



Universidad de las Ciencias Informáticas

Título: Módulo Fantasy Béisbol del Sitio de Béisbol de la UCI.

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

Autor:

Gianny Martínez Perdomo

Tutor:

Ing. José Leandro Cabrera Pérez.

Ciudad de La Habana, 2010.

Año 52 de la Revolución

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Autor

Tutor.



“Pero la juventud tiene que crear. Una juventud que no crea es una anomalía realmente.”

Ché.

DATOS DE CONTACTO

Ing. José Leandro Cabrera Pérez.

Graduado de Ingeniero en Ciencias Informáticas en el año 2007 y Especialista General con 3 años de experiencia en la Universidad de las Ciencias Informáticas (UCI) .

AGRADECIMIENTOS

Quiero agradecer a mi mamá, por educarme y quererme de la forma en que lo ha hecho, por estar siempre ahí cuando la necesité, por guiarme a través de todo y enseñarme como crecer, sin dudas, la razón por la cual hoy estoy aquí.

A mi familia por el apoyo y orgullo que me hacen sentir, a mi hermana, mi inconsciente conciencia, a mi Tía Lourdes y a mi papá, que aún no viviendo conmigo han sido invisibles razones de compromiso para seguir adelante,

A mi novia Jeydis por todo el amor, la ayuda y comprensión que en estos meses me ha brindado.

A mi tutor José Leandro, por tanta atención y por ser guía y promotor fundamental en los 2 principales acontecimientos de mi vida universitaria, y a todos los profesores que anónimamente han sido responsables de mi preparación como persona y futuro ingeniero.

A mis amigos, por estos inmejorables 5 años, Fernando, Walny, Joel, el Misa, el rosado y a todos los del inolvidable grupo 5106 en primer año. A los amigos de los demás grupos y del apartamento que han sabido ayudar y ser como hermanos en estos 5 años.

DEDICATORIA

Primeramente a mi madre, han sido varios años de estudio y sacrificio, pero finalmente este es mi regalo por luchar siempre conmigo y mi hermana, espero que me duren por siempre.

A mi abuela que aunque no está la recordaré siempre y me hubiera gustado que me acompañara hoy. A mi familia toda. A mi novia Jeydis y a mis amigos que los llevaré conmigo durante toda mi vida.

RESUMEN

El desarrollo de portales web especializados en deportes en Cuba se ha visto incrementado en los últimos años, como parte del auge tomado también a nivel mundial. Específicamente los portales relacionados con el béisbol han tomado mucha popularidad en el mundo, ya que además de brindar estadísticas y noticias especializadas, los servicios de gestión de deportes de fantasía que estos portales brindan a sus usuarios conllevan a que estos servicios sean los que más tráfico de internet generen luego de las redes sociales.

En la actualidad el portal web de béisbol de la Universidad de las Ciencias Informáticas (UCI) no cuenta con un sistema que gestione deportes de fantasía del pasatiempo nacional, por lo que los usuarios del sitio de béisbol de la UCI no cuentan con las funcionalidades que estos sistemas brindan, provocando un éxodo de éstos usuarios hacia portales foráneos y una ostensible disminución de la cantidad de visitas al mismo.

El presente trabajo de diploma tiene como objetivo desarrollar un sistema que permita la gestión eficiente de deportes de fantasía, específicamente de béisbol, dentro del sitio de béisbol de la UCI. Para el desarrollo del sistema propuesto se analizaron las necesidades actuales de los usuarios del sitio de béisbol y se tuvieron en cuenta las metodologías y tecnologías actuales propuestas por la Universidad de las Ciencias Informáticas para el desarrollo de aplicaciones web. Se seleccionó la tecnología PHP como lenguaje de programación utilizando el framework Symfony para su desarrollo.

PALABRAS CLAVE:

Deportes de Fantasía, Symfony, Web.

Contenido

INTRODUCCIÓN	1
INTRODUCCIÓN	5
1.1 BÉISBOL DE FANTASÍA	5
1.1.1 <i>Historia de los Juegos de Fantasía</i>	5
1.1.2 <i>Estadísticas Sabermétricas</i>	6
1.2 ANÁLISIS DE SOLUCIONES EXISTENTES	7
1.2.1 <i>Béisbol de Fantasía en el Mundo</i>	7
1.2.2 <i>Béisbol de Fantasía en Cuba</i>	8
1.2.3 <i>Béisbol de Fantasía en la Universidad</i>	8
1.2.4 <i>Ejemplos de Interfaz de Béisbol de Fantasía</i>	9
1.3 METODOLOGÍAS A UTILIZAR.....	11
1.3.1 <i>Programación Extrema XP</i>	11
1.3.2 <i>SCRUM</i>	12
1.3.3 <i>Proceso Unificado de Rational(RUP)</i>	12
1.3.4 <i>¿Por qué RUP?</i>	12
1.4 LENGUAJES DE MODELADO	13
1.4.1 <i>Lenguaje de Modelado OO i*</i>	13
1.4.2 <i>Lenguaje de Modelado OO UML</i>	13
1.4.3 <i>¿Por qué UML?</i>	14
1.5 LENGUAJES DE PROGRAMACIÓN.....	14
1.5.1 <i>HyperText Markup Language (HTML)</i>	15
1.5.2 <i>Java Script</i>	15
1.5.3 <i>Cascade StyleSheet (CSS)</i>	16
1.5.4 <i>Hypertext Preprocessor (PHP)</i>	17
1.6 FRAMEWORK DE DESARROLLO.....	18
1.6.1 <i>Symfony</i>	19
1.7 GESTOR DE BASE DE DATOS POSTGRESQL.....	22
1.8 CONSIDERACIONES FINALES	24
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	25
2.1 MODELO DEL DOMINIO	25
2.2 PROPUESTA DEL SISTEMA	26
2.3 ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE	26
2.3.1 <i>Requerimientos Funcionales</i>	26
2.3.2 <i>Requerimientos no Funcionales</i>	28

2.4 MODELO DE CASOS DE USO DEL SISTEMA.....	29
2.4.1 Definición de actores.....	29
2.4.2 Diagrama de Casos de Uso del Sistema	30
2.4.3 Descripción de los Casos de Uso del Sistema.....	30
CONSIDERACIONES FINALES.....	58
CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA	59
3.1 INTRODUCCIÓN	59
3.2 ANÁLISIS.....	59
3.2.1 Clases del Análisis	59
3.2.2 Diagrama de Colaboración.....	60
3.3 DISEÑO.....	64
3.3.1 Clases del Diseño.....	64
3.3.2 Seguridad	64
3.3.3 Patrones de diseño que se emplean.....	65
3.3.4 Métodos de validación y manejo de errores.....	68
3.4 DISEÑO DE BASE DE DATOS.....	69
3.4.1 Descripción de las Tablas	69
3.5 DIAGRAMA DE COMPONENTES	70
3.6 DIAGRAMA DE DESPLIEGUE	70
3.7 ESTÁNDARES DE CODIFICACIÓN.....	71
3.8 CONSIDERACIONES FINALES	72
CONCLUSIONES.....	73
RECOMENDACIONES	74
BIBLIOGRAFÍA.....	75
REFERENCIAS BIBLIOGRÁFICAS.....	77
ANEXOS.....	78
GLOSARIO DE TÉRMINOS	110

INTRODUCCIÓN

Desde el surgimiento de la internet la red de redes ha evolucionado de recurso exclusivo con fines militares o académicos utilizado por pequeños grupos de usuarios hasta lo que es hoy en día, una verdadera sociedad digital que permite la confluencia de millones de usuarios que buscan acceso de forma eficaz a los distintos servicios que hay disponible en la red.

Al configurarse paulatinamente el actual estado de conectividad mundial a las redes, muchas empresas comenzaron a ofrecer servicios en esta nueva plataforma para mejorar su economía, surgiendo lo que hoy se conoce como comercio electrónico. En este nuevo mercado es conocido que la variedad de dominios que obtienen jugosos ingresos es muy grande y la factibilidad de estos dominios depende en alta medida de la cantidad de usuarios que tengan acceso a ellos por lo que estar en el destino del tráfico en internet es una cuestión vital para los mismos.

A nivel internacional los 5 servicios informáticos que más tráfico de internet generan, son las redes sociales, los deportes de fantasía, los videos en línea, el correo electrónico y el chat, siendo los deportes de fantasía una de las de mayor popularidad ya que aunque no cobra a los usuarios directamente por acceder a sus servicios, es muy rentable ya que por el gran tráfico de internet que recibe puede mostrar información a una gran comunidad, ya sea esta información pagada (publicidad) o en forma de noticias con un fin determinado.

Un deporte de fantasía se define como un juego virtual basado en la simulación informática de un deporte real mediante estadísticas generadas durante el mismo y generalmente se encuentran hospedadas en portales temáticos dedicados a un deporte, o indexadas a redes sociales.

En la Universidad de las Ciencias Informáticas existe un sitio web de prestigio nacional dedicado al béisbol que se encarga de la promoción de los distintos eventos de ese deporte, así como a la actualización online del estado de cada juego de la serie nacional de béisbol en tiempo real, y que además recoge todas las estadísticas de los jugadores a lo largo de la competencia pero no gestiona las estadísticas individuales obtenidas para la conformación de equipos o alguna forma de deportes de fantasía de béisbol.

Entonces la **situación problemática** surge de la inexistencia en el sitio de béisbol de la Universidad de las Ciencias Informáticas de una herramienta que permita con economía de

recursos de procesamiento, actualizar y dar soporte dinámicamente a las estadísticas recolectadas para gestionar un deporte de fantasía, limitando gran parte del potencial de usabilidad que este tipo de portales genera

De esto se deriva el siguiente **problema científico**: ¿Cómo desarrollar una herramienta para el sitio de béisbol de la UCI de modo que permita actualizar y dar soporte dinámicamente a los datos de juegos recogidos para una Fantasía de Béisbol?

Tal interrogante implica que el **objeto de estudio** sean los sistemas de deportes de fantasía. Y como **campo de acción**: proceso de desarrollo de una fantasía de béisbol para el sitio de béisbol de la Universidad de las Ciencias Informáticas.

Para dar solución al problema planteado se ha trazado como **objetivo general de la investigación**: Implementar un módulo fantasía béisbol para el sitio de beisbol de la Universidad de las Ciencias Informáticas.

Para dar cumplimiento al objetivo general se han trazado las siguientes

Tareas de Investigación:

- Estudio de los juegos de fantasía para la confección del estado del arte.
- Estudio de la estructura y funcionamiento del sitio de beisbol de la UCI para integrar módulos al mismo
- Estudio de las tendencias actuales de programación en lenguaje PHP para posibilitar la reusabilidad.
- Análisis de las herramientas del framework Symfony para su uso en portales Web.
- Definición del dominio en que se enmarca el problema para comprender el funcionamiento del sistema a desarrollar.
- Realización de la captura de requerimientos, para conocer las capacidades y condiciones que debe poseer el sistema.
- Estudio del empleo de patrones y estándares de codificación en aplicaciones web.

Para la realización del trabajo se tienen en cuenta algunos métodos tradicionales investigativos. A continuación se mencionarán cada uno de ellos y de qué forma se ponen de manifiesto en la investigación.

Métodos de investigación utilizados:

A nivel teórico:

Se utilizó el método **Analítico – Sintético** pues se realiza una revisión de documentos, sitios web y tesis, todos relacionados con el tema de deportes de fantasía de los cuales se obtuvo la información específica que se necesitaba para el desarrollo de la investigación.

A partir del **Análisis histórico – lógico** se pudo constatar teóricamente como ha evolucionado la visualización de la información y los deportes de fantasía, así como sus antecedentes y tendencias actuales permitiendo obtener mayor conocimiento sobre el tema y aplicarlo en la solución de problemas.

A nivel empírico:

A través de la **Observación** se pudo conocer la esencia de la problemática definida, lo que ayudó al planteamiento del problema científico, además de permitir conocer el proceso definido como objeto de estudio, lo cual influye a la hora de tener un conocimiento más detallado de lo que se quiere, lo que hace falta hacer y cómo hay que hacerlo.

Este trabajo está estructurado de la siguiente manera: Introducción, Desarrollo, dividido en tres capítulos, las Conclusiones, Recomendaciones, la Bibliografía, Referencias bibliográficas, los Anexos y el Glosario de Términos.

Capítulo #1: Fundamentación teórica: Contiene la fundamentación teórica del tema. Se realiza un estudio del estado del arte de manera general. Se aborda el tema de las tendencias, metodologías y tecnologías de desarrollo del software actuales y culmina con la fundamentación de las metodologías y tecnologías escogidas.

Capítulo #2: Características del sistema: Realiza una descripción de los procesos del dominio relacionados con el procesamiento de la información estadística. Se especifican los actores, los casos de uso del sistema, el modelo del dominio, se definen los requerimientos funcionales y no funcionales, y se hace una descripción textual de los casos de uso que serán implementados.

Capítulo #3: Construcción de la solución propuesta: En este capítulo se realizarán los flujos de análisis, diseño e implementación del sistema, donde se definen los diagramas de clases del análisis y del diseño por casos de uso, se muestra la interacción entre el actor y el sistema, mediante los diagramas de colaboración y se muestra el diseño de la base de datos además de una descripción de sus tablas. También se presentan los patrones de diseño utilizados, así como los métodos de validación y manejo de errores, cuestiones relacionadas con la seguridad

y la interfaz del módulo implementado. Se realiza el modelo de implementación el cual está compuesto por el diagrama de componentes y el diagrama de despliegue. Para finalizar se realiza una breve explicación de los estándares de codificación utilizados.

Capítulo 1: Fundamentación Teórica.

Introducción

En este capítulo se desarrollará un estudio general sobre los deportes de fantasía de béisbol así como temas relacionados con la entrada de datos necesaria para el funcionamiento de los mismos, su descripción, funcionalidades y componentes. Con el propósito de alcanzar un claro entendimiento del dominio del problema en cuestión, la terminología básica utilizada, además se definen todo un conjunto de herramientas, metodologías y tecnologías necesarias para la realización del presente trabajo de tesis.

1.1 Béisbol de Fantasía

Deporte de fantasía de béisbol en Internet o béisbol en línea, entre los deportes de simulación de computadora, es una simulación de liga de béisbol en Internet que le otorga el cargo de director de equipo a un usuario. El béisbol en línea (también conocido como rotisserie, roto, o deporte de cuento de hadas; o la simulación de dueño) es un juego donde los directores de fantasía construyen un equipo que compite contra otros dueños de fantasía basados en la estadística generada por jugadores individuales o equipos de un profesional [1].

1.1.1 Historia de los Juegos de Fantasía

Si bien no hay consenso acerca del origen de los deportes de fantasía, estos empezaron a llamar la atención a raíz de la publicación en 1984 de Rotisserie League Baseball, basada en una liga jugada por un grupo de personas de medios neoyorquinos y con reglas pautadas por el periodista Daniel Okrent [2].

El atractivo de las ligas de fantasía para los fanáticos del deporte es a menudo mencionado como un factor en el éxito inicial de "USA Today", que fue lanzado en 1982 con más estadísticas detalladas que otros periódicos de temas deportivos, dando inicio a la socialización comparativa de las estadísticas.

Luego de la publicación de Rotisserie League Baseball el fenómeno del béisbol de fantasía se dispersó a la par de la temporada regular de las grandes ligas de béisbol norteamericano, pero las ligas existentes no podían congregarse a demasiados usuarios por las limitaciones de comunicación hasta el surgimiento del internet.

Los deportes de fantasía dieron un salto exponencial a fines de la década del `90, cuando internet hizo que los resultados deportivos estuvieran disponibles a nivel masivo y sitios como

Capítulo 1: Fundamentación Teórica

Yahoo Inc. comenzaron a organizar eventos a cambio de una tarifa, tendencia que han adoptado portales de deportes, que brindan estadísticas y ofrecen deportes de fantasía para captar usuarios.

La tendencia actual de acceso físico a las ligas de fantasía es que los fanáticos usen sus teléfonos celulares para leer las noticias deportivas, revisar las puntuaciones de las disciplinas y ver el estatus de sus equipos. Actualmente existen varios sitios o consejeros de internet que son diariamente consultados los cuales basan sus predicciones en el cálculo de las llamadas estadísticas saber métricas.

1.1.2 Estadísticas Saber métricas

Saber métricas es el análisis del béisbol a través de evidencia objetiva, especialmente de las estadísticas del béisbol. El término se deriva del acrónimo SABR el cual significa Society for American Baseball Research (sociedad para la investigación del béisbol americano). Fue fundado por Bill James, quién fue su primer defensor y ha sido por mucho tiempo su más prominente figura.

Bill James la define como la búsqueda de conocimiento objetivo sobre el béisbol. Por lo tanto, las saber métricas intentan responder a preguntas objetivas sobre el béisbol, como que jugador es el que más ha contribuido a la ofensiva del equipo o ¿Cuántos jonrones un jugador puede dar el próximo año?.. No se puede hacer frente a los juicios subjetivos que también son importantes para el juego, tales como "¿Quién es tu jugador favorito?".[3]

Algunas estadísticas de este tipo son muy utilizadas en los Análisis para medir la calidad de un jugador o un equipo en una liga de fantasía tales como: [4]

- OPS (% Embasado + slugging)
- WHIP (Boletos + hits / innings lanzados).
- DIPS (Defensa independiente del pitcheo)
- LIPS (Situaciones de presión en últimos innings)
- PERA (Promedio de carreras limpias periférico)
- VORP (Valor sobre el jugador de reemplazo)
- WAR (Victorias por encima del reemplazo)

1.2 Análisis de Soluciones Existentes

A continuación se muestra un análisis de los sistemas de gestión de deportes de fantasía existentes a nivel internacional, en Cuba y en la Universidad de Ciencias Informáticas para tener un mejor entendimiento de cómo funcionan y cuales son sus principales características.

1.2.1 Béisbol de Fantasía en el Mundo

En el ámbito internacional los sitios de deportes de fantasía están teniendo una expansión exponencial explotando el gran grupo social que sigue diariamente las ligas deportivas alrededor del mundo y en especial las de béisbol.

Yahoo.com

La fantasía de béisbol del sitio Yahoo.com, es una de las más exitosas a nivel mundial y congrega usuarios de todo el mundo. Brinda la posibilidad de gestionar una Liga de fantasía basada en las estadísticas reales de las grandes ligas de béisbol estadounidense que cuenta con 162 partidos. Gestiona a los usuarios por roles ofreciendo la funcionalidad de unirse a una liga pública o crear una nueva liga e invitar a amigos a unirse y tener el control de una liga privada. Provee además varias herramientas extras para visualizar las estadísticas de equipos y jugadores mediante gráficas 3D.

ESPNdeportes.com

La fantasía de béisbol del sitio ESPNdeportes.com comparte el liderazgo con Yahoo.com y agrupa la mayoría de la atención en el público hispanohablante. Posee una serie de funcionalidades implementadas que la han hecho muy popular en las últimas temporadas tales como:

- Plataforma multilenguaje personalizable según la región del usuario.
- Asignación de roles, permitiendo tanto crear y gestionar una liga privada, como unirse como invitado a una liga anteriormente creada.
- Posibilita Fantasear con ligas de países latinoamericanos y con categorías inferiores de las grandes ligas, además de la MLB.
- Permite compartir estadísticas y estado de las ligas creadas en redes sociales como Facebook y Twitter.

MLB.com

Capítulo 1: Fundamentación Teórica

La fantasía de béisbol del sitio MLB.com es la más visitada en los Estados Unidos de América, el mayor mercado de este servicio de Internet, y se estima que cuenta con 27 millones de usuarios y genera más de 1000 millones de dólares anuales. Su popularidad debe en gran medida a las siguientes razones:

- Se encuentra ubicada en el sitio oficial de las grandes ligas de béisbol y cuenta con el patrocinio de esa entidad.
- Asignación de roles, permitiendo tanto crear y gestionar una liga privada, como unirse como invitado a una liga anteriormente creada.
- Implementa la funcionalidad de crear equipos con jugadores legendarios y simular series según las estadísticas de cuando eran jugadores activos.
- Ofrece estadísticas sabermétricas para conocer parámetros adicionales sobre el rendimiento de los jugadores.
- Brinda predicciones basadas en frecuencias, sobre las probabilidades de rendimiento de los jugadores y equipos.

1.2.2 Béisbol de Fantasía en Cuba

En la República de Cuba el deporte nacional es el béisbol y se tiene conocimiento de que el primer partido oficial de este deporte desarrollado en la isla data del 27 de diciembre de 1874 en el estadio Palmar de Junco. Con el objetivo de promocionar este deporte y de dar a conocer todos los eventos que del mismo se realizan, se han desarrollado una amplia gama de Sitios Web que actualizan diariamente las estadísticas de los campeonatos de las distintas categorías.

Entre ellos se encuentran el portal oficial de la Federación Cubana de Béisbol Amateur, el sitio oficial del Instituto Nacional de Deportes Educación Física y Deportes (INDER), y otros que se listan en el anexo1.

Ninguno de los sitios descritos que dan seguimiento al béisbol que se desarrolla en el país, tiene implementado un sistema de gestión de béisbol de fantasía por lo que se puede concluir que el módulo a desarrollar para el sitio de la UCI sería único a nivel nacional.

1.2.3 Béisbol de Fantasía en la Universidad.

En la Universidad de las Ciencias Informáticas se tiene como antecedente la implementación de forma experimental de una serie de páginas web con un objetivo común para el sitio

Capítulo 1: Fundamentación Teórica

beisbol.uci.cu en el período 2007-2008 que proveía de cierta manera a los usuarios de ese sitio con una fantasía béisbol que contaba con las siguientes opciones:

- Asignación de roles, permitiendo tanto crear y gestionar una liga privada, como unirse como invitado a una liga anteriormente creada.
- Gestión de jugadores.
- Visualización de estadísticas generales y específicas de la liga real.
- Visualización en paneles generales de las ligas con mayor cantidad de puntos.
- Visualización a nivel de ligas de los líderes por puntos y por departamentos.

La misma tenía los problemas que a continuación se muestran:

- Estaba desarrollada sobre una tecnología de software propietario.
- Consumía demasiados recursos de procesamiento en el servidor donde estaba instalado, causando intermitencias en el servicio.
- No utilizaba ningún tipo de framework o metodología que garantizara buenas prácticas de programación y de desarrollo en general.
- No gestionaba automáticamente todas las estadísticas recogidas.
- No se implementó utilizando una metodología definida por lo que no contaba con la documentación ingenieril adecuada para dar soporte al mismo.

