



Universidad de las Ciencias Informáticas

Facultad 4

Título: “Análisis y Diseño de la versión 2.0 del portal web de béisbol de la Universidad de las Ciencias Informáticas.”

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

Autora:

Yessenia Rodríguez Báez

Tutores:

Ing. José Leandro Cabrera Pérez

Ing. Katia Pardo Duarte

Ciudad de la Habana, 2011.

“Año 53 de la Revolución”

Declaración de Autoría

Declaro ser la autora del trabajo de diploma Análisis y Diseño de la versión 2.0 del portal web de béisbol de la Universidad de las Ciencias Informáticas, y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma con carácter exclusivo.

Para que así conste se firma la presente a los _____ días del mes de _____ del año _____.

Yessenia Rodríguez Báez

Ing. José Leandro Cabrera Pérez

Ing. Katia Pardo Duarte



*Una de las principales enfermedades del hombre es su inquieta curiosidad por
conocer lo que no puede llegar a saber.*

Blaise Pascal

Dedico este trabajo en general a toda mi familia por todo el apoyo y la confianza incondicional que siempre me han dado.

A mi abuelo Efraín Báez, el cual no se encuentra ya entre nosotros pero hasta el último momento de su vida me dio mucho amor, cariño y educación.

A mi abuelita Elisa Reyna, la cual ha sido y siempre será como mi madre, gracias por todo tu amor y comprensión.

A mi sobrinita Lissa Marie, mi hermana Elizabeth, mi abuela María de la Caridad y a mis tíos y primos.

A mi queridísimo novio Gerar, pues me ha dado su amor y apoyo desde mis primeros años en la universidad, a sus padres Gerardo y Marbelis, su sobrinito Jorgito, su cuñado Jorge y su hermana Keny, han sido una familia más para mí.

Y muy especial dedico este trabajo para quienes me dieron el derecho de nacer y quienes han estado todo este tiempo de mi vida a mi lado dándome amor, educación y mucho apoyo, a mis padres María del Carme Báez y Andrés Rodríguez.

Agradecimientos

Agradezco a toda mi familia por estar siempre apoyándome en cada momento.

A mi abuelita Elisa que ha asumido el papel de madre durante estos 5 años de la carrera, dando todo su amor, comprensión y apoyo en todo momento.

A mi mamita María del Carme pues siempre ha confiado en mí, me ha dado mucho apoyo, amor y dedicación para que lograra ser lo que soy en estos momentos... te quiero muchísimo Tatico.

A mi papito Andrés que de igual forma siempre ha confiado en mí, en que saldré adelante pese a los traspiés que se interponen en el camino, gracias por todo tu amor, apoyo y toda la educación que junto a mi mami me has dado... te quiero mucho...

A mi sobrinita que siempre me espera impaciente para vacacionar, a mi hermana por apoyarme y aconsejarme en todo momento, las quiero.

A Gerar, mi amor del alma, siempre me ha dado mucho apoyo en las buenas y malas, cariño y una familia inigualable, sus padres Marbelis y Gerardo, que son como mis padres, los quiero mucho, a su sobrinito Jorgito, su cuñado Jorge y su hermana Keny que ha sido siempre más que una hermana para mí... los quiero mucho a todos.

A todos mis vecinos y amigos de mi casa, a Marica y su familia que es mía también.

A mi amiga por siempre Roxi, gracias por creer siempre en mí y por todos los momentos que compartimos juntas. Te quiero.

A Lazara Barrizonte y a su esposo Hector Luis (Viti), por su sonrisa siempre para levantar mi ánimo y su apoyo incondicional en todo momento, a Adrianet por haberse convertido en alguien especial y que siempre ha estado ahí en los momentos que más la he necesitado, a Lili Bello, por su compañía, han sido de las mejores cosas que me han sucedido en la universidad. Las quiero.

Agradecimientos

A todas mis buenas amistades que nunca olvidaré pues de una forma u otra siempre me han dado apoyo y me han ayudado a llegar hasta aquí, por siempre los querré, Adrián, Lianet, Anita y Julio.

A la Revolución por darme la oportunidad de vivir estos momentos inolvidables.

A la Universidad de las Ciencias Informáticas y la facultad a la que pertenezco por tratar siempre de conducirnos por el mejor camino para llegar hasta la meta y poder obtener el título de Ingeniero en Ciencias Informáticas.

A mis tutores Katia Pardo Duarte y José Leandro Cabrera Pérez por su apoyo, confianza y disposición para cada momento en que los necesité.

A mis compañeros de grupo por los momentos que compartimos juntos durante toda la carrera.

A todos los profesores que me impartieron clases en los 5 años de la carrera y en especial a la Thaimy y Dailín por todo su amor y dedicación.

Los portales web actualmente son de gran ayuda e interés para quienes los visitan, gracias a la información que cada uno según la entidad a la que pertenezca, almacene en sus páginas. A partir de esta afirmación, y debido al interés de sus administradores y por la alta aceptación que tiene el béisbol en Cuba, nuestro país, es que se decide crear el portal web de béisbol de la Universidad de las Ciencias Informáticas (UCI). El sitio que actualmente presta servicios a la comunidad universitaria, se ha desarrollado con tecnologías consideradas obsoletas para estos tiempos, propietarias y además, el uso de las mismas dificulta obtener nuevas funcionalidades como las encuestas que se puedan publicar, no cuenta con el expediente de proyecto incluyendo los artefactos definidos en el mismo, es decir, no existe ninguna documentación que propicie el entendimiento a los nuevos desarrolladores del sistema versionado, por tales motivos surge la necesidad de crear un nuevo portal web con las expectativas actuales de la universidad y el país de migrar a software libre y que responda a gestión de jugadores, juegos, estadísticas, y otros aspectos de interés con el objetivo de mantener actualizado al que lo visita acerca de lo que acontece en los juegos de béisbol nacional.

Por tanto, el objetivo de este trabajo consiste en realizar el análisis y diseño del portal web de béisbol de la UCI en su versión 2.0, quedando aquí explícitos todos los artefactos que se generen durante cada flujo de trabajo del ciclo de desarrollo del software.

Se hace un estudio crítico y valorativo acerca de los tipos de portales web que existen en la actualidad, además, de un análisis de las metodologías y herramientas existentes para el desarrollo web haciendo una selección de las más adecuadas para el desarrollo de la aplicación. Además, se muestran los resultados del análisis y diseño de la propuesta de solución para este sistema, se concluye con la obtención de una documentación que permite un mejor entendimiento del software que se desea desarrollar.

Introducción.....	11
Capítulo 1	16
1. Introducción	16
1.1 Estado del arte.....	16
1.1.1 Portal Web.....	16
1.1.2 Tipos de Portales Web	17
1.1.3 Portales web semejantes al que se le realizará el análisis y diseño:	18
1.2 Metodologías de Desarrollo de Software.....	20
1.2.1 Scrum.....	20
1.2.2 Programación Extrema (XP).....	21
1.2.3 ICONIX.....	22
1.2.4 RUP (Proceso Unificado de Desarrollo de Software).....	22
1.3 Lenguajes de Modelado de Objetos	24
1.3.1 Lenguaje de Modelado Unificado (UML)	24
1.3.2 Lenguaje Orientado a Objetos para la Modelación de Aplicaciones Multimedia (OMMMA-L)	25
1.3.3 Lenguaje para la Modelación de Aplicaciones Educativas (ApEM-L)	25
1.4 Herramientas CASE	26
1.4.1 Rational Rose Enterprise	26
1.4.2 Visual Paradigm	27
1.5 Lenguajes de Programación Web	28
1.5.1 Lenguajes del lado del servidor	28
1.5.2 Lenguajes del lado del cliente.....	31
1.6 Sistemas Gestores de Base de Datos (SGBD)	32
1.6.1 MySQL.....	33
1.6.2 PostgreSQL.....	33
1.6.3 Oracle.....	34
1.6.4 Microsoft SQL Server 2000	34
1.7 Servidores Web.....	35
1.7.1 HTTP Apache	35

1.7.2 ZOPE	36
1.7.3 Internet Information Server (IIS)	36
1.8 Frameworks	37
1.8.1 Symfony	37
1.8.2 Zend Framework	38
1.8.3 JQuery	38
1.9 Entorno de Desarrollo Integrado	39
1.9.1 Eclipse	39
1.9.2 NetBeans	40
Conclusiones	40
Capítulo 2	41
2. Introducción	41
2.1 Objeto de automatización	41
2.2 Información que se maneja	41
2.3 Propuesta del sistema	42
2.4 Modelo del Dominio	42
2.4.1 Diagrama conceptual de Modelo de Dominio	45
2.5 Especificación de los requisitos del software	45
2.5.1 Requisitos Funcionales (RF)	45
2.5.2 Requisitos No Funcionales (RNF)	46
2.6 Definición de los casos de uso	48
2.6.1 Definición de los actores del sistema	48
2.6.2 Listado de los Casos de Uso del Sistema	49
2.6.3 Diagrama de Casos de Uso del sistema	53
2.6.4 Casos de Usos expandidos	53
Conclusiones	56
Capítulo 3	57
3. Introducción	57
3.1 Modelo de análisis	57
3.1.1 Diagramas de clases del análisis:	58
3.2 Patrones Arquitectónicos	60

3.2.1 Patrón Modelo Vista Controlador.....	60
3.2.2 Patrones de Diseño	61
3.3 Modelo de Diseño	63
3.3.1 Clases del Diseño	63
3.4 Diseño de la Base de Datos	66
3.4.1 Diagrama de Clases Persistentes	67
3.4.2 Diagrama Entidad Relación de la Base de Datos.....	68
3.4.3 Descripción de las tablas de la Base de Datos	68
Conclusiones	70
Conclusiones Generales.....	71
Recomendaciones:	72
Glosario de Términos	73
Referencia Bibliográfica	75

Introducción

En la actualidad cada vez resultan más significativos el auge y la importancia que ha tomado la ciencia y la técnica en todas las esferas de la sociedad a nivel mundial, desarrollándose a la par de esto, las Tecnologías de la Información y las Comunicaciones (TIC). Cuba a pesar de los problemas socioeconómicos que por años ha tenido que enfrentar, no se queda atrás en cuanto a los avances tecnológicos, para ello se creó bajo el calor de la Batalla de Ideas que se libra hoy en día, la Universidad de las Ciencias Informáticas (UCI), que tiene como objetivos principales informatizar el país y desarrollar la industria del Software para contribuir al desarrollo económico del mismo.

Dentro de todos los sectores que se han beneficiado con el desarrollo de la informática en Cuba, se encuentra el deporte, y específicamente dentro de esta esfera el béisbol cubano, que es considerado por la alta aceptación dentro del país como deporte Nacional.

Desde que surgió la UCI, muchos han sido los software que se han creado en beneficio de la propia universidad y para el país en general, uno de éstos es el sitio de béisbol que brinda servicios a la comunidad universitaria, el cual fue creado por dos estudiantes de la universidad como parte de su superación y para aplicar los conocimientos adquiridos en clases, ellos se dieron a la tarea de realizar un portal web, donde se reflejara todo lo referente a los juegos de pelota como son: las estadísticas de los jugadores, la posición por zona, el calendario de juegos, información de los jugadores por series, así como noticias de la actualidad beisbolera. Los creadores de esta aplicación, no planificaron que ese mismo año saliera el sitio de Béisbol Cubano realizado por el INDER, el cual tenía funcionalidades comunes al que se estaba desarrollando, pero aun así el Sitio Nacional no cumplía con las expectativas de los clientes por lo que se decidió continuar desarrollando del portal web de Béisbol de la UCI.

En ese tiempo, por el nivel en que se encontraban los desarrolladores, estudiantes de segundo año de la carrera, apenas con unos pocos conocimientos de programación web, sin haber realizado un estudio del estado del arte previo y apoyándose más en los pocos conocimientos de la tecnología ASP y el lenguaje Visual Basic Script, desarrollaron el Sitio de Béisbol de la UCI. Además, esta aplicación que presta servicios a la comunidad universitaria, no posee la documentación ingenieril necesaria, como para que en un futuro pueda ser entendible para nuevos administradores, tampoco es factible seguir desarrollándolo con las mismas tecnologías que se consideran obsoletas, debido a que después del surgimiento de éstas, se han creado nuevos lenguajes de programación que permiten realizar un mejor desarrollo y lógicamente

obtener un producto con un alto grado de rendimiento y calidad. Con las actuales tecnologías de desarrollo web, como son los framework, el lenguaje PHP, entre otros, es posible optimizar las aplicaciones, ganando en rapidez y rendimiento, brindando una mejor prestación de los servicios al usuario, por lo que resulta conveniente realizar el desarrollo de un nuevo portal web de Béisbol para la UCI, aunque el desarrollo de este trabajo se centrará en obtener la documentación quedando así sentadas las bases para la implementación del sistema, debido a que se cuenta con un plazo de tiempo corto como para llegar a obtener la implementación de la aplicación.

Debido a la existencia de un portal web de Béisbol en la UCI desarrollado con tecnología obsoleta (ASP+ Visual Basic Script), ineficiente y sin poderle brindar mantenimiento y soporte, posee una BD que no presenta suficiente soporte para las nuevas funcionalidades para optimizar el mismo, debido a que los desarrolladores del portal no diseñaron una BD lo suficientemente potente como para incluir más datos de los que tiene almacenados, además, las tecnologías utilizadas para su desarrollo son herramientas propietarias y en general no presenta la suficiente documentación ingenieril como para que nuevos administradores y desarrolladores puedan entender lo que se ha desarrollado y de qué forma. Por lo antes planteado surge el siguiente **problema a resolver**: ¿Cómo modelar el portal web de béisbol en su versión 2.0 brindándole nuevos servicios al usuario?

Para dar cumplimiento al problema a resolver, se desea obtener una aplicación web con todas las funcionalidades definidas, logrando así un mejor mantenimiento y soporte para el mismo, pero debido a que su desarrollo es de gran alcance y se cuenta con un plazo corto de tiempo, en este trabajo solo se llegará a realizar el análisis y diseño quedando así sentadas las bases para la futura implementación y prueba de la aplicación que se desea obtener. Por tales motivos, se plantea como **objetivo general** desarrollar el Análisis y Diseño del Portal Web de Béisbol de la UCI en su versión 2.0 para apoyar los procesos de gestión de información.

Se define como **objeto de estudio** las aplicaciones web orientadas a brindar información sobre el béisbol y el **campo de acción** comprende el proceso de Análisis y Diseño del Portal Web de Béisbol en su versión 2.0 con las nuevas tecnologías existentes para la gestión de la información.

El objetivo general que se plantea para el desarrollo del presente trabajo se desglosa en los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación y definir la posición teórica del investigador.
- Identificar las deficiencias y limitaciones que presenta el sitio que existe actualmente en la universidad que justifican el desarrollo de un nuevo portal.
- Identificar los servicios a incluir en el nuevo portal.
- Definir y realizar los diferentes artefactos del flujo de trabajo de Análisis y Diseño.

Tareas de la Investigación:

- Entrevista a los creadores y actuales administradores del sitio para la definición de los nuevos requisitos del mismo.
- Análisis del portal web actual de béisbol de la UCI para la identificación de los requisitos generales.
- Realización del estudio de las metodologías y tecnologías de desarrollo web para seleccionarlas más adecuadas.
- Obtención de los artefactos del ciclo de vida del sistema, como son: el diagrama de casos de uso del sistema, diagrama de clases del análisis y los diagramas de clases del diseño empleando estereotipos web.
- Documentación de todo el proceso de desarrollo del portal web de béisbol.

Posibles Resultados:

La obtención del análisis y diseño, sustentado por la generación de varios artefactos, dejando sentadas las bases para el desarrollo del Portal Web en su versión multiplataforma que permita un constante soporte y mantenimiento al mismo, permitiéndoles a los usuarios una mejor interacción y brindándole la mayor cantidad de información posible.

