JavaScript y PHP4 o superior. Como gestor de Base Datos se utilizará MySQL, se



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

desarrollará sobre un sistema GNU/Linux, utilizándose herramientas como el Zend Studio, phpmyadmin, Visual Paradigm.

Portabilidad:

- ✓ Es un sistema multiplataforma. Podrá ser montado sobre los sistemas operativos: Unix, Linux, Windows, entre otros.

Seguridad:

- ✓ Verificar la autenticidad y niveles de privilegio del usuario antes de hacer cualquier operación dentro del sistema.
- ✓ Verificar que las funcionalidades sean mostradas en correspondencia con los niveles de privilegios.
- ✓ Protección contra operaciones no autorizadas que en algún momento puedan afectar la integridad de los datos.

Soporte:

- ✓ El sistema debe ser de fácil instalación y configuración, con vista a poder darle un mantenimiento relativamente fácil y sencillo en caso de fallos.

Legales:

- ✓ La plataforma está basada en la licencia GNU/GPL.
- ✓ El sistema es reconocido y autorizado por entidades superiores tales como la directiva de la UCI.

Ayuda y documentación en línea

- ✓ Ofrecerá una documentación de ayuda para el uso del sistema por los usuarios, la cual estará disponible desde cualquier parte del mismo para aclarar cualquier duda que el usuario presente a la hora de manejar y hacer uso de la aplicación web.

Confiabilidad:

- ✓ Realiza control de errores, recuperación ante fallos.

A continuación se presentan las cualidades que debe tener el sistema propuesto:



Módulo para la Gestión de Información de Torneos de Ajedrez. Características del Sistema

2.18 Actores del Sistema

Lo que se modela como actor del sistema es el rol que juega cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa directamente con el sistema intercambiando información y beneficiándose de sus resultados.

Actor	Descripción
Usuario	-Es todo aquel que acceda al sitio para hacer uso de él.
Organizador del Torneo	-Es el responsable de organizar el evento e invitar a los participantes. -Es el administrador del sitio, quien se encarga de la creación y administración del mismo.
WebMaster	-Es el responsable de las actualizaciones de la información del sitio.
Jugador del Torneo	-Protagonista del evento, quien juega el torneo.
Servidor de Correo	-Es un sistema que interactúa con el sistema propuesto para hacer llegar los correos correspondientes a sus destinatarios.
Árbitro del Torneo	-Es el responsable de velar por el cumplimiento de las reglas del torneo.

Tabla 5: Actores del Sistema

2.19 Casos de Uso del Sistema

Los Casos de Uso del Sistema son artefactos narrativos a través de los cuales se describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. En otras palabras, los Casos de Uso del Sistema representan una serie de acciones que realiza el sistema dado una determinada acción de un actor, ofreciéndole a este un resultado observable y representando los requisitos funcionales del sistema.

2.20 Diagrama de casos de uso del sistema

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



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

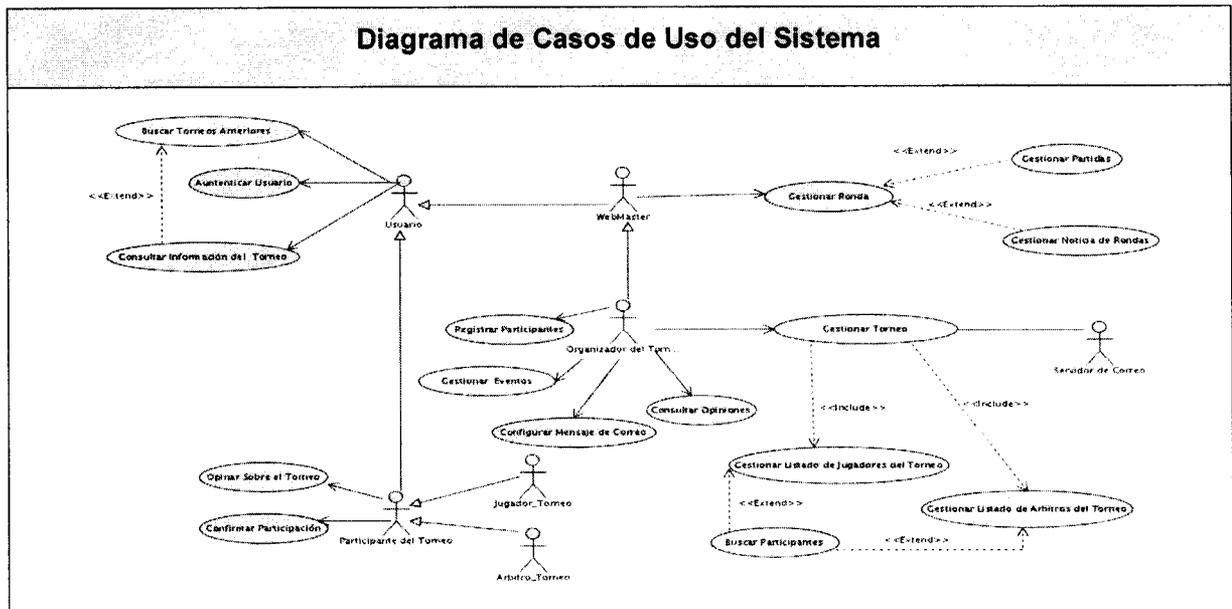


Figura 5: Diagrama de Casos de Uso del Sistema

2.21 Descripción de los casos de uso del sistema

Es necesario comentar el proceso de acción y reacción del sistema, lo cual se realiza a través de la descripción textual de los Casos de Uso del Sistema.

2.21.1 Descripción Textual del Caso de Uso del Sistema “Autenticar Usuario”

Caso de Uso:	Autenticar Usuario
Actores:	Usuario(inicia)
Resumen:	El CUS se inicia cuando el usuario desea acceder a la aplicación para hacer uso de ella, se autentica o simplemente se registra, el sistema otorga los permisos correspondientes para el acceso a la información y termina el CUS.
Referencia:	RF1
CU asociados:	
Precondiciones	El sistema visualiza el sitio.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
Escenario 1: Autenticar Usuario	
1. El usuario accede al sistema	1.1. El sistema muestra la página principal del sitio con un formulario para la entrada de los datos



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

	necesarios para la identificación del usuario y otorgar los permisos correspondientes a su rol (usuario y contraseña).
2. El usuario entra los datos solicitados por el sistema y selecciona la opción "Entrar".	2.1. El sistema comprueba los datos entrados por el usuario para la identificación del mismo.
	2-2-Si los datos son válidos y el usuario ya existe en la base de datos, el sistema muestra la información del sitio a la que el tipo de usuario autenticado tiene acceso y termina el CUS.
Flujo Alternativo	
2a-Datos incorrectos.	2a.1-Si los datos no son válidos el sistema muestra un mensaje de error, señala donde está el error e indica al usuario que retorne a la acción 2.
Escenario 2: Registrar Usuario	
1. El usuario accede al sistema	1.1. El sistema muestra un formulario para la entrada de los datos necesarios para que el usuario se registre (usuario, contraseña y confirmación de contraseña).
2. El usuario entra los datos solicitados por el sistema y selecciona la opción "Entrar".	2.1. El sistema verifica si los datos introducidos son válidos y si el usuario no está en la Base de Datos.
	2.2-Si los datos del usuario son válidos y este no está registrado en la base de datos, el sistema lo adiciona a la base de datos y muestra el formulario para que el usuario acceda al sistema (se autentique) y termina el CUS.
Flujo Alternativo	
2a-Datos incorrectos o incompletos	2a.1-El sistema verifica que hay error en los datos entrados por el usuario. Indica donde está el error y que regrese a la acción 2 del flujo normal.
2b-Usuario existente	2b.1-El sistema verifica que el usuario existe en la Base de Datos, emite un mensaje de error y termina el CUS.
Escenario 3: Cerrar Sección	
1-El Usuario selecciona la opción "Cerrar Sección".	1.1-La aplicación se cancela y termina el CUS.
Poscondiciones :	-Usuario autenticado.



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

	<ul style="list-style-type: none"> -Usuario registrado en la Base de Datos. -El sistema le da acceso al usuario a la información que le es permitida según su rol. -El sistema cierra la sesión de trabajo del usuario autenticado.
Prioridad:	Crítica

Tabla 6: Descripción Textual del Caso de Uso del Sistema “Autenticar Usuario”

2.21.2 Descripción Textual del Caso de Uso del Sistema “Gestionar Eventos”

Caso de Uso:	Gestionar Eventos
Actores:	Organizador del Torneo(inicia)
Resumen:	El CUS inicia cuando el Organizador del Torneo accede al sistema para confeccionar el cronograma de actividades, modificar o eliminar alguna actividad, el sistema ejecuta la acción realizada por el Organizador y termina el CUS.
Referencia:	RF6
CU asociados:	
Precondiciones :	-Usuario autenticado como Organizador del Torneo, para que el sistema le asigne los permisos correspondientes.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El Organizador del Torneo ejecuta el sistema y selecciona la opción “Gestionar Eventos”.	1.1-El sistema muestra un listado con los eventos que existen y la opción de adicionar otros, modificar o eliminar alguno.
Escenario 1: Adicionar Eventos	
1-El Organizador del Torneo selecciona la opción “Adicionar Evento”.	1.1-El sistema muestra un formulario con los aspectos a llenar para la adición del evento. (Nombre de la Actividad, Fecha de Inicio, Fecha de Fin).
2- El Organizador del Torneo introduce los aspectos requeridos por el sistema para adicionar de evento y selecciona la opción “Adicionar”.	2.1-El sistema verifica los datos introducidos por el Organizador.
	2.2- Si los datos son correctos el sistema adiciona el evento a la Base de Datos, visualiza el listado de los



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

	eventos actualizado y finaliza el CUS
Flujo Alterno	
2a-Datos incorrectos	2a.1-Si los datos son incorrectos el sistema muestra un mensaje de error, indica donde está el error, y que retorne a la acción 2.
Escenario 2: Modificar Eventos	
1- El Organizador del Torneo escoge del listado el evento a modificar y selecciona la opción "Modificar".	1.1- El sistema localiza los datos del evento especificado (nombre, fecha inicio, fecha fin) y los muestra listos para ser modificados.
2- El Organizador del Torneo realiza los cambios en el evento y selecciona la opción "Guardar".	2.1-El sistema verifica que los cambios realizados sean válidos.
	2.2-Si los datos son correctos el sistema guarda los cambios realizados por el Organizador en el evento seleccionado y termina el CUS.
Flujo Alterno	
1a- El Organizador del Torneo selecciona "Cancelar".	1a.1-El sistema no ejecuta ninguna acción y termina el CUS.
2a-Datos incorrectos	2a.1-Si los datos son incorrectos el sistema muestra un mensaje de error, indica donde está el error, y que retorne a la acción 2.
Escenario 3: Eliminar Eventos	
1- El Organizador del Torneo selecciona del listado un evento y selecciona la opción "Eliminar".	1.1-El sistema muestra un mensaje de advertencia antes de ejecutar la acción.
2- El Organizador del Torneo confirma si desea o no eliminar el evento seleccionado.	2.1-Si el Organizador acepta el sistema elimina el evento, actualiza los eventos existentes, muestra el listado y termina el CUS.
Flujo Alterno	
1a- El Organizador del Torneo selecciona "Cancelar".	1a.1-El sistema no ejecuta ninguna acción y termina el CUS.
2a-Eliminación cancelada.	2a.1-Si el Organizador cancela, el sistema no ejecuta ninguna acción y termina el CUS.
Poscondiciones :	-Evento adicionada



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

	- Evento modificada - Evento eliminada
Prioridad:	Crítica

Tabla 7: Descripción Textual del Caso de Uso del Sistema “Gestionar Eventos”

2.21.3 Descripción Textual del Caso de Uso del Sistema “Gestionar Torneo”

Caso de Uso:	Gestionar Torneo
Actores:	Organizador del Torneo(inicia)
Resumen:	El CUS se inicia cuando el Organizador del Torneo accede al sistema para crear un nuevo torneo, modificar el ya existente, procesar el torneo, ejecutar el torneo, terminar el torneo o eliminar el mismo, el sistema ejecuta la acción realizada por el Organizador y finaliza el CUS.
Referencia:	RF2
CU asociados:	- Gestionar Listado de Jugadores del Torneo(CUS include) - Gestionar Listado de Árbitros del Torneo(CUS include)
Precondiciones:	-El usuario debe estar autenticado como Organizador del Torneo para que le sean otorgados los permisos de administración.
Flujo Normal de Eventos	
1-El Organizador del Torneo accede al sistema.	1.1-El sistema muestra la opción: -Crear Torneo.
Escenario 1: Crear Torneo	
Acción del Actor	Respuesta del Sistema
1-El Organizador del Torneo selecciona la opción “Crear Torneo”.	1.1-El sistema muestra un formulario para que el organizador entre los datos del torneo(Nombre del torneo, Fecha de inicio, Fecha de Finalización, Provincia, Nombre del Organizador).
2- El Organizador del Torneo introduce los datos y selecciona la opción “Adicionar”	2.1-Con todos los datos introducidos el sistema consulta en la Base de Datos el calendario de eventos.
	2.2-El sistema comprueba que se puede realizar el torneo, que no hay eventos que obstaculicen el desarrollo del mismo. El sistema crea el torneo, cambia el estado del torneo a “creado” y muestra las