Estos problemas relacionados a la pionera de las fantasías de béisbol en la universidad conllevaron a la suspensión de este servicio, lo cual acarrió un sensible descenso de la cantidad de usuarios que hacían uso de las ofertas del sitio de béisbol.

1.2.4 Ejemplos de Interfaz de Béisbol de Fantasía.

En las figuras (fig)1, 2 y 3 se muestran ejemplos de los portales de béisbol en el mundo que mayor cantidad de usuarios acaparan con sus sistemas de béisbol de fantasía.

Capítulo 1: Fundamentación Teórica



Fig. 1 Portal Yahoo.com

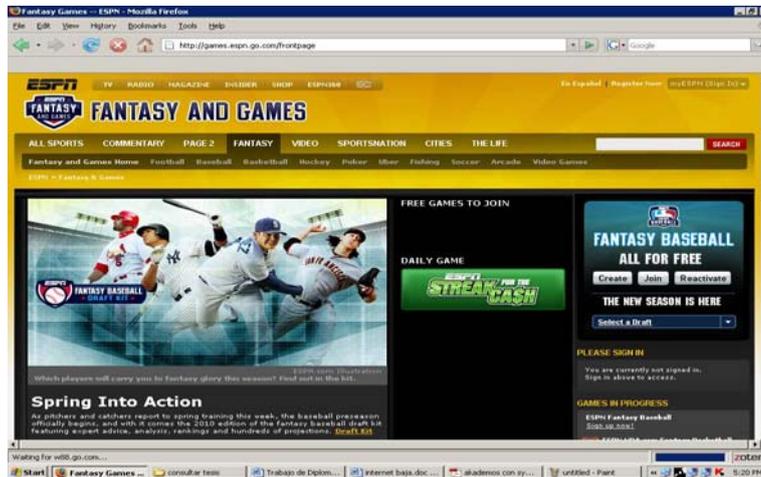


Fig. 2 Portal Espn.com



Fig.3 Portal Mlb.com

Capítulo 1: Fundamentación Teórica

Luego de la investigación realizada sobre los deportes de fantasía se puede afirmar, que a nivel mundial existen disímiles portales web que gestionan este tipo de servicios, pero ninguno de los mismo ha sido desarrollado para ser utilizado paralelamente a la serie nacional de béisbol cubana, ya que mayoritariamente gestionan estadísticas de las grandes ligas estadounidenses por lo que ninguna puede ser utilizada en Cuba.

1.3 Metodologías a utilizar

El objetivo de un proceso de desarrollo es subir la calidad del software (en todas las fases por las que pasa) a través de una mayor transparencia y control sobre el proceso. Es labor del proceso de desarrollo hacer que esas medidas para aumentar la calidad sean reproducibles en cada desarrollo.

En los últimos tiempos la cantidad y variedad de los procesos de desarrollo ha aumentado de forma impresionante, sobre todo teniendo en cuenta el tiempo que estuvo en vigor como ley única el famoso desarrollo en cascada. Se podría decir que en estos últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo, los llamados métodos pesados y los métodos ligeros. La diferencia fundamental entre ambos es que mientras los métodos pesados intentan conseguir el objetivo común por medio del orden y documentación, los métodos ligeros (también llamados métodos ágiles), tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso.

El sistema que se desarrollará tendrá un enfoque orientado a objetos (OO). Existen varias metodologías OO basadas en UML, que será el lenguaje de modelado a usar, que se pueden utilizar.

1.3.1 Programación Extrema XP

La metodología Extreme Programming (XP) o Programación Extrema, es una de variantes de las metodologías ágiles con más aceptación en la comunidad internacional de desarrollo. Los fundamentos de la misma según su creador son: mejorar la comunicación, buscar la simplicidad, buscar retroalimentación en que también va el trabajo y que siempre hay que proceder con valentía.

Una de las herramientas más importantes en la metodología XP es el desarrollo orientado a pruebas, que utiliza las pruebas unitarias como eje de todo desarrollo. Las interacciones suelen ser muy cortas y se promueve a los programadores a buscar

soluciones y experiencia con ellas, programar sin miedo a descomponer el sistema.

1.3.2 SCRUM

SCRUM es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza ejecutable que incorpora una nueva funcionalidad.

En SCRUM el equipo se enfoca solamente en construir software de calidad. La gestión de proyecto se focaliza en definir las características del software a construir y remover cualquier obstáculo que pudiera entorpecer el trabajo del equipo de desarrollo. Se busca que los equipos de desarrollo sean lo más productivos posible. [5]

1.3.3 Proceso Unificado de Rational(RUP)

El Rational Unified Process (RUP) es un proceso de amplio marco que ofrece las mejores prácticas para el software y los sistemas de entrega, la aplicación eficaz y la gestión de proyectos. Constituye una de las metodologías estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es un proceso de desarrollo de software, que se clasifica como un proceso pesado, se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (Unified Modeling Language) como herramienta principal. Se caracteriza por ser iterativo e incremental e incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). [6]

1.3.4 ¿Por qué RUP?

La metodología a utilizar será RUP ya que es un proceso para el desarrollo de un proyecto que tiene claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Tiene tres características esenciales está dirigido por casos de uso: que orientan el proyecto a la importancia para el usuario y lo que este quiere, está centrado en la arquitectura: que realiza la toma de decisiones que indica cómo tiene que ser construido el sistema y en qué orden y es iterativo e incremental: donde divide el desarrollo en mini proyectos donde los casos de uso y la arquitectura cumplen su objetivo de manera

más depurada. Está diseñada para proyectos de gran envergadura, esta metodología genera gran cantidad de documentación permitiendo un mejor entendimiento del proyecto.

1.4 Lenguajes de Modelado

Con el transcurso de los años se han ido desarrollando las técnicas de programación orientada a objetos (POO). Además se han desarrollado diferentes lenguajes de modelación orientada a objetos. Algunos ejemplos de estos lenguajes de modelado son I* y UML.

1.4.1 Lenguaje de Modelado OO i*

La notación i* fue creada a en la primera mitad de la década de los 90. Permite expresar de forma clara y sencilla los objetivos de los actores que aparecen en los modelos y la dependencia entre ellos. Consta con una notación gráfica que permite tener una visión intuitiva y unificada del entorno modelado mostrando tales actores y dependencias. Tiene la desventaja de no tener una definición única del lenguaje. Además, las definiciones existentes no son tan claras como se desearía ya que contienen ambigüedades y contradicciones

1.4.2 Lenguaje de Modelado OO UML

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Además capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. La estructura estática define los tipos de objetos importantes para un sistema y para su implementación, así como las relaciones entre los objetos. El comportamiento dinámico define la historia de los objetos en el tiempo y la comunicación entre objetos para cumplir sus objetivos. El modelar un sistema desde varios puntos de vista, separados pero relacionados, permite entenderlo para diferentes propósitos. [7]

Las herramientas que provee UML para el desarrollo de sistemas orientados a objetos son:

- ✓ Diagrama de casos de uso
- ✓ Diagrama de clases

- ✓ Diagrama de estados
- ✓ Diagrama de secuencias
- ✓ Diagrama de actividades
- ✓ Diagrama de colaboraciones
- ✓ Diagrama de componentes
- ✓ Diagrama de distribución

(Torres, 2005)

1.4.2.1 Características de UML

- Permite adaptarse fácilmente a los usuarios, así como a otros usuarios de otros métodos.
- Permite modelar nuevas cosas en el proyecto que se desarrolla.
- UML provee una expresividad e integridad holística mejorada, respecto a otros lenguajes de moldeamiento visual.
- UML provee una expresividad e integridad holística mejorada, respecto a otros lenguajes de moldeamiento visual.
- UML es fácil de aprender y usar, ya sea respecto a las técnicas más avanzadas, es decir, estereotipos y propiedades, así como algunos cambios en la anotación y semánticas

1.4.3 ¿Por qué UML?

Debido al estudio de las características de los lenguajes de modelado especificadas anteriormente se determinó como el lenguaje de modelado que será utilizado para dar solución a los objetivos planteados es UML ya que sus características se corresponden con lo que se quiere lograr del sistema y además es el lenguaje de modelado más utilizado en la UCI y se tiene experiencia con la utilización del mismo.

1.5 Lenguajes de programación

Los lenguajes de programación intentan conservar una similitud con el lenguaje humano, con la finalidad de que sean más naturales a quienes los usan. Establecen un conjunto de reglas sintácticas y semánticas, las cuales rigen la estructura del programa de computación que se

escribe o edita. De esta forma, permiten a los programadores o desarrolladores, poder especificar de forma precisa los datos sobre los que se va a actuar, su almacenamiento, transmisión y demás acciones a realizar bajo las distintas circunstancias consideradas. A continuación se describen una serie de lenguajes que se utilizan durante el desarrollo de este trabajo de tesis que no requieren de compilación alguna en el servidor de aplicaciones web excepto el lenguaje PHP.

1.5.1 HyperText Markup Language (HTML)

HTML son las siglas de HyperText Markup Language (del español, Lenguaje Marcado de Hipertexto).

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. [8]

El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global". Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica. [8]

1.5.2 Java Script

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. [9]

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, es una marca registrada de la empresa Sun Microsystems, [9]

Capítulo 1: Fundamentación Teórica

El código JavaScript puede ser integrado dentro de las páginas web. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseñó un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento). [9]

Ventajas:

- Los script tienen capacidades limitadas, por razones de seguridad.
- El código JavaScript se ejecuta en el cliente.
- La mayoría de los navegadores en sus últimas versiones interpretan código JavaScript
- Lenguaje de scripting seguro y fiable.

Desventajas:

- Código visible por cualquier usuario.
- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).
- No es un lenguaje orientado a objetos, el mismo no dispone de herencias.

1.5.3 Cascade StyleSheet (CSS)

El impulso de los lenguajes de hojas de estilos se produce a partir del auge y el crecimiento del uso de Internet y el incremento del lenguaje HTML para la creación de documentos electrónicos. La competencia entre los navegadores y la falta de un estándar para la definición de los estilos afectaban la creación de documentos con la misma apariencia.

En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS (del inglés, Cascading Style Sheets) y lo añadió a su grupo de trabajo de HTML. A finales de 1996, el W3C publicó la primera recomendación oficial, conocida como "CSS nivel 1".

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. [10]

El lenguaje CSS se utiliza para definir el aspecto o propiedades de cada elemento del contenido en una página web, tales como el color, tamaño, tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

1.5.4 Hypertext Preprocessor (PHP)

PHP (acrónimo de Hypertext Preprocessor,) es un lenguaje "del lado del servidor" (esto significa que PHP funciona en un servidor remoto que procesa la página Web antes de que sea abierta por el navegador del usuario) especialmente creado para el desarrollo de páginas Web dinámicas. Puede ser incluido con facilidad dentro del código HTML.

PHP es un lenguaje de script interpretado en el lado del servidor que no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS (Internet Information Server) con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. En la figura 4 se muestra su interacción en la web.



Fig.4 Interpretación lenguaje PHP

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.

Capítulo 1: Fundamentación Teórica

- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas:

- Se necesita instalar un servidor web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.

1.6 Framework de Desarrollo

Hoy en día la mayoría de los proyectos realizados tanto en ambiente web como desktop son desarrollados utilizando las facilidades que brindan los frameworks de desarrollo. Los frameworks son similares a las bibliotecas de software, ya que son abstracciones reutilizables de código definidos en una API. A diferencia de las bibliotecas, el control del flujo del programa no es dictado por el usuario que ejecuta determinada acción, sino por el propio framework. Esta inversión de control es su principal característica.

Un framework es una mini arquitectura que proporciona la estructura genérica y el comportamiento para una familia de abstracciones a lo largo de un contexto de metáforas que especifican su colaboración y es usado en un dominio dado. [11]

El framework codifica el contexto en una clase de máquina virtual mientras hace las abstracciones abiertas. Un framework no es generalmente una aplicación específica. Una aplicación, en cambio puede ser construida con uno o más frameworks insertando dicha funcionalidad. Una posible definición de framework es: conjunto de clases cooperantes que hace reusable un diseño para una clase específica de software. Un framework proporciona una guía arquitectónica para dividir el diseño en clases y definir sus responsabilidades y colaboraciones. Un framework dicta la arquitectura de la aplicación. (11)

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (12)

El bajo costo de aprendizaje de los fundamentos de PHP, le dio al lenguaje una reputación como el "idioma de aficionados". Poco a poco la situación cambió con la llegada de los frameworks MVC y otros frameworks RAD (desarrollo rápido de aplicaciones) y las bibliotecas o librerías diseñados para facilitar y acelerar el desarrollo. ROR (Ruby on Rails, framework escrito en lenguaje Ruby) trajo una nueva forma de ver el desarrollo Web: más rápido, más sencillo, más eficaz. Unos pocos años después de la aparición de ROR, salieron a la luz un gran número de frameworks de desarrollo rápido.

1.6.1 Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de dominio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

Características de Symfony.

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- 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.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.

Capítulo 1: Fundamentación Teórica

- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Symfony incorpora varios entornos de desarrollo diferentes e incluye varias herramientas que permiten automatizar las tareas comunes de la ingeniería del software:

- Las herramientas que generan automáticamente código han sido diseñadas para hacer prototipos de aplicaciones y para crear fácilmente la parte de gestión de las aplicaciones.
- El framework de desarrollo de pruebas unitarias y funcionales proporciona las herramientas ideales para el desarrollo basado en pruebas (“test-driven development”).
- La barra de depuración web simplifica la depuración de las aplicaciones, ya que muestra toda la información que los programadores necesitan sobre la página en la que están trabajando.
- La interfaz de línea de comandos automatiza la instalación de las aplicaciones entre servidores.
- Es posible realizar cambios “en caliente” de la configuración (sin necesidad de reiniciar el servidor).
- El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación.

En la actualidad existen otros frameworks de desarrollo muy populares, tales como CakePHP, CodeIgniter, Kumbia entre muchos otros. Para la implementación del módulo a desarrollar se ha utilizado como framework de desarrollo Symfony, después de hacer una minuciosa y exhaustiva investigación. Los resultados de la misma y el por qué se seleccionó Symfony para el desarrollo de este trabajo de diploma se muestra en la siguiente tabla.

Característica	Kumbia	Symfony	Cake
----------------	--------	---------	------

Capítulo 1: Fundamentación Teórica

MVC Nativo	X	X	X
Enrutamiento Avanzado	X	X	X
Scaffold(Generadores de código)	X	X	X
Scaffold Avanzado (Generadores de Formularios)	X		
Correo Electrónico	X	X	X
Mapeo Objeto-Relacional	X	X	X
Asociaciones ORM	X	X	X
Eventos ORM	X	X	X
Sistema de Plantillas	X	X	X
Integración con Smarty	X	X	X
Generación de Reportes (Múltiples Formatos)	X		
HTML Helpers	X	X	X
Plug-Ins (Integración Terceros)	X	X	X
i18n (Internacionalización)		X	X
Efectos Visuales	X	X	X
Integración AJAX	X	X	X
Componente de Sesiones	X	X	X
Loggers	X	X	
Documentación en Inglés		X	X

Documentación en Español	X	X	
ACL (Access Control Lists)	X	X	X
Soporte MySQL	X	X	X
Soporte PostgreSQL	X	X	X
Soporte Oracle	X	X	
Soporte SQL Server		X	
Soporte SQLite		X	X

Tabla 1 Comparación Frameworks de Desarrollo

1.7 Gestor de Base de Datos PostgreSQL

Una de las principales características que hace de PostgreSQL tan popular es precisamente que se distribuye bajo licencia GPL. Además provee las siguientes características:

- PostgreSQL, no tiene costo asociado a la licencia del software.
- Instalación Ilimitada
- Modelos de dominios más rentables con instalaciones a gran escala.
- No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
- Cumple completamente con ACID
- Cumple con ANSI SQL
- Integridad referencial
- Replicación (soluciones comerciales y no comerciales) que permiten la duplicación de bases de datos maestras en múltiples sitios de replica
- Interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby
- Reglas

- Vistas
- Triggers
- Unicode
- Secuencias
- Herencia
- Outer Joins.
- Sub-selects.
- Una API abierta.
- Procedimientos almacenados.
- Soporte nativo SSL.
- Lenguajes procedurales.
- Respaldo en caliente.
- Bloqueo a nivel mejor-que-fila.
- Índices parciales y funcionales.
- Autenticación Kerberos nativa.
- Soporte para consultas con UNION, UNION ALL y EXCEPT.
- Extensiones para SHA1, MD5, XML y otras funcionalidades.
- Herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL.
- Sistema de tipos de datos extensible para proveer tipos de datos definidos por el usuario, y rápido desarrollo de nuevos tipos.
- Funciones de compatibilidad para ayudar en la transición desde otros sistemas menos compatibles con SQL.

Extensible

El código fuente está disponible para todos sin costo.

Multiplataforma

PostgreSQL está disponible en casi cualquier plataforma Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.

Interfaz con diversos lenguajes

C, C++, Java, Delphi, Python, Perl, PHP, Bash entre otros.

La versión 8.2 es la que se utiliza para darle cumplimiento a los objetivos trazados en un inicio, ésta es una versión estable.

1.8 Consideraciones Finales

En este capítulo se realizó un breve estudio sobre los deportes de fantasía, su desarrollo a nivel mundial y en el ámbito nacional. Se realizó un estudio de las técnicas actuales de programación, tendencias de la programación web, así como algunas técnicas y metodologías utilizadas para la confección de la aplicación. Se realizó un estudio de los framework basados en MVC y se definieron algunas de las ventajas que brinda el framework escogido para la implementación.

Capítulo 2: Características del Sistema

En este capítulo, se realiza una breve descripción de los procesos del dominio relacionados con el procesamiento de la información estadística del módulo fantasía beisbol del sitio de beisbol de la UCI. Se especifican los actores del dominio y se definen los requerimientos funcionales y no funcionales, a partir de los cuales se representan los casos de uso del sistema, y se hace una descripción textual de los mismos.

2.1 Modelo del Dominio

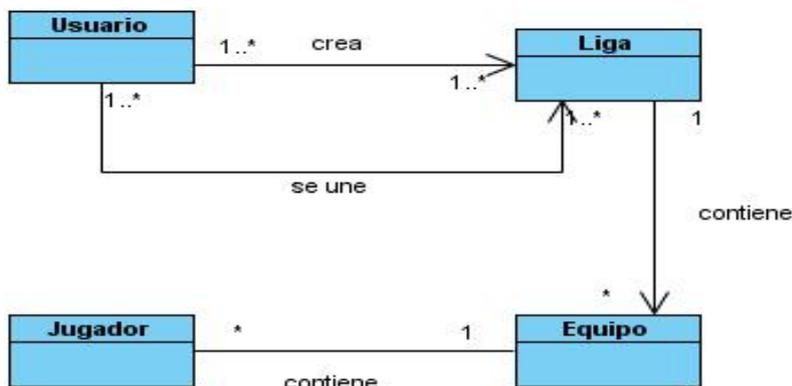


Fig.6 Modelo del Dominio

En la fantasía de béisbol se hace uso de un conjunto de documentos que son imprescindibles para el correcto funcionamiento de la misma. Algunos de estos, se utilizaron como apoyo para llevar a cabo el modelado de los procesos que fueron identificados como objeto de automatización:

Jugador: Recoge toda la información referente a los jugadores: archiva un identificador numérico, el nombre, los apellidos, el peso, su altura, una descripción sobre el mismo, si está activo, imágenes sobre él en caso de existir, mano de batear, mano de lanzar, equipo en el que comenzó, equipo en el que milita, y provincia a la que pertenece.

Equipo: Contiene información significativa sobre los equipos: archiva un identificador numérico, la sigla del equipo, el nombre del equipo y la imagen oficial.

Evento: Recoge toda la información sobre los eventos de béisbol que se puedan desarrollar: identificador numérico del evento, el nombre del mismo, la notación oficial, un identificador de etapas, un identificador de tipo de evento y si está activo.

Jugador Equipo: Recoge la información que define a un jugador en un equipo: contiene un

Capítulo 2: Características del Sistema

identificador numérico, si es novato, el número de uniforme, identificador del evento, identificador del jugador así como un identificador de la posición en la que juega.

Equipo Evento: Recoge la información principal de la participación de un equipo en un evento dado: contiene un identificador numérico, los juegos ganados como visitador, juegos ganados como home club, juegos perdidos como visitador, juegos perdidos como home club, identificador del evento, identificador del equipo e identificador del grupo.

2.2 Propuesta del Sistema

El módulo que se propone en el trabajo se acoplará al sitio de béisbol de la UCI que se encontrará brindando servicios sobre la red interna de la UCI, a través de ella todos los nodos tendrán la posibilidad de acceder al mismo. Es imprescindible que el usuario se autentique en el sitio para de esta forma interactuar con el módulo a través de la red siguiendo dos flujos que de forma general le permite acceder a reportes estadísticos y gestión de información de sus ligas, equipos y jugadores. El usuario autenticado en el módulo tiene la posibilidad de crear una liga con la un máximo de 16 participantes y una vez creada, debe esperar a que cada participante conforme su equipo para comenzar la liga. Cuando sea comenzada la liga los administradores de los equipos tienen la posibilidad de editar la información de sus equipos, agregar o eliminar jugadores que prevean puedan reportar la mayor cantidad de puntos durante cada jornada. El sistema brindará una serie de estadísticas que utilizadas inteligentemente por los usuarios puede aumentar su competitividad y rendimiento. El módulo que se propone, permitirá a un mismo usuario crear y unirse a ligas sin restricciones de tipo cuantitativo.

2.3 Especificación de los requisitos de software

Los requisitos de software en general, son condiciones o capacidades que necesitan los usuarios para resolver un problema o alcanzar un objetivo. Los mismos se pueden clasificar en requisitos funcionales y no funcionales. A continuación se muestra la captura de requisitos realizada para desarrollar la solución propuesta

2.3.1 Requerimientos Funcionales

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. A partir de los procesos de dominio estudiados y las actividades a automatizar identificadas se pueden definir los siguientes requisitos funcionales:

Capítulo 2: Características del Sistema

RF1: Gestionar liga

RF1.1: Crear liga: nombre, contraseña, cantidad de equipos de la liga

RF1.2: Modificar liga: nombre, contraseña y cantidad de equipos de la liga.

RF1.3: Eliminar liga a partir de un identificador.

RF2: Gestionar equipos

RF2.1: Crear equipos: nombre e imagen del equipo.

RF2.2: Modificar Equipos: nombre e imagen del equipo.