Métodos Científicos:

Para el desarrollo del trabajo se seleccionaron los métodos investigativos más apropiados, los cuales serán mencionados a continuación, dando una explicación de cómo se utilizan.

Teóricos:

Analítico-Sintético: Al utilizar este método se hizo una descomposición por partes del sistema en cuestión, para un mejor entendimiento del trabajo a desarrollar. Además, permitió la obtención de los elementos más importantes que deben ser implementados.

Histórico –Lógico: La utilización de este método permitió un análisis de las diferentes tecnologías, metodologías y herramientas existentes para el desarrollo de aplicaciones web, además, de analizar portales web preferentemente relacionados con temáticas deportivas, con el objetivo de desarrollar la presente investigación sobre las herramientas y tecnologías más adecuadas.

Modelación: Con la utilización de este método se hizo una representación simplificada, de lo que se modelará en la aplicación, a través, de diagramas se representarán los requisitos funcionales y no funcionales de la aplicación que se desarrollará.

Empíricos:

Observación: Mediante la utilización de este método se obtiene el problema a resolver, y de ahí detallar el objeto de estudio, lo cual permitió tener una visión más clara de lo que se haría, y cómo se haría.

Entrevista: Mediante este método, se obtuvo información de los administradores de la aplicación que existe actualmente, además, permitió identificar qué información se puede mostrar en la nueva aplicación de forma tal que sea del agrado y beneficio para quienes lo visiten.

Estructura capitular: El presente trabajo se encuentra estructurado por tres capítulos:

En el **Capítulo 1** “Fundamentación Teórica”, se incluye el estado del arte de la investigación donde se hace un análisis de las metodologías, tecnologías y herramientas que existen para desarrollar aplicaciones web, seleccionando por último las más adecuadas, además, se hace un estudio de portales web que existen sobre béisbol con el fin de encontrar similitudes que brinden soporte al trabajo en cuestión.

En el **Capítulo 2** “Características del Sistema”, se describen las características del sistema, así como los requisitos funcionales y no funcionales con los que debe cumplir, se realiza el modelado del negocio del sistema a través de un modelo de dominio, abordando con mayor profundidad el problema a resolver y la situación problemática, se genera además, en esta etapa el diagrama de Casos de Uso del Sistema.

En el **Capítulo 3** “Análisis y Diseño”, se identifican clases, métodos e interfaces que conformarán la propuesta de solución, definiendo el modelo de análisis del sistema y partiendo de éste, el modelo de clases, mediante estos artefactos generados es posible obtener un entendimiento claro del funcionamiento del software mostrando los detalles descriptivos de la solución propuesta, quedando toda la documentación detallada para una posterior implementación.

Capítulo 1

1. Introducción

En el presente capítulo se refleja el resultado de las investigaciones realizadas, se incluye el estado del arte de la investigación del trabajo tanto en el ámbito nacional como internacional. Se hará una valoración de las herramientas, tecnologías y metodologías para el desarrollo de aplicaciones web, seleccionando las más adecuadas para el desarrollo del presente trabajo, además, se analizará el portal web de béisbol de la UCI, permitiendo así seleccionar las mejores técnicas para desarrollo web que se pueden utilizar en este trabajo.

1.1 Estado del arte

Debido a los grandes avances que ha alcanzado la informática, han surgido las aplicaciones o portales web, con el objetivo de gestionar gran número de información, acerca de la empresa o entidad para el que ha sido creado, pues de este modo agiliza la mayor cantidad de acciones posibles que se llevan a cabo en dichas entidades.

1.1.1 Portal Web

Primeramente se dará una explicación de lo que son básicamente los portales web, sus características y sus clasificaciones, con el objetivo de que haya un mejor entendimiento de qué es lo que se desarrollará en este trabajo de diploma y como se podrá clasificar el sistema que se desea obtener.

¿Qué es un portal web?

Un portal web está basado en un Sistema de Administración de Contenidos, lo cual permite que se publique de forma fácil y rápida la información en la web, permitiéndole así a los usuarios que hagan uso de él, que obtengan la mayor cantidad de información posible, brindando accesos a diversos servicios y recursos de forma amigable lo que hace que sea del agrado de los usuarios. Además, facilita el acceso a información de diversas fuentes, la integración de herramientas de colaboración como son los grupos de trabajo, comunidades, foros o grupos de discusión y chats, y también brinda acceso a otros sitios de interés en internet [1].

Propósito de un Portal

Los portales web se encargan de satisfacer una amplia gama de necesidades tanto de negocios como personales; estos se basan en el uso de tecnologías web, pues presentan la mejor forma de acceso a los contenidos del negocio, clientes, proveedores y empleados. Básicamente el propósito de crear portales web es proveer acceso personalizado, filtrado

automático de contenidos según las preferencias del usuario, interacción amistosa, integración de múltiples sistemas, escalabilidad, administración de contenidos, seguridad, en fin un portal que presente al menos estas cualidades y que esté bien diseñado permite al usuario tener diariamente información actualizada de lo que sucede por día, lo cual es una gran ventaja para el portal, pues esto constituye un factor clave para sostener las ventajas competitivas[1].

1.1.2 Tipos de Portales Web

Los portales web presentan tres características fundamentales que son las que atraen a los usuarios, éstas son: información, participación y comodidad. Debido a esto, internet ha llegado a clasificarlos por tipos (horizontales, verticales, especializados y móviles) [1].

➤ Portales Horizontales

Es un portal de carácter general orientado a todo tipo de usuario, ofreciendo contenidos que son demandados por el público. En estos se incluyen servicios como son las comunidades virtuales, chat, correo, espacio web gratuito, noticias, buscadores, entre otros elementos de importancia. Estos son portales esencialmente para personas no muy experimentadas en el mundo de Internet [1].

➤ Portales Verticales

Son portales que se especializan en determinados temas, que buscan público y tienen un objetivo bien determinado. En función de su objetivo se pueden clasificar en [2]:

1. Portal Intranet: Comunicación corporativa para los empleados.
2. Portal Extranet: Comunicación corporativa para los proveedores.

➤ Portales Especializados

Es un portal donde la información que brinda es más específica acerca de un tema, a diferencia de los portales horizontales y verticales, los cuales almacenan información demasiado general o superficiales, es por eso que se han creado los portales especializados o exclusivos, como los de noticias, medicina, educación, entre otros [2].

➤ Portales Móviles

Este tipo de portal web permite a los usuarios establecer una conexión ya sea desde internet como de un teléfono móvil, estos portales se están abriendo camino a nivel de empresas para realizar negocios ofreciendo productos o servicios a los usuarios. En los

momentos actuales este tipo de portales web está en sus inicios debido al desarrollo de la telefonía móvil que ya existe, por lo que se le augura un fructífero y amplio futuro [2].

1.1.3 Portales web semejantes al que se le realizará el análisis y diseño:



Figura 1. Portal Web para el béisbol en Cuba

El portal web de béisbol de Cuba empezó a brindar sus servicios a los cibernautas a partir del 19 de noviembre del 2004, día en que se inauguró en La Habana la XLIV serie nacional del principal pasatiempo nacional. En la mencionada página, los usuarios encontrarán disímiles datos estadísticos acerca de cada uno de los participantes en los clásicos, así como de cada uno de los juegos que se estén realizando en cada momento. También, se beneficiarán entrenadores y peloteros quienes tendrán acceso a cursos y temas pedagógicos, para incrementar el nivel en la disciplina. En fin, el portal web creado para informatizar el deporte nacional cuenta con diversos servicios de interés y agrado para quienes lo visiten; sin embargo no se puede dejar de mencionar que por su parte técnica, como un producto, presenta una de las deficiencias que hoy en día trata el país de eliminar y es que ha sido desarrollado sobre la herramienta propietaria ASP (Active Server Pages), que imposibilita el uso legal de la misma.



Figura 2. Portal Web para el béisbol en la UCI

El portal web de béisbol de la UCI fue creado antes del portal nacional, por dos estudiantes de la universidad quienes se dieron a la tarea de realizar una aplicación web de béisbol para la comunidad universitaria, el cual refleja las estadísticas de los jugadores, la posición por zona, un calendario de juegos, información de los jugadores por series así como noticias de la actualidad beisbolera. La aplicación que actualmente presta servicios a la comunidad universitaria, fue desarrollada sobre los lenguajes ASP y Visual Basic Script, ambas son propietarias y se consideran obsoletas, además, el uso de estos lenguajes para el desarrollo de aplicaciones hace que los usuarios se vean obligados al uso del navegador Internet Explorer para obtener todas sus funcionalidades, lo cual afecta al proceso de migración en el que está inmersa la universidad y el país, además, de las tecnologías empleadas ya se han creado nuevas versiones con las que se obtienen aplicaciones más eficientes, ganando en rapidez y rendimiento, brindando una mejor prestación de los servicios al usuario.

El Portal Web de Béisbol de la UCI en su versión 2.0, mantendrá a los usuarios al tanto del acontecer beisbolero a nivel nacional, por lo que se puede clasificar como un portal de tipo especializado. Además, a diferencia de los portales web anteriormente analizados se desarrollará usando herramientas libres, con el objetivo de cumplir con las políticas que promueve el país de migrar al software libre.

1.2 Metodologías de Desarrollo de Software

Las Metodologías de Desarrollo de Software surgen debido a la necesidad de emplear una serie de procedimientos y técnicas a la hora de desarrollar un producto. Estas han sido creadas con el propósito de brindarle una guía al desarrollador a la hora de crear una nueva aplicación. Debido a que no todos los sistemas que se desarrollan tienen la misma complejidad, existen una gran variedad de metodologías para la creación de los mismos, están las Metodologías Pesadas, que son aquellas que establecen rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán y además, las Metodologías Ligeras, que se refieren a una mayor interacción del cliente con el desarrollador del software, mostrándole versiones funcionales del producto en intervalos de tiempo cortos, para que éste pueda evaluar y sugerir cambios según se va desarrollando el sistema [3].

1.2.1 Scrum

Es una metodología ágil la cual define un marco para la gestión de proyectos. Esta metodología se recomienda que sea utilizada para proyectos con un rápido cambio de requisitos. Tiene dos características fundamentales que podemos decir son las que la definen: el desarrollo de software que se realiza es mediante iteraciones, denominadas sprints, con una duración de 30 días, el resultado de cada iteración es un incremento ejecutable que se le muestra al cliente, y su segunda característica importante se basa en pequeñas reuniones que se realizan diariamente con todo el equipo de desarrollo para la coordinación e integración de lo que se ha desarrollado y de lo que posteriormente se va a desarrollar [4].

Scrum también es empleado para situaciones en las que se ha demorado demasiado la entrega de un producto, cuando no se está entregando al cliente lo que necesita, o cuando los costos se elevan. Scrum para controlar la evolución del proyecto utiliza las siguientes prácticas:

- Revisión de Iteraciones
- Desarrollo incremental
- Desarrollo evolutivo
- Auto-Organización
- Colaboración

Finalmente se puede decir que Scrum es una metodología ágil empleada en proyectos pequeños o de corta duración, y que para obtener un mejor producto, realiza reuniones de

alrededor de 15 minutos diarios, donde se analiza lo que se ha hecho, los problemas o dificultades encontradas, y lo que se va a realizar. Algunas de las desventajas que presenta esta metodología es que si no se define una fecha de finalización del proyecto es posible que se siga solicitando, y añadiendo nuevas funcionalidades; además, es una metodología que difiere del resto de las metodologías por lo que causa cierta resistencia en su aplicación por algunas personas.

Ciclo de Vida de Scrum. [Ver Anexo 1](#)

1.2.2 Programación Extrema (XP)

XP es una metodología ágil la cual se centra en potenciar las relaciones interpersonales como pilar fundamental para un exitoso desarrollo de software, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen entorno de trabajo. Éste se basa en una continua comunicación entre el cliente y el equipo de desarrollo con el objetivo de que todo marche según lo convenido y si existiera algún cambio poderlo solucionar de la mejor forma posible. Es una metodología muy utilizada para proyectos de corta duración, con posibles cambios constantes y donde exista un alto riesgo técnico [5].

Esta metodología propone que se realice la programación en pareja con el objetivo de lograr mayores resultados y que sean menos los errores encontrados. Define las Historias de Usuarios como base del software y estas historias las escribe el cliente, describiendo escenarios sobre el funcionamiento del sistema a desarrollar. En su diseño no tiene importancia la fase de análisis como lo define la metodología RUP, es decir, esta fase para XP no es necesaria desarrollarla, debido a que se trabaja en función de las necesidades que se presentan en cada momento.

Independientemente de las ventajas que proporciona XP, como son: está preparado para el cambio con el objetivo de reducir los costos, trata de mantener la menor tasa de errores, permite definir en cada iteración cuales son los objetivos de la siguiente; tiene también sus desventajas pues necesita un alto esfuerzo de desarrolladores y en ocasiones es posible que el prototipo por retrasos se convierta en parte del producto. Se debe delimitar el alcance del proyecto con los clientes.

Ver el ciclo de vida de la metodología XP en los anexos. [Ver Anexo 2](#)

1.2.3 ICONIX

ICONIX es una metodología simplificada en comparación a otras más tradicionales, la cual unifica un conjunto de métodos de orientación a objetos con el objetivo de tener un control estricto sobre todo el ciclo de vida del producto a realizar, cuenta con una secuencia de pasos que se deben seguir y determina claramente las actividades a desarrollar en cada etapa del ciclo de vida del proyecto que la utilice [6].

Características de ICONIX

- Iterativo e Incremental: durante el desarrollo del modelo del dominio y la definición de los casos de uso se producen varias iteraciones.
- Trazabilidad: Cada paso que se realiza está definido por un requisito, se define la trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos de software producidos.
- Dinámica del UML¹: Ofrece un uso dinámico del UML porque utiliza algunos diagramas UML, sin exigir la utilización de todos, como en el caso de RUP.

Esta metodología es la definición de un proceso ágil para poder obtener la especificación de requerimientos y poder modelar el sistema haciendo uso del Lenguaje de Modelamiento Unificado (UML).

La principal desventaja de esta metodología es que necesita información rápida y puntual de los requisitos, del diseño y de las estimaciones, además, es una metodología que no debe ser usada en proyectos de larga duración.

1.2.4 RUP (Proceso Unificado de Desarrollo de Software)

El Proceso Unificado de Desarrollo de Software (RUP) es una metodología pesada o tradicional, mediante la cual se les asignan roles y responsabilidades a los integrantes del equipo de desarrollo. Esto sucede durante el ciclo de vida del desarrollo del software, trazándose como objetivo final la obtención de un producto que cumpla con las expectativas del cliente. RUP se obtuvo como resultado de integrar los mejores elementos de diferentes metodologías de desarrollo, su enfoque es orientado a objetos, contando para ello con las mejores prácticas del modelado de software, entre los que se citan:

¹ Lenguaje de Modelamiento Unificado

- Desarrollo de software iterativo e incremental.
- Administración de requerimientos.
- Arquitectura basada en componentes.
- Modelar visualmente.
- Verificar la calidad del producto.
- Controlar los cambios durante el ciclo de vida del software.

RUP ha definido 9 flujos de trabajo, los cuales son: Modelamiento del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba o Testeo, Instalación, Administración del proyecto, Administración de Configuración, y por último Cambios y Ambiente [7].

RUP está caracterizado por tres elementos fundamentales, ser iterativo e incremental, dirigido por casos de usos y centrado en la arquitectura, además, está organizado en cuatro fases: Inicio, en la cual se define alcance y objetivos del proyecto, Elaboración, en esta fase se define la arquitectura del sistema, Construcción, de aquí se obtiene el producto con todas sus funcionalidades y de forma tal que esté listo para que en la fase de Transición se pueda liberar a sus clientes. Éste es utilizado en proyectos de larga duración y para cualquier cantidad de personas que integren el equipo de trabajo ya sean muchos o pocos.