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

	<p>opciones relacionada con la creación del torneo.</p> <ul style="list-style-type: none"> -Completar Torneo. - Gestionar Listado de Jugadores del Torneo. - Gestionar Listado de Árbitros del Torneo. -Modificar Datos del Torneo. -Eliminar Torneo. -Procesar Torneo. -Ejecutar Torneo. -Terminar Torneo
3-El Organizador selecciona la opción "Completar Torneo".	3.1-El sistema muestra un formulario para continuar la entrada de los datos del torneo (Fecha de Fin de la Inscripción, Elo máximo, Elo mínimo, Objetivos del torneo, Bases, Historia, Biografía).
4-El Organizador del Torneo entra los datos y selecciona la opción "Aceptar".	4.1-El sistema comprueba que los datos sean válidos y actualiza los datos del torneo.
5-El Organizador del Torneo selecciona la opción "Gestionar Jugadores del Torneo".	5.1-El sistema ejecuta el CUS "Gestionar Listado de Jugadores del Torneo".
6- El Organizador del Torneo selecciona la opción "Gestionar Árbitros del Torneo".	6.1- El sistema ejecuta el CUS "Gestionar Listado de Árbitros del Torneo".
7-El Organizador del Torneo selecciona la opción "Salir".	7.1-El sistema termina el CUS.
Flujo Alterno	
2a-El Organizador del Torneo selecciona la opción "Cancelar".	2a.1-El sistema cancela la creación del torneo y finaliza el CUS.
2b-El Organizador del Torneo selecciona la opción "Adicionar" pero faltan datos.	2b.1-El sistema verifica que hay campos vacíos, emite un mensaje "Faltan datos por entrar" e indica que regrese a la acción 2 del flujo normal.
2c: Existencia de eventos importantes que obstaculizan el desarrollo del torneo.	<p>2c.1-Si el sistema comprueba que la realización del torneo no se puede efectuar porque la fecha coincide con eventos importantes, emite un mensaje.</p> <p>-"No se puede realizar el torneo porque se está realizando el evento "" el cual es de gran importancia para la Universidad".</p>
3a- El Organizador del Torneo selecciona la opción "Salir".	3a.1-El sistema termina el CUS.



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

3b-El Organizador del Torneo selecciona la opción "Gestionar Jugadores del Torneo".	3b.1-El sistema ejecuta el CUS "Gestionar Listado de Jugadores del Torneo".
3c- El Organizador del Torneo selecciona la opción "Gestionar Árbitros del Torneo".	3c.1- El sistema ejecuta el CUS "Gestionar Listado de Árbitros del Torneo".
5a-El Organizador del Torneo selecciona la opción "Salir".	5a.1-El sistema termina el CUS.
5b- El Organizador del Torneo selecciona la opción "Gestionar Árbitros del Torneo".	5b.1- El sistema ejecuta el CUS "Gestionar Listado de Árbitros del Torneo".
6a-El Organizador del Torneo selecciona la opción "Salir".	6a.1-El sistema termina el CUS.
Escenario 2: Modificar Datos del Torneo	
1-El Organizador del Torneo selecciona la opción "Modificar Torneo".	1.1-El sistema localiza los datos del torneo que pueden ser modificados y los muestra listos para modificar: -Objetivos. -Historia del Torneo. -Biografía de la persona que da nombre al torneo. -Premios. -Fecha de Fin de Inscripción.
2-El Organizador del Torneo introduce los datos y selecciona la opción "Guardar"	2.1-El sistema realiza los cambios, actualiza los datos y termina el CUS.
Escenario 3: Eliminar Torneo	
1-El Organizador del Torneo selecciona la opción "Eliminar Torneo".	1.1-El sistema emite un mensaje de advertencia antes de ejecutar la acción.
2-El Organizador del Torneo confirma si desea o no eliminar el torneo.	2.1-Si acepta el sistema elimina el torneo de la base de datos y termina el CUS.
Flujo Alternativo	
2a-Eliminación cancelada.	2a.1-Si cancela el sistema no ejecuta nada y termina el CUS.
Escenario 4: Terminar Torneo	
1-El Organizador del Torneo selecciona la opción "Terminar Torneo".	1.1-El sistema cambia el estado del torneo de "ejecutado" a "terminado, emite un mensaje "El torneo está terminado" y termina el CUS



Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

Escenario 5: Procesar Torneo	
1-El Organizador del Torneo selecciona la opción "Procesar Torneo".	1.1-El sistema verifica los aspectos necesarios para la ejecución del torneo.
	1.2-El sistema cambia el estado del torneo de "creado" a "procesado", localiza la dirección de correo electrónico del listado de invitados (jugadores y árbitros), localiza el mensaje de invitación, realiza el envío de e-mail y termina el CUS.
Flujo Alternativo	
1a-Faltan datos necesarios por gestionar	1a.1-El sistema emite un mensaje "el torneo no cumple con los requisitos para la ejecución", indica donde está el error y termina el CUS.
Escenario 6: Ejecutar Torneo	
1-El Organizador del Torneo selecciona la opción "Ejecutar Torneo".	1.1-El sistema cambia el estado del torneo de "procesado" a "ejecutado", los jugadores que no confirmaron su participación los pone en estado "sin estado", le cambia el rol de "jugador" a "usuario" y muestra las opciones correspondientes a la entrada de datos y resultados del torneo y termina el CUS.
Poscondiciones	<ul style="list-style-type: none"> -Torneo creado. -Torneo eliminado -Torneo terminado -Torneo ejecutado -Torneo modificado -Torneo procesado
Prioridad:	Crítica

Tabla 8: Descripción Textual del Caso de Uso del Sistema "Gestionar Torneo"

Anteriormente solo se expusieron las descripciones de algunos de los casos de uso críticos del sistema, las restantes descripciones se muestran en el Anexo 2.

2.22 Conclusiones

En el transcurso de este capítulo se identificaron los actores y trabajadores del negocio así como las entidades y procesos con los que interactúan los mismos, además se analizó el flujo actual de los procesos del negocio y los procesos a automatizar, los cuales sirvieron de apoyo para resaltar las características con las que contará el sistema propuesto.



Capítulo2

Módulo para la Gestión de Información de Torneos de Ajedrez Características del Sistema

Por otra parte se identificaron los requisitos funcionales y no funcionales que deberá cumplir y tener el sistema correspondiente, mediante los mismos se definieron los procesos del sistema, y se expuso la correspondiente descripción textual de cada uno de ellos, mediante la cual se ofrece una panorámica general de lo que se pretende que el sistema realice.

Una vez detallados todos estos aspectos se puede pasar a describir cómo implementar el sistema para darle solución a los problemas existentes.



Capítulo 3

Análisis y Diseño del Sistema

3.1 Introducción

En el presente capítulo se abordará todo lo referente al análisis y diseño de la aplicación propuesta. Teniendo en cuenta que el análisis consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace el sistema, de manera que su interés se centra solamente en los requisitos funcionales; y que el diseño representa un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en pocas palabras, el diseño se preocupa en CÓMO cumple el sistema sus objetivos y debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

3.2 Diagrama de clases del análisis

Se realiza el diagrama de clases del análisis para conseguir una comprensión más precisa de los requisitos, refinarlos y estructurarlos y de esta forma proporcionar una visión general del sistema. En el análisis las clases se clasifican en:

- ✓ **Clases Entidad o Sistema:** Las clases de entidad se utilizan para modelar información que posee larga vida y que es a menudo persistente. Modelan la información y el comportamiento asociado de algún fenómeno o concepto.
- ✓ **Clases Interfaz o Frontera:** Las clases de interfaz son empleadas para modelar la interacción entre el sistema y sus actores, la cual a menudo incluye recibir información y peticiones de los usuarios y sistemas externos.
- ✓ **Clases Controladora o Software:** Las clases de control representan la coordinación, secuencia, transacciones, y control de otros objetos y se usan frecuentemente para encapsular el control de un caso de uso en concreto.

En el sistema propuesto se cuenta con 16 casos de uso por lo que exponer todos los diagramas del análisis haría sumamente extenso el informe, por ello a continuación se exponen solamente los diagramas de clases del análisis que responden a algunos de los casos de uso críticos del sistema. Los diagramas del análisis de los restantes casos de uso se presentan en el Anexo 3.

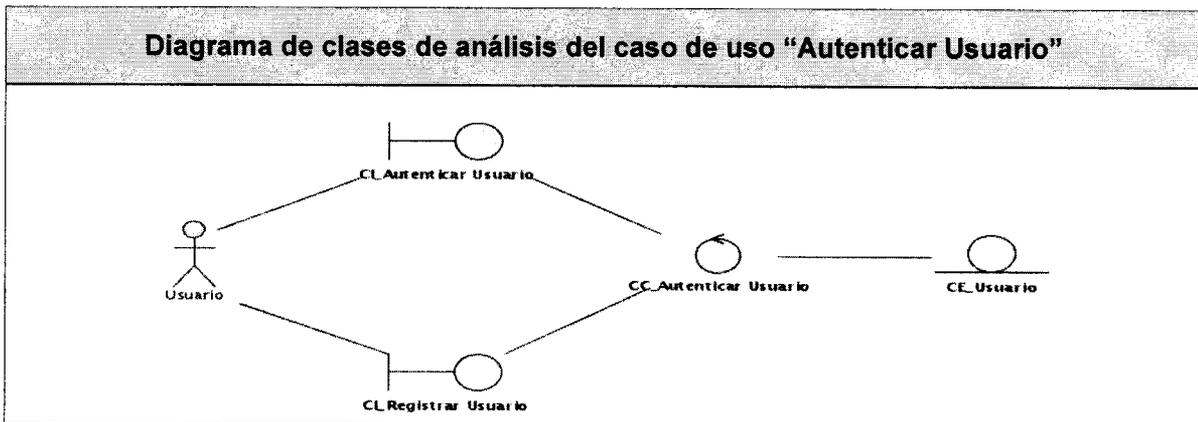


Figura 6: Diagrama de clases de análisis del caso de uso "Autenticar Usuario"

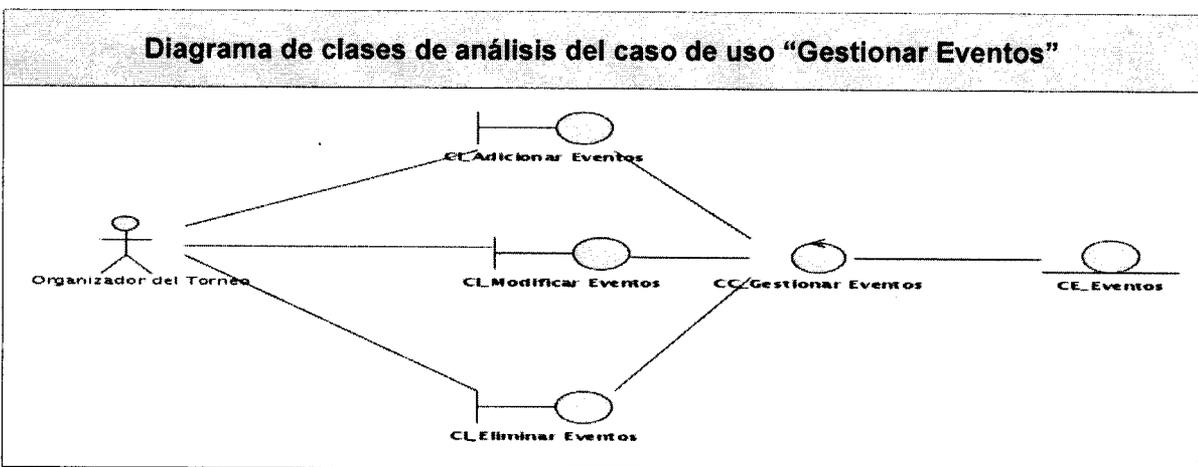


Figura 7: Diagrama de clases de análisis del caso de uso "Gestionar Eventos"