RF2.3: Eliminar equipos a partir de un identificador.

RF2.4: Mostrar equipos: nombre, puntos.

RF3: Gestionar Jugadores

RF3.1: Agregar Jugadores: lanzadores y bateadores a un equipo mediante un id.

RF3.2: Eliminar Jugadores a partir de un identificador.

RF3.3: Mostrar Jugadores: nombre, puntos, posición.

RF4: Realizar Preequipo: nombre de jugadores.

RF5: Integrar Liga: nombre de liga y contraseña.

RF6: Mostrar Mejores bateadores del día: nombre, estadísticas de bateo.

RF7: Mostrar mejores lanzadores del día: nombre, estadísticas de pitcheo.

RF8: Mostrar Ligas: nombre.

RF9: Mostrar Ligas con mejor promedio en total: nombre, puntos.

RF10: Mostrar Ligas con mejor promedio del día: nombre, puntos

RF11: Mostrar Equipos por Liga: nombre.

RF12: Mostrar Cambios realizados: identificador de cambio, jugador sale, jugador entra.

RF13: Mostrar Jugadores por Equipo: nombres, estadísticas individuales.

RF14: Mostrar Estadísticas del Equipo: nombre, puntos.

2.3.2 Requerimientos no Funcionales

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

Requerimiento de hardware en servidores.

La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables. La aplicación desarrollada será utilizada en los servidores de la Universidad de las Ciencias Informáticas que poseen las siguientes prestaciones:

Servidores de Base de datos: 1 HP-Blade, Procesador Intel® Core™2 *Extreme* Cuad-Core 32GB de memoria y 2x72GB de disco.

Servidores de Aplicaciones: 1 HP-Blade, Procesador Intel® Core™2 *Extreme* Cuad-Core 32GB de memoria y 2x72GB de disco.

Requerimiento de hardware en estaciones de trabajo.

Se seleccionaron estaciones de trabajo de 256Mb de memoria RAM y un microprocesador de 2.0Hz.

Usabilidad.

El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido.

Seguridad.

La información manejada por el sistema contará de protección ante intrusos y acceso no autorizados, será vista únicamente por aquellos usuarios que tengan derecho a verla. El sistema controlará los diferentes niveles de acceso y funcionalidad de usuarios al sitio, o sea, prioriza la identificación del usuario antes de que sea capaz de realizar cualquier acción sobre el sistema

Rendimiento.

Capítulo 2: Características del Sistema

El sistema deberá dar respuesta de menos de 3000 milisegundos para las operaciones que impliquen reportes y de menos de 1500 milisegundos para las operaciones de almacenamiento o eliminación.

Software.

El servidor debe correr en sistemas operativos Windows, Unix y Linux. Los clientes deberán disponer de un navegador web, estos pueden ser IE 7 o superior, Opera 9 superior, Google chrome 1 o superior y Firefox 2 o superior.

Restricciones de diseño e implementación.

El sistema se debe desarrollar utilizando el lenguaje de programación PHP 5.1 enmarcado en el framework de desarrollo Symfony 1.4. Como gestor de base de datos se utilizará PostgreSql 8.4 y para mostrar los datos generados se utilizará el lenguaje XHTML formateado por los estilos CCS. El servidor de aplicaciones web debe ser Apache 2.x.

Interfaz de Usuario.

El diseño de las interfaces que se propone para el módulo a implementar está basado en la premisa de organización y sencillez de la información a mostrar. Las interfaces están conformadas por un banner en la parte superior alegórico al béisbol, en el panel de la derecha se mantienen informaciones sobre los juegos del día y en el área central de la pantalla se muestran las informaciones del módulo utilizando principalmente el formato de tablas con encabezados con imágenes que alternan los colores negro y rojo utilizando el color blanco para los textos, para los datos de las tablas no se utilizan imágenes de fondo y la letra utilizada es de color automático. El sistema se desarrollará para una resolución de pantalla estándar de 1024 X 768 píxeles.

2.4 Modelo de Casos de Uso del Sistema

A continuación se muestran una serie de artefactos que componen el modelo de casos de uso del sistema, los cuales tienen como objetivo describir las relaciones y dependencias entre un grupo de casos de uso y los actores participantes en el proceso.

2.4.1 Definición de actores

Actor	Descripción
-------	-------------

Capítulo 2: Características del Sistema

<p>Usuario</p>	<p>Es el único actor que interactúa dentro del módulo. Una vez autenticado es el que maneja los flujos de la fantasía al hacer uso de todas sus funcionalidades, tanto de gestión como de visualización de estadísticas.</p>
----------------	--

2.4.2 Diagrama de Casos de Uso del Sistema

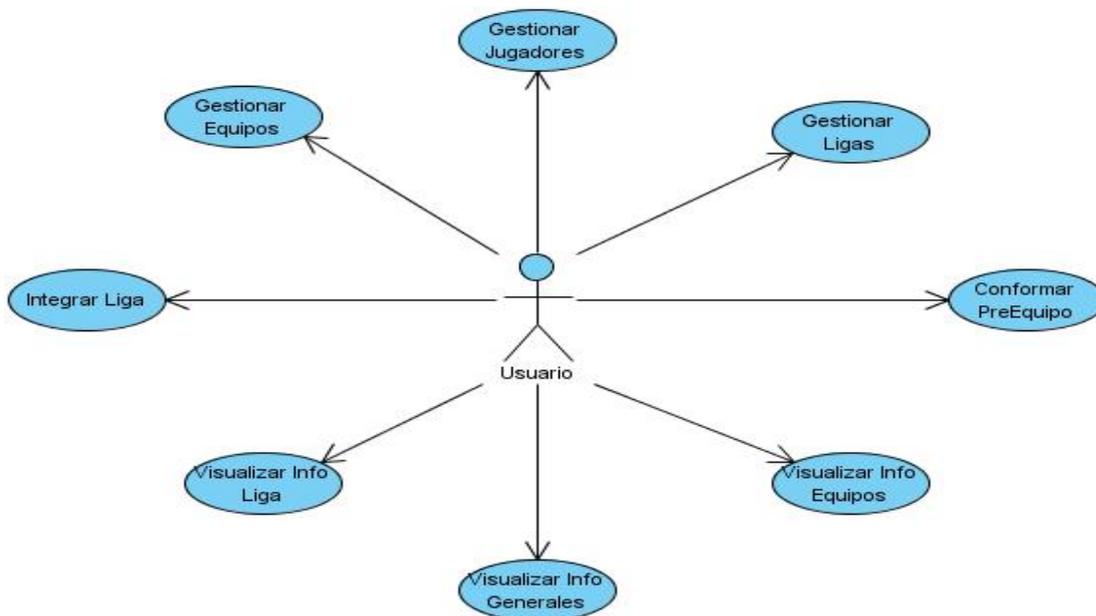


Fig.7 Diagrama de Casos de Uso del Sistema

2.4.3 Descripción de los Casos de Uso del Sistema

Las descripciones de casos de uso son reseñas textuales del caso de uso. Normalmente tienen el formato de una nota o un documento relacionado de alguna manera con el caso de uso, y explica los procesos o actividades que tienen lugar en el caso de uso. A continuación se muestran las descripciones de los casos de uso del sistema que se desarrolla.

CU Gestionar Liga

<p>Caso de Uso:</p>	<p>Gestionar Liga.</p>
<p>Actores:</p>	<p>Usuario (Inicia)</p>

Capítulo 2: Características del Sistema

Propósito	Permitir a los usuarios crear, eliminar o modificar ligas.
Resumen:	El caso de uso inicia cuando el usuario selecciona la opción crear liga, eliminar liga, o modificar ligas y termina cuando se ejecuta una de estas opciones.
Precondiciones:	Que el usuario esté autenticado.
Referencias	RF1
Prioridad	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 El usuario necesita Crear, Eliminar, y Modificar todos los datos de una liga	<p>El sistema ejecuta una de las siguientes acciones.</p> <ul style="list-style-type: none"> a) Si decide crear una liga ir a la sección "Crear Liga". b) Si decide eliminar una liga ir a la sección de "Eliminar Liga". c) Si decide modificar una liga ir a la sección "Modificar Liga".
Sección "Crear Liga."	
Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la opción crear Liga en la pantalla principal.	1.1 El sistema muestra el formulario a llenar para inscribir la nueva liga.
2 El usuario entra los datos correspondientes a la nueva liga	<p>2.1 El sistema verifica la validez de los datos.</p> <p>2.2 El sistema almacena los datos de la liga.</p> <p>2.3 El sistema redirecciona al usuario hacia la interfaz "mis ligas" y termina el caso de uso.</p>

Capítulo 2: Características del Sistema

Prototipo de Interfaz

The screenshot displays the 'Beisbol Cubano' website interface. At the top, it features a navigation bar with the text 'SITIO OFICIAL DEL BEISBOL EN LA UCI', 'BIENVENIDO: GIANNY MARTINEZ PERDOMO [SALIR]', and 'DESARROLLADO POR'. Below this is a header section with a background image of baseballs and the 'SNB' logo. The main title 'Beisbol Cubano' is prominently displayed, along with the tagline 'El mejor sitio web del beisbol cubano'. A secondary navigation menu includes links for 'PORTADA', 'NOTICIAS', 'LIDERES', 'CALENDARIO', 'POSICION', 'JUGADORES', 'FANTASY', and 'SERIE VIRTUAL'. A login form is present with fields for 'Nombre', 'Password', and 'Cant equipos', accompanied by an 'Agregar' button. The footer contains the website URL 'BEISBOL.uci.cu', a circular logo for the 'SERIE NACIONAL DE BEISBOL CUBA', and a copyright notice: '© 2004-2009 Universidad de Ciencias Informáticas. Todos los derechos reservados'.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	2.1 En caso de no ser correctos los datos el sistema informa que deben ser corregidos (nombre, contraseña o cantidad de usuarios).

Prototipo de Interfaz

Capítulo 2: Características del Sistema



Pos condiciones

La liga ha sido creada.

Flujo Normal de Eventos

Sección "Eliminar Liga."

Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la opción eliminar liga en el menú principal.	1.1 El sistema muestra las ligas que el usuario ha creado.
2 El usuario selecciona la liga a eliminar.	2.1 El sistema muestra el mensaje de confirmación "Está seguro de eliminar la liga" y en caso afirmativo elimina la liga. 2.2 Redirecciona hacia la página mis ligas y termina el caso de uso.

Prototipo de Interfaz

Capítulo 2: Características del Sistema

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	2.1 En caso de seleccionar el botón “Cancelar” no se realiza acción y se reenvía hacia la página “mis ligas”.

Pos condiciones Se ha eliminado la liga

Flujo Normal de Eventos

Sección “Modificar Liga.”

Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la opción modificar liga en la pantalla principal.	1.1 El sistema muestra los campos editables de la liga de la cuales es creador el usuario

Capítulo 2: Características del Sistema

2 El usuario modifica la información deseada (cantidad de equipos, nombre y contraseña)

2.1 El sistema verifica que los datos sean correctos.

2.2 El sistema modifica la liga.

2.3 El sistema informa sobre la correcta modificación de la liga y termina el caso de uso.

Prototipo de Interfaz

The screenshot shows the 'Beisbol Cubano' website interface. At the top, there is a navigation bar with links: PORTADA, NOTICIAS, LIDERES, CALENDARIO, POSICION, JUGADORES, FANTASY, and SERIE VIRTUAL. Below this is a table for league management:

Lugar	Equipo	Cambios	U.Jornada	Puntos	Pre-Equipo
1	Santiago de Cuba		15	30	[+]
2	Guantánamo		0	0	[+]

Below the table is a section for 'Cambios Realizados:' with columns for Fecha, Equipo, Entra, and Sale. To the right, there is a 'MIS LIGAS' sidebar with links for Liga 2, Santiago de Cuba, Historia, Liga SNB, and Holguin. The footer contains the website URL 'BEISBOL.uci.cu' and copyright information: '© 2004-2009 Universidad de Ciencias Informáticas. Todos los derechos reservados'.

Flujos Alternos

Acción del Actor

Respuesta del Sistema

2.1 En caso de no ser correctos los datos, el sistema informa que deben ser corregidos.

Prototipo de Interfaz

Capítulo 2: Características del Sistema

SITIO OFICIAL DEL BEISBOL EN LA UCI BIENVENIDO: JOSE LEANDRO CABRERA PEREZ [SALIR] DESARROLLADO POR

Beisbol Cubano
El mejor sitio web del beisbol cubano

PORTADA NOTICIAS LIDERES CALENDARIO POSICION JUGADORES FANTASY SERIE VIRTUAL

Nombre

Password
• Required.

Cant equipos

BEISBOL.uci.cu

© 2004-2009 Universidad de Ciencias Informáticas. Todos los derechos reservados

Pos condiciones	Se ha modificado la liga.
------------------------	---------------------------

CU Gestionar Equipo

Caso de Uso:	Gestionar equipo.
Actores:	Usuario (Inicia)
Propósito	Permitir a los usuarios crear, eliminar o modificar equipos.
Resumen:	El caso de uso inicia cuando el usuario selecciona la opción crear equipo, eliminar equipo, o modificar equipo y termina cuando es ejecutada una de esas opciones.
Precondiciones:	Usuario ya autenticado y dentro de una liga.
Referencias	RF2
Prioridad	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 El usuario necesita Crear, Eliminar, y Modificar los datos de un equipo	El sistema ejecuta una de las siguientes acciones.

Capítulo 2: Características del Sistema

	<ul style="list-style-type: none">d) Si decide crear un equipo ir a la sección "Crear Equipo".e) Si decide eliminar un equipo ir a la sección de "Eliminar Equipo".f) Si decide modificar un equipo ir a la sección "Modificar Equipo".
Sección "Crear Equipo."	
Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la opción crear Equipo dentro del área de ligas	1.2 El sistema muestra el formulario a llenar para inscribir el nuevo equipo.
2 El usuario entra los datos correspondientes al nuevo equipo (nombre e imagen).	<ul style="list-style-type: none">2.1 El sistema verifica que los datos sean correctos.2.2 El sistema almacena los datos del equipo.2.3 El sistema redirecciona al usuario hacia la página principal y termina el caso de uso.
Prototipo de Interfaz	

Capítulo 2: Características del Sistema



Flujos Alternos

Acción del Actor	Respuesta del Sistema
	2.1 En caso de no ser correctos los datos el sistema informa que deben ser corregidos como se muestra en el prototipo de interfaz.

Prototipo de Interfaz



Capítulo 2: Características del Sistema

Pos condiciones	El equipo ha sido creado.
Flujo Normal de Eventos	
Sección “Eliminar Equipo.”	
Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la liga de la cual es creador en la que quiere eliminar equipo	1.1 El sistema muestra los equipos de la liga seleccionada.
2 El usuario selecciona el equipo a eliminar.	2.1 El sistema pide confirmación sobre eliminación del equipo. 2.2 El sistema elimina el equipo y termina el caso de uso.
Prototipo de Interfaz	
 <p>The screenshot shows the website 'Beisbol Cubano' with a navigation menu (PORTADA, NOTICIAS, LIDERES, CALENDARIO, POSICION, JUGADORES, FANTASY, SERIE VIRTUAL). A table lists teams: Liga SNB, Lugar 1 (Pinar del Rio, 20 U.Jorn), Lugar 2 (Holguin, 5 U.Jorn). A modal dialog box asks 'Estas seguro de eliminar el Equipo?' with 'Aceptar' and 'Cancelar' buttons. A sidebar shows 'MIS LIGAS' with links for Santiago de Cuba, cpt, Historia, Liga SNB, and Holguin. The footer includes 'BEISBOL.uci.cu' and '© 2004-2009 Universidad de Ciencias Informáticas. Todos los derechos reservados'.</p>	
Pos condiciones	Se ha eliminado el Equipo.
Flujo Normal de Eventos	

Capítulo 2: Características del Sistema

Sección "Modificar Equipo."

Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la opción modificar equipo en la sección de ligas.	1.1 El sistema muestra los campos editables del equipo del cual es creador el usuario
2 El usuario modifica la información deseada (nombre e imagen).	2.1 El sistema verifica que los datos sean correctos. 2.2 El sistema modifica el equipo y termina el caso de uso.

Prototipo de Interfaz

The screenshot shows the website 'Beisbol Cubano' with a navigation menu and a table of teams. A callout points to a pencil icon in the 'Pre-Equipo' column of the table, labeled 'Modificar Equipo'.

Fecha	Equipo	Entra	Sale
Cambios Realizados:			

Lugar	Equipo	Cambios	U. Jornada	Puntos	Pre-Equipo
1	Pinar del Rio		20	25	 Listo
2	Holguin		5	10	Listo

Flujos Alternos

Capítulo 2: Características del Sistema

Acción del Actor	Respuesta del Sistema
	2.1 En caso de no ser correctos los datos el sistema informa que deben ser corregidos como se muestra en el prototipo de interfaz.

Prototipo de Interfaz



Pos condiciones	Se ha modificado la liga.
------------------------	---------------------------

CU Gestionar Jugadores

Caso de Uso:	Gestionar jugadores.
Actores:	Usuario (Inicia)
Propósito	Permitir a los usuarios agregar o eliminar jugadores.
Resumen:	El caso de uso inicia cuando el usuario selecciona la opción agregar jugadores, o eliminar jugadores con el objetivo de realizar cambios.
Precondiciones:	Usuario ya autenticado y dentro de un equipo

Capítulo 2: Características del Sistema

Referencias	RF3
Prioridad	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 El usuario necesita Agregar y Eliminar jugadores a un equipo	<p>El sistema ejecuta una de las siguientes acciones.</p> <p>g) Si decide insertar un jugador ir a la sección "Agregar Jugador".</p> <p>h) Si decide eliminar un jugador ir a la sección de "Eliminar Jugador".</p>
Sección "Agregar Jugador."	
Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la opción agregar jugador dentro de su equipo	1.3 El sistema muestra la lista de todos los peloteros disponibles.
2 El usuario selecciona el jugador a insertar en su equipo	<p>2.1 El sistema verifica que haya la capacidad en el roster del equipo para agregar otro jugador.</p> <p>2.2 El sistema verifica que la posición del jugador no está ocupada en el roster actual.</p> <p>2.3 Agrega el jugador al equipo y termina el caso de uso.</p>
Prototipo de Interfaz	

Capítulo 2: Características del Sistema



Pos condiciones El jugador ha sido agregado.

Flujo Normal de Eventos

Sección "Eliminar Jugador."

Acción del Actor	Respuesta del Sistema
1 El usuario selecciona el equipo del que es creador y quiere eliminar un jugador	1.1 El sistema muestra todos los jugadores que posee en su roster.
2 El usuario selecciona el jugador a eliminar.	2.1 El sistema elimina el jugador del roster. 2.2 El sistema redirecciona hacia la página "mi equipo" y termina el caso de uso.

Prototipo de Interfaz

Capítulo 2: Características del Sistema

Nombre y Apellidos	Puntos	Posición	Eliminar
Ernesto Molinet	0	Jugador de Cuadro	X
Yariel Duque	0	Jugador de Cuadro	X
Yuniet Dita	0	Jugador de Cuadro	X
Yoandry Garlobo	0	Jugador de Cuadro	X
Pavel Quezada	0	Jugador de Cuadro	X
Luis Felipe Rivera	0	Jugador de Cuadro	X
Rafael Orta	0	Jardinero	X
Serguei Perez	0	Jardinero	X
Ariel Hernandez	0	Jardinero	X
Wilfredo Aroche	0	Jardinero	X
Alexei Bell Quintero	0	Jardinero	X
Gerardo Miranda	0	Lanzador	X
Yadier pedroso	0	Lanzador	X
Ian Rendon	0	Lanzador	X
Yasmani Hernandez	0	Lanzador	X

Pos condiciones Se ha eliminado el jugador.

Flujo Normal de Eventos

CU Realizar Preequipo

Caso de Uso:	Realizar Preequipo
Actores:	Usuario (Inicia)
Propósito	Permitir a los usuarios conformar su preequipo antes de iniciar la liga.
Resumen:	El caso de uso inicia cuando el usuario selecciona la opción realizar preequipo y se termina una vez que confirma el preequipo realizado.
Precondiciones:	Usuario ya autenticado y dentro de un equipo
Referencias	RF4
Prioridad	Crítico.

Capítulo 2: Características del Sistema

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la opción conformar preequipo.	1.2 El sistema muestra la lista de todos los peloteros disponibles.
2 El usuario selecciona los jugadores que quiere tener en su equipo.	2.1 Se guardan los jugadores seleccionados en una lista temporal.
3 El usuario selecciona la opción Confirmar Preequipo	3.1 El sistema guarda el preequipo del usuario y termina el caso de uso.

Prototipo de Interfaz

The screenshot shows the website 'Beisbol Cubano' with a navigation menu and a player selection interface. The navigation menu includes: PORTADA, NOTICIAS, LIDERES, CALENDARIO, POSICION, JUGADORES, FANTASY, and SERIE VIRTUAL. The main content area is divided into two columns: 'Jugadores Disponibles' and 'Mi Pre - Equipo'. The 'Jugadores Disponibles' column lists 10 players with their names, teams, and positions, each with an 'AGREGAR >>' button. The 'Mi Pre - Equipo' column lists 4 players with their names, teams, and positions, each with an 'ELIMINAR' button. A 'CONFIRMAR PREEQUIPO' button is located below the 'Mi Pre - Equipo' column. The website header includes the text 'SITIO OFICIAL DEL BEISBOL EN LA UCI', 'BIENVENIDO: JOSE LEANDRO CABRERA PEREZ [SALIR]', 'DESARROLLADO POR BEISBOL.UCI.CU', and 'http://beisbol.cu'. The footer includes 'El mejor sitio web del beisbol cubano' and 'MIERCOLES 1 ESPERA O JUGANDO'.

Pos condiciones El preequipo ha sido conformado

CU Integrar Liga

Caso de Uso:	Integrar Liga
Actores:	Usuario (Inicia)
Propósito	Permitir a los usuarios unirse a una liga ya creada, introduciendole el nombre y la contraseña de la misma.

Capítulo 2: Características del Sistema

Resumen:	El caso de uso inicia cuando el usuario selecciona la opción integrar Liga y se termina una vez integrado a la liga o con el mensaje de error en caso de que los datos sean incorrectos.
Precondiciones:	Usuario ya autenticado y dentro de un equipo
Referencias	RF5
Prioridad	Crítico.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1 El usuario elige la opción de unirse a una liga	1.1 El sistema muestra la interfaz “unirse a liga”.
2 El usuario introduce los datos (nombre de liga y contraseña).	2.1 El sistema verifica que los datos introducidos sean correctos y el usuario pasa a pertenecer a la liga. El sistema redirecciona hacia la página “crear equipo” y termina el caso de uso.