RUP lleva a cabo un profundo levantamiento de requerimientos, buscando detectar defectos en las fases iniciales con el propósito de reducir tanto como sea posible el número de cambios en el producto que se espera obtener. Se realiza el análisis y diseño lo más completo posible y a diferencia de XP en RUP el cliente interactúa con el equipo de desarrollo mediante reuniones.

Ciclo de Vida de RUP. [Ver Anexo 3](#)

Selección de la Metodología de Desarrollo de Software

Para controlar, y concebir la propuesta de este sistema, se selecciona como metodología de desarrollo de software, RUP, por todas las ventajas de organización que brinda y además, permite que se generen una serie de artefactos que sustentan el proceso de construcción, aunque no es necesario que para todos los software que se desarrollan utilizando dicha metodología, se generen todos los artefactos, o se realicen todas las actividades, se utilizan

solo los necesarios. Además, RUP utiliza el Lenguaje de Modelado UML y es considerada como una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo de software y es la metodología que se imparte en la carrera lo cual permite no tener que emplear tiempo en la capacitación.

1.3 Lenguajes de Modelado de Objetos

1.3.1 Lenguaje de Modelado Unificado (UML)

Un modelado es una simplificación de la realidad, esencialmente en la construcción de software, se utiliza para comunicar la estructura de un sistema complejo, especificar el comportamiento deseado del sistema, comprender mejor lo que estamos construyendo y descubrir oportunidades de simplificación y reutilización. Un modelado proporciona gráficamente cómo debe realizarse y quedar el producto a realizar, mostrando a través de sus gráficas los elementos más relevantes en cada momento [8].

El UML es un estándar diseñado para visualizar, especificar, construir y documentar software orientado a objetos. Se centra en la representación gráfica de un sistema.

UML cuenta con muchos objetivos, pero estos se pueden sintetizar en los siguientes [8]:

- Visualizar: Expresa gráficamente un sistema entendible para cualquier persona que lo consulte.
- Especificar: Especifica cuales son las características de un sistema antes de su construcción.
- Construir: Teniendo en cuenta los modelos previamente especificados, se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Por último es importante destacar que a pesar de que UML está creado principalmente para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajos en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware. Es además, un lenguaje que ha sido aceptado como estándar por la OMG (Object

Management Group en español es Grupo de Gestión de Objetos) y prácticamente todas las herramientas CASE² y de desarrollo la han adoptado como lenguaje de modelado [8].

1.3.2 Lenguaje Orientado a Objetos para la Modelación de Aplicaciones Multimedia (OMMMA-L)

El lenguaje OMMMA-L tiene como base UML e integra el comportamiento interactivo de las aplicaciones multimedia con el de procedimientos temporales, para lograr la descripción de aplicaciones [9].

OMMMA-L Es una extensión del UML, el cual permite modelar aspectos estructurales, funcionales y dinámicos de un sistema interactivo y su interfaz de usuario. Presenta análisis parecidos a RMM (Relationship Management Metodology o Metodología de Administración de Relaciones) haciendo uso original de la semántica de UML. Es robusto y altamente descriptivo, refleja el proceso en todas sus etapas y hereda de RUP el de vida basado en iteraciones y el flujo de trabajo iterativo e incremental, centrado en la arquitectura y dirigido por casos de uso.

1.3.3 Lenguaje para la Modelación de Aplicaciones Educativas (ApEM-L)

El lenguaje ApEM-L se presenta como una extensión de UML que permite incorporar a éste los elementos fundamentales del proceso productivo UCI. En ApEM-L se incorporan los elementos más importantes del lenguaje OMMMA-L, con el objetivo de obtener una extensión consistente para la modelación de aplicaciones educativas [9].

Esencialmente las características de ApEM-L se agrupan en los siguientes conceptos:

- Estructura Lógica: está compuesto por la Vista Estática y la Vista de Arquitectura.
- Comportamiento dinámico: el comportamiento de la aplicación está descrito por la vista de comportamiento, y está compuesta por los diagramas de actividades, secuencia, colaboración y de estados.
- Gestión del Modelo: este aspecto ha cambiado tanto su carácter semántico como sintáctico, con la incorporación de estereotipos restrictivos en todos los diagramas a partir de nuevos conceptos incorporados a los diagramas de clases originales de UML.

² Computer Aided Software Engineering, en español se traduce a: Ingeniería de Software Asistida por Computadoras

Selección del Lenguaje de Modelado

Después de haber analizado los Lenguajes de Modelado anteriormente mencionados, se ha seleccionado para el desarrollo de este trabajo, el UML, pues además, de las características ya mencionadas anteriormente, podemos decir que está considerado como el lenguaje estándar para realizar el análisis y diseño de sistemas de cómputo. Mediante su uso es posible obtener una serie de requerimientos y estructuras necesarias para desarrollar un sistema de software previo al proceso de codificación. Además, los otros lenguajes de modelado analizados son una extensión del UML, por lo que podemos decir que el lenguaje de modelado seleccionado es el más completo de los tres analizados.

1.4 Herramientas CASE

Las herramientas **CASE** (**C**omputer **A**ided **S**oftware **E**ngineering, Ingeniería de Software Asistida por Computadoras) son un conjunto de programas y ayudas que sirven de apoyo a los analistas, ingenieros de software y desarrolladores, durante el ciclo de vida de desarrollo de un software, es decir, en las etapas de inicio, análisis, diseño, implementación e instalación.

1.4.1 Rational Rose Enterprise

Es una herramienta de modelado visual que junto con otras herramientas cubren el ciclo de vida de desarrollo de un software. Es considerado el producto más completo de la familia Rational Rose. Una de sus características esenciales es que utiliza como lenguaje de modelado UML, lo cual permite el diseño de base de datos y requerimientos de aplicaciones a través de diseños lógicos y físicos. Rational Rose Enterprise, está caracterizada por un conjunto de cualidades que como herramienta la hacen una de las más poderosas y efectivas [9].

Características:

- Genera código Ada, ANSI C ++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo-códigos configurables.
- Características de control por separado de componentes, modelo que permite una administración por partes y el uso de modelos.
- Soporte Enterprise Java Beans 2.0.
- Capacidad de análisis de calidad de código.

- Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant³, incluyendo a Rational Clear Case.
- Integración con otras herramientas de desarrollo de Rational.
- Se puede usar en plataformas Windows, Linux y Unix.
- Modelado UML para trabajar en un diseño de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicaciones a través de diseños lógicos y físicos.

Rational Rose es una herramienta propietaria por lo que para hacer uso de la misma se necesita de su licencia, algo que la pone en desventaja para emplearla en el desarrollo del producto que proponemos, además, su entorno gráfico no es amigable para los usuarios y necesita de alta capacidad de procesamiento.

1.4.2 Visual Paradigm

Es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software, es decir, el análisis y diseño orientado a objetos, construcción, pruebas y despliegue. El UML le ayuda a una rápida construcción de aplicaciones de calidad y a un menor coste. Posee una interfaz amigable y se puede modelar en varios idiomas. Es una herramienta colaborativa, pues tiene soporte para que varios usuarios puedan trabajar sobre el mismo proyecto. Soporta la notación UML 2.1, ingeniería inversa, generador de informes en varios formatos (Web o .pdf), editor de figuras, aunque los reportes e imágenes generadas no son de alta calidad, integración con IDE⁴, con Visual Studio, IntelliJ IDEA, Eclipse, NetBeans y otros [9].

Otras características:

- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de integrarse en los principales IDEs.

³ Source Code Control en español Control de Código Fuente significa que un programa utiliza una interfaz en particular definida por Microsoft para Visual SourceSafe. El SCC denota control de código fuente.

⁴ Entorno de Desarrollo Integrado.

- Disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP o Vista), Linux, Mac OS X, Solaris o Java.
- Posibilita la generación de código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir del código.
- Posibilita documentar todo el trabajo sin tener que hacer uso de herramientas externas.

Selección de la Herramienta CASE

Anteriormente se analizaron algunas de las herramientas CASE existentes en el mundo para el desarrollo de sistemas de software, y teniendo en cuenta que se desea desarrollar un trabajo bajo las políticas de software libre, se selecciona como herramienta CASE a Visual Paradigm, aunque esta herramienta no es libre, pero la universidad posee su licencia para modelar proyectos, además, que ésta soporta el ciclo completo de vida del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Es fácil de instalar y actualizar y es compatible entre ediciones.

1.5 Lenguajes de Programación Web

En los momentos actuales debido al gran desarrollo que ha alcanzado la informática en el mundo y a las tendencias y necesidades de las plataformas de desarrollo, han surgido diferentes lenguajes de programación para el desarrollo de aplicaciones web.

Aparejado al inicio de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. Con el paso del tiempo fueron surgiendo nuevos problemas al respecto, a los cuales había que darles solución, por lo que surgen los lenguajes de programación web dinámicos, que permitirían interactuar con los usuarios, utilizando sistemas de Bases de Datos.

1.5.1 Lenguajes del lado del servidor

PHP

PHP (Hypertext Pre-processor) es un lenguaje de programación utilizado para crear sitios web. Surge en 1995 y fue desarrollado por PHP Group. Es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en

páginas HTML y ejecutadas en el servidor. Para su funcionamiento se necesita tener instalado el Apache o IIS⁵ con las librerías de PHP [10].

Ventajas de PHP:

- Es un lenguaje fácil de aprender y muy rápido.
- Puede soportar la orientación a objetos, es decir, clases y herencia.
- Es multiplataforma: Linux, Windows, etc.
- Se puede conectar con la mayoría de los gestores de bases de datos como MySQL, PostgreSQL, Oracle, MS SQL Server, entre otros.
- Es libre lo cual lo convierte en una de las mejores alternativas de fácil acceso para todos.
- Incluye gran cantidad de funciones.

Desventajas de PHP:

- Se necesita instalar un servidor web para su uso.
- La legibilidad del código en ocasiones se ve afectada al mezclar HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.

Python

Python es uno de los lenguajes más populares para el desarrollo de aplicaciones de escritorio para GNU/Linux, aunque también es usado para la programación web por importantes empresas como Google. Su principal arma es que es mejorado por una gran comunidad alrededor del mundo, quienes le han añadido soporte para hacer de todo, desde la computación del entretenimiento hasta la computación científica. La principal desventaja de Python es el rendimiento de las aplicaciones, Python es lento en tiempo de ejecución porque es interpretado en vez de compilado. Es por eso que muchos programadores siempre se han cuestionado la viabilidad de programar en Python cuando se esfuerzan en construir

⁵ **Internet Information Services** o **IIS** es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows.

aplicaciones de alto rendimiento. Entre los principales IDEs con soporte para Python se pueden mencionar el Eclipse y Geany [10].

Ventajas:

- Lenguaje de propósito general.
- Gran cantidad de funciones y librerías.
- Multiplataforma y licencia de código abierto (Opensource).

Desventaja:

- Su principal desventaja es que se vuelve lento por ser un lenguaje interpretado.

Practical Extracting and Reporting Language (Perl)

Es un lenguaje gratuito que actualmente está disponible para sistemas operativos de tipo Unix y otros como Windows. Es un lenguaje muy práctico, pues con su uso se hace fácil la manipulación de información de archivos de texto y administración de sistemas, programación en red, y desarrollo de GUI (Interfaz Gráfica de Usuarios). Es eficiente y completo, está enfocado hacia un desarrollador con conocimientos sobre el lenguaje y es ideal para desarrollo rápido de aplicaciones. Soporta tanto la programación estructurada como la programación orientada a objetos y la programación funcional. Una de sus grandes ventajas es la amplia comunidad de usuarios donde se hace sencillo plantear y resolver las dudas que surjan en el desarrollo de una aplicación [10]; aunque también presenta algunas **desventajas**:

- Presenta complejo aprendizaje sobre su uso.
- Y al igual que Python es un lenguaje que se vuelve lento por ser interpretado.

Selección de la propuesta del lenguaje del lado del servidor

Luego de analizar algunos de los lenguajes de programación del lado del servidor, de ver de cada uno de ellos sus características, ventajas y desventajas que proporcionan su uso, se ha determinado usar PHP pues es un lenguaje fácil de aprender, es recomendable para el desarrollo de aplicaciones web con tiempo de duración corto, posee documentación en páginas oficiales en las que se incluyen descripciones y ejemplos de cada una de sus funcionalidades. Además, y algo muy importante para el desarrollo de este trabajo es que es multiplataforma lo que lo convierte en una alternativa de fácil acceso para todos.

1.5.2 Lenguajes del lado del cliente

Los lenguajes de programación del lado del cliente que se analizan a continuación, han sido los seleccionados para el desarrollo de este trabajo, es decir, se hará uso de los tres, debido a las características que se exponen a continuación.

JavaScript

Es un lenguaje de tipo script compacto, basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones con la tecnología web. Los programas JavaScript van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear mensajes, crear animaciones, comprobar campos. Es similar al Java, aunque. La mayoría de los navegadores en sus últimas versiones interpretan código JavaScript. El código JavaScript puede ser integrado dentro de las páginas web y para evitar incompatibilidades el World Wide Web Consortium (W3C) diseñó un estándar denominado DOM (Document Object Model, en español Modelos de Objetos del Documento) [10].

Otras de sus ventajas son:

- Los script tienen capacidades limitadas, por razones de seguridad.
- El código JavaScript se ejecuta en el cliente.

XHTML (eXtensible Hyper Text Markup)

Lenguaje de Marcado de Hipertexto Extensible (XHTML) es una versión más actualizada de HTML, el cual surge con el objetivo de reemplazar a este último. XHTML está orientado al uso de un etiquetado correcto, exige una serie de requisitos básicos a cumplir en lo que a código se refiere, entre los requisitos básicos se puede mencionar una estructura coherente dentro del documento donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente, atributos de valores entrecomillados [10]

Otras de las características de XHTML son:

- Es simple, multiplataforma.
- Está diseñado para definir la estructura y apariencia básica de textos y documentos, permitiendo que estos se visualicen correctamente al margen del contexto y el navegador utilizado.

- Es fácil de aprender, además, se hace fácil actualizar los contenidos utilizando estilos en formato CSS (Cascading Style Sheets).

CSS

CSS (**Cascading Style Sheets**) en español Hojas de Estilo en Cascada. Es lenguaje de estilos creada para controlar la forma de presentación de los documentos definidos con HTML y XHTML. Define el estilo visual de un sitio entero sin necesidad de hacerlo etiqueta por etiqueta, de manera que un cambio, por sencillo que éste sea, a un estilo CSS, se reflejará automáticamente en todos los elementos que sean referidos a éste. De esta forma se reduce la complejidad de su mantenimiento, evita que los archivos sean demasiado pesados y permite visualizar el mismo documento en infinidad de dispositivos diferentes (ordenadores, PDAs⁶, teléfonos móviles, entre otros) [10].

1.6 Sistemas Gestores de Base de Datos (SGBD)

Los SGBD son un tipo de software muy específico, los cuales sirven de interfaz entre las bases de datos y las aplicaciones que la utilizan. Su principal propósito se centra en manejar de forma sencilla, clara y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización; además, estos se relacionan con la creación y mantenimiento de la base de datos, el control de acceso, la manipulación de datos de acuerdo con las necesidades del usuario, así como evitar redundancias e inconsistencias y mantener la integridad [11].

Características fundamentales de los SGBD [11]:

- **Abstracción de la información:** los detalles de almacenamiento son transparentes al usuario.
- **Redundancia mínima:** evitar la aparición de información repetida o redundante.
- **Consistencia:** cuando no se ha logrado obtener una redundancia nula, es necesario tener en cuenta que aquella información que aparece repetida se actualice de forma coherente.

⁶ PDAs (del inglés personal digital assistant), también conocido como ordenador de bolsillo, originalmente diseñado como agenda electrónica.

- **Seguridad:** la información almacenada puede tener un gran valor, por lo que se deben garantizar que esta se encuentre segura frente a usuarios con intenciones de tergiversar el contenido mostrado.
- **Integridad:** proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra ocurrencia capaz de dañar la información almacenada.