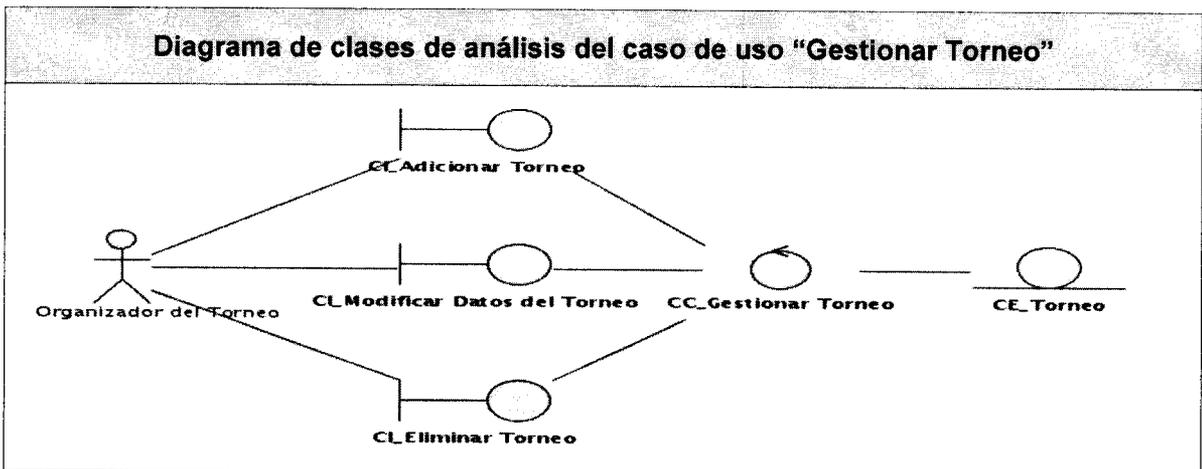


Figura 8: Diagrama de clases de análisis del caso de uso "Gestionar Torneo"

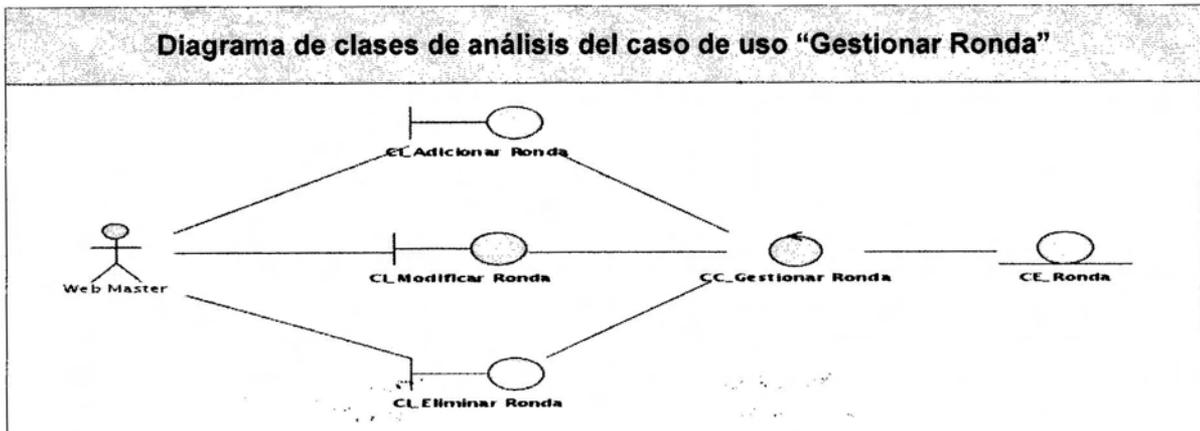


Figura 9: Diagrama de clases de análisis del caso de uso "Gestionar Rondas"

3.3 Diagrama de Clases del Diseño

En el diseño se modela el sistema y se encuentra su forma para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen.

En un diagrama de clases se presentan las clases del sistema con sus relaciones estructurales y de herencia. En el caso específico de las aplicaciones web, el diagrama de clases representa las relaciones que existen entre las páginas, donde cada página lógica puede ser representada como una clase.

En el diagrama de clases del diseño las clases se identifican en:

- ✓ **Página Cliente o Interfaz:** Una instancia de Página Cliente es una página web, con formato HTML. Es una mezcla de datos, presentación y lógica, los cuales son interpretados por el navegador.
- ✓ **Formulario:** Una clase Formulario representa una colección de elementos de entrada que son parte de una página cliente.
- ✓ **Página Servidora o Controladora:** La clase Página Servidora representa la página web que tiene código que se ejecuta en el servidor.

Seguidamente solo se exponen los diagramas de clases del diseño de algunos de los casos de uso críticos del sistema. Los diagramas del diseño de los restantes casos de uso se exponen en el Anexo 4.

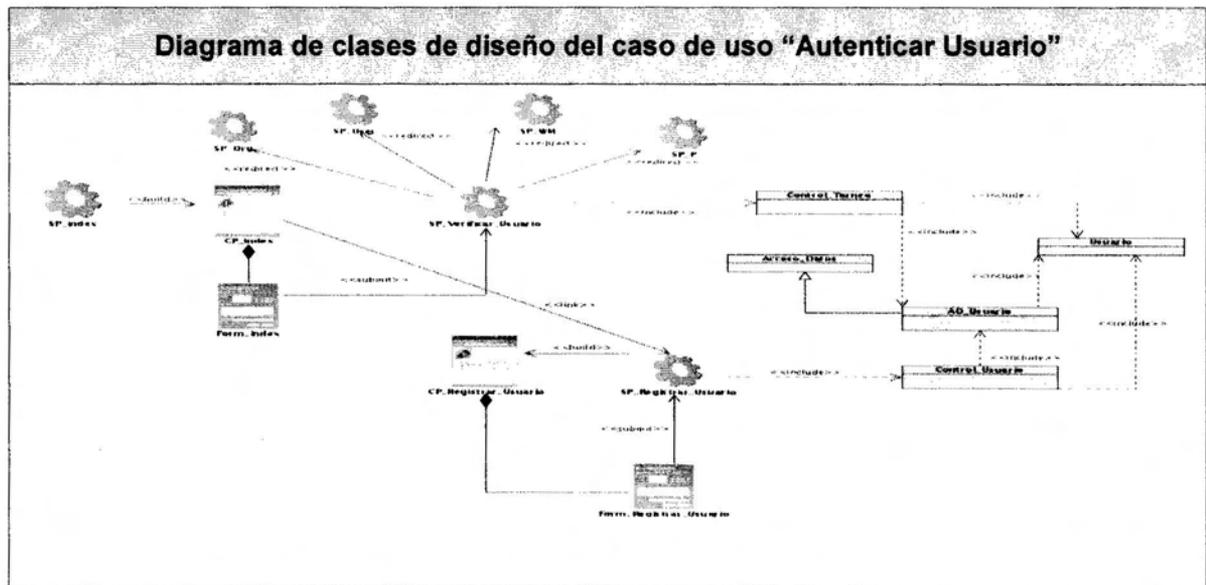


Figura 10: Diagrama de clases Web para el caso de uso "Autenticar usuario"

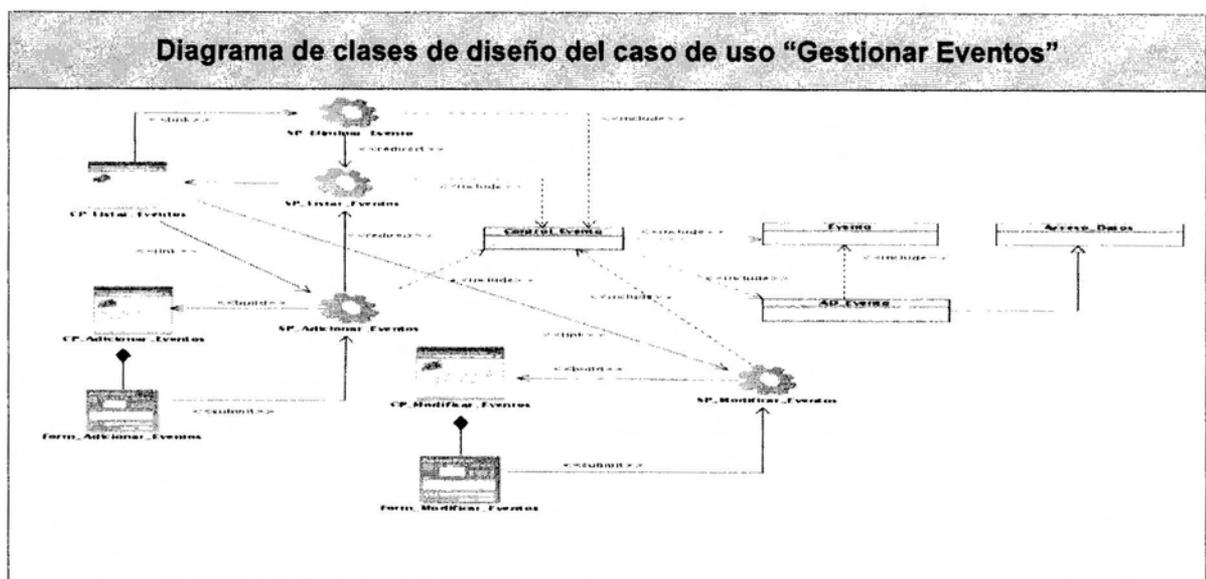


Figura 11: Diagrama de clases Web para el caso de uso "Gestionar Eventos"

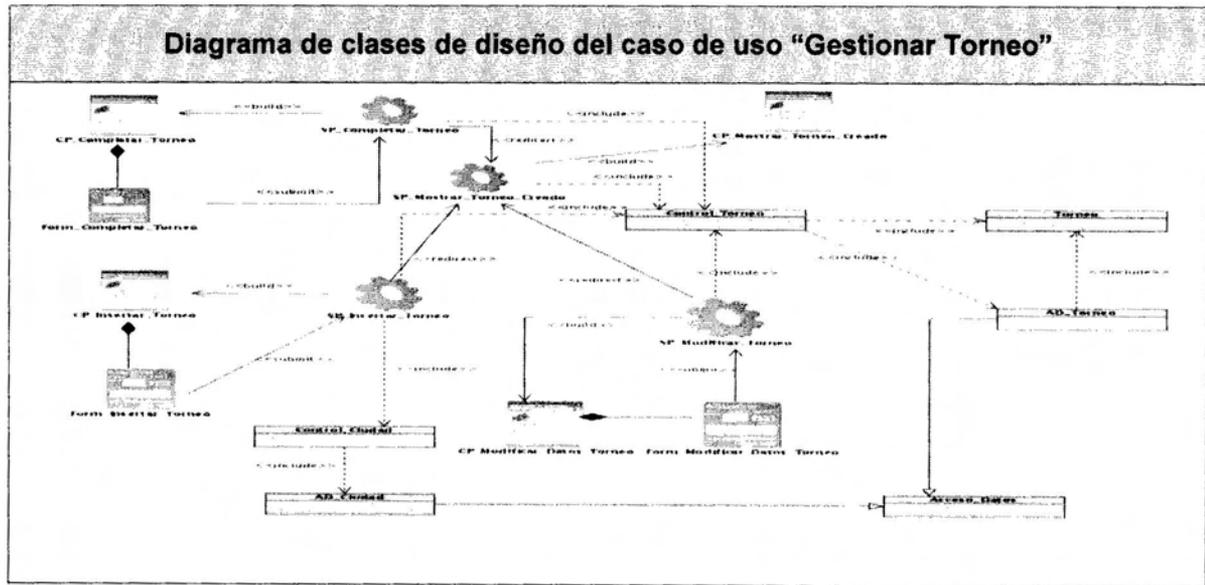


Figura 12: Diagrama de clases Web para el caso de uso "Gestionar Torneo"

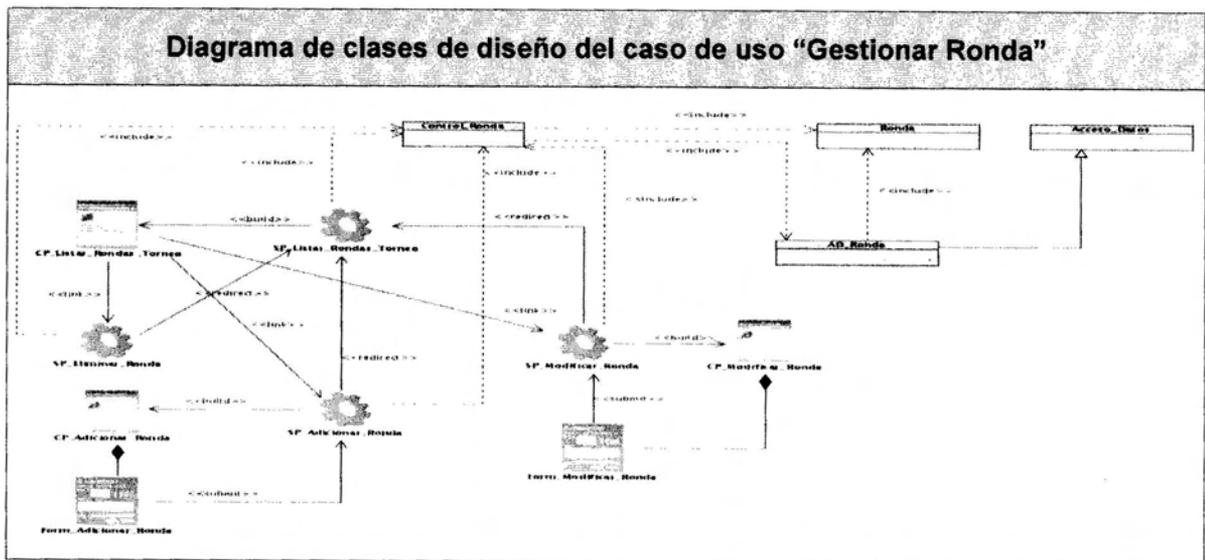


Figura 13: Diagrama de clases Web para el caso de uso "Gestionar Ronda"

3.4 Diagrama de Secuencia del Diseño:

Los diagramas de secuencia se emplean para modelar los aspectos dinámicos de un sistema y destacan la ordenación temporal de los mensajes. A continuación se exponen los diagramas de secuencia de algunos de los casos de uso más importantes para el sistema, los diagramas de los restantes casos de uso se presentan en el Anexo 5.