Prototipo de Interfaz



Flujos Alternos

Capítulo 2: Características del Sistema

Acción del Actor	Respuesta del Sistema
	3.1 En caso de no cumplir con la contraseña o de no existir la liga se muestra un mensaje de error al usuario

Prototipo de Interfaz



Pos condiciones	Se ha unido a la liga
------------------------	-----------------------

CU Visualizar Informaciones Generales

Caso de Uso:	Visualizar Informaciones Generales
Actores:	Usuario (Inicia)
Propósito	Permitir a los usuarios conocer estadísticas generales para manejar su liga y su equipo.
Resumen:	El caso de uso inicia cuando el usuario selecciona la opción Mostrar mejores lanzadores del día, Mostrar Mejores bateadores del día, Mostrar Ligas, Mostrar Ligas con mejor promedio en total, Mostrar Ligas con mejor promedio del día

Capítulo 2: Características del Sistema

Precondiciones:	Que el usuario esté autenticado
Referencias	RF6, RF7, RF8, RF9, RF10
Prioridad	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 El usuario necesita consultar los mejores lanzadores o bateadores del día, mostrar las ligas existentes, mostrar las mejores ligas en un día específico y en total.	<p>El sistema ejecuta una de las siguientes acciones.</p> <ul style="list-style-type: none"> i) Si decide mostrar mejores lanzadores del día ir a la sección “Lanzadores del día”. j) Si decide mostrar mejores bateadores del día ir a la sección “Bateadores del día”. k) Si decide mostrar las ligas existentes ir a la sección “Mostrar Ligas”. l) Si decide mostrar las mejores ligas del día ir a la sección “Mejores Ligas del día”. m) Si decide mostrar ligas con más puntos ir a la sección “Mejores Ligas Total”.
Sección “Lanzadores del día.”	
Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la opción mostrar mejores lanzadores del día.	1.2 El sistema busca la información sobre los lanzadores del día y la muestra al usuario.

Capítulo 2: Características del Sistema

Prototipo de Interfaz

PRIMEROS LANZADORES		
LUGAR	BATEADOR	PUNTOS
1	Gonzalez Ledesma	83
2	Vera	63
3	Assiel Alvarez	23

Flujos Alternos

Acción del Actor

Respuesta del Sistema

1.2 En caso de ser una jornada de descanso se informa que no hay estadísticas para el día.

Prototipo de Interfaz

Pos condiciones

Se han mostrado los mejores lanzadores del día.

Flujo Normal de Eventos

Sección "Bateadores del día."

Acción del Actor

Respuesta del Sistema

1 El usuario selecciona la opción mostrar mejores bateadores del día.

1.2 El sistema busca la información sobre los bateadores del día y la muestra al usuario.

Prototipo de Interfaz

RENDIMIENTOS DE LOS JUGADORES

PRIMEROS BATEADORES		
LUGAR	BATEADOR	PUNTOS
1	Aguilera Portilla	3
2	Malleta	5
3	Gourriel	9

Flujos Alternos

Capítulo 2: Características del Sistema

Acción del Actor	Respuesta del Sistema
	1.2 En caso de ser una jornada de descanso se informa que no hay estadísticas para el día.

Pos condiciones	Se han mostrado los mejores bateadores del día.
------------------------	---

Flujo Normal de Eventos

Sección “Mostrar Ligas.”

Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la opción mostrar ligas	1.1 El sistema muestra la lista de todas las ligas creadas en el sitio.

Prototipo de Interfaz

Primeras Ligas	
Liga SNB	35
Liga 2	30
cpt	0
lokilla	0

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	1.1 En caso de no haber ligas creadas, se muestra el mensaje “No existen Ligas”.

Prototipo de Interfaz

Sección “Mejores Ligas del día.”

Acción del Actor	Respuesta del Sistema

Capítulo 2: Características del Sistema

<p>1 El usuario selecciona la opción mostrar mejores ligas del día.</p>	<p>1.1 El sistema calcula los puntos obtenidos para todas las ligas existentes en el sitio.</p> <p>1.2 El sistema muestra las mejores ligas por puntos del día al usuario y termina el caso de uso.</p>
---	---

Prototipo de Interfaz

SEGUIMIENTOS DE LOS EQUIPOS EN LAS LIGAS

ULTIMA JORNADA		
LUGAR	EQUIPO	PUNTOS
1	Pinar del Rio	20
2	Santiago de Cuba	15
3	Holguin	5
4	Guantánamo	0
5	Algebra	0
6	Historia	0
7	SistemaOp	0

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	<p>1.2 En caso de ser una jornada de descanso se informa que no hay estadísticas para el día.</p>

Pos condiciones Se han mostrado las mejores ligas del día.

Flujo Normal de Eventos

Sección "Mejores Ligas Total."

Acción del Actor	Respuesta del Sistema
<p>1 El usuario selecciona la opción mostrar mejores ligas en total.</p>	<p>1.1 El sistema suma los puntos obtenidos durante todas las jornadas para todas las ligas existentes en el sitio.</p>

Capítulo 2: Características del Sistema

	1.2 El sistema muestra las mejores ligas por puntos al usuario y termina el caso de uso.																											
Flujos Alternos																												
Acción del Actor	Respuesta del Sistema																											
	1.2 En caso de no existir ligas se le informa al usuario que debe crear ligas primero																											
Prototipo de Interfaz																												
<table border="1"> <thead> <tr> <th colspan="3">GENERAL</th> </tr> <tr> <th>LUGAR</th> <th>EQUIPO</th> <th>PUNTOS</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Santiago de Cuba</td> <td>30</td> </tr> <tr> <td>2</td> <td>Pinar del Rio</td> <td>25</td> </tr> <tr> <td>3</td> <td>Holguin</td> <td>10</td> </tr> <tr> <td>4</td> <td>Guantánamo</td> <td>0</td> </tr> <tr> <td>5</td> <td>Algebra</td> <td>0</td> </tr> <tr> <td>6</td> <td>Historia</td> <td>0</td> </tr> <tr> <td>7</td> <td>SistemaOp</td> <td>0</td> </tr> </tbody> </table>		GENERAL			LUGAR	EQUIPO	PUNTOS	1	Santiago de Cuba	30	2	Pinar del Rio	25	3	Holguin	10	4	Guantánamo	0	5	Algebra	0	6	Historia	0	7	SistemaOp	0
GENERAL																												
LUGAR	EQUIPO	PUNTOS																										
1	Santiago de Cuba	30																										
2	Pinar del Rio	25																										
3	Holguin	10																										
4	Guantánamo	0																										
5	Algebra	0																										
6	Historia	0																										
7	SistemaOp	0																										
Pos condiciones	Se han mostrado las mejores Ligas																											

CU Visualizar Informaciones Liga

Caso de Uso:	Visualizar Informaciones Liga
Actores:	Usuario (Inicia)
Propósito	Permitir a los usuarios conocer estadísticas sobre la liga a la que pertenece, cambios y equipos de la misma.
Resumen:	El caso de uso inicia cuando el usuario selecciona la opción mostrar equipos por ligas o mostrar cambios realizados y termina cuando se visualizan estas informaciones.
Precondiciones:	Que el usuario esté autenticado y tenga un equipo dentro de una liga
Referencias	RF11, RF12

Capítulo 2: Características del Sistema

Prioridad	Crítico.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1 El usuario elige la opción Visualizar Informaciones de Liga.	<p>El sistema ejecuta una de las siguientes acciones.</p> <ul style="list-style-type: none"> a) Si decide ver equipos de una Liga ir a la sección “Mostrar Equipos por Ligas”. b) Si decide ver cambios realizados en su liga ir a la sección “Mostrar Cambios realizados”. 	
Sección “Mostrar Equipos por Ligas.”		
Acción del Actor	Respuesta del Sistema	
1 El usuario selecciona la opción mostrar equipos por ligas.	1.1 El sistema muestra la lista de todas las ligas creadas en el sitio.	
2 El usuario selecciona una liga determinada.	2.1 El sistema consulta y muestra los equipos que contiene esa liga y termina el caso de uso.	
Prototipo de Interfaz		

Capítulo 2: Características del Sistema

Debug toolbar 1.4.1 Config config View Layer view Log logs Memory 5632.0 KB Time 1292 m

Beisbol Cubano
El mejor sitio web del beisbol cubano

PORTADA NOTICIAS LIDERES CALENDARIO POSICION JUGADORES FANTASY SERIE VIRTUAL

Liga: cpt [EDITAR LIGA]

Lugar	Equipo	Cambios	U.Jornada	Puntos	Pre-Equipo
1	Algebra		0	0	Listo
2	Historia		0	0	Listo
3	SistemaOp		0	0	Listo

MIS LIGAS
cpt
>> Algebra

Pos condiciones Se han mostrado los equipos de la liga

Flujo Normal de Eventos

Sección "Mostrar Cambios realizados."

Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la liga a la que pertenece	1. 1 El sistema muestra la interfaz de Liga donde aparecen todos los equipos.
2 El usuario selecciona la opción ver cambios realizados	2.1 El sistema busca para esa liga, los cambios realizados por cada equipo. 2.2 Visualiza los cambios por fecha en que se realizaron y termina el caso de uso.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	2.1 En caso de no haberse efectuado ningún cambio en los equipos se le informa de esto al usuario.

Prototipo de Interfaz

Capítulo 2: Características del Sistema

Cambios Realizados:			
Fecha	Equipo	Entra	Sale
2010-06-18	Historia	Yoandry Garlobo	Yoandry Garlobo
2010-06-18	Historia	Alexei Bell Quintero	Yoandry Urgelles
2010-06-18	Historia	Pavel Quezada	Leris Aguilera Portilla
2010-06-18	Historia	Luis Felipe Rivera	Leris Aguilera Portilla
2010-06-19	Historia	Leris Aguilera Portilla	Ernesto Molinet
2010-06-19	Algebra	Yoandry Urgelles	Yoelvis Leiva

 BEISBOL.uci.cu

Pos condiciones Se han mostrado los cambios en la liga

Capítulo 2: Características del Sistema

CU Visualizar Informaciones Equipos

Caso de Uso:	Visualizar Informaciones Equipos
Actores:	Usuario (Inicia)
Propósito	Permitir a los usuarios conocer datos sobre sus equipos y los jugadores que lo integran.
Resumen:	El caso de uso inicia cuando el usuario selecciona la opción mostrar jugadores del equipo o mostrar estadísticas del equipo y finaliza cuando el sistema visualiza estas informaciones.
Precondiciones:	Que el usuario esté autenticado y tenga un equipo dentro de una liga
Referencias	RF13, RF14
Prioridad	Crítico.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1 El usuario elige la opción Visualizar Informaciones de Equipo.	El sistema ejecuta una de las siguientes acciones. c) Si decide ver jugadores del equipo ir a la sección "Mostrar Jugadores Equipo". d) Si decide ver estadísticas de los jugadores ir a la sección "Mostrar Estadísticas Equipo".

Sección "Mostrar Jugadores Equipo."

Capítulo 2: Características del Sistema

Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la opción mostrar jugadores del equipo.	1.1 El sistema muestra la lista de todos los equipos de la liga.
2 El usuario selecciona un equipo determinado.	2.1 El sistema consulta y muestra los jugadores que contiene ese equipo y termina el caso de uso.

Prototipo de Interfaz

The screenshot shows the official website of Cuban Baseball (Beisbol Cubano). The header includes the site name, a welcome message for Jose Leandro Cabrera Perez, and the developer information (BEISBOL.uci.cu). The main navigation menu includes: PORTADA, NOTICIAS, LIDERES, CALENDARIO, POSICION, JUGADORES, FANTASY, and SERIE VIRTUAL. The current page displays a list of players with the following data:

Nombre y Apellidos	Puntos	Posición	Eliminar
Ernesto Molinet	0	Jugador de Cuadro	X
Yariel Duque	0	Jugador de Cuadro	X
Yuniet Dita	0	Jugador de Cuadro	X
Yoandry Garlobo	0	Jugador de Cuadro	X
Pavel Quezada	0	Jugador de Cuadro	X
Luis Felipe Rivera	0	Jugador de Cuadro	X
Rafael Orta	0	Jardinero	X
Serguei Perez	0	Jardinero	X
Ariel Hernandez	0	Jardinero	X
Wilfredo Aroche	0	Jardinero	X
Alexei Bell Quintero	0	Jardinero	X
Gerardo Miranda	0	Lanzador	X
Yadier pedroso	0	Lanzador	X
Ian Rendon	0	Lanzador	X
Yasmani Hernandez	0	Lanzador	X

The interface also shows a search bar for the league name, a date indicator (MIERCOLES 11), and a status indicator (ESPERA / JUGANDO).

Pos condiciones Se han mostrado los jugadores del equipo

Flujo Normal de Eventos

Sección "Mostrar Estadísticas Equipo."

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

Capítulo 2: Características del Sistema

1 El usuario selecciona la opción mostrar estadísticas del equipo.	1. 1 El sistema muestra la lista de todos los equipos de la liga.
2 El usuario selecciona un equipo determinado.	2.1 El sistema calcula para ese equipo, las estadísticas conjuntas. 2.2 Visualiza las estadísticas del equipo y termina el caso de uso.

Prototipo de Interfaz

The screenshot shows the 'Beisbol Cubano' website interface. At the top, there is a navigation menu with options: PORTADA, NOTICIAS, LIDERES, CALENDARIO, POSICION, JUGADORES, FANTASY, and SERIE VIRTUAL. Below the menu, the page displays the following information:

- Nombre de la Liga: cpt
- Selección de Jugadores para el día: 20:06:10
- Director: Gianni
- Equipo: Algebra
- Puntos: 0

The interface is divided into two main sections: BATEADORES (Batters) and LANZADORES (Pitchers).

BATEADORES

Pos	Jugador	VB	H	2B	3B	HR	CI	C	BR	CR	E	PUNTOS
Receptor	Dennis Laza Spencer											
Jardinero	Dennis Laza Spencer											
Jugador de Cuadro	Dayan Garcia											
Jugador de Cuadro	Roberto Zulueta											
Jardinero	Carlos A Tabares padilla											
Jugador de Cuadro	Rudy Reyes											
Jugador de Cuadro	Leugim Barroso											
Jardinero	Vladimir Garcia											
Jardinero	Joan Chaviano											
Jardinero	Yoandry Urgelles											

LANZADORES

Pos	Jugador	INN	JG	JP	JS	L	SO	BB	H	CL	PUNTOS
Lanzador	Miguel Alfredo Gonzalez										
Lanzador	Jonder Martinez										
Lanzador	Armando Rivero										
Lanzador	Freddy Assiel Alvarez										

At the bottom right, it shows 'Puntos ultima jornada 0'.

Pos condiciones Se han mostrado las estadísticas del equipo.

Consideraciones Finales

En el capítulo se estableció y describió como esta compuesto el dominio en que se enmarca el sistema, y los requisitos funcionales y no funcionales. Se identificaron los actores del sistema y se describieron los casos de uso del sistema.

Capítulo 3: Construcción de la solución propuesta

3.1 Introducción

En este capítulo se realizarán los flujos de análisis, diseño e implementación del sistema, donde se definen los diagramas de clases del análisis y del diseño por casos de uso, se muestra la interacción entre el actor y el sistema, mediante los diagramas de colaboración y se muestra el diseño de la base de datos además de una descripción de sus tablas. También se presentan los patrones de diseño utilizados, así como los métodos de validación y manejo de errores, cuestiones relacionadas con la seguridad y la interfaz del módulo implementado. Se realiza el diagrama de componentes y el diagrama de despliegue y para finalizar se realiza una breve explicación de los estándares de codificación utilizados.

3.2 Análisis

3.2.1 Clases del Análisis

Como resultado del desarrollo de cada uno de los casos de usos más fundamentales se han desarrollado los diagramas de clases de análisis. Estos representan como se lleva a cabo el desarrollo de los diferentes casos de uso dentro de la aplicación. A continuación se muestran las clases del análisis de los casos de uso Gestionar Equipo, Integrar Liga y Gestionar Jugadores. El resto de los diagramas se pueden localizar en el anexo # 2 del documento.

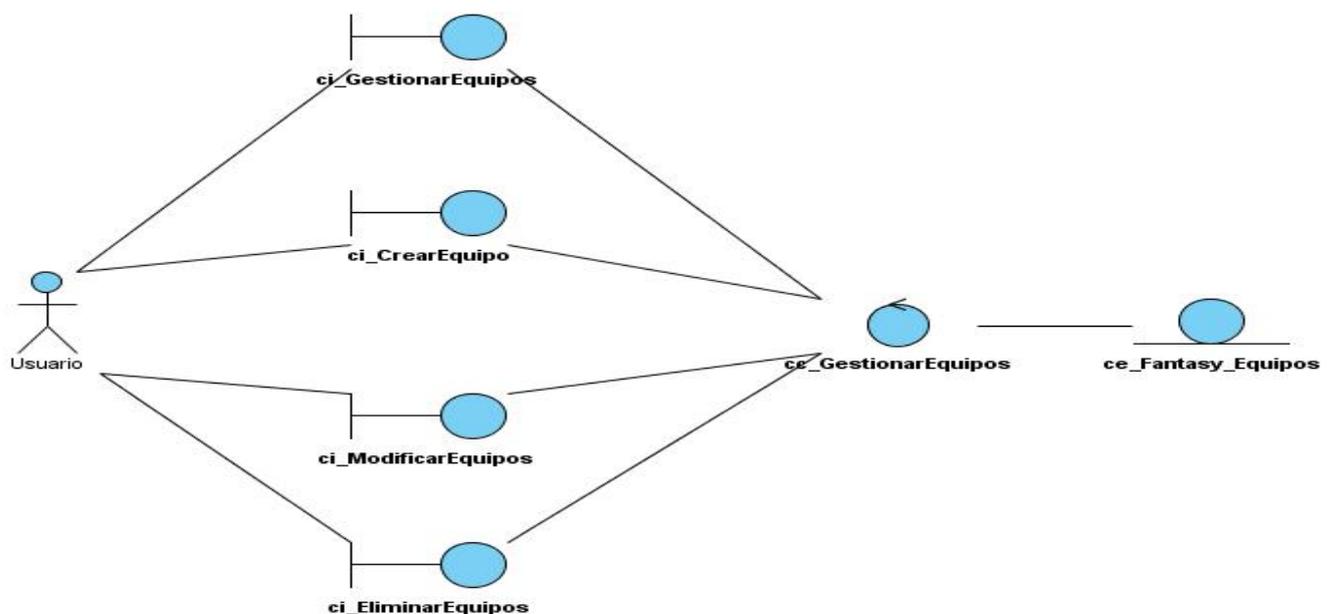


Fig8. DCA Gestionar Equipos



Fig9. DCA Integrar Liga

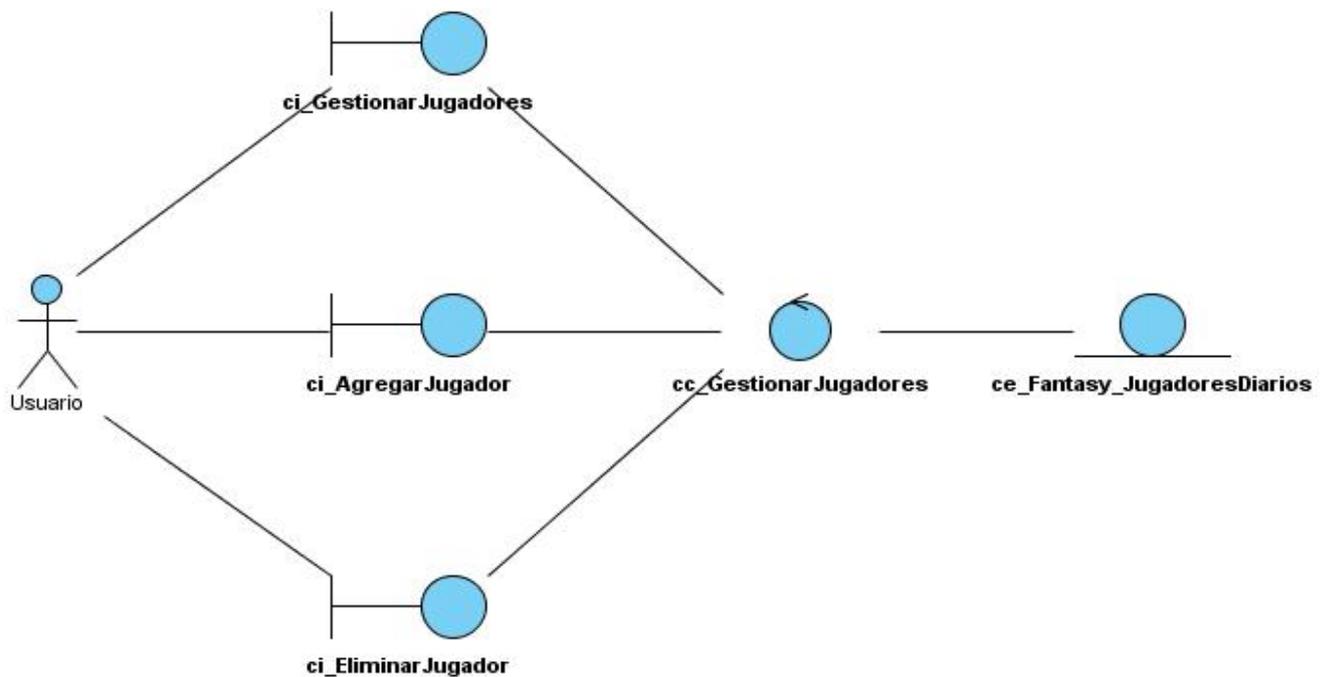


Fig10. DCA Gestionar Jugadores

3.2.2 Diagrama de Colaboración

Para modelar los aspectos dinámicos de este sistema se confeccionaron los diagramas de colaboración por cada sección de los casos de uso. A continuación se incorporan los diagramas de colaboración de los casos de uso Gestionar Equipo, Integrar Liga y Gestionar Jugadores; cada uno de ellos desglosados en sus respectivas secciones. El resto de los diagramas se pueden localizar en el anexo #3.