1.6.1 MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Constituye el servidor de bases de datos relacionales más popular y es desarrollado por la empresa MySQL AB⁷, la cual desde enero del 2008 pertenece a Sun Microsystems⁸. MySQL es desarrollado como software libre siguiendo un esquema de licenciamiento dual, o sea, se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia. Sin embargo si no se está de acuerdo con la misma o se necesita incorporar MySQL en una aplicación comercial es posible comprar una licencia específica que permita éste uso. Contrario a otros proyectos de software libre, como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código [11].

1.6.2 PostgreSQL

Publicado bajo la licencia BSD⁹, proporciona un gran número de características que normalmente sólo se encontraban en los SGBD comerciales de alto calibre tales como Oracle. Es un SGBD objeto-relacional, ya que aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Su avanzada funcionalidad se pone de manifiesto con las consultas SQL declarativas, el control de concurrencia multiversión, soporte multiusuario, transacciones, optimización de consultas, herencia y valores no atómicos (atributos basados en vectores y conjuntos). Sirve de soporte a los lenguajes más populares como PHP, C, C++, Java, Python, Ruby, entre otros, además, de soportar un número ilimitado de bases de datos [11].

⁷ **MySQL AB** de las más grandes compañías de software libre del mundo.

⁸ **Sun Microsystems** es una empresa informática de Silicon Valley, fabricante de semiconductores y software.

⁹ La licencia **BSD** es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Pertenece al grupo de licencias de software Libre.

1.6.3 Oracle

Conocido como Relational Data Base Management System por su acrónimo en inglés, es un sistema de bases de datos considerado como uno de los más completos por su estabilidad, soporte de transiciones, escalabilidad y ser multiplataforma, sin embargo es privativo y su precio es considerablemente costoso. Otro aspecto que ha sido motivo de crítica para este sistema gestor de bases de datos es la seguridad de la plataforma, y las políticas de suministros de parches de seguridad, modificadas a comienzos del 2005 y que incrementan el nivel de exposición de los usuarios. Oracle es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo sea utilizado en empresas multinacionales y grandes, por norma general [12].

1.6.4 Microsoft SQL Server 2000

Es un sistema gestor de bases de datos producido por Microsoft basado en el modelo relacional, constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o PostgreSQL o MySQL, es totalmente habilitado para la web, exhibiendo sellos de referencia en cuanto a escalabilidad, confiabilidad, estabilidad y seguridad; incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente. Se puede decir también, que este gestor permite lograr una gran velocidad en el procesamiento de transacciones, y agilidad en todas sus operaciones [12].

Selección del Sistema Gestor de Bases de Datos

Los SGBD analizados anteriormente cuentan con un magnífico soporte para la web, MySQL se ve favorecido cuando el factor velocidad es primordial, Oracle es uno de los más completos por su estabilidad, y Microsoft SQL Server también, permite lograr una gran velocidad en el procesamiento de transacciones, pero sin embargo se selecciona PostgreSQL como SGBD para el desarrollo del presente trabajo ya que con el uso de éste se garantiza la estabilidad, integridad y seguridad del sistema, además, es válido mencionar que PostgreSQL ante otros SGBD cuenta con las siguientes características:

- Gran escalabilidad: es ajustable al número de procesadores y a la cantidad de memoria que posee el sistema de forma eficiente, por tal motivo es capaz de soportar una mayor cantidad de peticiones simultáneas.
- Tiene la capacidad de almacenar procedimientos en la propia base de datos.
- Multiusuario, con arquitectura cliente- servidor y control de privilegios de acceso.

1.7 Servidores Web

Servidor Web se refiere a un programa que muestra datos a usuarios en forma de páginas Web, hipertextos o páginas XHTML mediante la implementación del protocolo HTTP (Hypertext Transfer Protocol). De este modo, un servidor Web se mantiene a la espera de peticiones HTTP, que son ejecutadas por el cliente HTTP; lo que se conoce como navegador Web. El navegador realiza una petición al servidor y éste da como respuesta lo que ha solicitado el cliente.

1.7.1 HTTP Apache

Es un servidor de páginas Web, de software libre, que posibilita el acceso a las páginas Web alojadas en un ordenador. Su diseño le permite ser un servidor Web potente y flexible, funcional en una amplia gama de plataformas y entornos. Precisamente las mencionadas cualidades lo han convertido en uno de los primeros en las preferencias de los desarrolladores, logrando así ser uno de los servidores Web más utilizados en el mundo [11].

El servidor HTTP Apache se ubica, hoy en día, por encima de sus competidores, tanto libres como comerciales. El diseño modular que lo caracteriza les permite a los administradores de sitios Web elegir que funcionalidades serán incluidas en el servidor seleccionando que módulos se van a usar. Los módulos de Apache se encuentran clasificados en tres categorías:

- **Base:** Módulo con las funciones básicas del Apache.
- **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.
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales están definidas dentro del módulo base mientras que el módulo multiproceso se hace necesario para manejar las peticiones. Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

Las funcionalidades restantes del servidor se satisfacen por medio de módulos adicionales. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

Este servidor tiene capacidad para servir tanto páginas estáticas como dinámicas a través de otras herramientas soportadas que facilitan la actualización de los contenidos usando bases de datos, ficheros u otras fuentes de información.

1.7.2 ZOPE

Es un servidor de aplicaciones Web, de software libre, desarrollado fundamentalmente en Python. Entre las características que lo distinguen se encuentra su base de datos orientada a objetos denominada ZODB (Zope Object Database), la cual almacena objetos ordenados en un sistema similar a un sistema de ficheros, pero cada objeto tiene propiedades, métodos u otros objetos. Esta aproximación difiere bastante de las bases de datos relacionales actuales. No obstante, Zope posee la ventaja de contar con múltiples conectores para las diferentes bases de datos relacionales ofreciendo sistemas básicos de conexión y consulta abstrayéndolos como objetos [12].

1.7.3 Internet Information Server (IIS)

El servidor web de Microsoft, el IIS (Internet Information Server), es el motor que ofrece esta compañía a modo profesional, con él es posible programar en ASP (Active Server Pages, Páginas de Servidor Activo) las cuales vienen a ser algo similares al PHP. Este servidor posee componentes programables desde ASP accediendo a cada uno de sus módulos para una función específica [13].

Selección de la propuesta del Servidor Web

Teniendo en cuenta las características antes analizadas de algunos servidores web, se puede decir que son muy buenas opciones para el desarrollo Web. Sin embargo se selecciona HTTP Apache ya que éste se ve favorecido por su estructuración en módulos, tiene gran compatibilidad con el lenguaje de programación PHP y es muy utilizado tanto en el mundo como en la UCI, además, cuenta con características específicas que lo hacen diferente a los demás como son:

- Modular.
- Multiplataforma.
- Software Libre.
- Extensible.
- Popular.

- Gratuito.

1.8 Frameworks

Un framework simplifica el desarrollo de aplicaciones mediante la automatización de algunos de los patrones más utilizados. Además, brinda una estructura al código fuente, obligando a los desarrolladores a escribir código legible, limpio y fácil de mantener. Por otra parte un framework facilita la programación de las aplicaciones debido a que encapsula operaciones complejas en instrucciones sencillas [14].

Siempre se ha argumentado que PHP genera código difícil de mantener porque al crecer la aplicación el código se vuelve más complicado, confuso, difícil de seguir y entender. Sin embargo, una aplicación programada con la utilización de un framework para PHP no presenta estos problemas dado que aprovecha al máximo las ventajas de PHP mientras que, de forma paralela, facilita las herramientas para reducir los problemas que presentan las aplicaciones desarrolladas en este lenguaje.

1.8.1 Symfony

Symfony es un framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones Web. Está desarrollado con PHP5, por lo cual es completamente orientado a objetos. Ha sido probado en numerosos proyectos reales y se utiliza en sitios Web de comercio electrónico de primer nivel. Vale destacar que Symfony es compatible con la mayoría de los sistemas gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Además, implementa la mayoría de las mejores prácticas y patrones de diseño para la programación Web. [14]

Principales características [14]:

- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la Web.
- Código fácil de leer que incluye comentarios de phpDocumentor y permite un mantenimiento muy sencillo.

- Fácil de extender, lo que permite su integración con las librerías de otros fabricantes.

1.8.2 Zend Framework

Es un framework de código abierto para aplicación Web, orientada a objetos, desarrollado en PHP 5 y registrado bajo la nueva licencia BSD. Zend Framework (a menudo denominado ZF) se desarrolla con el objetivo de simplificar el desarrollo Web y al mismo tiempo promover las mejores prácticas en la comunidad de desarrolladores de PHP [15].

Una de sus principales ventajas es que fue lanzado por la casa Zend, impulsora de la tecnología de servidor PHP, por lo cual se complementa de manera excelente con los entornos de desarrollo creados por la misma, sin embargo a su vez esta opción se convierte en una desventaja, pues dichos entornos no clasifican dentro de las herramientas de software libre.

1.8.3 JQuery

Es una biblioteca rápida y concisa de JavaScript que simplifica el documento HTML, manejo de eventos, animación y las interacciones AJAX para el desarrollo de una aplicación web en poco tiempo. JQuery está diseñado para cambiar la forma en que se escribe JavaScript. [15]

La librería jQuery en síntesis ofrece las siguientes ventajas:

- Ahorro de líneas de código.
- Es soportada por cualquiera de los principales navegadores web como Internet Explorer, Firefox, Opera, Safari y Chrome.
- Provee un mecanismo para la captura de eventos.
- Soporta muchos selectores, tanto CSS como para la manipulación del Document Object Model (DOM).
- Proporciona un conjunto de funciones para animar el contenido de la página.
- Posee una amplia comunidad creadora de componentes o plugins, lo que proporciona una manera más fácil de encontrar soluciones ya creadas para desarrollar elementos de interfaz de usuario, galerías de imágenes, validadores entre otras cosas.

Selección del Framework PHP

Tanto Symfony como Zend Framework son frameworks PHP5 con gran éxito entre los desarrolladores. Dos de las ventajas más importantes con las que cuenta Symfony son la

calidad de su código fuente y la gran cantidad de documentación disponible. Además, Symfony es una herramienta ideal para el desarrollo de aplicaciones rápidas y se adapta fácilmente a los cambios que puedan surgir durante el desarrollo de software. Tiene también a su favor que en la UCI varias facultades cuentan con experiencia en el desarrollo de aplicaciones Web mediante su uso. Por tanto, se selecciona Symfony como framework PHP y JQuery como framework para JavaScript pues este último cuenta además, con las siguientes características: integra funcionalidades para trabajar con Asynchronous JavaScript And XML (AJAX), es estable, contiene amplia documentación y tiene gran aceptación por parte de los desarrolladores ya que es una librería basada en JavaScript, del cual se tiene un conocimiento amplio.

1.9 Entorno de Desarrollo Integrado

Un Ambiente Integrado de Desarrollo (IDE, sus siglas significan Integrated Development Environment) es un entorno de programación que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica [15].

1.9.1 Eclipse.

Eclipse es una plataforma de software de código abierto independiente y que es y ha sido muy utilizada para crear entornos integrados de desarrollo (IDE), pero del mismo modo se puede usar para otros tipos de aplicaciones cliente [15].

Características fundamentales:

- Tiene una gran comunidad de usuarios extendiendo constantemente las aplicaciones.
- Editor sensible al contexto, el cual provee resaltamiento de errores y autocompletado de código.
- Permite inspeccionar el uso de las vistas del contorno del fichero y del proyecto, así como la nueva vista PHP Explorer.
- Soporte para el debug incremental del código de PHP.

- Extensos frameworks y APIs¹⁰ que permiten a los desarrolladores e ISVs (vendedores de software independientes) fácilmente extender PDT¹¹ para crear nuevas e interesantes herramientas orientadas al desarrollo de PHP [16].

1.9.2 NetBeans

NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. El NetBeans es un IDE de código abierto escrito completamente en Java. Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web y aplicaciones móviles). También tiene soporte para varios lenguajes como PHP, JavaScript, HTML, CSS y en su versión 6.8 tienen soporte para el framework Symfony [16].

Otras características de NetBeans son:

- Incluye CVS (control de versiones) y Ant (compilación avanzada).
- Posibilidad de usar otras versiones de compiladores y depuradores.
- Creación visual de componentes gráficos.

Selección del Entorno de Desarrollo Integrado

Luego de analizados dos de los IDE, y teniendo en cuenta sus características, se selecciona para el desarrollo de este trabajo a NetBeans ya que además de las características antes mencionadas se puede decir también, que es un producto libre y gratuito sin restricciones de uso.

Conclusiones.

En este capítulo se definieron una serie de conceptos importantes relacionados con el tema investigativo, los cuales servirán de soporte al lector para adentrarse en el tema en cuestión. Se abordaron los principios, características, ventajas y desventajas de las metodologías, herramientas y lenguajes de modelado y de programación con el propósito de hacer la mejor selección para el desarrollo del sistema propuesto, justificando cada elección realizada.

¹⁰ APIs (del inglés Application Programming Interface) y su acrónimo en español interfaz de programación de aplicaciones; es un conjunto de funciones y procedimientos en la programación orientada a objeto, el cual ofrece bibliotecas para ser usado por otros software como capa de abstracción.

¹¹ PDT (PHP Development Tools), es un plugin de desarrollo de PHP para el IDE Eclipse y para proyectos de código abierto que se desarrollan.

Capítulo 2

2. Introducción

En el siguiente capítulo se abordan las características que tendrá el sistema, para ello se ha realizado un estudio sobre todos los servicios y funcionalidades que presentará el Portal Web de Béisbol de la Universidad de las Ciencias Informáticas en su versión 2.0, los objetivos y procesos que lo soportarán. Se realiza la descripción de los procesos a automatizar. Se determina los actores, trabajadores y se obtiene los requisitos funcionales y no funcionales con los que debe cumplir el sistema, así como el diagrama de casos de uso del sistema.

2.1 Objeto de automatización.

Los principales procesos que serán objeto de automatización son:

- La publicación de noticias, efemérides, comentarios deportivos y comentarios realizados por los usuarios, y además, hechos relevantes, todos estos con previa aprobación.
- La gestión de usuarios, que recogerá información del usuario registrado en el portal; es decir, sus datos personales, como: usuario, contraseña, nombre, apellidos, etc.
- La gestión de jugadores, que recogerá información de cada uno de los jugadores ya sean retirados o activos, es decir, su nombre, edad, talla, peso, provincia a la que pertenece, equipo en el que juega, etc.
- La gestión de juegos, que recogerá información de cada juego que se haya realizado o que se esté realizando, lugar, hora, equipos que se enfrentan, etc.

2.2 Información que se maneja.

En el portal web de béisbol de la UCI en su versión 2.0 se manejará información acerca de lo que acontece en el béisbol en el país, tales como:

- Información de todos los jugadores, ya sean retirados o activos.
- Información de los juegos realizados y los que se estén realizando en dependencia este último elemento del momento en que se visite.
- Noticias de lo que acontece en cuanto respecta al béisbol, los jugadores, equipos, jugadas, momentos importantes, etc.

- Publicación de efemérides.
- Publicación de avisos.
- Publicación de juegos que se han realizado o que están por realizarse, a través del calendario de juego.
- Estadísticas de los jugadores, en cuanto a todo lo que realiza un jugador en cada juego, ya sea bateo, pitcheo, entre otros.

2.3 Propuesta del sistema.

El sistema que se desarrollará debe ser accesible por todos los usuarios que naveguen en él, por lo que los administradores del portal tendrán como objetivos principales garantizar que el mismo funcione lo mejor posible, brindándole a los usuarios los hechos más relevantes durante un período determinado; los datos de los peloteros, los equipos y los juegos tendrán que estar lo más actualizado posible.

Además, el sistema estará centrado en realizar los siguientes procesos: publicar noticias, efemérides, buscar información contenida en el portal, así como mostrar una estadística del portal orientada a la cantidad de usuarios conectados y usuarios registrados; aunque también mostrará estadísticas de los juegos realizados en una fecha dada, siempre mostrará la de los últimos juegos, pero permitirá al usuario seleccionar el día del cual desea ver la información.