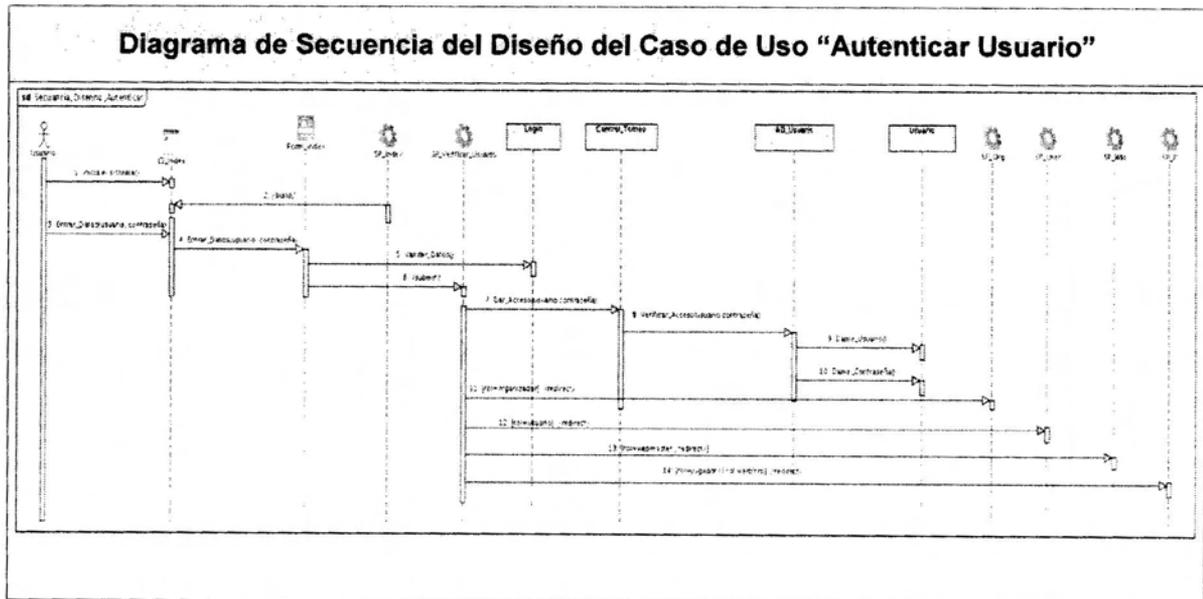


Figura 14: Diagrama de Secuencia del Diseño del Caso de Uso "Autenticar Usuario"

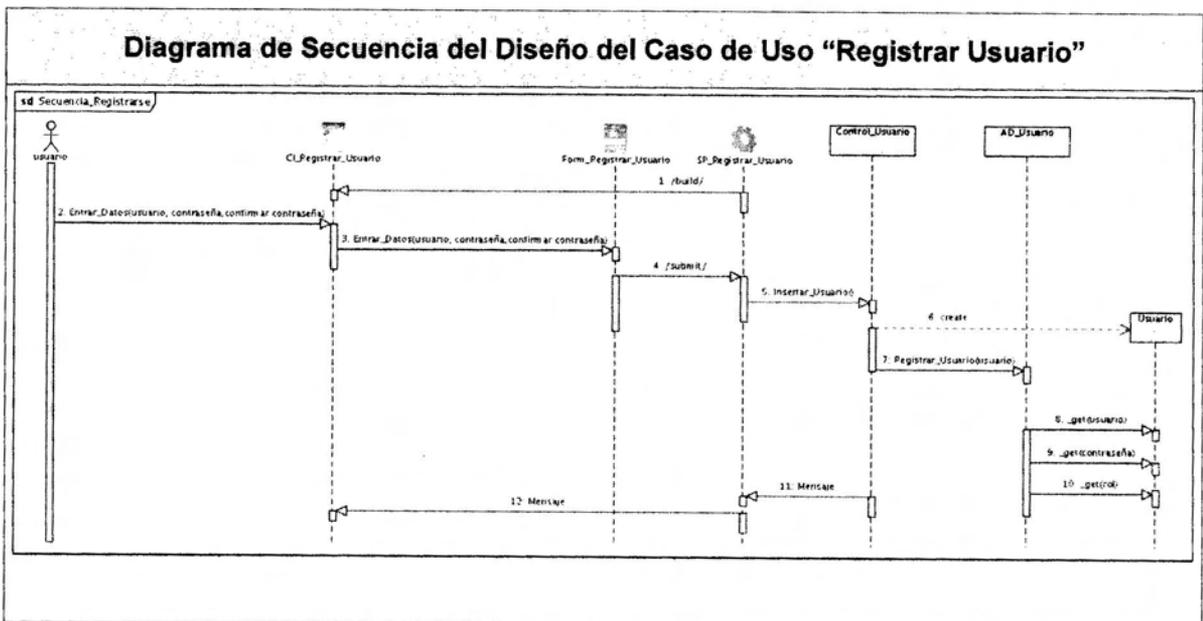


Figura 15: Diagrama de Secuencia del Diseño del Caso de Uso "Registrar Usuario"

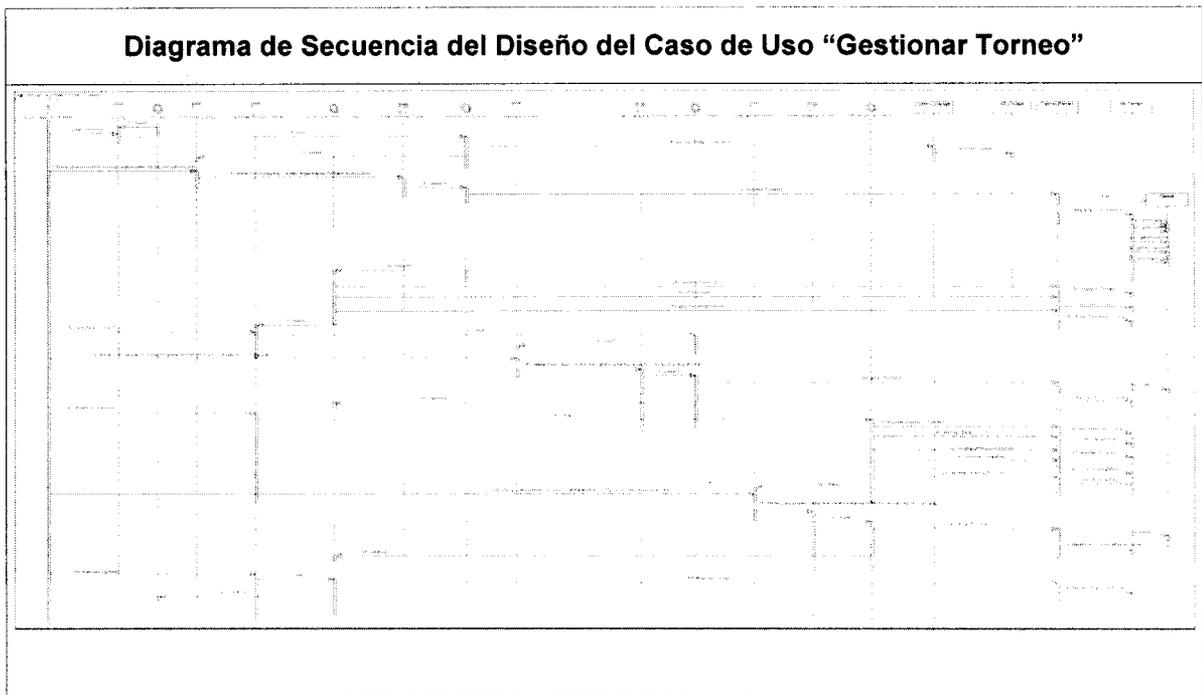


Figura 16: Diagrama de Secuencia del Diseño del Caso de Uso "Gestionar Torneo"

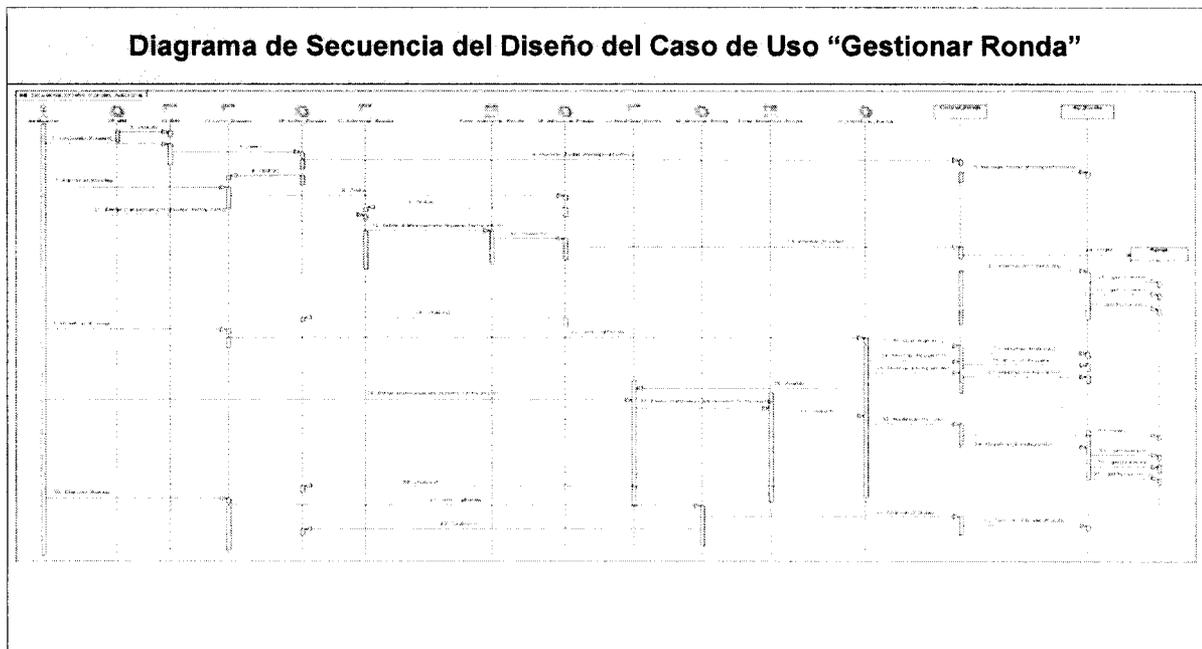


Figura 17: Diagrama de Secuencia del Diseño del Caso de Uso "Gestionar Ronda"



3.5 Diagrama de clases del subsistema de “Acceso a Datos”

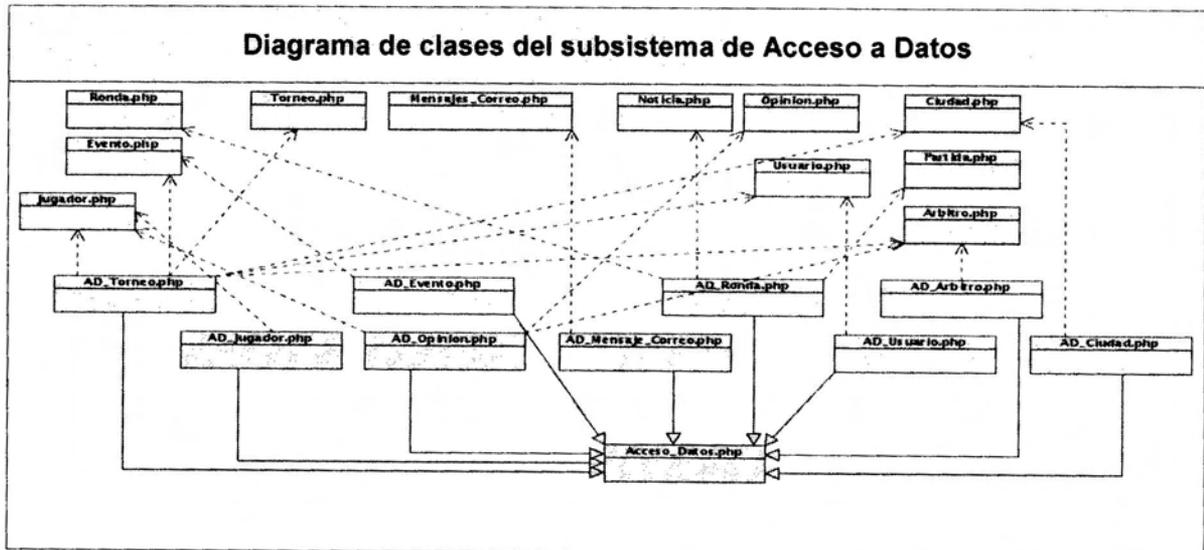


Figura 18: Diagrama de clases del subsistema de “Acceso a Datos”

3.6 Descripción textual de las clases de diseño web

La descripción textual de cada una de las clases que se modelan en el diagrama de clases del diseño posibilita comprender fácilmente cada una de las funcionalidades de dichas clases, ya que en la descripción se especifican claramente cuales son sus responsabilidades.