Capítulo 3: Construcción de la solución propuesta

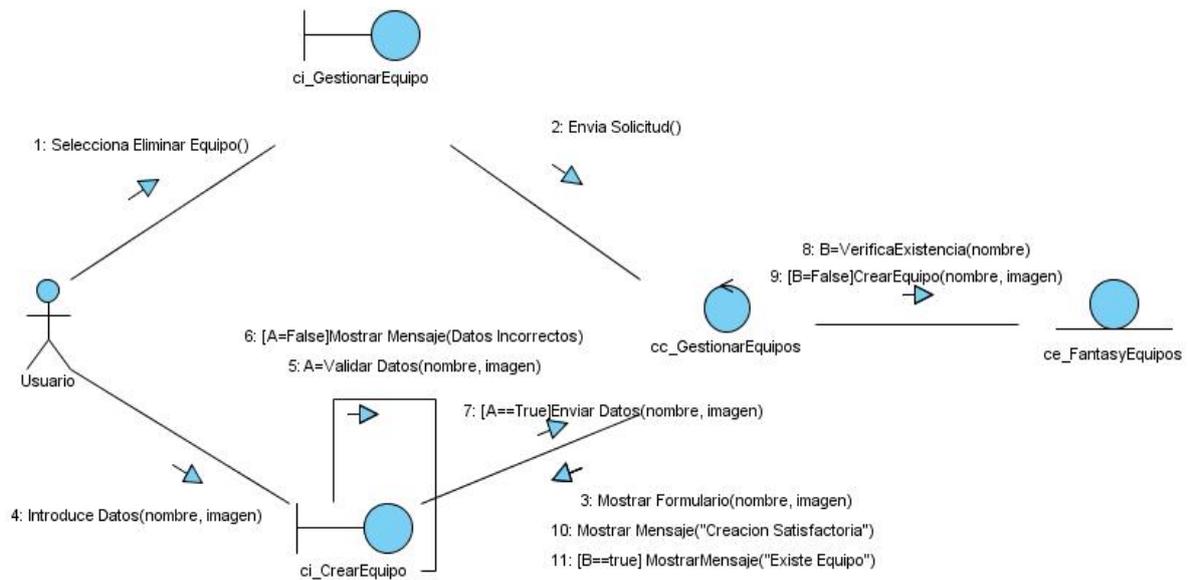


Fig.11 DC Gestionar Equipo Sección Crear

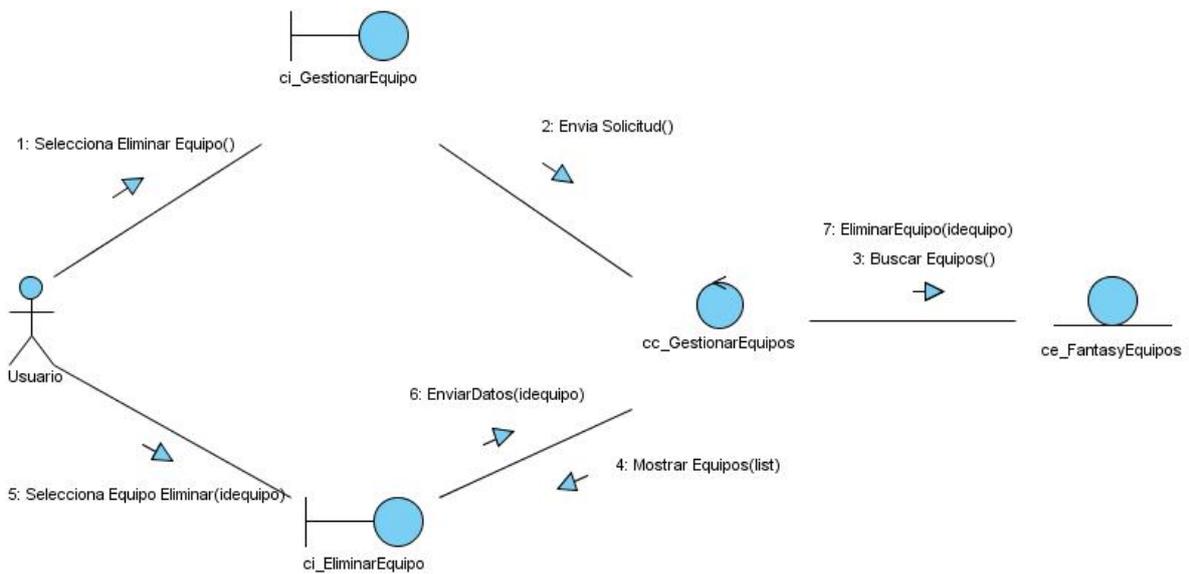


Fig.12 DC Gestionar Equipo Sección Eliminar

Capítulo 3: Construcción de la solución propuesta

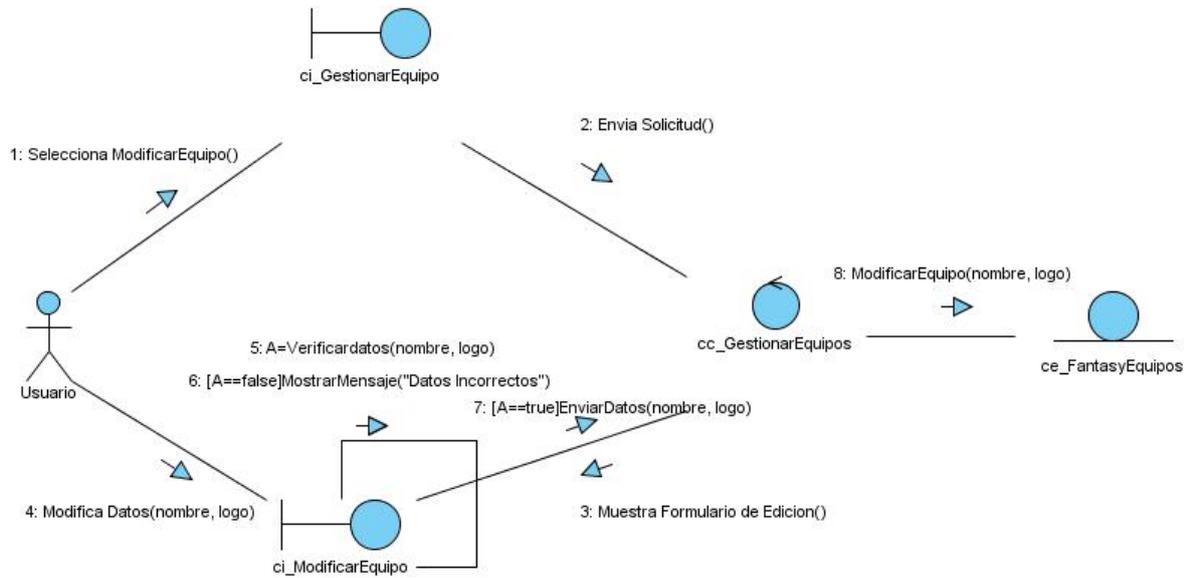


Fig. 13 DC Gestionar Equipo Sección Modificar

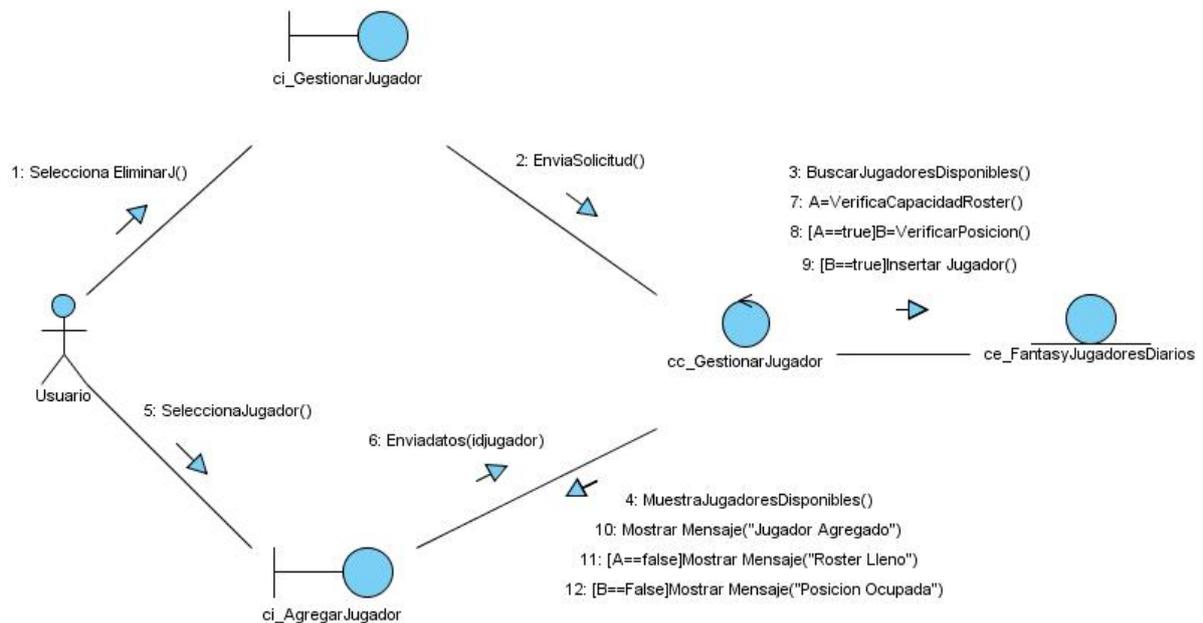


Fig. 14 DC Gestionar Jugadores Sección Crear

Capítulo 3: Construcción de la solución propuesta

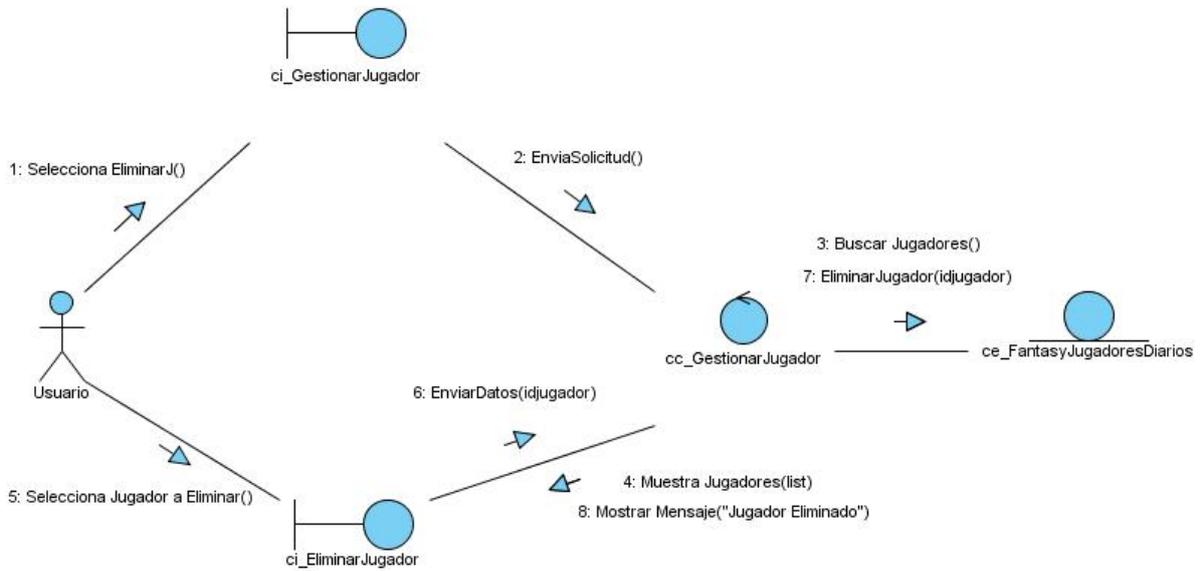


Fig.15DC Gestionar Jugadores Sección Eliminar

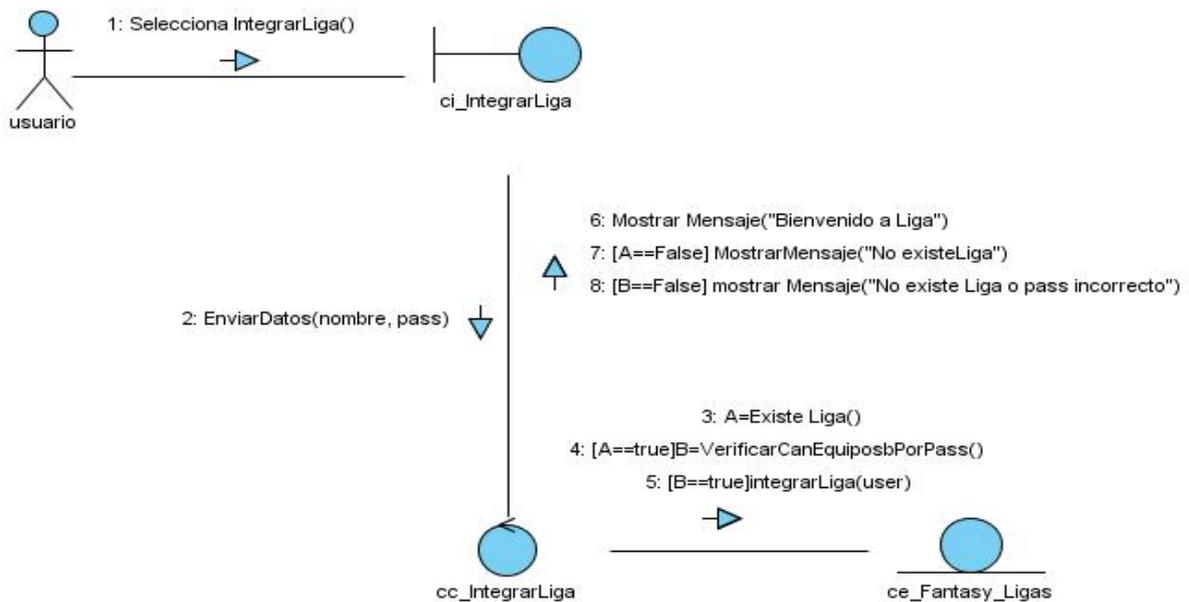


Fig.16 DC Integrar Liga

3.3 Diseño

3.3.1 Clases del Diseño

El Modelo de Diseño es un proceso para la definición detallada de un sistema con el fin de la realización física de los casos de uso para cubrir las funciones que realizará el sistema y otras restricciones del entorno de implementación que tienen impacto en el mismo, por tanto en él se definen las clases del diseño que conformarán el sistema que se va a implementar.

Páginas clientes: Son las páginas encargadas de permitir a los usuarios interactuar con el sistema tanto para hacer solicitudes como para que sean mostradas las respuestas a las mismas.

Páginas servidoras: Son las encargadas de la construcción de forma dinámica de las páginas clientes y sirven de enlace entre estas y el resto de las clases.

Los diagramas de clases del diseño que componen el modelo de diseño se muestran en el anexo # 4.

3.3.2 Seguridad

La seguridad se lleva a cabo mediante la autenticación por parte de los usuarios que trabajan con la aplicación. Como precondition para acceder al módulo implementado es requerida la autenticación de usuarios. La contraseña de los usuarios será encriptada para evitar que pueda ser copiada fácilmente. Además el framework de desarrollo utiliza implícitamente mecanismos de seguridad a través del controlador frontal que carga una serie de políticas de seguridad almacenados en el archivo security.yml de la configuración del módulo que permite definir a que acciones tiene derecho cada usuario. En symfony el mecanismo de seguridad puede ser entendido como una cadena de filtros, por el que debe pasar cada petición antes de ejecutar la acción y en cada filtro se van depurando posibles vulnerabilidades.

Filtros

1. sfRenderingFilter: encargado de renderizar la vista.
2. sfBasicSecurityFilter: chequea la seguridad de cada petición.
3. sfCacheFilter: controla el mecanismo de cache del framework.
4. sfCommonFilter: añade los ficheros javascript y css a la respuesta.
5. sfFlashFilter: elimina las variables flash de la sesión.

6. `sfExecutionFilter`: se encarga de validar los parámetros de la petición, la ejecución de la acción y de la vista.

Symfony elimina internamente, debido a su estructura de filtros los riesgos de la vulnerabilidad más atacada en los sistemas web, XSS (cross-site scripting). Los ataques CSRF también son muy populares, por lo que a partir de la versión 1.1 de Symfony también se ha incluido un mecanismo de protección contra los ataques CSRF mediante el filtro CSRF.

3.3.3 Patrones de diseño que se emplean

Patrón Modelo-Vista-Controlador(MVC)

MVC (model-view-controller o modelo-vista-controlador) es una arquitectura que integra tres niveles de diseño siguientes:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de dominio.
- La vista 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 realiza los cambios apropiados en el modelo o en la vista.

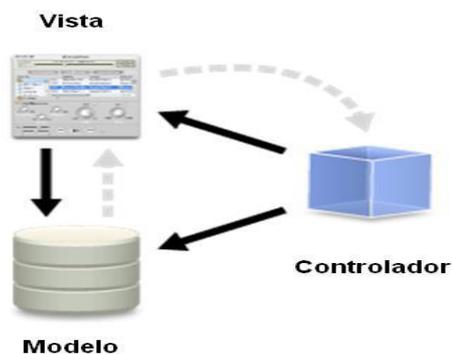


Fig.16 Modelo Vista Controlador

Como se aprecia en la figura 16 la arquitectura MVC separa la lógica de dominio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a

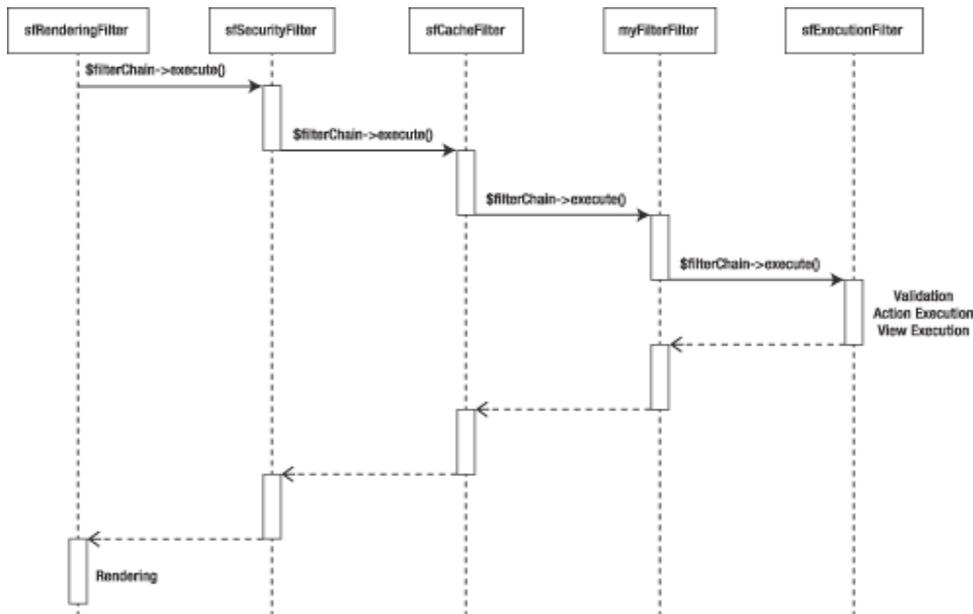


Fig.18 Secuencia de Filtros en Symfony

Patrón Controlador Frontal

Un patrón que es utilizado implícitamente en las aplicaciones desarrolladas con el framework de desarrollo symfony es el patrón Controlador Frontal. Que consiste en la existencia de un único controlador en el sistema que soluciona el problema de la descentralización presente en el patrón MVC. Es el único punto de entrada a la aplicación y realiza las siguientes operaciones:

1. Define las constantes del núcleo.
2. Localiza las librerías de Symfony.
3. Carga e inicializa las clases del núcleo del framework.
4. Carga la configuración.
5. Decodifica la URL de la petición para determinar la acción a ejecutar y los parámetros de la petición.
6. Si la acción no existe, redirecciona a la acción error 404.
7. Activa los filtros.
8. Ejecuta los filtros, primera pasada.

9. Ejecuta la acción y produce la vista.
10. Ejecuta los filtros, segunda pasada.
11. Muestra la respuesta.

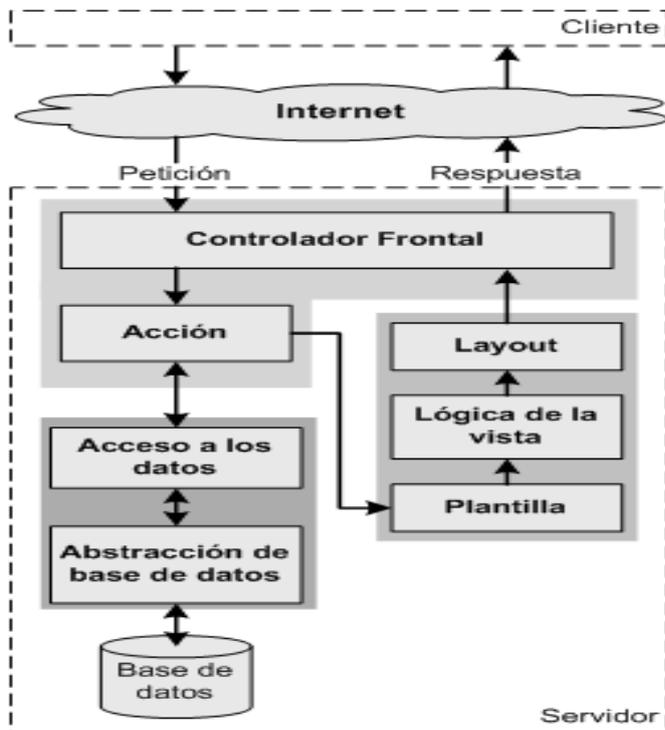


Fig. 19 Flujo básico de una petición a symfony.

3.3.4 Métodos de validación y manejo de errores

Symfony posee un mecanismo de validación en el servidor a través del uso de ficheros de validación y una serie de validadores, llamados `sfvalidators`, que permite hacer este proceso más fácil y extensible, ya que desde fases tempranas de la implementación se pueden detectar posibles errores principalmente en cuanto a la entrada de datos por parte de los usuarios. No obstante hay casos especiales donde es necesario hacer la validación manualmente o combinar ambos métodos. Una característica importante de Symfony en este sentido es que permite el relleno automático de datos (*repopulation*), lo que asegura la obtención de datos correctos y mejora la experiencia de los usuarios. Tal como se mencionó anteriormente las validaciones se realizan como parte de la ejecución del filtro `sfExecutionFilter`.

Para el tratamiento de errores se utilizará el mecanismo provisto por PHP y las clases que para este fin Symfony posee, principalmente los métodos `hasError()` y `getError()` del objeto `sfrequest`.