También los usuarios registrados en él, podrán realizar comentarios con respecto a las noticias que se publiquen.

Se podrán realizar búsquedas de cualquier información que respecte al acontecer beisbolero, tanto de peloteros como de jugadas.

2.4 Modelo del Dominio

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del sistema representan las cosas que existen o los eventos que se suceden en el entorno en el que trabaja el sistema [17].

Luego de realizar un estudio de los procesos que se van a efectuar en el sistema, se propone un modelo de dominio, pues el de negocio presenta un bajo nivel de estructuración, y con el de dominio se ayuda a los usuarios, clientes, desarrolladores y demás personal vinculado con el

sistema a utilizar un vocabulario común para poder entender en contexto en que se desarrolla la aplicación.

Para que la captura de requisitos se haga de la mejor forma posible y el sistema cumpla con las expectativas del usuario, es preciso tener claro cómo debe funcionar el proceso en cuestión. El modelo de dominio va a contribuir con la identificación de clases que posteriormente serán utilizadas en el sistema, este modelo es descrito mediante diagramas UML, específicamente con un diagrama de clases conceptuales significativas en el dominio del problema.

A continuación, se muestran una serie de conceptos que son muy importantes para el entendimiento del modelado:

- **Portal web de Béisbol:** Es un sitio web cuya característica fundamental es servir de puerta de entrada a los usuarios para ofrecerle una serie de servicios y temas de forma fácil e integrada sobre los juegos de béisbol que se realizan en las series nacionales.
- **Módulo Prensa:** Departamento donde se gestiona la noticia, las efemérides, los hechos relevantes y las encuestas, por el periodista o el administrador del sistema.
- **Módulo Nomencladores:** Departamento donde se gestiona las etapas, los tipos de eventos, y los estadios, donde se juegan los encuentros deportivos, por el estadístico o el administrador del sistema.
- **Módulo Seguridad:** Departamento donde se gestiona los usuarios, permisos, roles y las combinaciones entre ellos, por el administrador del sistema.
- **Módulo Estadísticas:** Departamento donde se gestiona el calendario de juego, jugadores, equipos, eventos y juegos, por el estadístico y el administrador del sistema.
- **Administrador:** Es la persona que administra el portal, y tiene privilegios ilimitados para realizar cualquier operación en el sistema.
- **Estadístico:** Es la persona que se encarga de gestionar los jugadores, equipos, calendario de Juego, y otros elementos para llevar el control de lo que sucede en una serie nacional de béisbol.
- **Periodista:** Es la persona encargada de gestionar las noticias, los hechos relevantes, las efemérides y las encuestas que se publican en el sistema.

- **Efemérides:** Acontecimientos históricos ocurridos en el día en curso, los cuales se mostrarán a través de textos breves.
- **Noticias:** Eventualidades o actividades ocurridas en el día en curso o en días cercanos.
- **Hechos relevantes:** Acontecimiento o acontecimientos ocurridos durante un juego o que esté relacionado con el béisbol, el cual será de relevante importancia.
- **Encuesta:** Preguntas que se realizan para medir el conocimiento de los usuarios acerca de béisbol.
- **Calendario:** Listado de los juegos que se planifican para un evento.
- **Jugadores:** Se muestra una ficha técnica de cada jugador en la que se incluyen: datos personales, breve reseña y datos estadísticos por jornada y series en las que ha participado.
- **Eventos:** Es en lo que participan los distintos equipos, por ejemplo la serie nacional de béisbol.
- **Equipos:** Conjunto de jugadores.
- **Juegos:** Los enfrentamientos entre equipos.
- **Usuario:** Persona que interactúa con el sistema.
- **Roles:** Es el papel que desempeña cada tipo de usuario que interactúa con el sistema.
- **Permisos:** Son privilegios que da el administrador a los usuarios en dependencia del rol que este tenga en el sistema.
- **Etapas:** Son los momentos por los que pasa una serie nacional desde que empieza hasta que termina.
- **Estadio:** Lugar donde se realizan los juegos.
- **Tipo de Evento:** Es en lo que se clasifican los eventos.

2.4.1 Diagrama conceptual de Modelo de Dominio.

Para facilitar la comprensión de este trabajo se incluyó el diagrama conceptual de Modelo de Dominio en los Anexos. [Ver Anexo 4](#)

2.5 Especificación de los requisitos del software.

El flujo de especificación de los requisitos del software radica su mayor esfuerzo en el reconocimiento del problema como lo ve el usuario. Se define una evaluación del problema y se darán respuestas a las funcionalidades que tendrá el sistema. Con él se pretende entender el comportamiento de restricciones adicionales de diseño [12].

Estos reflejan las ideas que tienen los usuarios y equipo de proyecto sobre lo que el sistema debe hacer. Además, se clasifican por funcionales y no funcionales.

2.5.1 Requisitos Funcionales (RF).

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, se mantienen invariables. Son el punto de partida para identificar qué debe hacer el sistema [20].

A continuación, se relacionan algunos de los requisitos funcionales con los que debe cumplir el portal web de béisbol en su versión 2.0, en esta parte solo se especifican las funcionalidades, para un mejor entendimiento se recomienda ver los anexos, donde se exponen con sus respectivos atributos. [Ver Anexo 5](#)

Requisitos Funcionales:

RF1 Dar permisos a usuarios.

RF2 Listar Noticia.

RF3 Insertar Noticia.

RF4 Modificar Noticia.

RF5 Eliminar Noticia.

RF6 Mostrar Noticia.

RF7 Listar Hechos Relevante.

RF8 Insertar Hechos Relevantes.

RF9 Modificar Hechos Relevantes.

RF10 Eliminar Hechos Relevantes.

RF11 Listar Efemérides.

RF12 Insertar Efemérides.

RF13 Modificar Efemérides.

RF14 Eliminar Efemérides.

RF15 Listar Competencia.

RF16 Insertar Competencia.

RF17 Modificar Competencia.

RF18 Eliminar Competencia.

RF67 Registrarse.

RF68 Autenticar.

RF69 Actualizar Perfil de Usuario.

RF71 Mostrar Noticias Ampliada.

RF72 Realizar Comentario a Noticias.

RF82 Mostrar Comparación Virtual de Lanzadores.

RF83 Mostrar Rendimiento Individual por Equipo.

RF98 Actualizar Estadísticas de por Vida de los Jugadores.

2.5.2 Requisitos No Funcionales (RNF).

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener [20]. Estos no son más que las características que debe tener la aplicación para que sea del agrado de los usuarios, sea usable, rápido y confiable, están además, vinculados con los requerimientos funcionales del sistema y son fundamentales para el éxito del producto final.

RNF1 Apariencia e Interfaz externa

- El sistema debe contar con una interfaz amigable que le permita al usuario interactuar de forma cómoda con la misma, le agilice y facilite el trabajo con el software.
- El software debe interactuar con el usuario mediante una interfaz usable y funcional.

RNF2 Usabilidad

- El sistema debe contar con un ambiente sencillo y fácil de aprender principalmente para el uso de personas con pocos conocimientos de informática.

RNF3 Disponibilidad

- El software deberá estar disponible todo el día y la semana, además, debe garantizar un acceso de forma fácil y rápida para los usuarios.

RNF4 Soporte

- El software será actualizado por los administradores, de forma sistemática, garantizando además, su mantenimiento.

RNF5 Rendimiento

- El sistema deberá garantizar la rapidez de las respuestas a las solicitudes de los usuarios en el menor tiempo posible, es decir, en segundos.

RNF6 Portabilidad

- El software será multiplataforma.

RNF7 Seguridad Integridad

- El sistema contará con protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Se deben realizar copias periódicamente de la Base de datos, como respaldo de la información.
- Usar el protocolo seguro https para garantizar que los datos viajen de manera segura por la red.

RNF8 Confidencialidad

- Proteger la información que se maneje en el sistema para evitar accesos no autorizados, implementando mecanismos de autenticación por roles.
- Solicitar a los usuarios registrarse para interactuar con el sistema estableciendo los diferentes roles de usuario, con lo cual se garantiza que la información que se maneja en el sistema sea vista solamente por quien tenga derecho a hacerlo.

RNF9 Confiabilidad

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

RNF10 Software

Para poder instalar la aplicación, se debe disponer de:

- **Lado del Cliente:**
 - Cualquier sistema operativo con interfaz gráfica o red.
 - Navegador Web estándar.
 - Sistema Operativo Linux o Windows.
- **Lado del Servidor:**
 - Servidor de BD PostgreSQL.
 - Servidor Web Apache.

- PHP 5 o superior como lenguaje de programación.

RNF11 Hardware

- El servidor deberá contar como mínimo con:
 - Pentium IV
 - Un microprocesador a 2.0 GHz
 - 1 GB de memoria RAM
 - 80 GB de disco duro
 - La velocidad de la red deberá ser rápida a 100 Mbps

RNF12 Legales

- El sistema contará con normas de estilos y pautas de diseño para cada información que brinde el mismo, además, se usarán herramientas de software libre bajo la licencia GNU/GPL.

2.6 Definición de los casos de uso.

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y respuestas, el comportamiento del sistema. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. Son considerados las actividades a automatizar [12].

2.6.1 Definición de los actores del sistema.

Las personas relacionadas con el sistema son todas las que obtienen un resultado de uno o varios procesos que se ejecutan en el mismo. En un sistema a estas personas se les conoce como actores, y estos no son más que un rol que un usuario juega con respecto al sistema. La palabra rol es importante usarla en estos casos, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema [8].

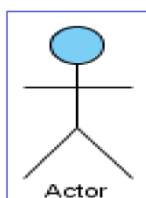


Figura 3. Actor

Tabla: Actores del sistema

Actor	Descripción
Invitado	Usuario que navega por el sistema sin haber creado aún una cuenta de usuario, navega dentro de éste sin privilegios de acceder a los recursos que se definen para los usuarios autenticados.
Usuario Registrado	Usuario que se encuentra registrado en el sistema, y debido al rol que se le asigna, tendrá acceso a diferentes funcionalidades, es decir, a todas a las que tiene acceso el usuario invitado, y otras más a las que su rol le permite acceder.
Periodista	Usuario encargado de gestionar las noticias, hechos relevantes, efemérides y encuestas, teniendo en cuenta los arreglos, la edición, las normas y estilos establecidos para efectuar dichas funcionalidades.
Estadístico	Usuario encargado de gestionar todos los datos estadísticos que se publican en el portal, es decir, gestiona los jugadores, juegos, equipos, eventos, calendario de juego, etapas, estadios, etc.
Administrador	Usuario que dispone de posibilidades ilimitadas para ejecutar todas las funciones administrativas dentro del sistema, como son: gestionar usuarios, permisos, juegos, jugadores, efemérides, encuestas, etc. Es decir, puede realizar las acciones que realizan el periodista y el estadístico.

2.6.2 Listado de los Casos de Uso del Sistema.

Caso de Uso: Es una operación o tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.



Figura 4. Casos de Uso

Tabla: Casos de Usos (CU) del sistema

No. CU	Nombre del CU	Actor	Referencia
CU-1	Gestionar Permisos de Usuario	Administrador	RF1
CU-2	Gestionar Noticia	Periodista y Administrador	RF2, RF3, RF4, RF5, RF6
CU-3	Gestionar Hechos Relevantes	Periodista y Administrador	RF7, RF8, RF9, RF10
CU-4	Gestionar Efemérides	Periodista y Administrador	RF11, RF12, RF13, RF14
CU-5	Gestionar Competencia	Estadístico y Administrador	RF15, RF16, RF17, RF18
CU-6	Gestionar Evento	Estadístico y Administrador	RF19, RF20, RF21, RF22
CU-7	Gestionar Equipo	Estadístico y Administrador	RF23, RF24, RF25, RF26
CU-8	Gestionar Jugador	Estadístico y Administrador	RF27, RF28, RF29, RF30
CU-9	Gestionar Directivo	Estadístico y Administrador	RF31, RF32, RF33, RF34

Características del Sistema

CU-10	Gestionar Equipo a Evento	Estadístico Administrador	y	RF35, RF36, RF37, RF38
CU-11	Gestionar Jugador a Equipo	Estadístico Administrador	y	RF39, RF40, RF41, RF42
CU-12	Gestionar Directivo a Equipo	Estadístico Administrador	y	RF43, RF44, RF45, RF46
CU-13	Gestionar Juego	Estadístico Administrador	y	RF47, RF48, RF49, RF50
CU-14	Gestionar Etapa	Estadístico Administrador	y	RF51, RF52, RF53, RF54
CU-15	Gestionar Tipo de Evento	Estadístico Administrador	y	RF55, RF56, RF57, RF58
CU-16	Gestionar Estadio	Estadístico Administrador	y	RF59, RF60, RF61, RF62
CU-17	Gestionar Encuesta	Periodista Administrador	y	RF63, RF64, RF65, RF66
CU-18	Registrarse	Invitado		RF67
CU-19	Autenticar	Usuario Registrado		RF68
CU-20	Actualizar Perfil de Usuario	Usuario Registrado		RF69
CU-21	Ver Resumen de Noticia	Invitado y Usuario Registrado		RF70
CU-22	Ver Noticia Ampliada	Usuario Registrado		RF71
CU-23	Realizar Comentario a Noticias	Usuario Registrado		RF72
CU-24	Ver Estadísticas	Invitado y Usuario		RF73, RF75, RF77, RF79, RF81, RF82,

		Registrado	RF83, RF84, RF85, RF86
CU-25	Ver Más Líderes	Usuario Registrado	RF74
CU-26	Ver Rendimiento Individual de Bateadores	Usuario Registrado	RF76
CU-27	Ver Rendimiento Individual de Lanzadores	Usuario Registrado	RF78
CU-28	Ver Rendimiento Colectivo por Equipos	Usuario Registrado	RF80
CU-29	Ver Calendario	Invitado y Usuario Registrado	RF87
CU-30	Buscar Jugador	Invitado y Usuario Registrado	RF88
CU-31	Ver Ficha de Jugador	Invitado y Usuario Registrado	RF89
CU-32	Ver Juegos Diarios	Invitado y Usuario Registrado	RF90
CU-33	Ver los Mejores Jugadores	Invitado y Usuario Registrado	RF91
CU-34	Ver Jugadores Todos Estrellas	Invitado y Usuario Registrado	RF92
CU-35	Ver Hechos Relevantes	Invitado y Usuario Registrado	RF93
CU-36	Ver Efemérides	Invitado y Usuario Registrado	RF94

CU-37	Realizar Encuesta	Usuario Registrado	RF95
CU-38	Actualizar Estadísticas de Jugadores	Estadístico y Administrador	RF96, RF97, RF98

2.6.3 Diagrama de Casos de Uso del sistema.

El diagrama de caso de uso representa la forma en como un cliente (actor) interactúa con el sistema en desarrollo, además, de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso). Estos diagramas constan de actores, casos de uso y las relaciones entre ellos [8].

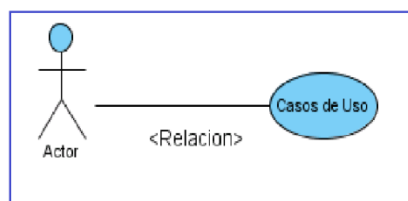


Figura 5. Relación entre actores y casos de uso.

Para un mejor entendimiento de la relación que existe entre los actores del sistema y la aplicación consulte el diagrama de casos de usos del sistema el cual lo puede ver en los anexos. [Ver Anexos 6](#)

2.6.4 Casos de Usos expandidos.

A continuación, solo se exponen el resumen de los casos de uso más significativos para el sistema, permitiendo así un mejor entendimiento acerca de los mismos.