Como resultado de los diagramas de clases del diseño se obtuvieron varias clases de las cuales solo se exponen seguidamente las descripciones de las clases que responden a algunos de los casos de uso más importantes del sistema. Las restantes clases se detallan en el Anexo 6.

3.6.1 Descripción de la clase entidad “Torneo”

Nombre	Torneo
Tipo de Clase	Entidad
Atributo	Tipo
Nombre Torneo	String
Ciudad	String



Fecha Inicio	Date
Fecha Fin	Date
Organizador	String
Bases	Text
Objetivos	Text
Historia	Text
Fecha Inscripción	Date
Premio	Text
Elo Máximo	INT
Elo Mínimo	INT
Biografía	Text
Estado	String
Responsabilidad	
Nombre	_construct (nombre, ciudad, fecha inicio, fecha fin, organizador, bases, objetivos, historia, fecha inscripción, premio, elo máximo, elo mínimo,
	_destruct()
	_get(valor)
	_set(valor1,valor2)
Descripción	Clase entidad que equivale a la tabla Torneo en la base de datos y además posee las funciones para el acceso a sus datos.

Tabla 9: Descripción de la clase entidad "Torneo"

3.6.2 Descripción de la clase entidad "Ronda"

Nombre	Ronda
Tipo de Clase	Entidad
Atributo	Tipo
Numero Ronda	String



Fecha Inicio	Date
Resumen Ronda	Text
Responsabilidad	
Nombre	_construct(numero ronda, fecha inicio, resumen ronda)
	_destruct()
	-get(valor)
	_set(valor1,valor2)
Descripción	Clase entidad que equivale a la tabla Ronda en la base de datos y además posee las funciones para el acceso a sus datos.

Tabla 10: Descripción de la clase entidad "Ronda"

3.6.3 Descripción de la clase entidad "Árbitro"

Nombre	Árbitro
Tipo de Clase	Entidad
Atributo	Tipo
Id Arbitro	
Responsabilidad	
Nombre	_construct(id arbitro)
	_destruct()
	_get(valor)
	_set(valor1,valor2)
Descripción	Clase entidad que equivale a cierta información de la tabla Árbitro en la base de datos y además posee las funciones para el acceso a sus datos.

Tabla 11: Descripción de la clase entidad "Árbitro"

3.6.4 Descripción de la clase controladora "Control_Torneo"

Nombre	Control_Torneo
Tipo de Clase	Controladora



Atributo	Tipo
Responsabilidad	
Nombre	Insertar_Torneo()
	Eliminar_Torneo()
	Torneo_Completado()
	Completar_Torneo()
	Mostrar_Torneos_Anteriores()
	Torneo_Creado()
	Torneo_En_Proceso()
	Torneo_En_Ejecucion()
	Torneos_Terminados()
	Dar_acceso(usuario,contrasena)
	Procesar_Torneo()
	Ejecutar_Torneo()
	Terminar_Torneo()
	Nombre_Torneo()
	Estado_Torneo(id_torneo)
	Id_Torneo()
	Nombre_Torneo_Terminado(id_torneo)
	Buscar_Torneos_Anteriores()
	Fecha_Torneo(ano,lista_torneo)
	Mostrar_Bases_Torneo(id_torneo)
	Mostrar_Historia_Torneo(id_torneo)
	Mostrar_Biografia(\$d_torneo)
	Mostrar_Premios_Torneo(id_torneo)
	Mostrar_Objeticos_Torneo(id_torneo)
Inscipcion(id_torneo)	



Descripción	Clase que controla todo lo relacionado a los Torneo y maneja cierta información de los usuarios aunque no utiliza su tabla de la base de datos.
--------------------	---

Tabla 12: Descripción de la clase controladora "Control_Torneo"

3.6.5 Descripción de la clase controladora "Control_Ronda"

Nombre	Control_Ronda	
Tipo de Clase	Controladora	
Atributo	Tipo	
Responsabilidad		
Nombre	Insertar_Ronda()	
	Insertar_Partida(id_ronda)	
	Insertar_Noticia(id_ronda)	
	Modificar_Noticia(id_ronda,id_noticia)	
	Mostrar_Todas_Rondas(id_torneo)	
	Mostrar_Noticia_Ronda(id_ronda)	
	Mostrar_Todas_Partidas(id_ronda)	
	Mostrar_Partida_Id(id_partida)	
	Eliminar_Ronda(id_ronda)	
	Eliminar_Partida(id_partida)	
	Eliminar_Noticia(id_ronda,id_noticia)	
Descripción	Clase que controla toda la información relacionada a las rondas, partidas y noticias del torneo.	

Tabla 13: Descripción de la clase controladora "Control_Ronda"

3.6.6 Descripción de la clase controladora "Control_Árbitro"

Nombre	Control_Árbitro	
Tipo de Clase	Controladora	
Atributo	Tipo	



Responsabilidad	
Nombre	Mostrar_Todos_Arbitros()
	Mostrar_Todos_Arbitros_Torneo()
	Mostrar_Arbitros_Confirmados()
	Correos_Arbitros()
	Insertar_Arbitro_Torneo()
	Eliminar_Arbitro_Torneo()
	Buscar_Arbitros()
	Id_Arbitro_Usuario(id_usuario)
	Estado_Arbitro(id_arbitro)
	Confirmar_Arbitros_Organizador()
	Confirmar_Participacion_Arbitro()
	Rechazar_Arbitros_Organizador()
	Rechazar_Participacion_Arbitro()
	Mostrar_Arbitros_Pendientes()
Mostrar_Arbitros_Rechazados()	
Descripción	Clase que controla toda la información relacionada a los árbitros que existen en la base de datos.

Tabla 14: Descripción de la clase controladora "Control_Arbitro"

3.6.7 Descripción de la clase de acceso a datos "Acceso_Datos"

Nombre	Acceso_Datos
Tipo de Clase	AD
Atributo	Tipo
Host	String
Usuario	String



Contraseña	String
Base Dato	String
Mysqli	String
Responsabilidad	
Nombre	__construct()
	__get(valor)
	__set(valor1,valor2)
	__destruct()
	Execute_Query(query)
	Fetch_Array(result)
Descripción	Clase que se encarga de ejecutar la conexión y las consultas a la base de datos.

Tabla 15: Descripción de la clase de acceso a datos "Acceso_Datos"

3.6.8 Descripción de la clase de acceso a datos "AD_Torneo"

Nombre	AD_Torneo	
Tipo de Clase	AD	
Atributo	Tipo	
Responsabilidad		
Nombre	Registrar_Torneo(objeto)	
	Eliminar_Torneo(id_torneo)	
	Completar(id_torneo,objeto)	
	Mostrar_Torneos_Anteriores()	
	Torneo_Creado()	
	Torneo_En_Proceso()	
	Torneo_En_Ejecucion()	
	Procesar_Torneo()	



	Ejecutar_Torneo()
	Terminar_Torneo()
	Nombre_Torneo()
	Id_Torneo()
	Estado_Torneo(id_torneo)
	Nombre_Torneo_Terminado(id_torneo)
	Buscar_Torneos_Anteriores()
	Buscar_Torneos_Anteriores1(nombre)
	Buscar_Torneos_Anteriores2(ciudad)
	Buscar_Torneos_Anteriores3(nombre,ciudad)
	Mostrar_Bases(id_torneo)
	Mostrar_Historia(id_torneo)
	Mostrar_Biografia(id_torneo)
	Mostrar_Premios(id_torneo)
	Mostrar_Objeticivos(id_torneo)
	Fecha_Inscripcion(id_torneo)
Descripción	Clase responsable de realizar todas las consultas a la base de datos relacionadas al manejo de los torneos.

Tabla 16: Descripción de la clase de acceso a datos "AD_Torneo"

3.6.9 Descripción de la clase de acceso a datos "AD_Ronda"

Nombre	AD_Ronda	
Tipo de Clase	AD	
Atributo	Tipo	
Responsabilidad		
Nombre	Mostrar_Todas_Rondas(id_torneo)	
	Mostrar_Partidas(id_ronda)	



	Mostrar_Partida_Id(id_partida)
	Mostrar_Noticia_Ronda(id_ronda)
	Insertar_Ronda(objeto)
	Insertar_Partida(id_ronda,objeto)
	Insertar_Noticia(id_ronda,objeto)
	Modificar_Noticia(id_ronda,id_noticia,objeto)
	Eliminar_Ronda(id_ronda)
	Eliminar_Partida(id_partida)
	Eliminar_Noticia(id_ronda,id_noticia)
Descripción	Clase responsable de realizar todas las consultas a la base de datos relacionadas al manejo de las rondas, partidas y noticias del torneo.

Tabla 17: Descripción de la clase de acceso a datos "AD_Ronda"

3.7 Modelo de clases persistentes

Las clases persistentes son aquellas que almacenan información que necesita ser consultada frecuentemente. El diagrama de clases persistentes representa la interacción que existe entre estas clases.

A continuación se muestra el Diagrama de Clases Persistentes:

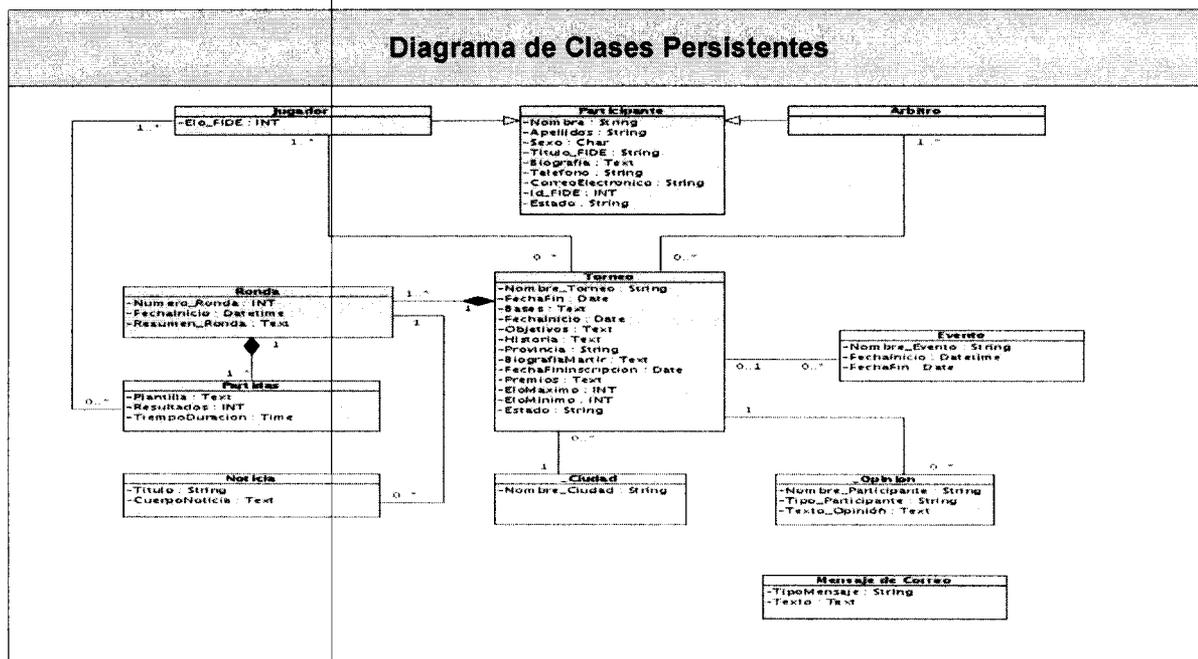


Figura 19: Diagrama de Clases Persistentes

3.8 Principios de Diseño

El diseño web consiste en la planificación, diseño e implementación de aplicaciones web y páginas web, y tiene como objetivo primordial la satisfacción de los usuarios. Este diseño debe contemplar la navegabilidad, interactividad, usabilidad, arquitectura de la información, entre otros aspectos. Siempre se debe tener en cuenta que una navegación engorrosa, puede provocar la pérdida casi inmediata de todos los posibles clientes-usuarios. En el momento de diseñar un sitio web se deben tener en cuenta ciertos principios de diseño enfocados completamente a los usuarios, tales como:

1. Proporcionar una interfaz agradable que contribuya al entendimiento por parte del usuario de la información presentada.
2. El usuario debe iniciar las acciones y controlar las tareas.
3. El usuario debe poder personalizar la interfaz.
4. El diseño debe lograr la mayor interactividad posible.
5. El diseño debe contemplar las posibilidades de error del usuario.
6. El diseño debe dar una respuesta apropiada a las acciones del usuario en un tiempo apropiado.