3.4 Diseño de Base de Datos

Los sistemas web que en la actualidad existen son desarrollados utilizando tecnologías cliente-servidor. Estos constan en la mayoría de los casos con bases de datos robustas que son las encargadas de almacenar todos los datos que en la misma se muestran. Debido a la complejidad y el cúmulo de información que se maneja se ha desarrollado una base de datos para almacenar toda la información que se controla dentro del módulo web. La misma viene incluida dentro de los anexos. (Ver Anexo # 6).

3.4.1 Descripción de las Tablas

Las principales entidades de la base de datos son descritas a continuación para un mejor entendimiento de los campos de las mismas. Para consultar la totalidad de las tablas descritas dirigirse al Anexo # 7

Nombre Fantasy_beisbol_cambios_jugadores		
Descripción Almacena los cambios realizados en la liga por los equipos participantes		
Atributo	Tipo	Descripción
Id	Serial	Identificador de la tabla
Fecha	Date	Almacena la fecha del cambio
Id_fantasy_beisbol_equipos	Integer	Id del equipo que realiza el cambio
Id_jugador_equipo_entra	Integer	Id del jugador que entra
Id_jugador_equipo_sale	Integer	Id del jugador que sale

Tabla 2 Fantasy_beisbol_cambios_jugadores

Nombre Fantasy_beisbol_equipos		
Descripción Contiene los equipos que pertenecen a una fantasy		
Atributo	Tipo	Descripción

Id	Serial	Identificador de la tabla
nombre	Character varying (255)	Nombre del equipo del creador
listo	Boolean(false)	Establece cuando el equipo este listo
imagen	Character varying (255)	Imagen del equipo
puntos	Integer	Puntos acumulados
Id_fantasy_beisbol_ligas	Integer	Id de la liga a que pertenece
Id_usuario	Integer	Id del usuario del equipo

Tabla 3 Fantasy_beisbol Equipos

3.5 Diagrama de Componentes

Un componente es una parte modular de un sistema, desplegable y reemplazable que típicamente contiene clases y puede ser implementado por uno o más artefactos (ficheros ejecutables, binarios, etc.). Es un elemento de implementación que representa algo físico, ya sea ficheros o archivos, y son creados para poner el código, ya sea código fuente, código binario o código ejecutable.

Los mismos son creados, modificados o eliminados en el proceso de implementación. Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. Es un diagrama que muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones. En el Anexo # 8 se muestra el diagrama de componentes de la aplicación.

3.6 Diagrama de Despliegue

El Diagrama de Despliegue es un grafo de nodos unidos por conexiones de comunicación que muestra las relaciones físicas entre los componentes de hardware que forman la topología sobre la que se ejecuta el sistema y la distribución de sus partes. Este refleja tanto la distribución del sistema con los nodos físicos (ordenadores) como la correspondencia que

tienen los componentes con los nodos. En el sistema a desarrollar se pueden identificar como procesadores uno para la base de datos, otro para que funcione como servidor Web y uno que representa en general todos los clientes que puedan conectarse a la aplicación. Los nodos que representan el servidor de Base de Datos y el Servidor Web deben ser capaces de brindar una serie de prestaciones de hardware como las que se plantean: Servidor1 HP-Blade, Procesador Intel® Core™2 *Extreme* Cuad-Core 32GB de memoria y 2x72GB de disco. En el Anexo # 5 se muestra el diagrama de Despliegue del módulo Web desarrollado.

3.7 Estándares de Codificación

Los estándares de codificación definen la forma en que se va a escribir el código durante la programación. Esto es de gran ayuda, pues contribuye a reducir los errores y propicia un mejor entendimiento y comprensión del código escrito. A continuación se muestra los estándares definidos por los desarrolladores del trabajo.

Declaraciones: los nombres utilizados en las declaraciones deben estar acorde con su significado en la aplicación.

Atributos: deben comenzar con letra minúscula, si son atributos compuestos comienzan con minúscula y la segunda palabra con mayúscula.

Ejemplos: atributo simple: string nombre

atributo compuesto: string nombreEquipo.

Métodos: comienzan con letra minúscula, si son compuestos comienzan con minúscula y la segunda palabra con mayúscula.

Ejemplos: simple: public function Adicionar(\$id)

Compuesto: public function existeEquipo(\$nombre)

Clases: deben comenzar con letra mayúscula.

Comentarios: para comentar una línea se utiliza //, si es un bloque de texto se comienza con /* y se termina con */.

Instrucciones: las llaves que se utilizan para abrir y cerrar un bloque de instrucciones deben estar al mismo nivel.

Variables de control de ciclos: las variables utilizadas para los ciclos serán i, j, k.

Estilo de codificación

Existen varios estilos reconocidos por equipos que se dedican al desarrollo de software, durante la implementación se utilizó el estilo K&R ya que es el más usado en el lenguaje C y PHP. Se trata de abrir la llave en la misma línea de declaración de la orden, indentando los siguientes pasos al mismo nivel que la llave y cerrando la llave en el mismo nivel que la declaración. Normalmente las tabulaciones en Windows son de 4 espacios, cuando las tabulaciones tienen 8 espacios se trata del estilo BSD o KNF, el código fuente del kernel de Unix y Linux está escrito en este estilo.

```
function saludar($val){
    if ($val == 1) {
        echo "Hola!";
    } else {
        echo "Chao!";
    }
}
```

3.8 Consideraciones Finales

En el capítulo recién finalizado se realizaron una serie de tareas correspondientes a las fases de análisis, diseño e implementación del sistema propuesto, deviniendo en un conjunto de artefactos que tributaron a la obtención de una primera versión funcional de la aplicación. Se realizó el modelo de implementación que contiene el diagrama de componentes que representa los elementos de software que conforman el sistema, el diagrama de despliegue que modela la distribución del hardware que se va a utilizar, los diagramas de clases del análisis y del diseño así como los diagramas de colaboración para definir la interacción del usuario con el sistema.

Fue definido el estándar de codificación a utilizar, con el objetivo de evitar errores y lograr una mejor comprensión del código generado.

Conclusiones

Durante el desarrollo del presente trabajo se llevaron a cabo los diferentes flujos de trabajo que propone la metodología utilizada para darle cumplimiento a las tareas planteadas, se investigó el estado del arte de las aplicaciones de deportes de fantasía y se analizó que no era posible utilizar ninguna solución anterior. Se estudiaron y seleccionaron las tendencias, tecnologías y herramientas más idóneas para su desarrollo y se realizó un profundo análisis de las herramientas que provee el framework symfony para desarrollo de aplicaciones web. Además se logró desarrollar el módulo Fantasía Béisbol e integrar el mismo al portal de beisbol de la UCI, lo cual aumentará las prestaciones de este portal web, ya que es el único de su tipo en el país que gestiona este tipo de servicios, dándole cumplimiento de esta forma al problema científico planteado en la investigación.

Recomendaciones

Luego de terminada la investigación, el autor recomienda:

- Estudiar la factibilidad de extender el sitio de beisbol de la UCI y de la Fantasía Béisbol implementada a nivel nacional.
- Actualizar la versión del framework empleado.
- Realizar pruebas unitarias al sistema para conocer los tiempos de respuesta de la base de datos en el servidor de despliegue.

Bibliografía

1. Jr, Frederick Brooks. The mythical man-month. s.l. : Addison-Wesley, 1975.
2. Perry, Dewayne y Wolf, Alexander. Foundations for the study of software architecture. s.l. : ACM SIGSOFT Software Engineering, 1992.
3. Introducción a la Arquitectura de Software. Reynoso, Carlos Billy. 2005.
4. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. Billy, Carlos y Kicillof, Nicolás. 2004.
5. Arquitectura Base sobre la Web. Pimentel, Luis y Perez, Yosev. 2008.
6. Jose A Vela Dávila. [En línea] Ponencia en Seminario, 2008. <http://www.cimat.mx/Eventos/seminariotecnologias08/javd.pdf>.
7. Metodologías de Desarrollo de Software. Sánchez, María A Mendoza. 2004.
8. Lenguajes de Descripción de Software. Reynoso, Carlos y Kicillof, Nicolás. 2004.
9. Applying UML and Patterns -An Introduction to Object- Oriented Analysis and Design and Iterative Development. Larman, Craig. 2005, Vol. tercera edición.
10. Gustavo F Torrealday. Informática. [En línea] <http://www.torrealday.com.ar/articulos/articulo006.htm>.
11. Álvarez, Miguel A. [En línea] 2004. <http://www.desarrolloweb.com/articulos/que-es-html.html>.
12. Monteiro, Juliana Lázaro. desarrolloweb.com. [En línea] <http://www.desarrolloweb.com/articulo/26.php>.
13. https://developer.mozilla.org/index.php?t=Es/Gu%C3%ADa_JavaScript_1.5/Concepto_de_JavaScript. mozilla <developer center>. [En línea] 2008
14. Ruby. A Programmer's Best Friend. [En línea] <http://www.ruby-lang.org/es/about/>.
15. Der, Christian Van. [En línea] 2003. <http://www.maestrosdelweb.com/editorial/phpintro/>.
16. Gutiérrez, Javier J. [En línea] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
17. CakePHP. [En línea] <http://cakephp.org/>.
18. symfony.es. [En línea] <http://www.symfony.es/que-es-symfony/>.
19. Potencier, Fabien. Symfony guía definitiva 1.2. 2009.

20. Bantam Books (1984). Rotisserie League Baseball. Ventura, CA, U.S.A. ISBN: 0553340816 / 0-553-34081-6.
21. [En línea] 2009. <http://uvfdatabases.wordpress.com/2009/02/07/terminos-de-repaso-introductorios-a-bds/>.
22. PostgreSQL. [En línea] <http://www.postgresql.org/about/>.
23. Erika Camacho, Fabio Cardeso, Gabriel Nuñez. Arquitecturas de Software. 2004.
24. Bosch, Jan. Design and Use of Software Architectures. s.l. : Addison Wesley, 2000.
25. Barbacci, Mario, y otros. Quality Attributes. Technical Report. Carnegie Mellon University. Pittsburgh, Pennsylvania 15213 : s.n., 1995.
26. Doctrine development team. [En línea] <http://www.doctrineproject.org/documentation>.
27. Salvador Gómez, Omar. Evaluando Arquitecturas de Software. Parte 1. Panorama General. s.l. : Brainworx S.A, 2007.
28. Introducción a Extreme Programming . Escribano, Gerardo Fernández. 2002.
29. Fowler, Martin. Patterns of Enterprise Application Architecture . s.l. : Addison Wesley, 2003.
30. Souders, Steve. Even Faster Web Sites. s.l. : O'REILLY, 2009.
31. Harmes, Ross y Díaz, Dustin. Pro JavaScript™ Design Patterns. 2008.
32. Stefanov, Stoyan. Object-Oriented JavaScript. 2008.
33. White, Alexei. JavaScript Programmer's Reference. s.l. : Wiley Publishing, 2009.
34. OLSSON, TOMMY y O'BRIEN, PAUL. THE ULTIMATE CSS REFERENCE. s.l. : SitePoint, 2008.
35. Holzner, Steve. Design Patterns For Dummies. s.l. : Wiley Publishing, 2006.

Referencias bibliográficas

- [1] Sheena-kay. Disponible en: <http://es.shvoong.com/internet-and-technologies/sport/1847602-deporte-b%C3%A9isbol-fantasia> [Citado el: 04/02/2010]
- [2] Rojas, Enrique. "Fantasy Basseball nowadays". Disponible en: <http://www.exitogarantizado.es.tl/DEPORTES-DE-FANTASIA.htm> [Citado el 05/02/2010]
- [3] Tortolero, Manuel. Disponible en: <http://planeta-beisbol.com/pizarra/viewtopic.php?f=7&t=280> [Citado el 06/02/2010]
- [4] Disponible en: <http://www.fangraphs.com/blogs/index.php/glossary/#winvalues> [Citado el 07/02/2010]
- [5] Schwaber, Ken 2004, AGILE PROJECT MANAGEMENT WITH SCRUM. [Citado el 07/02/2010]
- [6] Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Addison Wesley, 1999. [Citado el 07/02/2010]
- [7] Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Prentice Hall, 1998.
- [7] Martin fowler, kendall sccott, "UML Gota a Gota", 1999. [Citado el 10/02/2010]
- [8] Pérez, Javier Eguíluz. Introducción a XHTML. 2008. [Citado el 10/02/2010]
- [9] Pérez, Javier Eguíluz. Introducción a JavaScript. 2008. [Citado el 10/02/2010]
- [10] Pérez, Javier Eguíluz. Introducción a CSS. 2008. [Citado el 10/02/2010]
- [11] Reutilización basado en la tecnología de objetos. Berrospi, Eloy Bernabé. 2003. [Citado el 10/02/2010]
- [12] Potencier, Fabien y Zaninotto, François. Symfony, la guía definitiva. 2007. [Citado el 10/02/2010]
- [13] Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Addison

Anexos

Anexo1

Sitios Nacionales
http://beisbolcubano.cubasi.cu/
http://www.beisbolcubano.cu/
http://www.inder.cu/
http://www.terrenodepelota.com/
http://www.baseballdecuba.com/