Tabla: CU-2 Gestionar Noticia

CU-2	Gestionar Noticia
Actor	Periodista y Administrador
Resumen	El caso de uso se inicia cuando el periodista o el administrador acceden al sistema para gestionar una noticia. Aquí tienen la opción de listar, insertar, modificar, eliminar o ver una noticia publicada o que se desee publicar en el portal, para mantener

	informado a los usuarios que lo visiten. Fin del caso de uso.
Referencia	RF2, RF3, RF4, RF5, RF6
Prioridad	Crítica

Tabla: CU-4 Gestionar Efemérides

CU-4	Gestionar Efemérides
Actor	Periodista y Administrador
Resumen	El caso de uso se inicia cuando el periodista o el administrador acceden al sistema para gestionar una efeméride. Aquí tienen la opción de listar, insertar, modificar, o eliminar una efeméride publicada o que se desee publicar en el portal, para mantener informado a los usuarios que lo visiten. Fin del caso de uso.
Referencia	RF11, RF12, RF13, RF14
Prioridad	Crítica

Tabla: CU-6 Gestionar Evento

CU-6	Gestionar Evento
Actor	Estadístico y Administrador
Resumen	El caso de uso se inicia cuando el estadístico o el administrador acceden al sistema para gestionar un evento. Aquí tienen la opción de listar, insertar, modificar, o eliminar un evento publicado o que se desee publicar en el portal, para mantener informado a los usuarios que lo visiten. Fin del caso de uso.

Referencia	RF19, RF20,RF21, RF22
Prioridad	Crítica

Tabla: CU-17 Gestionar Encuesta

CU-17	Gestionar Encuesta
Actor	Estadístico y Administrador
Resumen	El caso de uso se inicia cuando el estadístico o el administrador acceden al sistema para gestionar una encuesta donde se comprueban los conocimientos que tienen los usuarios sobre los juegos de béisbol. Aquí tienen la opción de listar, insertar, modificar, o eliminar una encuesta que ya esté o que se desee incluir en el portal. Fin del caso de uso.
Referencia	RF63, RF64, RF65, RF66
Prioridad	Crítica

Tabla: CU-38 Actualizar Estadísticas de Jugadores

CU-38	Actualizar Estadísticas de Jugadores
Actor	Estadístico y Administrador
Resumen	El caso de uso se inicia cuando el estadístico o el administrador acceden al sistema para actualizar los datos estadísticos de un jugador luego de este haber participado en algún juego, es decir, a medida que se efectúe un juego el estadístico o el administrador

	del sistema deben actualizar los datos de cada jugador según su actuación en el juego que concluyó y que haya participado, y seguidamente tendrá que actualizar las estadísticas del jugador en cuanto a las series en las que ha participado, y por consiguiente, de por vida. Por tanto, al finalizar cada juego se podrá ver las estadísticas de la actuación de un jugador en los juegos, las series y de por vida, finalizando así el caso de uso.
Referencia	RF96, RF97, RF98
Prioridad	Crítica

Las descripciones textuales de los casos de uso se pueden consultar en los nexos. [Ver Anexo 7](#)

Conclusiones

En este capítulo se profundizó en la propuesta de solución, obteniéndose como resultados en el mismo, una lista de requerimientos funcionales, los cuales indican las actividades fundamentales del sistema, y una de los requerimientos no funcionales, los que indican la calidad y eficiencia del software, además, de la primera lista de las antes mencionadas, se obtiene el Diagrama de Caso de Uso del Sistema, y la descripción de los Casos de Uso que componen dicho diagrama, quedando así constituida la base para el análisis y diseño del sistema además, es también la base para empezar a construir software.

Capítulo 3

3. Introducción

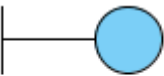

En este capítulo se comienza a construir la solución propuesta, la cual ha sido descrita en el capítulo anterior. Se proponen del flujo de trabajo análisis y diseño, los diagramas de clases de análisis, ya que esta es una fase opcional y además, en este trabajo se han identificado 38 casos de uso lo cual requiere mucho más tiempo del que se dispone para realizar este flujo completo, se obtiene también los diagramas de clases de diseño y el modelo de datos. Con las clases de diseño se presentará como quedará construido el sistema, el cual se realiza a partir de los Diagramas de Clases del Diseño, usando estereotipos web, los cuales permiten describir cómo será la interacción entre las distintas páginas de la aplicación. Finalmente, se obtiene el diagrama entidad relación con las descripciones pertinentes en cada caso.

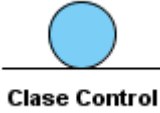
3.1 Modelo de análisis

Este modelo contiene las clases del análisis y sus objetos organizados en paquetes que colaboran [20].

Las clases del análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema, pues representan conceptos y relaciones del dominio. Estas tienen atributos y entre ellas se establecen relaciones de asociación, agregación/composición, generalización/especialización y tipos asociativos. La metodología RUP propone que las clases se clasifiquen y se representen de la siguiente forma:

Tabla: Clasificación y representación de las clases.

Nombre	Características	Representación
Interfaz	Modelan la interacción entre el sistema y sus actores.	 <p>Clase Interfaz</p>
Control	Coordinan los eventos necesarios para la realización o especificación del caso de uso, es decir, son las que ejecutan el caso	 <p>Clase Control</p>

	de uso.	
Entidad	Representan información que posee larga vida y que es a menudo persistente. Reflejan entidades del mundo real, las cuales son necesarias para realizar tareas internas del sistema.	

3.1.1 Diagramas de clases del análisis:

A continuación, se mostrarán algunos de los Diagramas de Clases del Análisis correspondiente a los casos de uso del sistema descritos en el capítulo anterior, los Diagramas de Clases del Análisis de los restantes casos de uso los puede consultar en los anexos. [Ver Anexo 8](#)

Al pie de cada imagen se usará la abreviatura DCA correspondiente a Diagrama de Clase del Análisis.

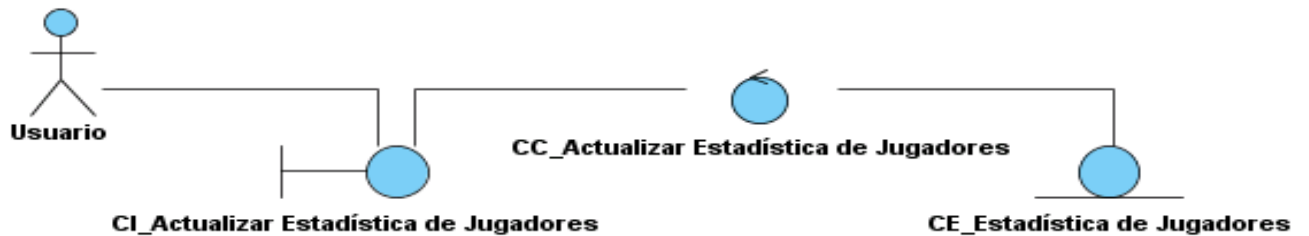


Figura 6. DCA Actualizar Estadísticas de Juegos



Figura 7. DCA Ver Calendario

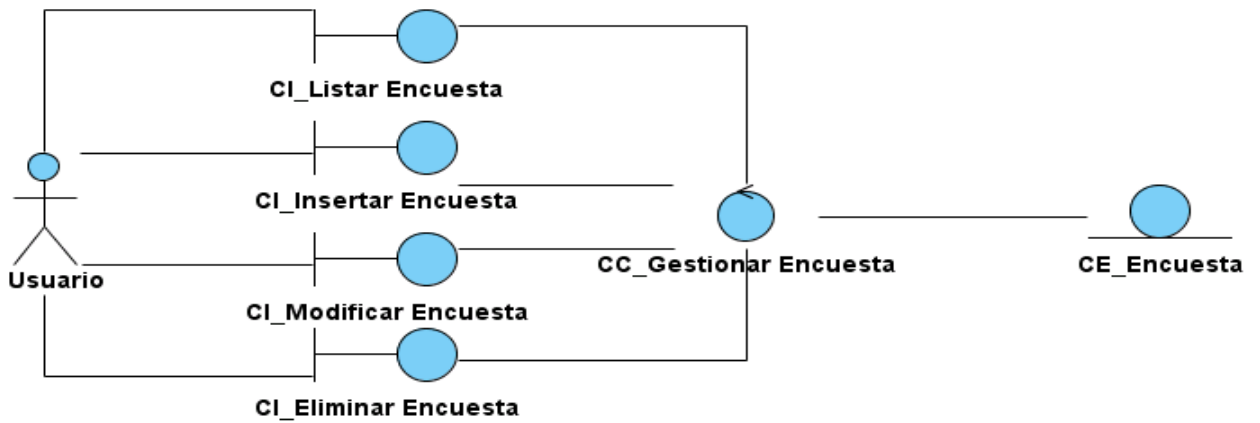


Figura 8. DCA Gestionar Encuesta

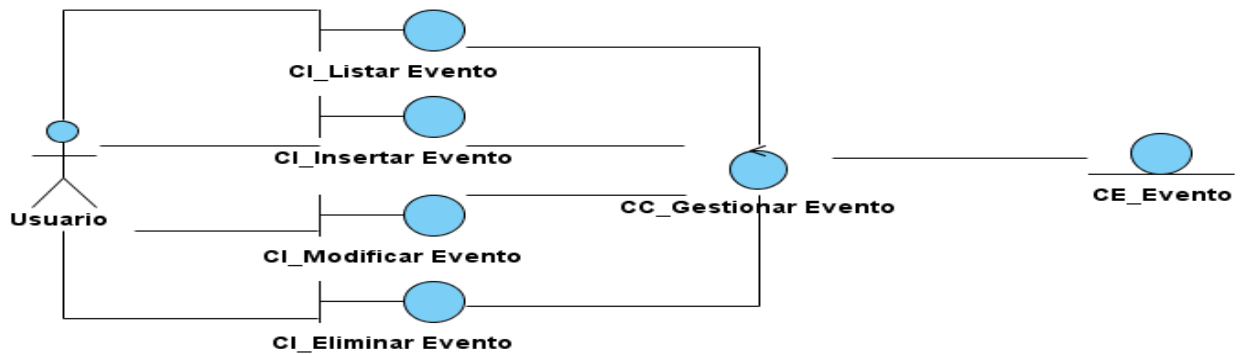


Figura 9. DCA Gestionar Evento

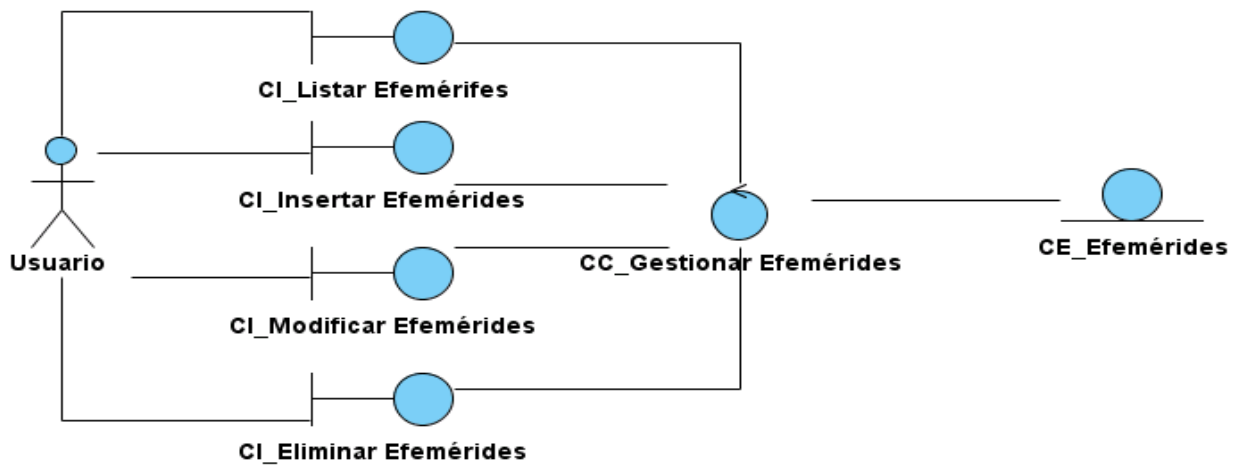


Figura 10. DCA Gestionar Efemérides

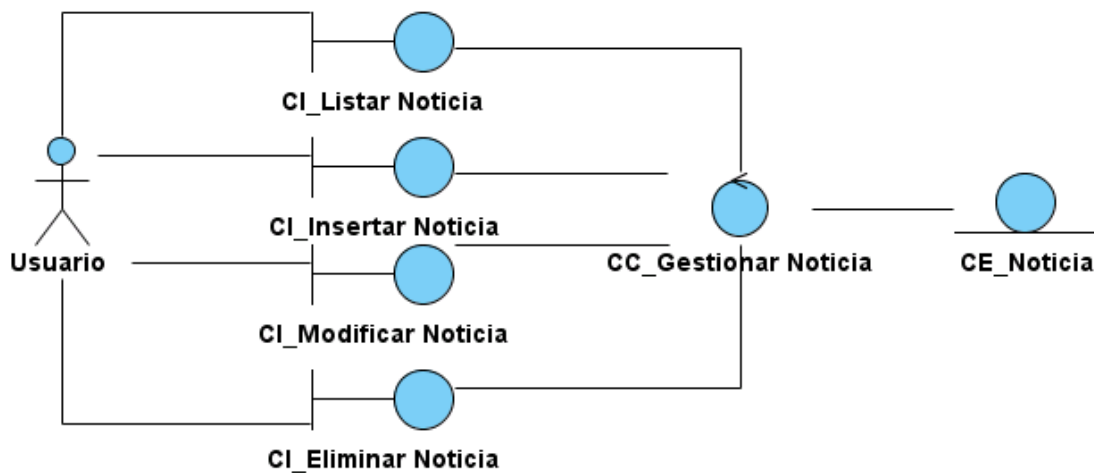


Figura 11. DCA Gestionar Noticia

3.2 Patrones Arquitectónicos

Los patrones arquitectónicos son patrones de software que solucionan problemas de arquitectura de software en ingeniería de software; éstos expresan un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. Uno de los aspectos más importantes de la aplicación de los patrones arquitectónicos es que permiten obtener productos de alta calidad; algunos representan soluciones a problemas de rendimiento y otros pueden ser utilizados en sistemas de alta disponibilidad [21].

Para el desarrollo de aplicaciones web utilizando Symfony se aplican varios patrones de diseño de software los cuales se sitúan en las capas de Modelo y Control que plantea el patrón arquitectónico Modelo Vista Controlador (MVC).

3.2.1 Patrón Modelo Vista Controlador

El patrón MVC, es uno de los patrones más usados para el desarrollo de aplicaciones web, éste es un patrón de diseño aportado originariamente por el lenguaje Small Talk a la Ingeniería de Software [21].

La arquitectura (MVC) separa la lógica del negocio de la presentación, posibilitando así que el mantenimiento a las aplicaciones sea de forma sencilla. El controlador es el encargado de separar el Modelo de la Vista.

El uso de un framework que utiliza MVC obliga a dividir y organizar el código según las normas establecidas por el mismo. El código de la presentación se guarda en la Vista, el código de manipulación de datos se guarda en el Modelo y la lógica de procesamiento de las peticiones constituye el Controlador.

Para que el desarrollo de aplicaciones web utilizando Symfony sea más rápido y sencillo, éste toma lo mejor de la arquitectura MVC, y para ello cuenta con tres niveles:

- El **Modelo**, que representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La **Vista**, esta transforma el modelo en una página Web que permite al usuario interactuar con ella.
- El **Controlador**, se encarga de procesar las interacciones del usuario y realizar los cambios apropiados en el modelo o en la vista.

Consultar el funcionamiento del patrón MVC en los anexos. [Ver Anexo 9.1.](#)

Ventajas del MVC

Este patrón tiene una gran ventaja ya que con su aplicación es posible separar la lógica del negocio y la presentación. Se le pueden aplicar opciones como el multilinguaje, distintos diseños de presentación, sin alterar la lógica del negocio [21].

La separación entre presentación, lógica del negocio y acceso a dato, que este patrón proporciona es muy importante para el desarrollo de arquitecturas consistentes, reutilizables y de fácil mantenimiento, lo que permite ahorrar tiempo en proyectos posteriores.