7. La interfaz debe ser simple, fácil de usar y aprender, con funcionalidades accesibles y bien definidas.
8. Se debe posibilitar la reversibilidad y recuperabilidad de las acciones.
9. El usuario puede realizar distintas búsquedas.

3.9 Interfaz de Usuario

La interfaz de usuario es una de las partes más importante de cualquier programa, la misma es un conjunto de comandos a través de los cuales se comunican el usuario y el sistema mediante el lenguaje visual. Representa los pasos a seguir por el usuario durante su estadía en el sistema y proporciona las respuestas a las distintas acciones de los usuarios, cuenta además con todo lo que el usuario podrá consultar o ejecutar en el programa.

Realizar el diseño de la interfaz de usuario exige gran dedicación y tener siempre en cuenta los principios del diseño web para garantizar:

- ✓ Una interfaz agradable
- ✓ Interactividad usuario/sistema
- ✓ Búsqueda estable.
- ✓ Respuestas apropiadas en tiempo apropiado.
- ✓ Retroceso de acciones y recuperación de la información.

3.10 Concepción general de la ayuda.

A pesar de que todo el trabajo realizado en el diseño e implementación del sistema se enfocó a lograr una aplicación sumamente sencilla para el usuario en cuanto a uso y manejo, siempre para este existen cosas que no están lo suficientemente claras, es por ello la importancia de proveer el sistema de una documentación de ayuda que sirva de guía a los usuarios en su navegación por el sistema, indicándole qué debe hacer para realizar cada una de las distintas operaciones, esta documentación estará disponible desde cualquier parte de la aplicación para aclarar cualquier duda que el usuario presente a la hora de manejar y hacer uso de la aplicación web.



3.11 Tratamiento de errores.

Para lograr la integridad de los datos y el buen funcionamiento del sistema realizado, es necesario llevar a cabo un correcto tratamiento de los posibles errores que se puedan generar de la interacción del usuario con el sistema en cuestión y de los problemas que se puedan presentar de un momento a otro.

3.12 Seguridad

La seguridad del sistema se centrará en verificar la autenticidad y los niveles de privilegio del usuario antes de que el mismo realice cualquier operación dentro del sistema. Es por ello que se accederá al sitio con usuario y contraseña, se otorgarán los permisos y se verificará que las funcionalidades sean mostradas en correspondencia con los niveles de privilegios asignados. Por otra parte se llevará a cabo la protección contra operaciones no autorizadas que en algún momento puedan afectar la integridad de la información almacenada.

3.13 Conclusiones

En el capítulo que concluye se expusieron de forma detallada los distintos elementos que reflejan cómo queda constituido el sistema propuesto en términos de clases de análisis y del diseño; y se ha logrado un resultado importante para la Implementación como son: principios del diseño, modelo de clases persistentes, entre otros. Con el análisis se obtuvo una visión general del sistema y se encontró con el diseño la forma del sistema para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen.

Se describieron además las clases de diseño a través de las cuales es posible comprender claramente cada una de las distintas responsabilidades de las mismas.

También se plantearon aspectos relacionados con: la interfaz de usuario del sistema, el tratamiento de errores, la seguridad del sistema y la concepción de una documentación de ayuda para los administradores de la web.



Capítulo 4

Implementación

4.1 Introducción

En este capítulo se describe cómo los elementos del modelo de diseño, como las clases, son implementados en términos de componentes. Los componentes representan algo físico, que se crea, se modifica o se elimina en el proceso de implementación.

4.2 Modelo de despliegue

El modelo de despliegue es un modelo de objetos que provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue del sistema propuesto, en pocas palabras, el diagrama de despliegue representa lo que le hace falta al sistema para su funcionamiento. Este modelo contiene elementos tales como:

- ✓ Procesadores: Nodos que tienen capacidad de procesamiento, computadoras por lo general: Máquinas Cliente, Servidores.
- ✓ Dispositivos: Nodos que no tienen capacidad de procesamiento: Impresoras, Scanner, entre otros.
- ✓ Protocolos: Estándares para efectuar cierta conexión.

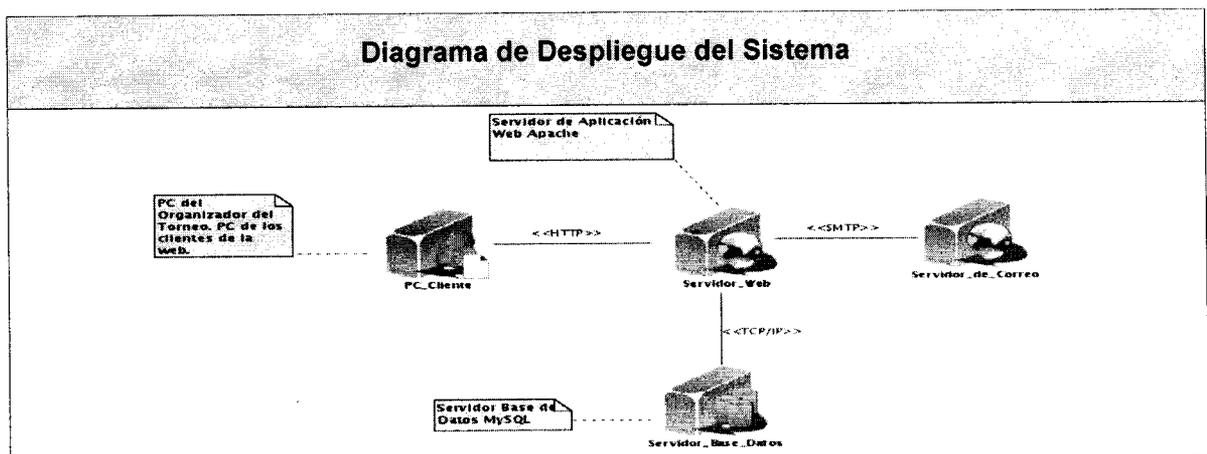


Figura 20: Modelo de Despliegue



4.3 Modelo de Implementación

En el modelo de implementación se describe cómo los elementos del diseño, o sea las clases, se implementan en términos de componentes, tales como: ficheros de código fuente, ejecutables, entre otros. Por otra parte se describe cómo están organizados los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación que se utiliza, y cómo dependen los componentes unos de otros.

Seguidamente se exponen el diagrama de componentes referente al sistema de forma general y los diagramas de algunos de los casos de uso que se consideran más importantes para el sistema. Los diagramas de componentes de los restantes casos de uso se exponen en el Anexo 7.

4.4 Diagrama de Componentes del Sistema

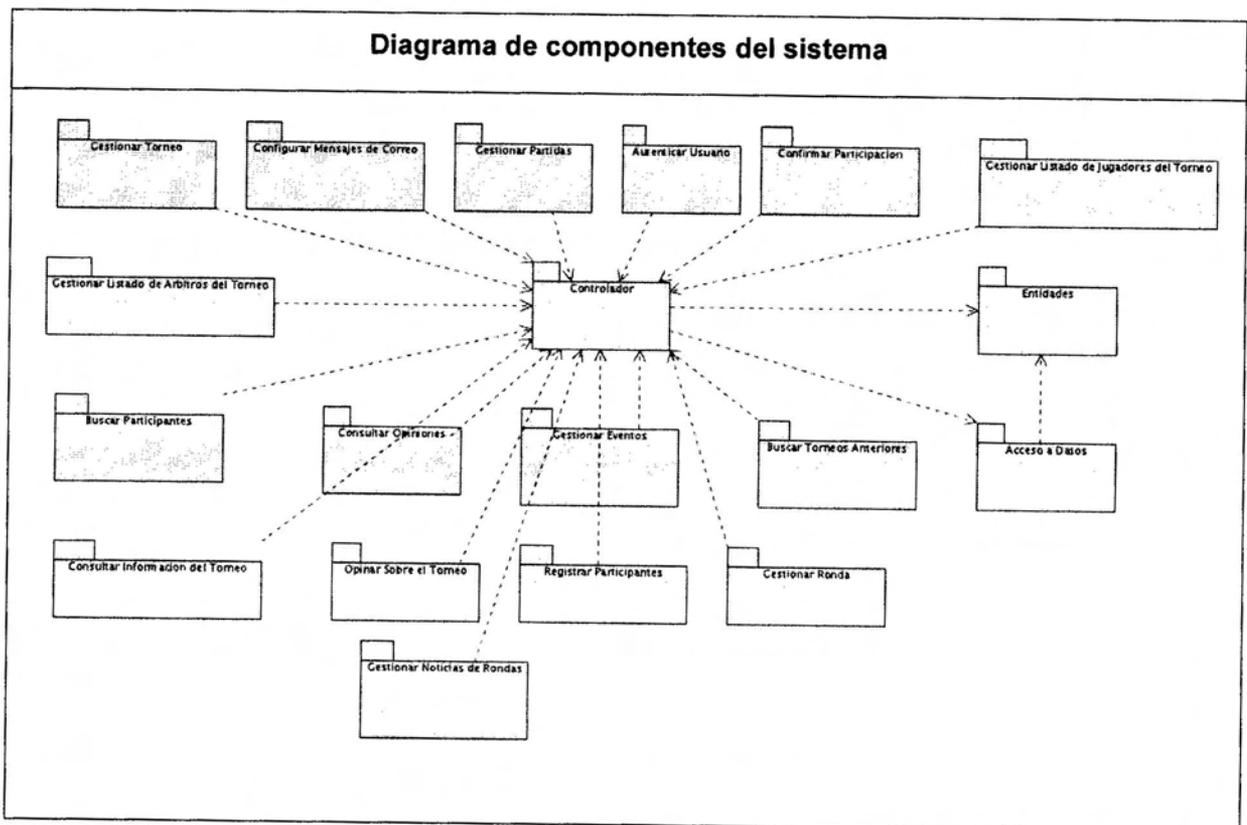


Figura 21: Diagrama de componentes del sistema



4.5 Diagrama de componentes de los casos de uso

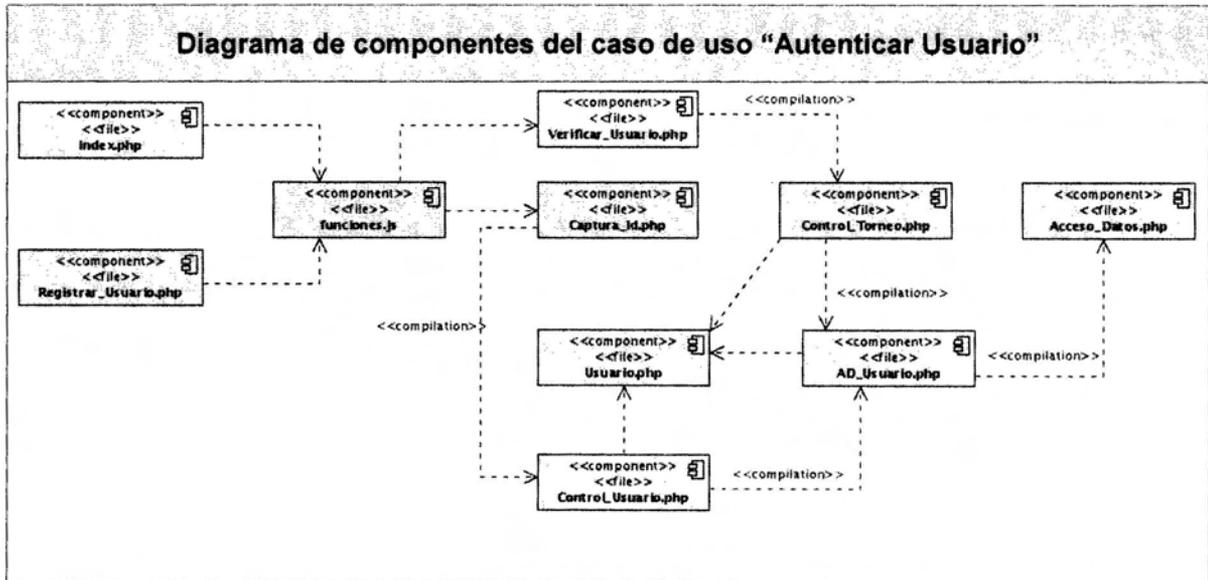


Figura 22: Diagrama de componentes del caso de uso "Autenticar Usuario"