Anexo2 Diagramas de Clases de Análisis

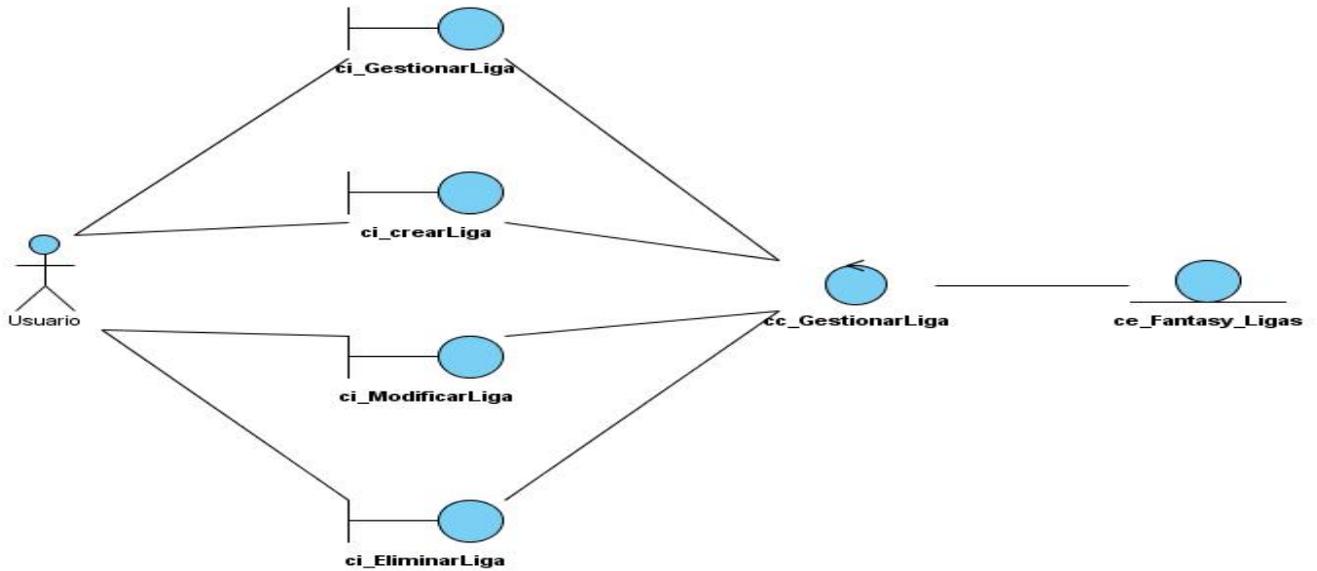


Fig19. DCA Gestionar Liga

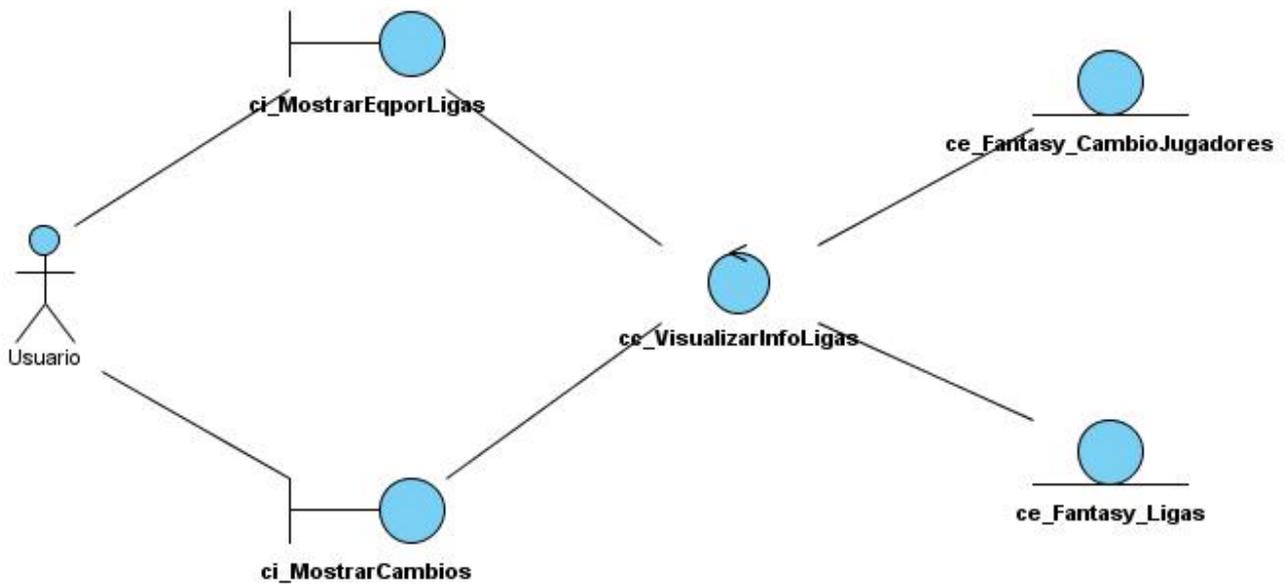


Fig20. DCA Visualizar Informaciones Liga

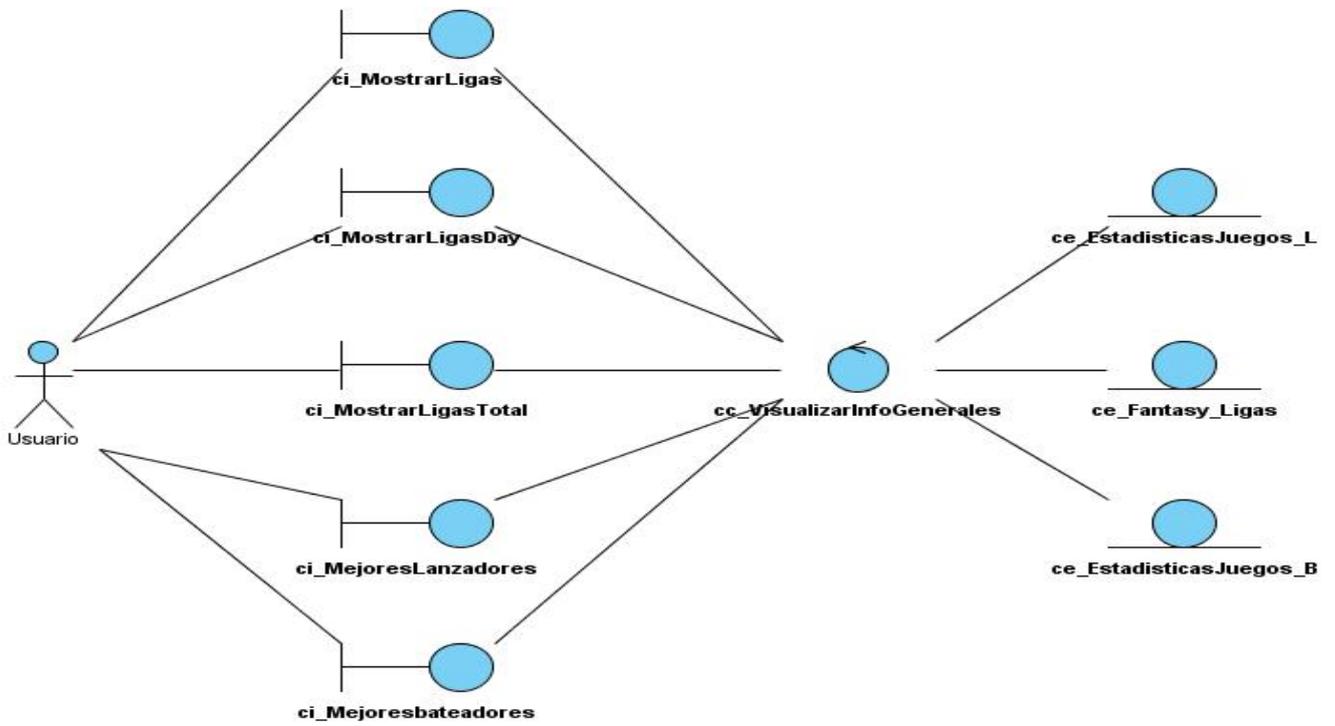


Fig21. DCA Visualizar Informaciones Generales



Fig23. DCA Conformar Pre Equipo

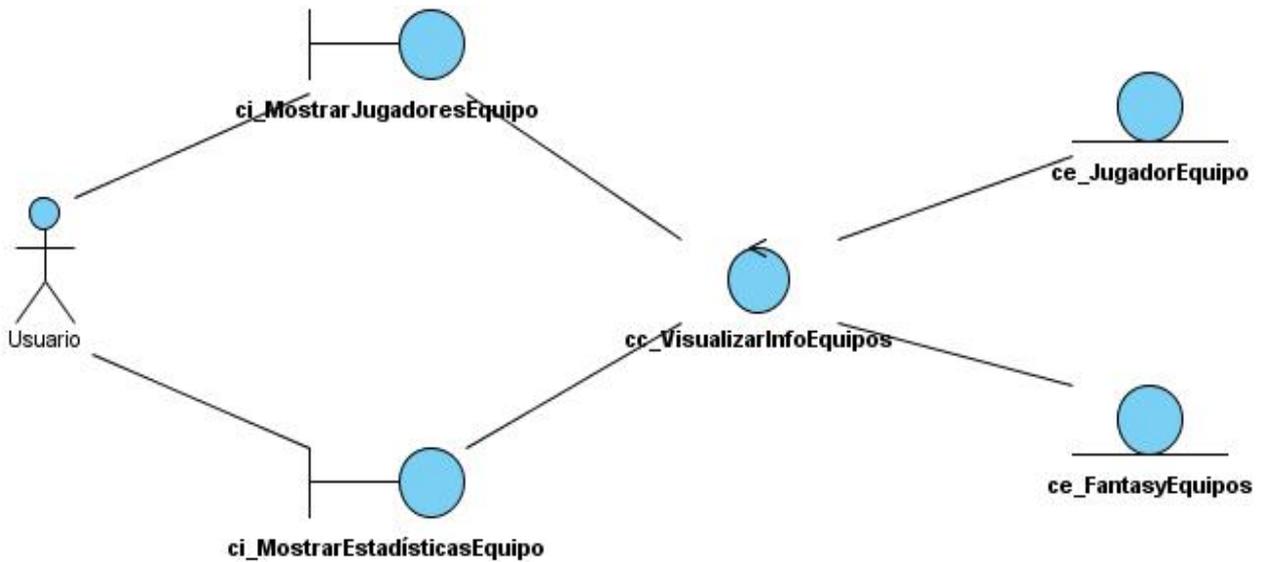


Fig24. DCA Visualizar Informaciones Equipo

Anexo3 Diagramas de Colaboración

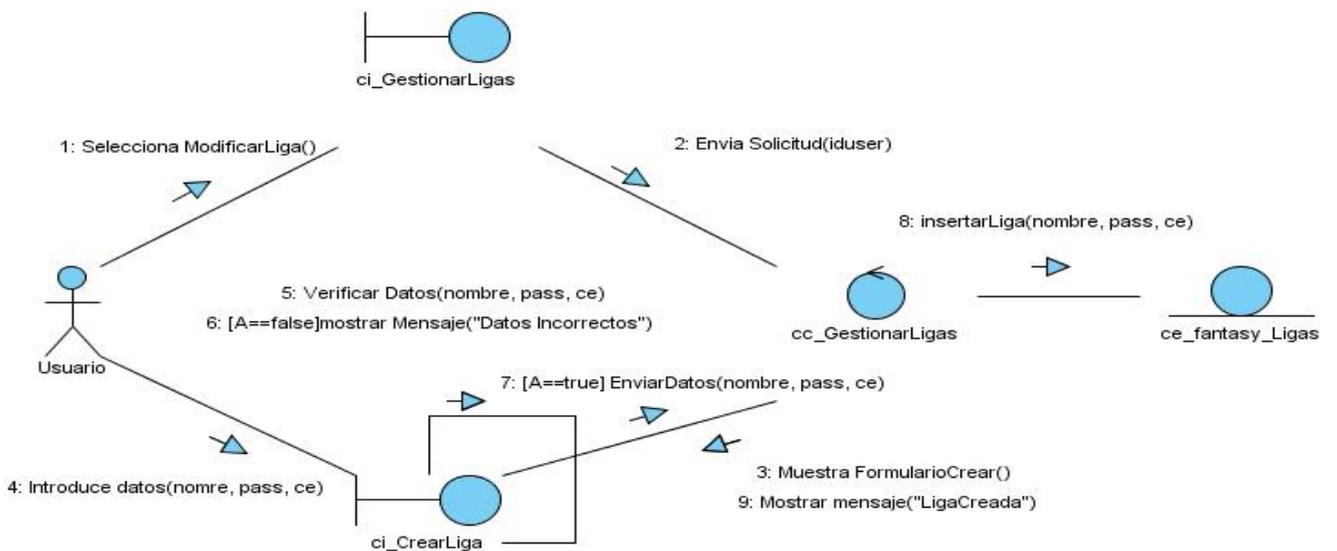


Fig25. DC Gestionar Ligas Sección Crear

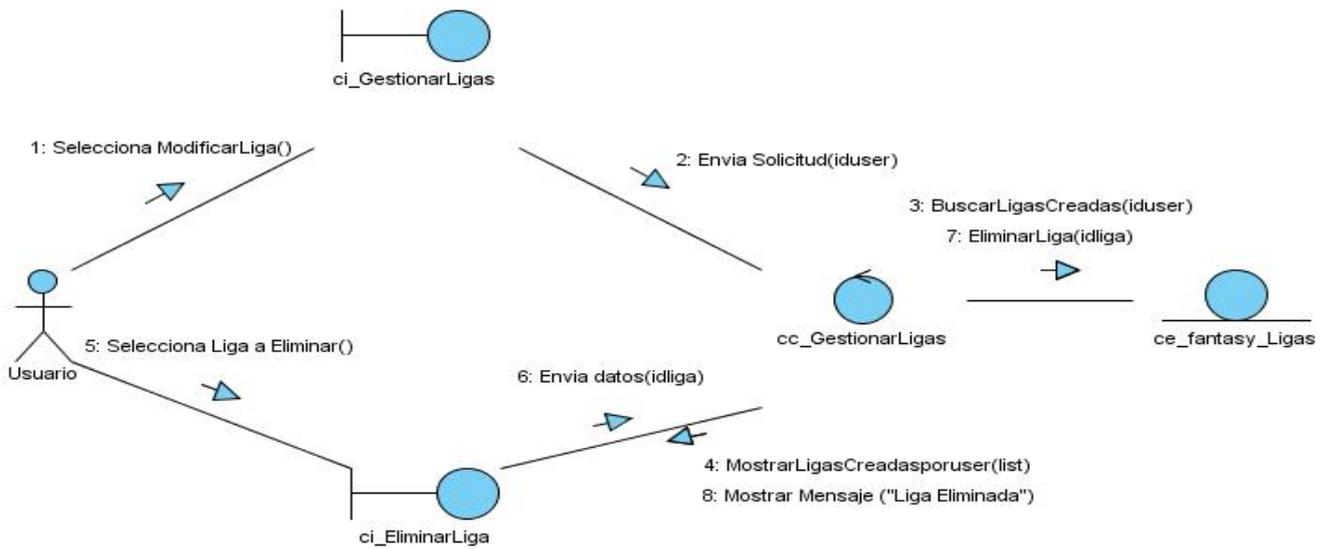


Fig26. DC Gestionar Liga Sección Eliminar

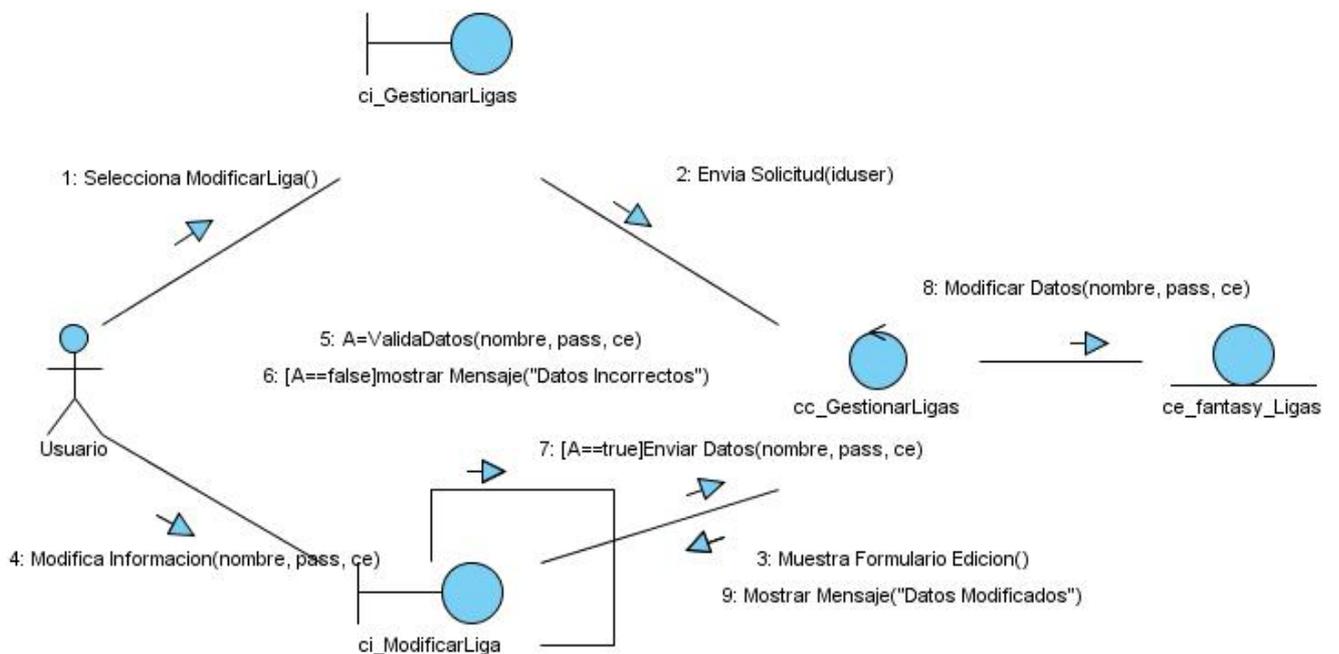


Fig27. DC Gestionar Liga Sección Modificar

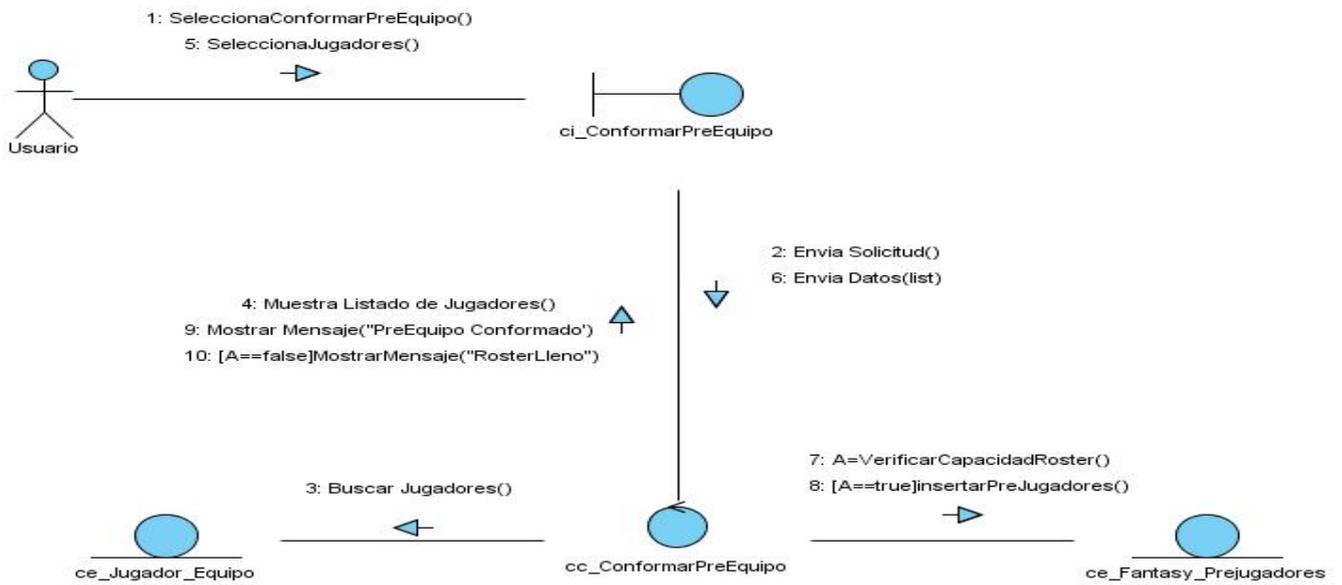


Fig28. DC Conformar Preequipo

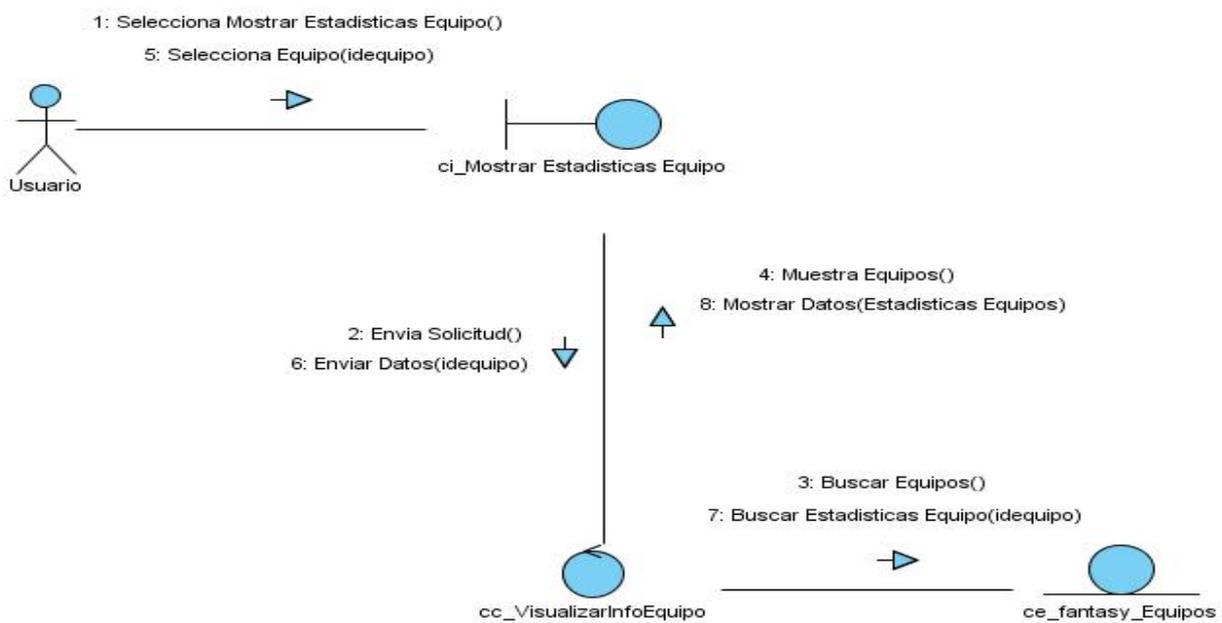


Fig29. DC Visualizar Informaciones Equipo Sección Mostrar Estadísticas Equipo

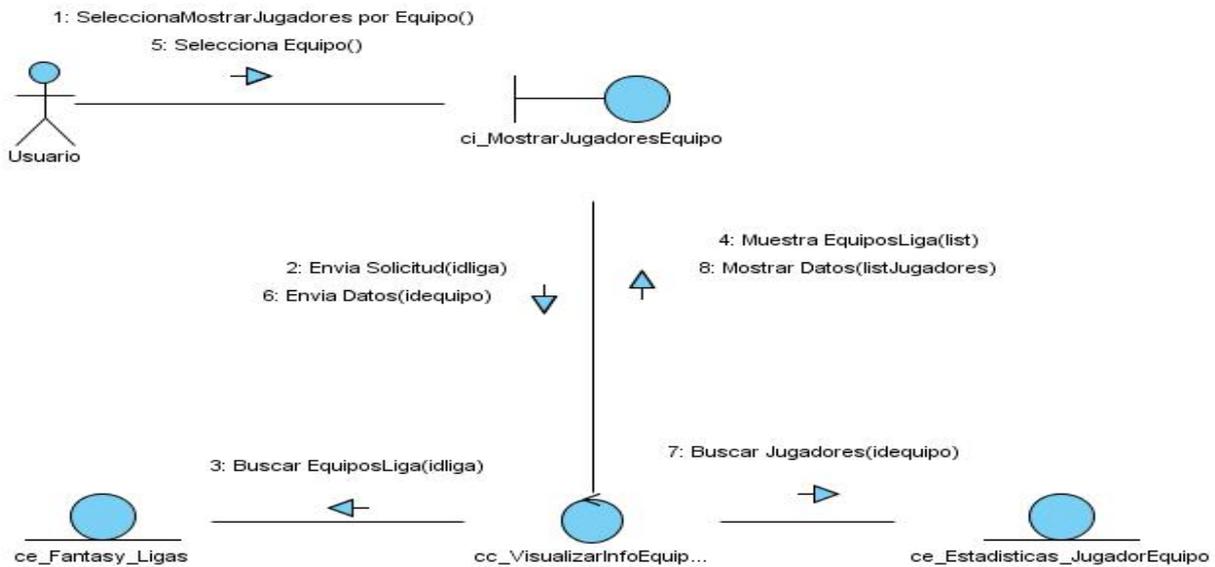


Fig30. DC Visualizar Informaciones Equipo Sección Mostrar Jugadores Equipo

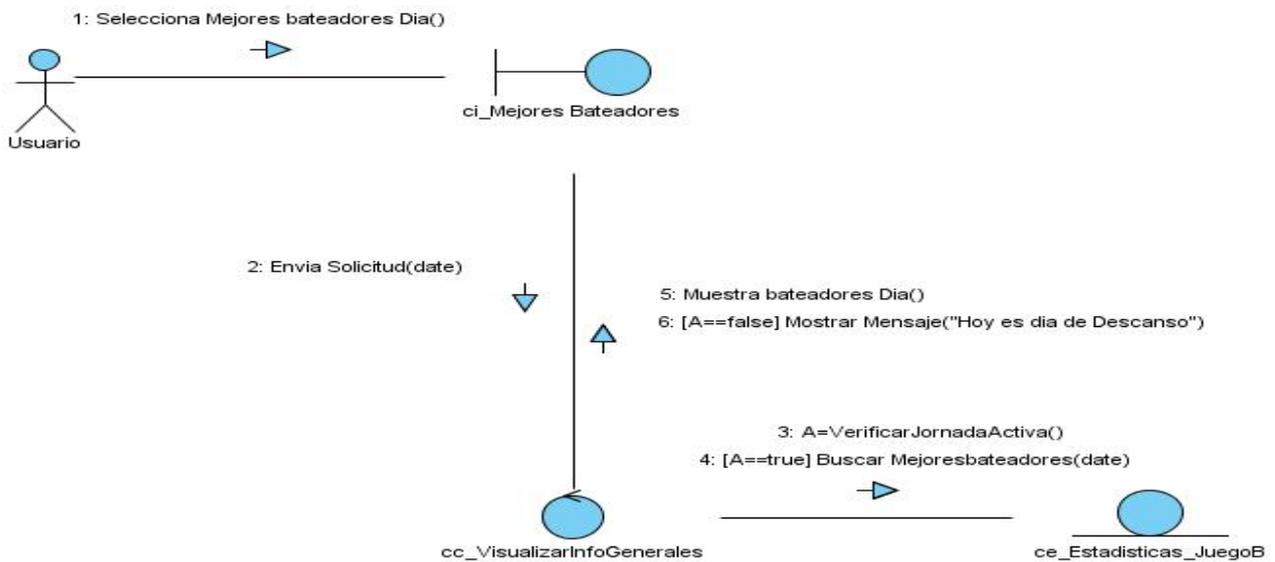