3.2.2 Patrones de Diseño

Los patrones de diseño o comúnmente conocidos como “Design Patterns”, son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Estos describen un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema [21].

Symfony hace uso de diferentes patrones de diseño para el desarrollo de aplicaciones web, entre los que podemos mencionar:

Patrones GRASP (General Responsibility Assignment Software Patterns) en español Patrones de Asignación General de Responsabilidades de Software.

- **Experto:** Es uno de los patrones más utilizados cuando se trabaja con Symfony, con la inclusión de la librería Propel¹² para mapear la Base de Datos. Symfony utiliza esta librería para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades [21].
- **Alta Cohesión:** A través de este patrón Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Lo antes dicho se afirma con la aplicación de la clase Actions, la cual está compuesta por varias funcionalidades que están estrechamente relacionadas, siendo esta la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades [21].

Se analizaron otros patrones GRASP, los cuales se pueden consultar en los anexos. [Ver Anexo 9.2](#)

Patrones GOF

- **Singleton (Instancia única):** La aplicación de este método se basa en el empleo de la clase sfRouting, la cual se utiliza en el controlador frontal, y es la encargada de enrutar todas las peticiones que se hagan a la aplicación. Con el uso de este patrón como tal se garantiza que una clase solo tenga una instancia, y que proporcione un punto de acceso global a ella [21].
- **Decorator (Envoltorio):** Este método se observa en la clase abstracta sfView, padre de todas las vistas, la cual contiene un decorador para permitir agregar funcionalidades dinámicamente. El archivo conocido como Layout.php es que contiene el Layout de la página. Este es un archivo que por sus características es conocido como plantilla global, pues guarda el código HTML que será común para todas las páginas de la aplicación, en otras palabras se puede afirmar que el Layout es el que decora la plantilla de la aplicación [21].

¹²Propel: Es un código abierto Mapeo Objeto-Relacional (ORM) para PHP5. Le permite acceder a su base de datos mediante un conjunto de objetos, proporcionando una API (interfaz de programación) sencilla para almacenar y recuperar datos.

3.3 Modelo de Diseño

Este es un modelo que describe como los casos de uso influyen en el sistema, es además, una abstracción de la implementación; es decir, mediante el Modelo de Diseño se puede especificar una solución funcional, que pueda posteriormente convertirse en código fuente.

3.3.1 Clases del Diseño

Para un mejor entendimiento de cómo funciona la relación en los Diagramas de Clases del Diseño, es preciso tener en cuenta que:

- Para distribuir y mostrar las responsabilidades de las clases en el patrón MVC, se usan paquetes.
- En el caso del subsistema Componentes Symfony, contiene todos los componentes (núcleo del framework, clases internas y ficheros) que proporciona Symfony.
- El subsistema Doctrine, en este caso representa todos los componentes que utiliza el ORM Doctrine.
- En el subsistema JavaScript se modela el diseño de todas las clases JavaScript.

3.3.2 Diagramas de Clases del Diseño

A continuación, se representan los Diagramas de Clases del Diseño correspondiente a alguno de los casos de uso, permitiendo así un mejor entendimiento de cómo funcionan estos diagramas, los demás, es decir, los restantes diagramas correspondientes a los otros casos de uso se pueden consultar en los anexos. [Ver Anexo 10](#)

Al pie de cada imagen se usará la abreviatura DCD correspondiente a Diagrama de Clase del Diseño.

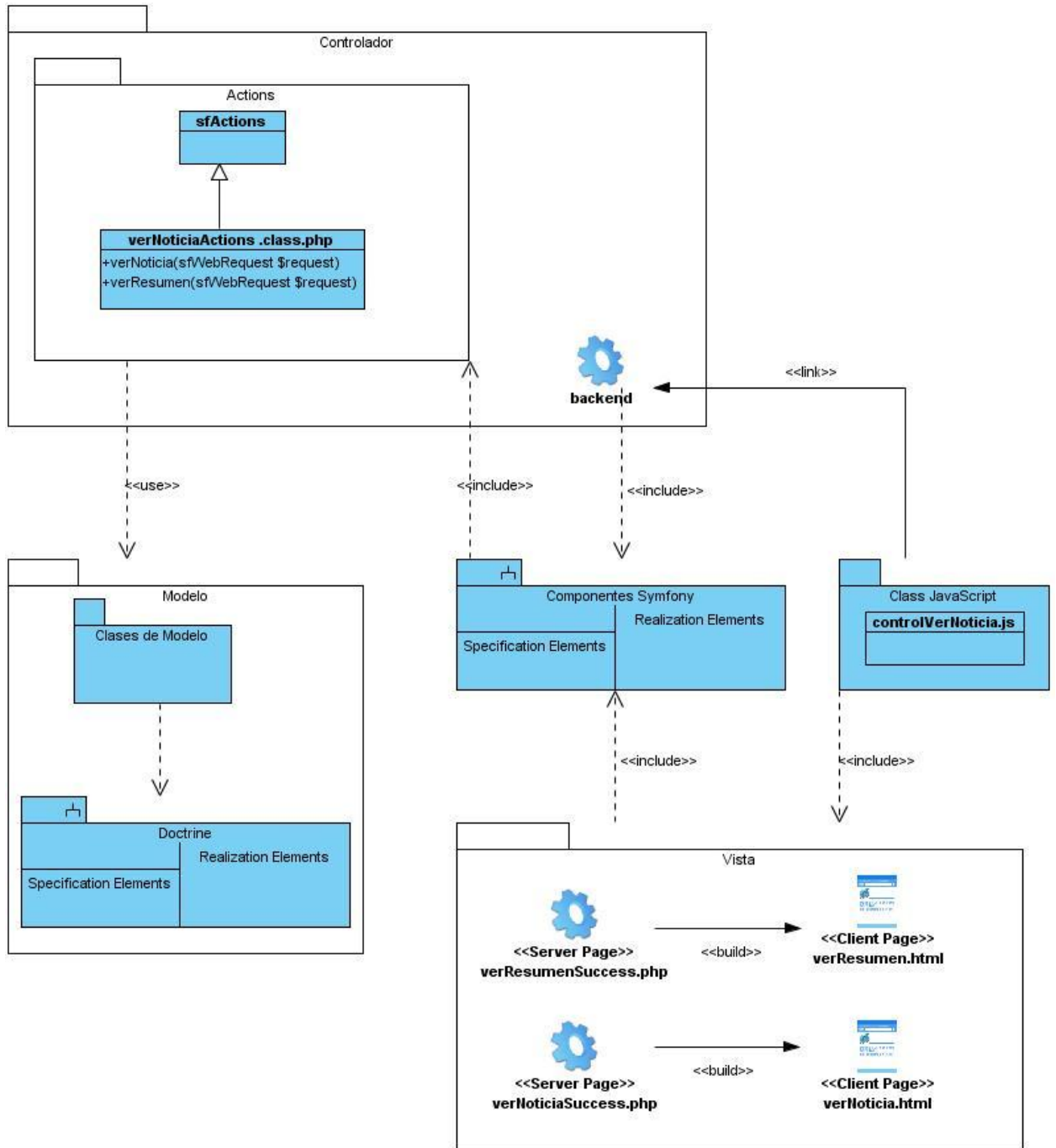


Figura 12. DCD Ver Noticia

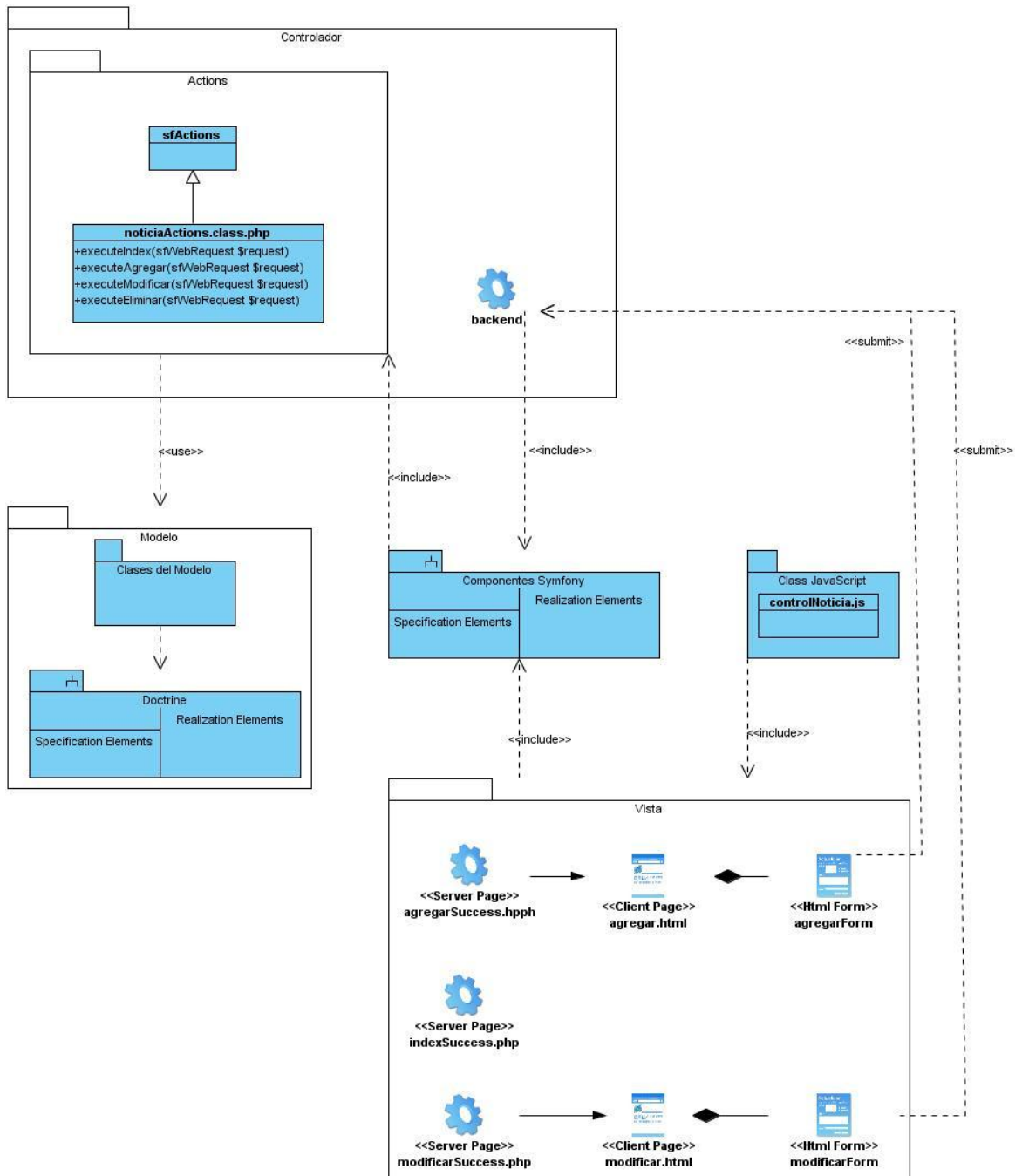


Figura 13. DCD Gestionar Noticia

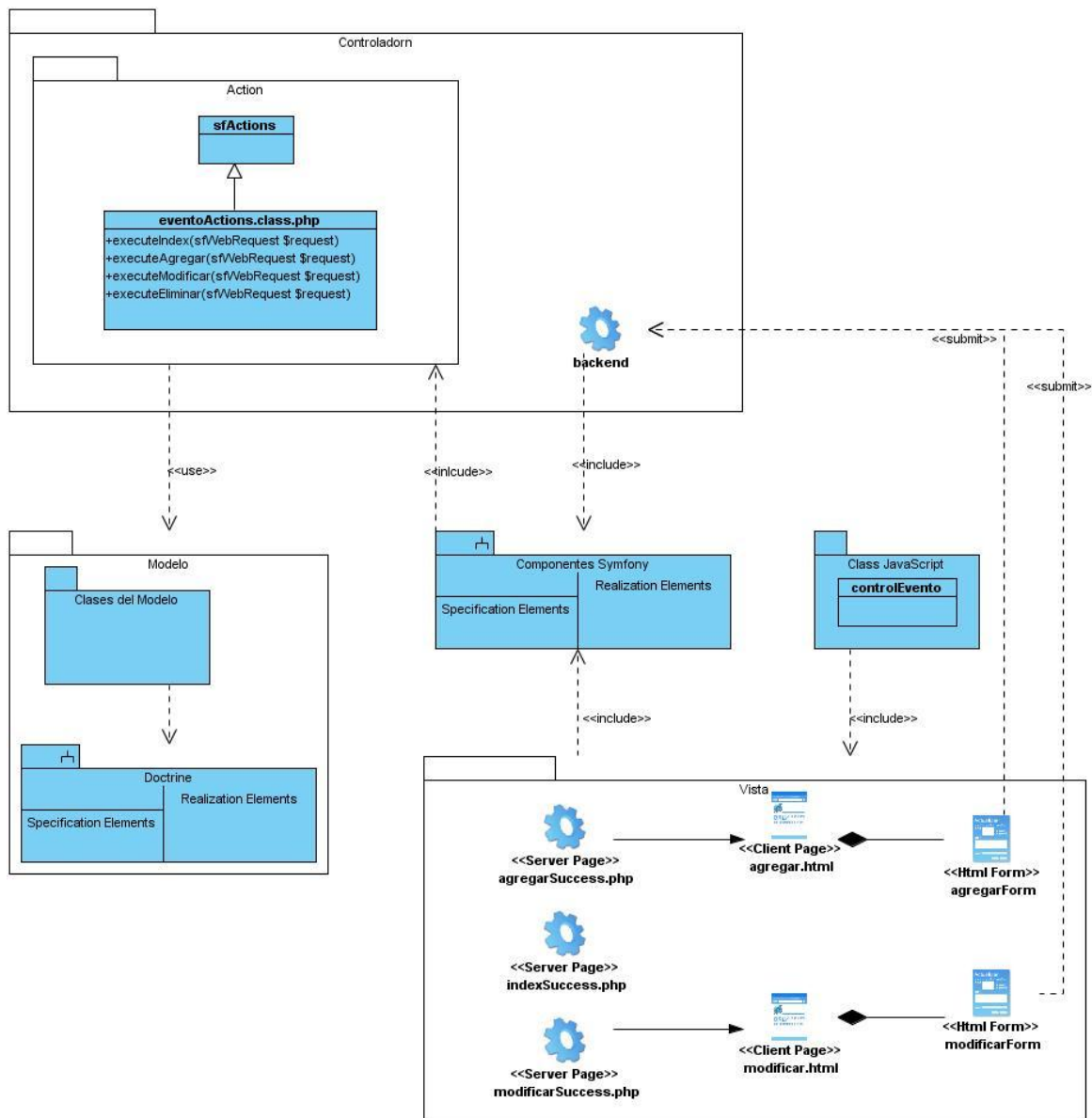


Figura 14. DCD Gestionar Evento

3.4 Diseño de la Base de Datos

El diseño de las bases de datos tiene como principal objetivo generar tablas que modelen los registros en los que se guardará la información del sistema que se desarrolla. Es muy importante evitar las redundancias en cuando se almacena la información para que los datos se puedan recuperar de forma rápida y eficiente.

3.4.1 Diagrama de Clases Persistentes

El diagrama de clases persistentes muestra todas las clases capaces de mantener su valor en el tiempo y espacio.