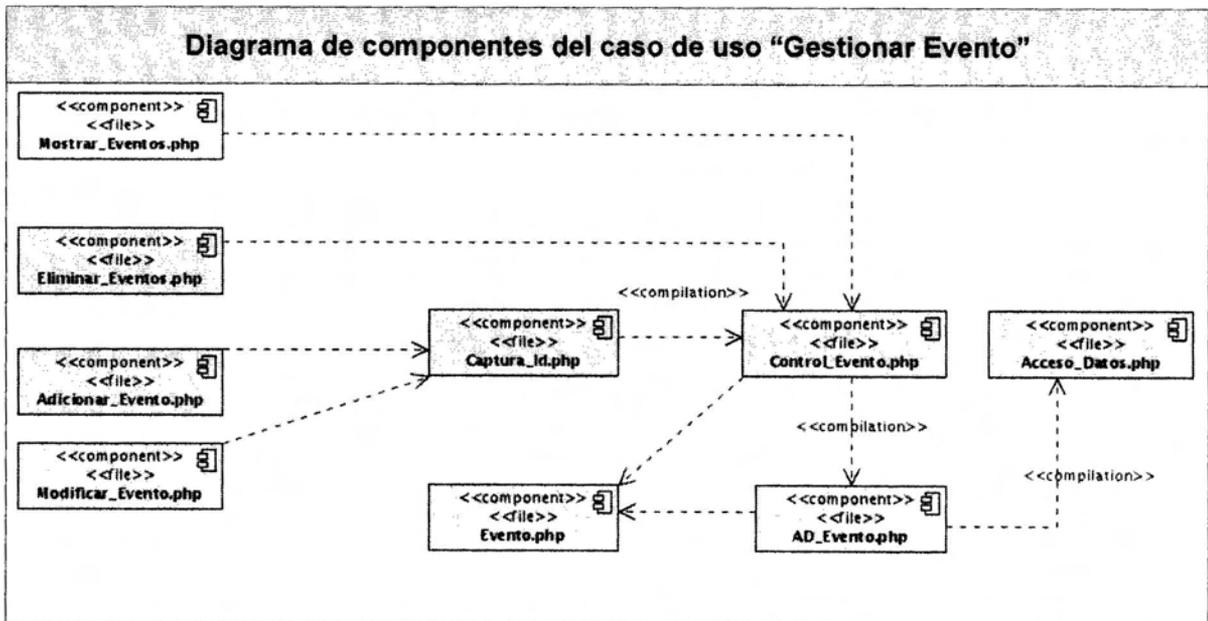


Figura 23: Diagrama de componentes del caso de uso "Gestionar Evento"

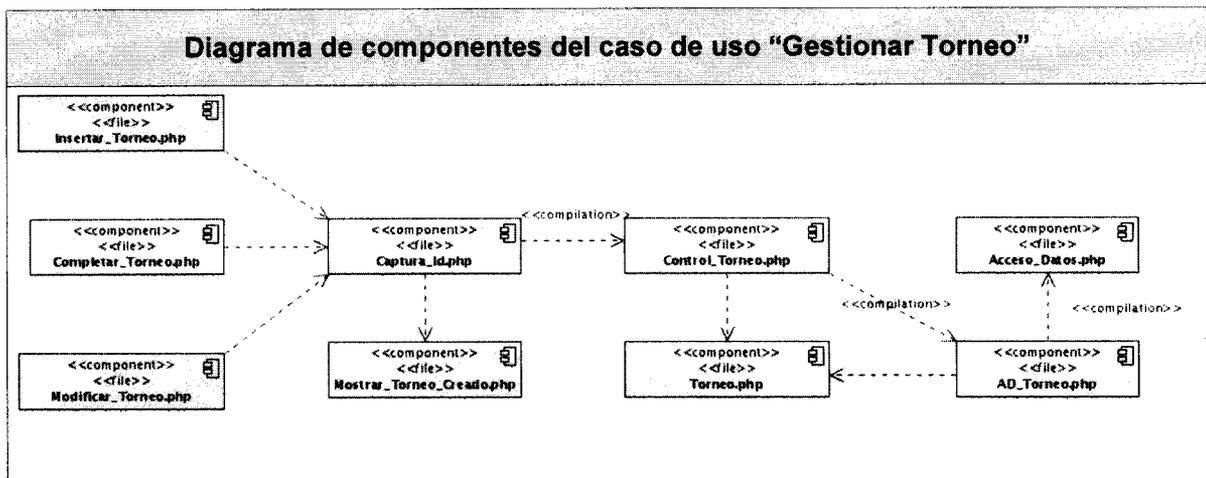


Figura 24: Diagrama de componentes del caso de uso "Gestionar Torneo"

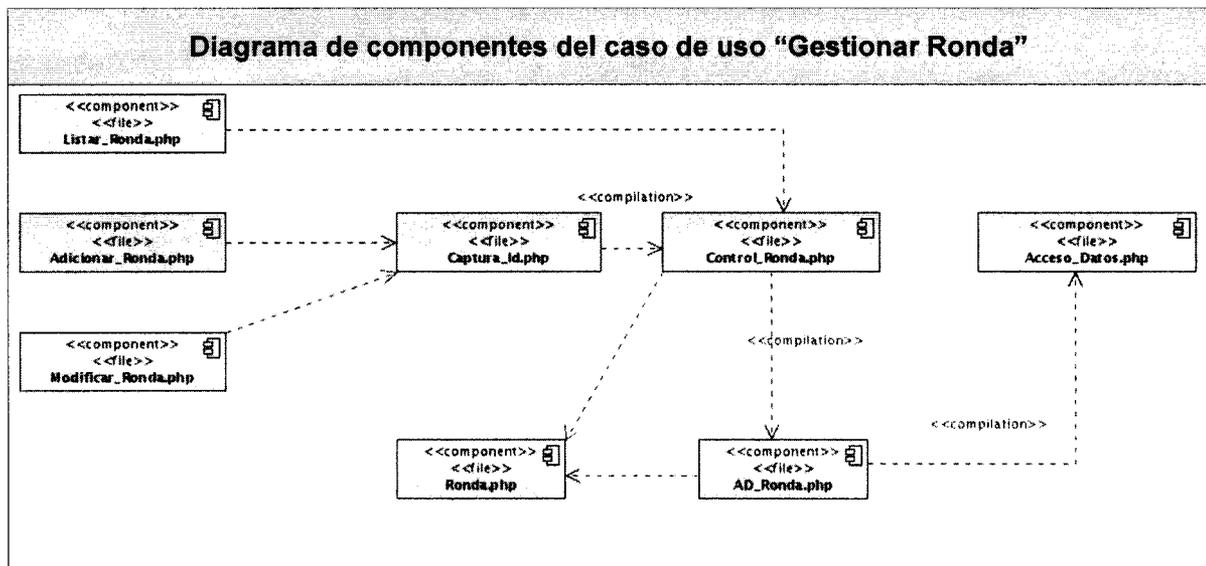


Figura 25: Diagrama de componentes del caso de uso "Gestionar Ronda"

Es necesario aclarar, para una mejor comprensión de los diagramas de componentes de este capítulo, lo siguiente:

En algunos diagramas aparece un componente que lleva por nombre "captura_id", el mismo es un fichero creado para manejar de forma fácil los métodos a los cuales referencian determinadas acciones.



Capítulo 5

Estudio de Factibilidad

5.1 Introducción

En el presente capítulo se realizará el estudio de factibilidad del sistema realizado, por ser fundamental para la realización de un proyecto la estimación del esfuerzo humano, el tiempo de desarrollo que se requiere para la ejecución del mismo y también su costo; dicho estudio se llevará a cabo empleando el método mediante el análisis de Puntos de Casos de Uso.

La especificación de los requerimientos mediante Casos de Uso ha probado ser uno de los métodos más efectivos para capturar la funcionalidad de un sistema.

La estimación por Puntos de Caso de Uso resulta muy efectiva para estimar el esfuerzo requerido en el desarrollo de los primeros Casos de Uso de un sistema, si se sigue una aproximación iterativa como el Proceso Unificado de Rational.

5.2 Puntos de Casos de Uso

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 estos.

5.3 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

5.3.1 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



Módulo para la Gestión de Información de Torneos de Ajedrez Estudio de Factibilidad

segundo lugar, la forma en la que el actor interactúa con el sistema. Analizando el sistema se tienen 5 actores de tipo complejo pues se trata de personas que interactúan con el sistema a desarrollar mediante una interfaz gráfica, a los cuales se le asigna peso 3. No se tienen actores de tipo simple, y se posee un actor medio, al cual se le asigna peso 2.

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	5x3

Tabla 18: Factor de Peso de los Actores sin Ajustar

El factor de peso de los actores sin ajustar sería:

$$UAW=0x1 + 1x2 + 5x3=17$$

5.3.2 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 el caso presente se cuenta con 16 casos de uso, de los cuales:

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	15x5
Medio	El Caso de Uso contiene de 4 a 7		



	transacciones.	10	1x10
Complejo	El Caso de Uso contiene más de 8 transacciones.	15	0x15

Tabla 19: Factor de Peso de los Casos de Uso sin Ajustar

El factor de peso de los Casos de Usos sin ajustar sería: $UUCW = 15 \times 5 + 1 \times 10 + 0 \times 15 = 85$

Como resultado final, los Puntos de Casos de Uso sin ajustar serían:

$$UUCP = UAW + UUCW = 17 + 85 = 102$$

5.4 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.

5.4.1 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.

Factor	Descripción	Peso	Valor asignado	Comentario
T1	Sistema distribuido.	2	5	Es un sistema es distribuido.
T2	Tiempo de respuesta.	1	4	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	3	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.

Tabla 20: Factor de Complejidad Técnica

El Factor de complejidad técnica sería:

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

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

$$TCF = 0.6 + 0.01 \times 53$$

$$TCF = 1.13$$



5.4.2 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.

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	4	El tiempo trabajado en la aplicación ha sido bastante.
E2	Experiencia en orientación a objeto.	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.

Tabla 21: Factor de Ambiente

El Factor de Ambiente sería:

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso}_i \times \text{Valor asignado}_i)$$

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

$$EF = 1.4 - 0.03 \times 23$$

$$EF = 0.71$$



Módulo para la Gestión de Información de Torneos de Ajedrez Estudio de Factibilidad

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

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

$$UCP = UUCP \times 1.13 \times 0.71$$

$$UCP = 102 \times 1.13 \times 0.71 = 81.8$$

5.5 Calcular el Esfuerzo del Flujo de Trabajo Implementación

$$E = UCP \times CF$$

Donde:

E: Esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de Conversión.

Para calcular el Factor de Conversión 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 de 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 de 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.

En resumen:

CF = 20 horas-hombre (si Total EF \leq 2)

CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

CF = abandonar o cambiar proyecto (si Total EF \geq 5)

Finalmente el Factor de Conversión sería: 20 horas-hombre.



Módulo para la Gestión de Información de Torneos de Ajedrez Estudio de Factibilidad

Calculando el esfuerzo necesario para el desarrollo de los casos de uso como: $E = UCP \times CF = 81.8 \times 20 = 1636$ Horas/Hombre

Es importante tener en cuenta que este 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 dura el proyecto en total, se debe añadir a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones del 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:

Actividad	Porcentaje	Horas/Hombre
Análisis	20%	654.4
Diseño	30%	981.6
Programación	50%	1636
Total	100%	3272

Tabla 22: Distribución del esfuerzo entre las diferentes actividades de un proyecto

Convirtiendo a 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:

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

$$\text{Quedando, } Et = 3272 \text{ Horas-Hombres} / 192 \text{ Horas-Mes} = 17 \text{ Mes-Hombres}$$

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

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

$$\text{Tiempo de desarrollo} = 17 / 2 = 8.5 = 8.5 \text{ meses}$$



Módulo para la Gestión de Información de Torneos de Ajedrez Estudio de Factibilidad

Finalmente el tiempo empleado entre dos hombres para el desarrollo de la aplicación es de 8 meses y medio.

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: **\$349**

5.6 Costo

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

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

$Ct = \$349 \times 2 \times 8.5 = \5933

5.7 Beneficios tangibles e intangibles

5.7.1 Tangibles

El producto “Módulo para la Gestión de Información de Torneos de Ajedrez” es un producto elaborado en software libre por lo que tiene como beneficio inicial el hecho de que esté adecuado a las necesidades de la institución y las aspiraciones de migración. Su objetivo es poner a disposición de todos los que lo necesiten una aplicación para el control, registro y visualización de toda la información generada en un torneo de ajedrez. La documentación generada puede servir de consulta para el desarrollo de soluciones similares, así como la escalabilidad y el seguimiento de patrones de responsabilidad, provee de una aplicación eficiente y reutilizable. El producto es de distribución gratuita, por tanto, no reporta beneficios económicos.

5.7.2 Intangibles

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

- ✓ Centralización de la información generada en los torneos de ajedrez.
- ✓ Control y registro de cada resultado obtenido.
- ✓ Almacenamiento de información de torneos pasados.
- ✓ Mayor organización a la hora de la gestión de un torneo.



Módulo para la Gestión de Información de Torneos de Ajedrez Estudio de Factibilidad

- ✓ Acceso casi inmediato a la información almacenada.
- ✓ Visualización de la información de los torneos.

5.8 Análisis costo-beneficio

Las características presentes en la aplicación, conllevan a reconocer que hace falta cierto tiempo para la realización del software en cuestión. No es un producto pequeño por lo que requiere algunos gastos de recursos.

La aplicación cuenta con una interfaz que favorece la navegación, ya que el usuario podrá vagar por ella sin la posibilidad de perderse. Es beneficioso realizar esta aplicación pues aumenta el control, visualización y acceso a la información generada en un torneo de ajedrez.