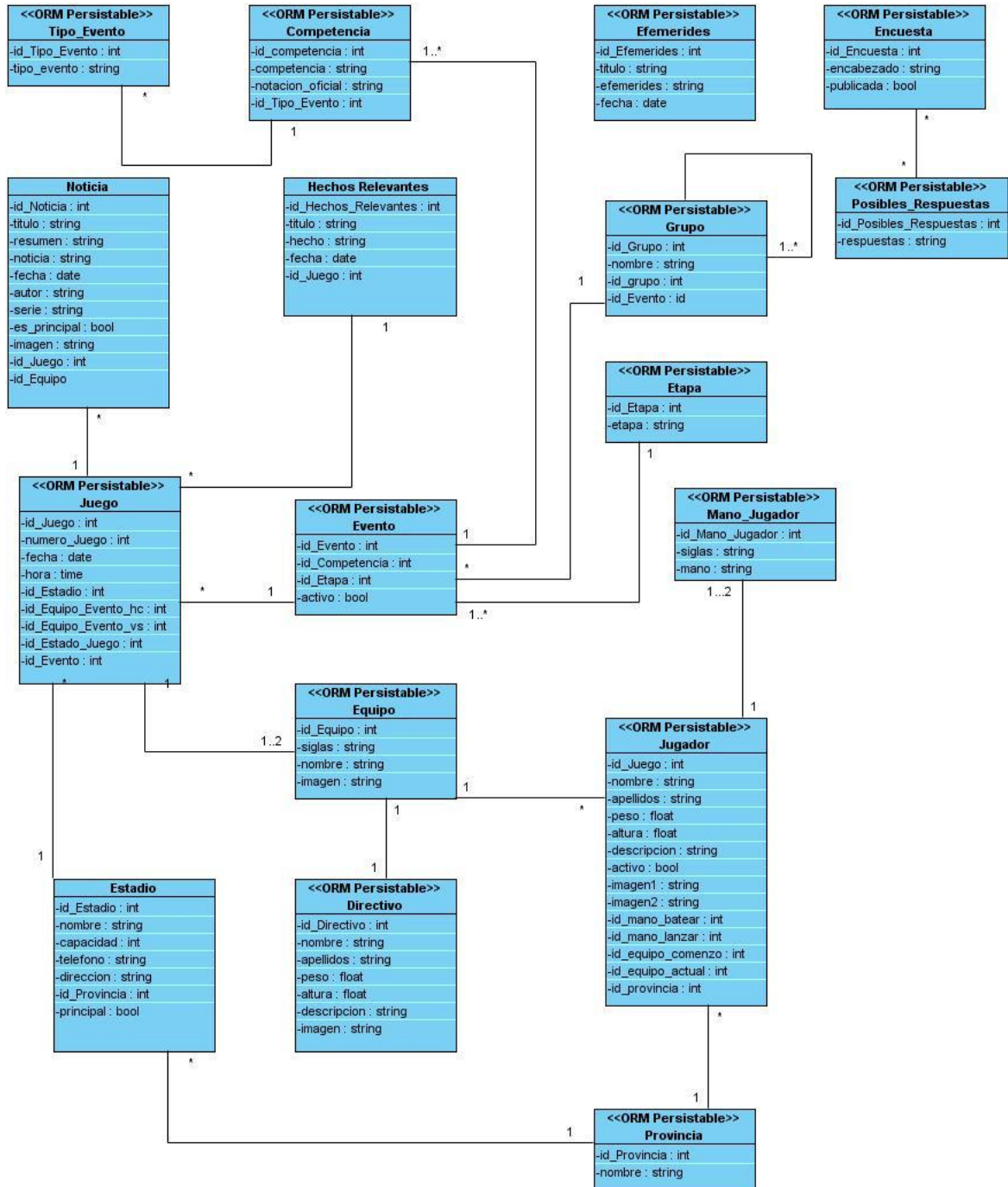


Figura 15. Diagrama de Clases Persistentes

3.4.2 Diagrama Entidad Relación de la Base de Datos

En este modelo se representan las entidades que fueron creadas para dar solución al problema que se modela.

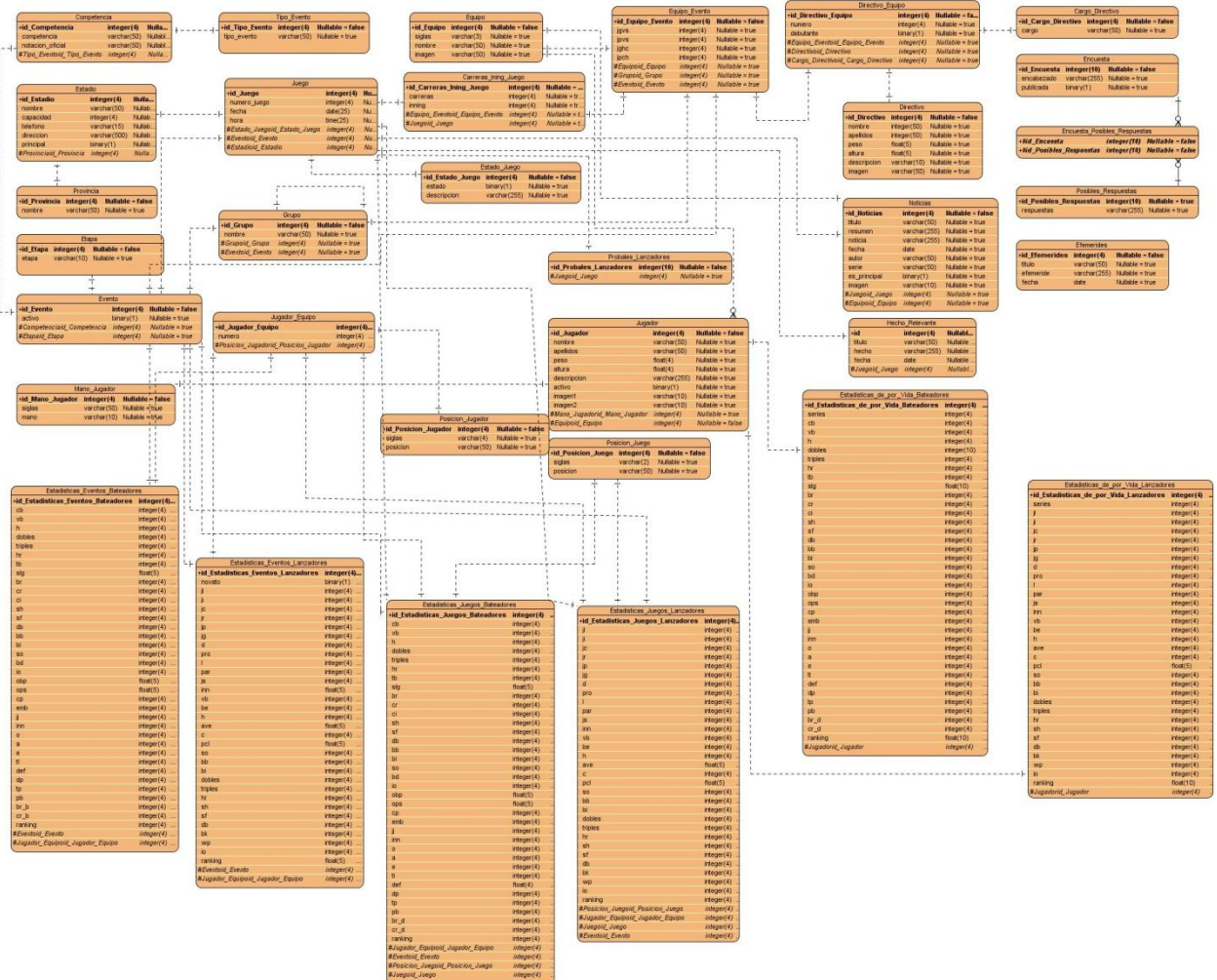


Figura 16. Diagrama Entidad – Relación de la BD

3.4.3 Descripción de las tablas de la Base de Datos

A continuación, se describen algunas de las tablas de la Base de Datos para un mejor entendimiento, las descripciones de las otras tablas se pueden consultar en los anexos.

Tabla	Competencia	
Tabla que almacena datos de las competencias		
Atributos	Tipo	Descripción
id_Competencia	int	Identificador de la competencia.
competencia	string	Nombre de la competencia
notacion_oficial	string	Notación Oficial
Id_Tipo_Evento	int	Tipo de evento al que pertenece

Tabla	Noticias	
Tabla que almacena datos de las noticias		
Atributos	Tipo	Descripción
id_Noticias	int	Identificador de la noticia
nombre	Varchar(50)	Título de la noticia
resumen	Varchar(255)	Resumen de la noticia
noticia	Varchar(255)	Contenido de la noticia
fecha	date	Fecha de la noticia
autor	Varchar(50)	Autor de la noticia
serie	Varchar(50)	Serie que se juega
es_principal	binary	Si la noticia es principal o no
imagen	Varchar(10)	Imagen sobre la noticia
Id_Juego	int	Identificador del juego
Id_Equipo	int	Identificador del equipo

Tabla	Efemerides	
Tabla que almacena datos de las efemérides		
Atributos	Tipo	Descripción
id_Efemerides	int	Identificador de la efeméride
titulo	Varchar(50)	Título de la efeméride
efemeride	Varchar(255)	Contenido de la efeméride
fecha	date	Fecha de la efeméride

Ver las demás descripciones de la Base de Datos en los anexos. [Ver Anexo 10.](#)

Conclusiones

En este capítulo se dio paso a construir la propuesta del sistema obtenido en el capítulo 2, se obtuvo los diagramas de clases de análisis, los diagramas de clases de diseño, así como el de entidad relación. Para realizar estos diagramas se utilizaron patrones de diseño los cuales ayudarán a desarrollar un sistema de calidad. Con todos los artefactos obtenidos en esta fase de desarrollo del sistema, quedan sentadas las bases para seguir trabajando en la implementación del Portal Web de Béisbol en su versión 2.0.

Conclusiones Generales

Luego de realizar las investigaciones y estudios requeridos para el desarrollo del portal web de béisbol de la UCI en su versión 2.0 y una vez concluidas las fases que propone RUP para obtener una amplia y detallada documentación, es decir, las fases de Inicio y Elaboración; se puede arribar a que:

- Las tecnologías y herramientas seleccionadas para la futura implementación del portal web de béisbol aportarán a la solución a desarrollar el valor de ser un producto libre.
- La selección de la metodología RUP como guía del proceso de desarrollo fue la correcta, pues permitió documentar el ciclo de desarrollo del portal web, quedando sentadas las bases para su posterior implementación.
- Los patrones de diseño seleccionados permitirán a los desarrolladores una mejor organización de cada una de las partes de la aplicación Web.

Recomendaciones:

Se recomienda:

- Implementar la solución propuesta para la versión 2.0 del Portal web de Béisbol.
- Desarrollar el módulo ayuda que propicie un mejor entendimiento a los usuarios que interactúen con el mismo.
- Tener en cuenta la documentación generada en este trabajo para proyectos futuros, con características similares.

Glosario de Términos

- **TIC:** Es un conjunto de herramientas, soporte y canales para el tratamiento y acceso a la información, para dar forma, registrar, almacenar y difundir contenidos digitalizados.
- **INDER:** Instituto Nacional de Deportes, Educación Física y Recreación, organismo que se encarga de la gestión deportiva, la educación física y la recreación en Cuba.
- **ASP:** Active Sever Page, también es conocido como ASP clásico, es una tecnología de Microsoft del tipo “lado del servidor” para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Services (IIS).
- **VBScript:** Abreviatura de Visual Basic Script Edition, es un lenguaje interpretado por el Windows Scripting Host de Microsoft. Su sintaxis refleja su origen como variación del lenguaje de programación visual Basic.
- **Internet Explorer:** Anteriormente conocido como Microsoft internet Explorer, y comúnmente como IE, es un navegador web desarrollado por Microsoft para el sistema operativo Microsoft Windows desde 1995.
- **RUP:** El Proceso Unificado de Desarrollo de Software (significado de los vocablos en inglés, Rational Unified Process), es un proceso de desarrollo de software y junto con el lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis implementación y documentación de sistemas orientados a objetos.
- **ICONIX:** Es un proceso de desarrollo de software que describe el análisis y diseño “lógicos” de la base que modelan proceso.
- **UML:** Lenguaje Modelado Unificado que permite modelar la representación conceptual y física de un sistema.
- **XP:** Extreme Programming y su significado en español, Programación Extrema, es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software.
- **CASE:** **C**omputer **A**ided **S**oftware **E**ngineering, en español Ingeniería de Software Asistida por Computadoras, son herramientas las cuales se definen como un conjunto

de programas que asisten a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un software.

- **OMMMA-L:** Extensión del lenguaje de modelado UML para la modelación Orientada a Objetos de Aplicaciones Multimedia.
- **Java Script:** Es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento.

Referencia Bibliográfica

ALEJANDRO LAVIN COELLO, HECTOR LUIS REYES PUPO. 2010. *Desarrollo de un módulo para gestionar los cuestionarios interactivos en la Colección El Navegante en su versión multiplataforma.* La Habana : s.n., 2010.

Amaro Calderón, Sarah Dámaris Valverde Rebaza, Jorge Carlos. 2007. *Metodologías Ágiles.* Trujillo, Perú : s.n., 2007.

Conferencia # 2. La Disciplina de Análisis y Diseño de RUP, Ingeniería de Software II.

Cornejo, José Enrique González. ¿Qué es UML? [En línea] [Citado el: 12 de enero de 2011.] <http://www.docirs.cl/uml.htm>.

Ecured. [En línea] [Citado el: 5 de mayo de 2011.] http://www.ecured.cu/index.php/Aplicaci%C3%B3n_de_patrones_en_Symfony.

Ecured. [En línea] [Citado el: 5 de mayo de 2011.] http://www.ecured.cu/index.php/Aplicaci%C3%B3n_de_patrones_en_Symfony..

Elizabet Rodríguez Cid, Michel Legrá Fleitas. 2010. *Sistema para la Gestión de la Información de los Juegos Deportivos Interfacultades.* La Habana : s.n., 2010.

Fernández, Carlos Alberto. 2005. *El Proceso Unificado para el Desarrollo de Software.* Huajuapán de León, Oaxaca : s.n., 2005.

Fernández, Leudys Romero. 2009. *ANÁLISIS Y DISEÑO DEL SOFTWARE EDUCATIVO CUENTOS INFANTILES PARA NIÑOS EN EDAD PREESCOLAR.* La Habana : s.n., 2009.

Isaías Carrillo Pérez, Rodrigo Pérez González y Aureliano David Rodríguez Martín. 2008. *Metodología de Desarrollo del Software.* 2008.

Ivar Jacobson, Grady Booch, James Rumbaugh. 2000. *El Proceso Unificado de Desarrollo de Software.* . España : s.n., 2000.

José H. Canós, Patricio Letelier y M^a Carmen Penadés. 2008. *Software, Metodologías Ágiles en el Desarrollo.* s.l. : DSIC -Universidad Politécnica de Valencia, 2008.

Lianelys Braña Verdeal, Yeinier Chacón Romero. 2007. *Análisis y Diseño del Sistema Intranet 2.* La Habana : s.n., 2007.

Nuevo Visual Paradigm for UML 3.2. [En línea] [Citado el: 10 de enero de 2011.] <http://www.programacion.com/noticia/1018/>.

Patricia, Carla Rebeca. 2007. *Uso de Metodología ICONIX*. San Martín, Oliva. : s.n., 2007.

Rodríguez Yordanis Rodríguez, Octavio Enrique Berroa Arias. 2010. *ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA VERSIÓN 2.0 DEL PORTAL DE LA FACULTAD 8*. La Habana : s.n., 2010.

SALVADOR, JUAN RODRIGUEZ. IIS [Internet]. [En línea] [Citado el: 12 de enero de 2011.]
<http://www.webtaller.com/construccion/lenguajes/php/lessons/phpiss/phpiss.php>.

Telepieza's Weblog. [En línea] [Citado el: 8 de enero de 2011.]
<http://www.telepieza.com/wordpress/2008/01/29/los-diferentes-tipos-de-portales-que-existen-en-internet/>.

Torre, Aníbal de la. Lenguajes del lado servidor o cliente. [En línea] [Citado el: 12 de enero de 2011.]
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.

Zaninotto, François. 2008. *Symfony la guía definitiva*. 2008.