5.9 Conclusiones

Luego de haber concluido con el estudio de factibilidad se destacan los costos a incurrir, los recursos humanos implicados, el tiempo de desarrollo y los beneficios que arroja la terminación del producto "Módulo para la Gestión de Información de Torneos de Ajedrez". Expresándose además en este capítulo la ventaja que la realización de este producto implica.

Parámetros	Valores
Esfuerzo	17 mes-hombres
Tiempo de desarrollo	8.5 meses
Cantidad de hombres	2 hombres
Salario	\$349
Costo Total MN	\$5933
Costo Total CUC	\$237

Tabla 23: Estimación Final



Conclusiones

Se desarrolló una aplicación web a través de la cual se da solución a las necesidades que existen en la Cátedra de Ajedrez de la UCI. Para ello, se realizó un estudio profundo de los sitios más prestigiosos vinculados con el juego ciencia. Se analizaron las tendencias y tecnologías actuales para el desarrollo de aplicaciones web en software libre y se seleccionaron las más adecuadas para la construcción del sistema.

El sistema informático obtenido propicia la gestión, registro, control, visualización y acceso eficiente a la información almacenada de los torneos de ajedrez que se realizan en la Universidad, dando respuesta, en un alto porcentaje, a la problemática planteada.



Recomendaciones

- ✓ Vincular este módulo con el módulo de arbitraje, ya que ambos están estrechamente relacionados.
- ✓ Integrar a la plataforma Infodrez, para hacer uso de una especializada gestión de noticias e imágenes; así como contar con el visor de las partidas y la descarga de las mismas.
- ✓ Dar la posibilidad de soportar en la aplicación todos los sistemas de juego de los torneos de ajedrez.
- ✓ Realizar un estudio de cómo podría la aplicación migrar sus datos de MySQL a PostgreSQL: actualmente la aplicación web realizada está enfocada solamente a resolver los problemas de la Cátedra de Ajedrez de la Universidad, pero se espera que el alcance de la misma se extienda a todo el país, es por ello que se recomienda dar la posibilidad de que el gestor de base de datos sea PostgreSQL, ya que soporta la concurrencia de usuarios con mayor eficiencia.



Referencias Bibliográficas

1. Graham, P. "Beneficios De Las Aplicaciones Basadas En Web Y El Anuncio De Microsoft De La Era "En Vivo": Disponible en:
http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones_Basadas_En%20_Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.htm
2. Álvarez, R. "Lenguajes de lado servidor o cliente.": Disponible en:
<http://www.desarrolloweb.com/articulos/239.php>
3. Álvarez, Á. M. "Introducción a PHP 5.": Disponible en:
<http://www.desarrolloweb.com/articulos/1696.php>
4. Luebsgrafic (2008). "Ventajas y Usos de PHP.": Disponible en:
<http://www.metodophp.com/2008/02/27/ventajas-y-usos-de-php/>
5. Álvarez, R. "Lenguajes de lado servidor.": Disponible en:
<http://www.desarrolloweb.com/articulos/243.php>
6. Álvarez, Á. M. "Qué es Python": Disponible en:
<http://www.desarrolloweb.com/articulos/1325.php>
7. Pérez, V. D. (2007). "Los diferentes lenguajes de programación para la web": Disponible en: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>
8. "Aspectos y características generales": Disponible en:
http://www.htmlpoint.com/javascript/corso/js_02.htm
9. Introducción a AJAX: Disponible en: <http://www.librosweb.es/ajax/capitulo1.html>.
10. Peralta, F S. (2006). "El servidor Web. Arquitectura y funcionamiento": Disponible en:
<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=366>.
11. "Apache": Disponible en: <http://hospedaje.cinfonet.com/?web=apache>.
12. (2007). "Módulos": Disponible en:
<http://www.osmosislatina.com/apache/modulos.htm>.



Módulo para la Gestión de Información de Torneos de Ajedrez Referencias Bibliográficas

13. Cuenca, C L. Arquitectura del servidor Apache: Disponible en:
<http://www.desarrolloweb.com/articulos/1112.php>
14. (2006). "Ventajas del Uso de Apache"http://www.abartiateam.com/posicionamiento-en-google/200601_ventajas-del-uso-de-apache
15. "MySQL": Disponible en: http://www.salnet.com.ar/inv_mysql/pag01_intro.htm.
16. Abraham. "MySQL vs PostgreSQL ¿cuándo emplear cada una de ellas?": Disponible en:
http://www.javahispano.org/contenidos/es/mysql_vs_postgresql_cuando_emplear_cada_una_de_ellas_11/?menuld=otro
17. "Servidor de correo SMTP Software": Disponible en:
<http://translate.google.com/cu/translate?hl=es&sl=en&u=http://www.lsoft.com/products/smtpserver.asp&sa=X&oi=translate&resnum=3&ct=result&prev=/search%3Fq%3DSMTP%2BEmail%2Bserver%26hl%3Des>.
18. "Extreme Programming": Disponible en:
http://www.planetacodigo.com/wiki/glosario:extreme_programming
19. "Visual Paradigm for UML": Disponible en :
[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/)



Bibliografía

1. Abraham "MySQL vs PostgreSQL ¿cuándo emplear cada una de ellas?": Disponible en:http://www.javahispano.org/contenidos/es/mysql_vs_postgresql_cuando_emplear_cada_una_de_ellas_11/?menuId=otro.
2. Álvarez, Á. M. "Introducción a PHP 5.": Disponible en:
<http://www.desarrolloweb.com/articulos/1696.php>.
3. Álvarez, R. "Lenguajes de lado servidor.": Disponible en:
<http://www.desarrolloweb.com/articulos/243.php>.
4. Álvarez, R. "Lenguajes de lado servidor o cliente.": Disponible en:
<http://www.desarrolloweb.com/articulos/239.php>.
5. Anónimo."Apache." : Disponible en:
<http://www.manualesapache.com.ar/apache/mpm.html>.
6. Anónimo."Aplicaciones Web.": Disponible en :
<http://www.vincolo.com/es/services.php?id=17>
7. Anónimo "Cuba utilizará software libre.": Disponible en:
http://es.wikinews.org/wiki/Cuba_utilizar%C3%A1_software_libre
8. Anónimo "Herramientas Case Libres.": Disponible en:
<http://ceds.nauta.es/informes/case04.htm>.
9. Anónimo "MySQL.": Disponible en:
http://www.salnet.com.ar/inv_mysql/pag01_intro.htm.
10. Anónimo "Tecnologías.": Disponible en: <http://elmanusito.wordpress.com>.
11. Anónimo "Tutorial de PostgreSQL.": Disponible en: <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/x56.html>.
12. Anónimo (2007). "Apache.": Disponible en:
<http://www.osmosislatina.com/apache/modulos.htm>.
13. Anónimo."Principios, Prototipos y Heurísticas para Evaluación.": Disponible en:
<http://www.monografias.com/trabajos11/heuri/heuri.shtml>.



14. Anónimo. (2008). "Lenguajes de Programación Web": Disponible en:
<http://www.mundote.info/Foro/lenguajes-de-programacion-web-t12780.0.html;msg38482>.
15. Anónimo. (2006). "Ventajas del Uso de Apache" http://www.abartiateam.com/posicionamiento-en-google/200601_ventajas-del-uso-de-apache.
16. Anónimo. "Visual Paradigm for UML": Disponible en:
[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).
17. Claudio, P. M. Herramientas Case: Disponible en:
<http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>
18. Cuenca, C. L. "Arquitectura del servidor Apache.": Disponible en:
<http://www.desarrolloweb.com/articulos/1112.php>.
19. Draving (2007). "8 lenguajes de programación/php c java ajax ruby asp perl que deberías aprender.": Disponible en: <http://www.joltivan.com/noticia7846-8-lenguajes-de-programacion-php-c-java-ajax-ruby-asp-perl-que-deberias-aprender.html>.
20. Espinosa, R. A. H. "Reflexiones sobre el uso del Software Libre en Cuba. Ventajas.": Disponible en: <http://www.monografias.com/trabajos40/software-cuba/software-cuba.shtml>.
21. Fernández, S. P. (2006). "El servidor Web. Arquitectura y funcionamiento.": Disponible en:
<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=366>.
22. Garrett, J. J. (2005). "Ajax un nuevo acercamiento a Aplicaciones Web.": Disponible en: <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>.
23. Gómez, L. S. M. "Diseño de Interfaces de Usuario: Disponible en:
<http://www.monografias.com/trabajos11/heuri/heuri.shtml>.
24. González, L. (2007). "Lenguajes del lado servidor y del lado cliente.": Disponible en:
<http://eats.wordpress.com/2007/01/17/lenguajes-del-lado-servidor-y-del-lado-cliente/>.



25. Graham, P. "Beneficios De Las Aplicaciones Basadas En Web Y El Anuncio De Microsoft De La Era "En Vivo".": Disponible en:
http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones_Basadas_En%20_Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.htm.
26. Heredia, M. G. (2007). "Zend Studio.": Disponible en: <http://www.tufuncion.com/zend-studio>.
27. Hidalgo, A. F. P. (2006). "Descripción de los aspectos fundamentales del lenguaje de programación Perl.": Disponible en: <http://www.di-mare.com/adolfo/cursos/2007-1/pp-Perl.pdf>.
28. Ibarra, S. "HOWTO: El servidor web ligero: Lighttpd + PHP5 +MySQL5.": Disponible en: <http://www.saghul.net/blog/2007/08/14/howto-un-servidor-web-ligero-lighttpd-php5-mysql5/>.
29. Jacobson, I, Grady B, James R. El Proceso Unificado de Desarrollo de Software. Félix Varela.2004.438.
30. Ku, A. A. P. "Ruby on Rails: Web Development Frameworks.": Disponible en: <http://anatoly.ragesys.net/2007/07/25/ruby-on-rails-web-development-framework/>.
31. Luebsgrafic (2008). "Ventajas y Usos de PHP.": Disponible en: Disponible en : <http://www.metodophp.com/2008/02/27/ventajas-y-usos-de-php/>.
32. Martínez, A. (2008). "Probando PostgreSQL.": Disponible en: <http://www.antonioimtz.org/?q=postgresql>.
33. Martínez, S. G. "¿Qué es el ISLA?": Disponible en: <http://www.cuba.cu/ajedrez/isla.html>.
34. Obregón, R. (2006). "Lenguajes interpretados vs Lenguajes Compilados: Desidia, capricho o tendencia.": Disponible en: <http://robregonm.blogspot.com/2006/02/lenguajes-interpretados-vs-lenguajes.html>.
35. Peralta, M. "Estimación del esfuerzo basada en Casos de Uso.": Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=9007>.
36. Torre, A. d. I. (2006). "Lenguajes del lado servidor o cliente.": Disponible en: http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.



Glosario de Términos

1. **CVS**: es un **Sistema Concurrente de Versiones**, una forma de trabajo habitualmente utilizada para almacenar el código fuente de grandes proyectos de software. CVS almacena todas las versiones de todos los ficheros de tal forma que nada es nunca perdido, y su utilización por varias personas es registrada. También proporciona una forma de combinar el código de dos o más personas que estén trabajando simultáneamente en el mismo fichero. Todo el código y sus versiones son almacenados en un servidor central.
2. **Gedit** es el editor de texto libre oficial del escritorio GNOME. Este editor se caracteriza principalmente por su facilidad de uso, conseguida en gran parte gracias a una interfaz gráfica clara y limpia, mostrando únicamente las funcionalidades principales que suelen requerir la mayoría de usuarios.
3. **TLS** (Transport Layer Security): es un protocolo para el establecimiento de una conexión segura entre un cliente y un servidor. TLS (Transport Layer Security) es capaz de autenticar el cliente y el servidor y la creación de una conexión cifrada entre los dos. El TLS (Transport Layer Security) es el protocolo extensible, en el sentido de que los nuevos algoritmos se pueden agregar para cualquiera de estos fines, siempre y cuando el servidor y el cliente son conscientes de los nuevos algoritmos.
4. **Drag & Drop**: Operación de pinchar y arrastrar que facilita la utilización del ratón para realizar operaciones de forma más sencilla y rápida.
5. **IDE**: Integrado de doble canal reforzado.
6. **Elo**: El sistema de puntuación Elo (o, incorrectamente, sistema de puntuación ELO) es un método para calcular la fuerza relativa de los jugadores de juegos como el ajedrez. Fue inventado para mejorar el sistema de clasificación vigente de los jugadores de ajedrez.
7. **Sistema Gestor de Contenido**: Un Sistema de Gestión de Contenidos (Content Management System, en inglés, abreviado CMS) permite la creación y administración de contenidos principalmente en páginas web.
8. **Sistema Todos Contra Todos**: Sistema de Juego donde todos los participantes del torneo se enfrentan entre ellos en un número constante de oportunidades (generalmente, una o dos si es a doble vuelta).



Módulo para la Gestión de Información de Torneos de Ajedrez *Glosario de Términos*

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

10. **Sistema Scheveningen:** Se forman dos equipos, donde cada integrante de un equipo A juega con cada integrante del equipo B alternando totalmente los colores para cada equipo.