Fig31. DC Visualizar Informaciones Generales Sección mejores Bateadores del día

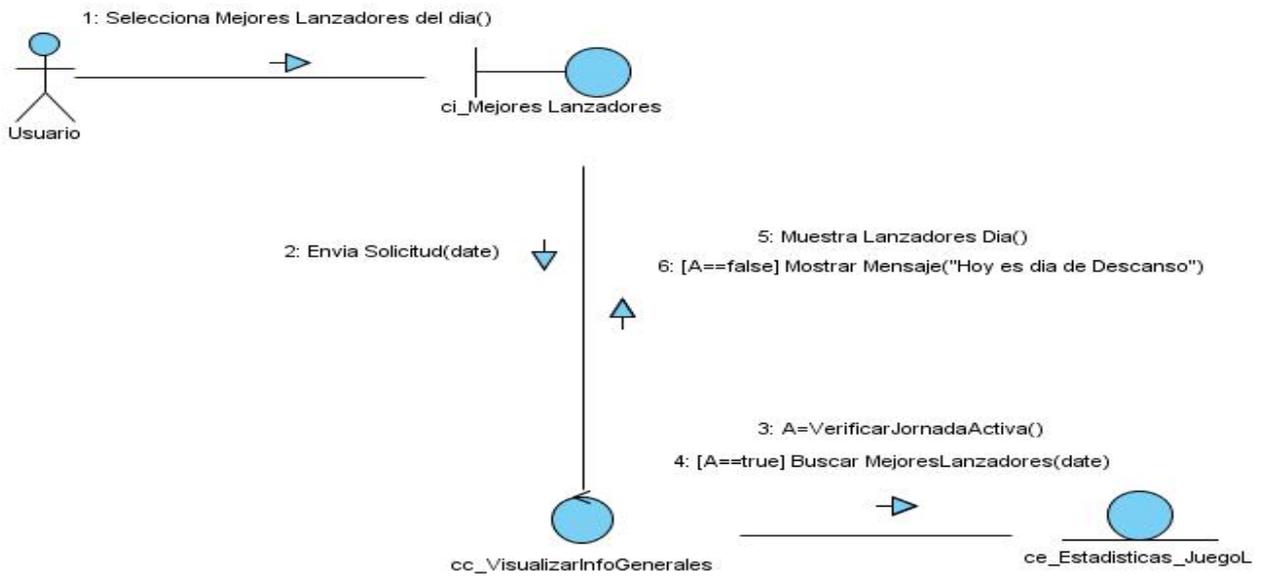


Fig32. DC Visualizar Informaciones Generales Sección mejores Lanzadores del día

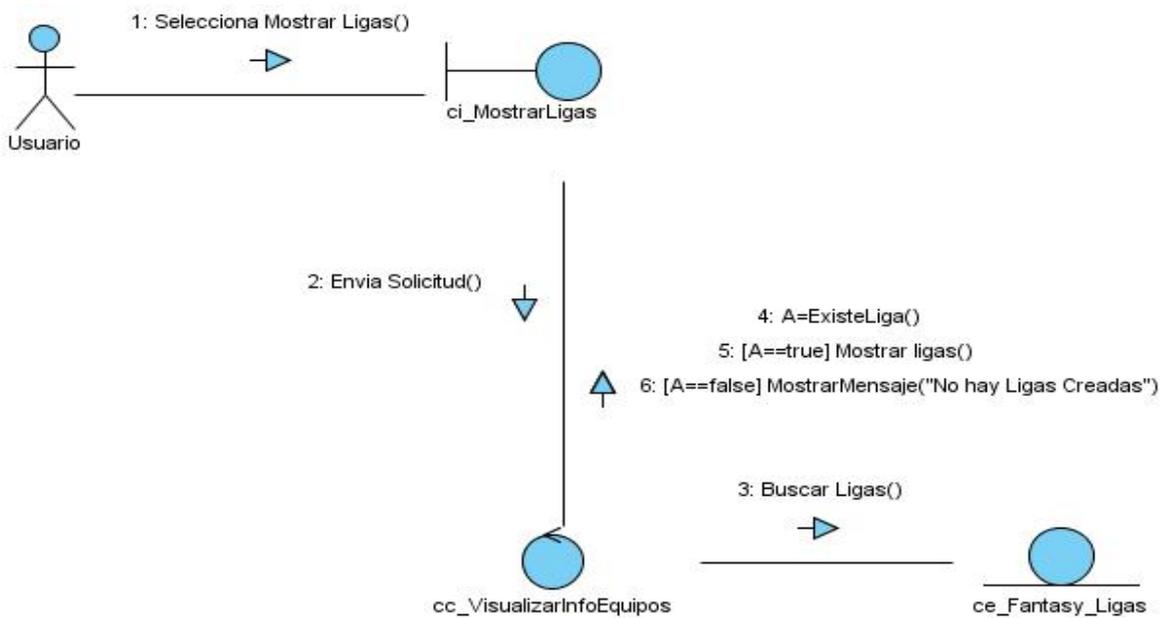


Fig33. DC Visualizar Informaciones Generales Sección Mostrar Ligas

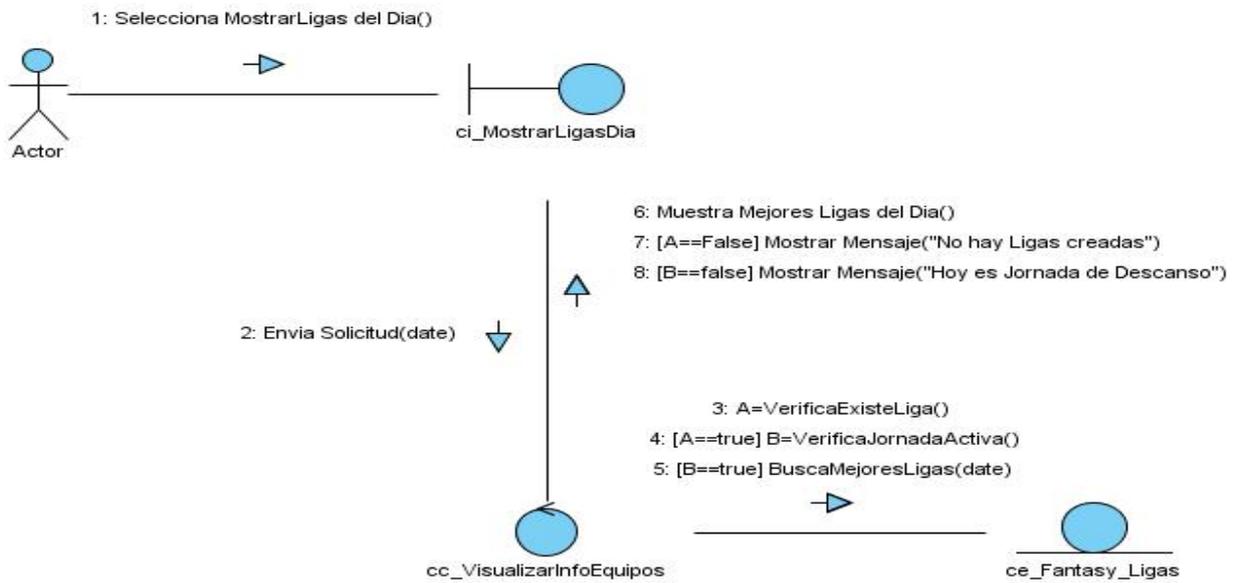


Fig34. DC Visualizar Informaciones Generales Sección mostrar ligas del día

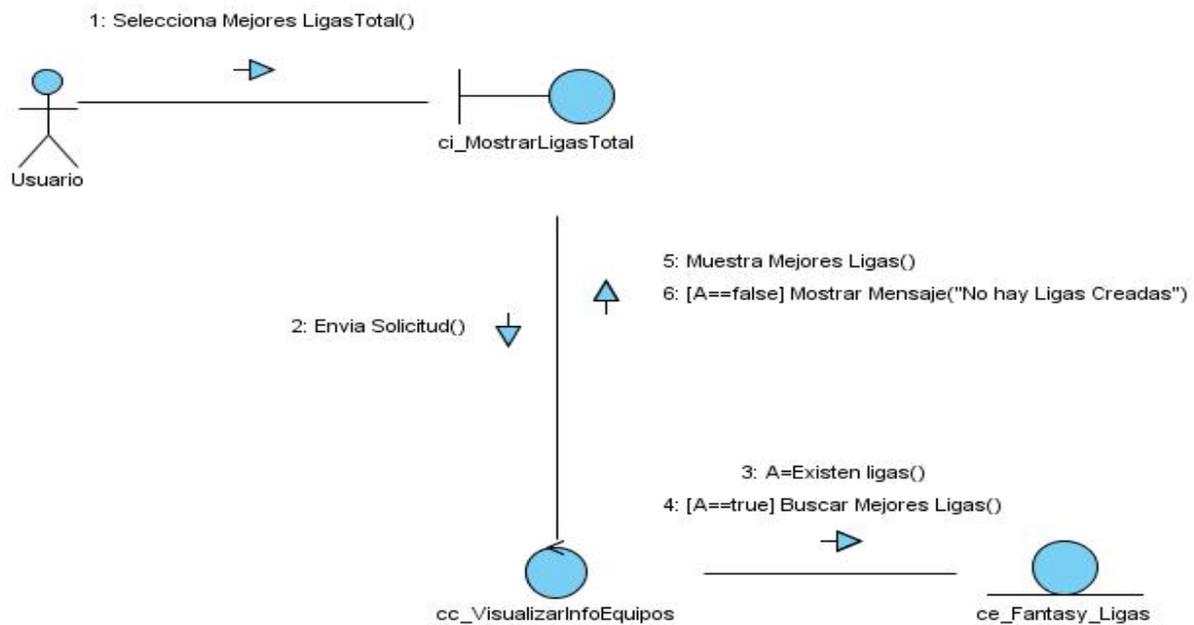


Fig35. DC Visualizar Informaciones Generales Sección mejores Ligas total

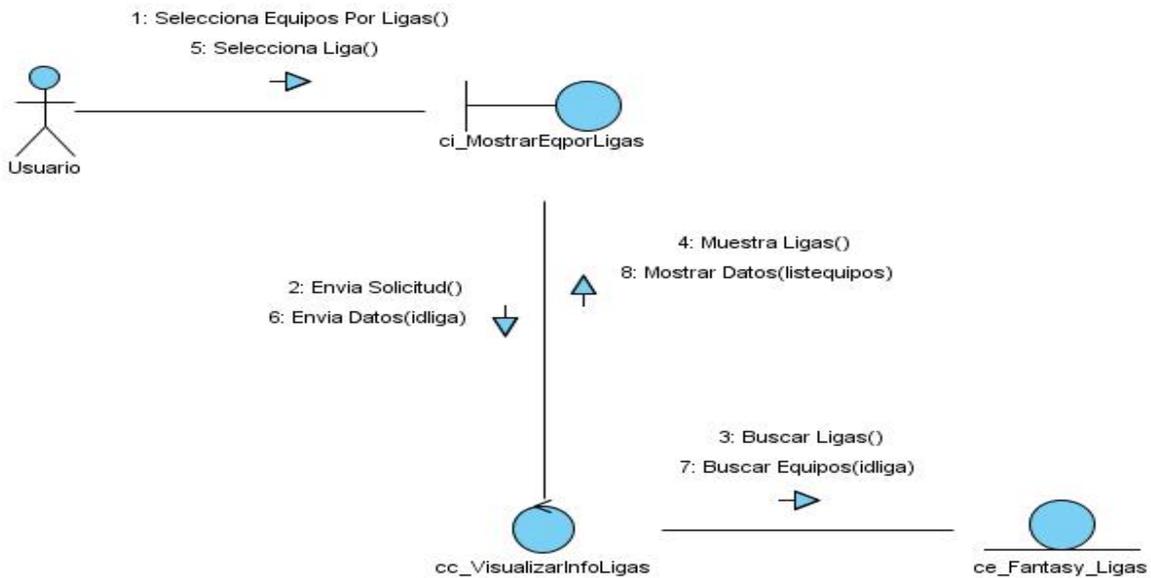


Fig36. DC Visualizar Informaciones Liga Sección Equipos por Liga

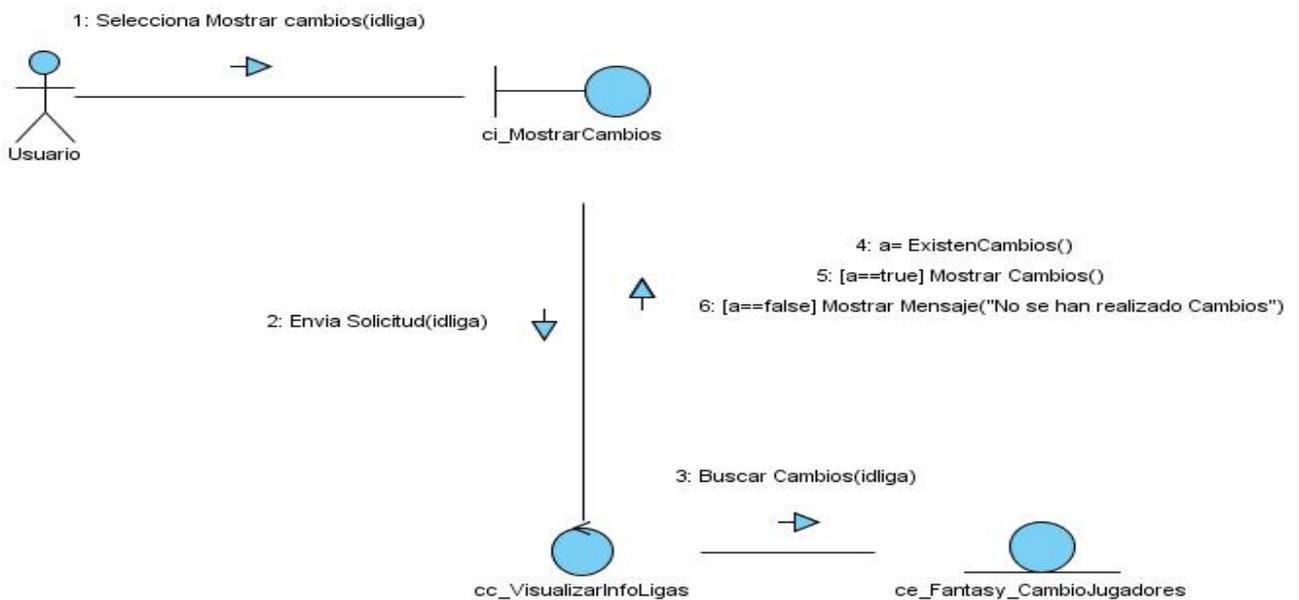


Fig37. DC Visualizar Informaciones Liga Sección mostrar cambios

Anexo4 Clases del Diseño

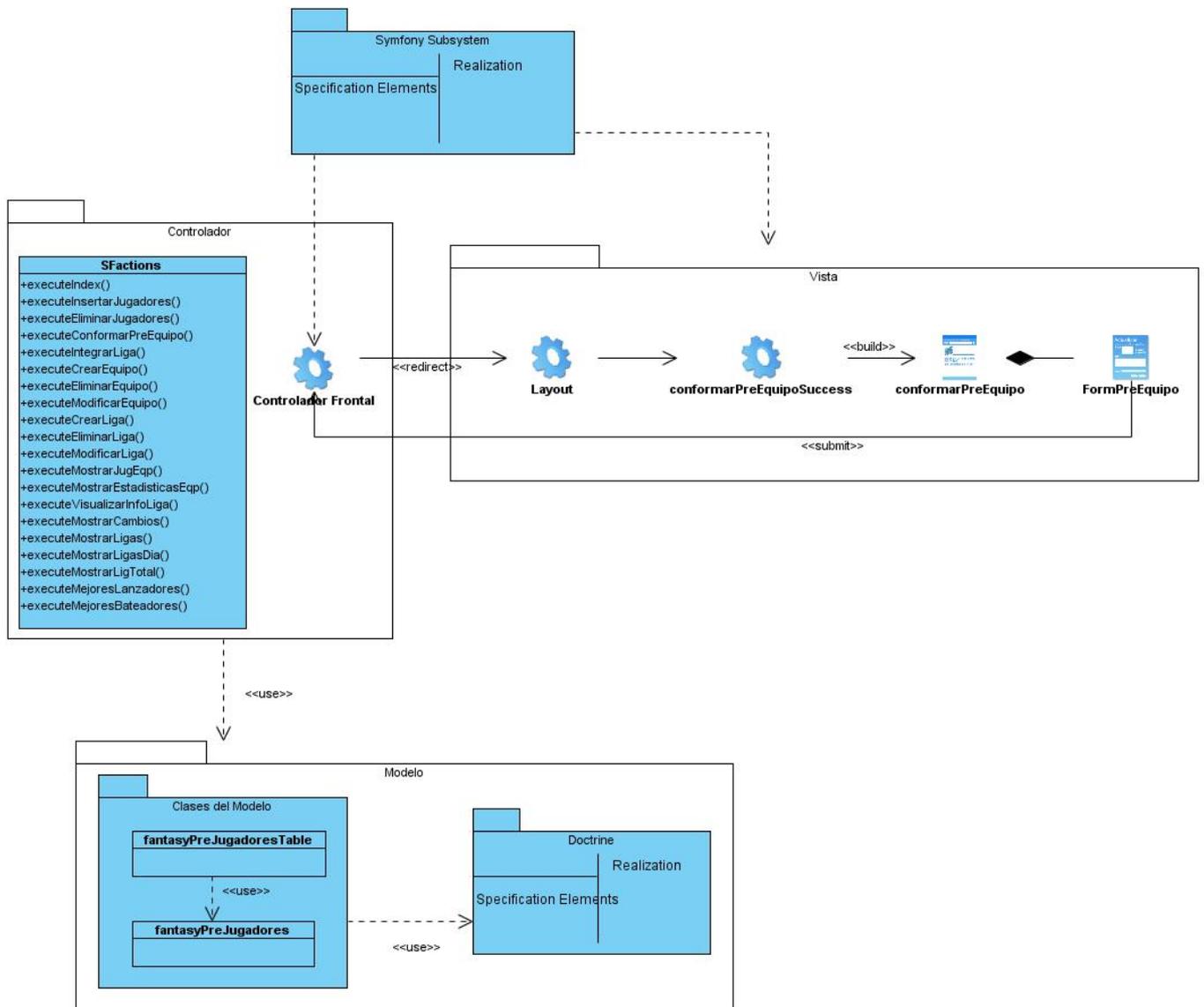


Fig38. DCD Conformar PreEquipo

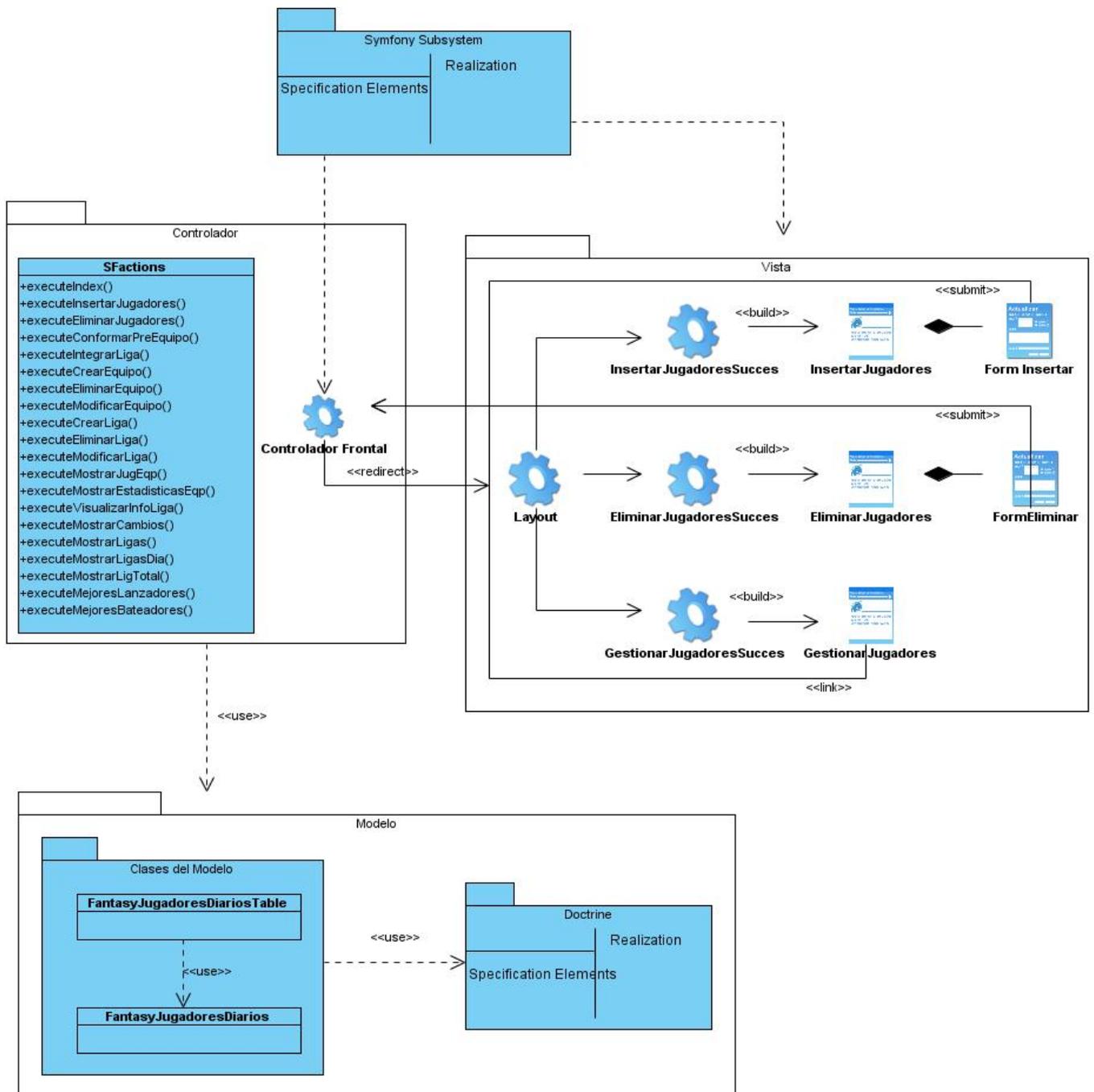


Fig39. DCD Gestionar Jugadores

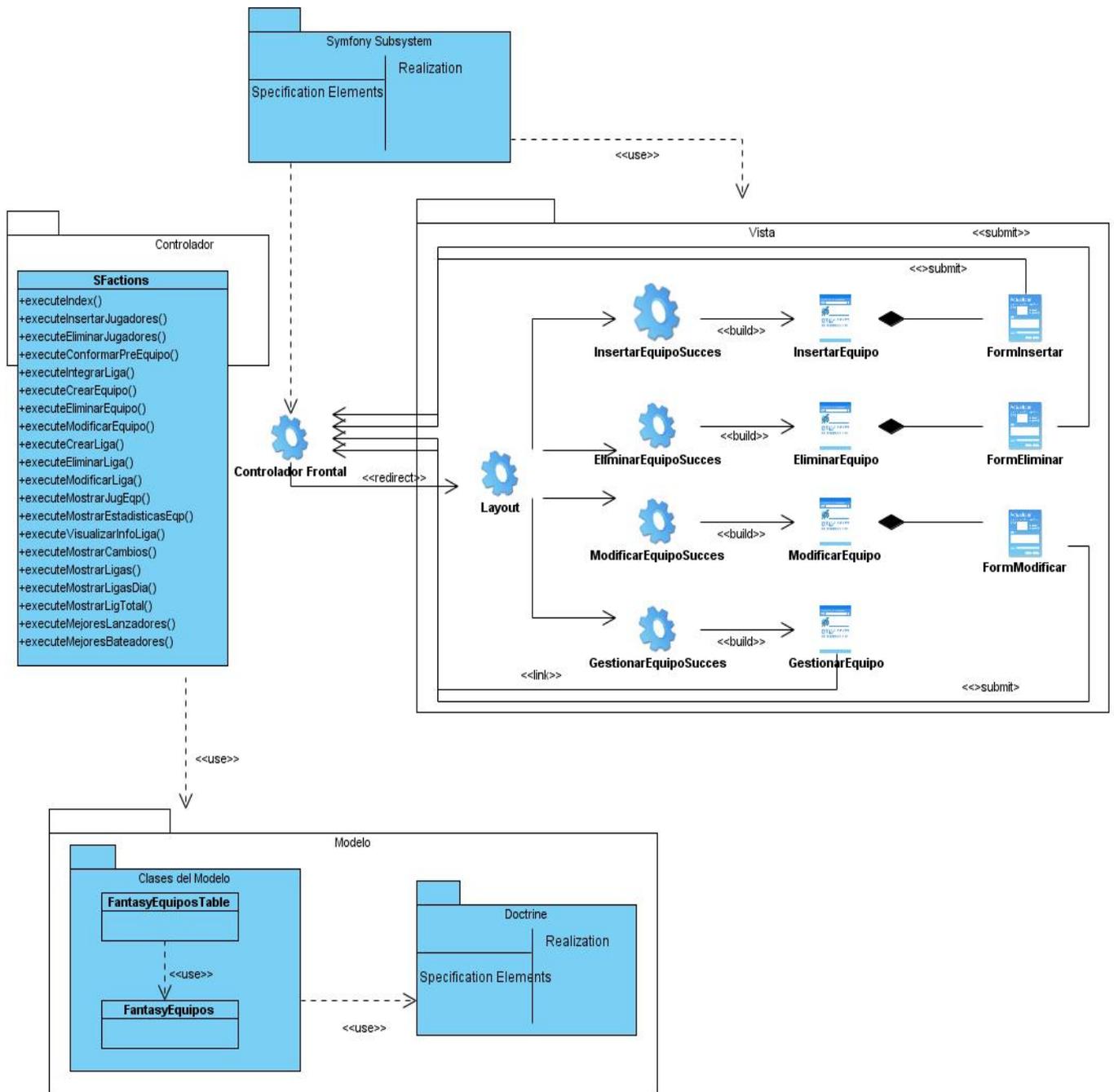


Fig40. DCD Gestionar Equipos

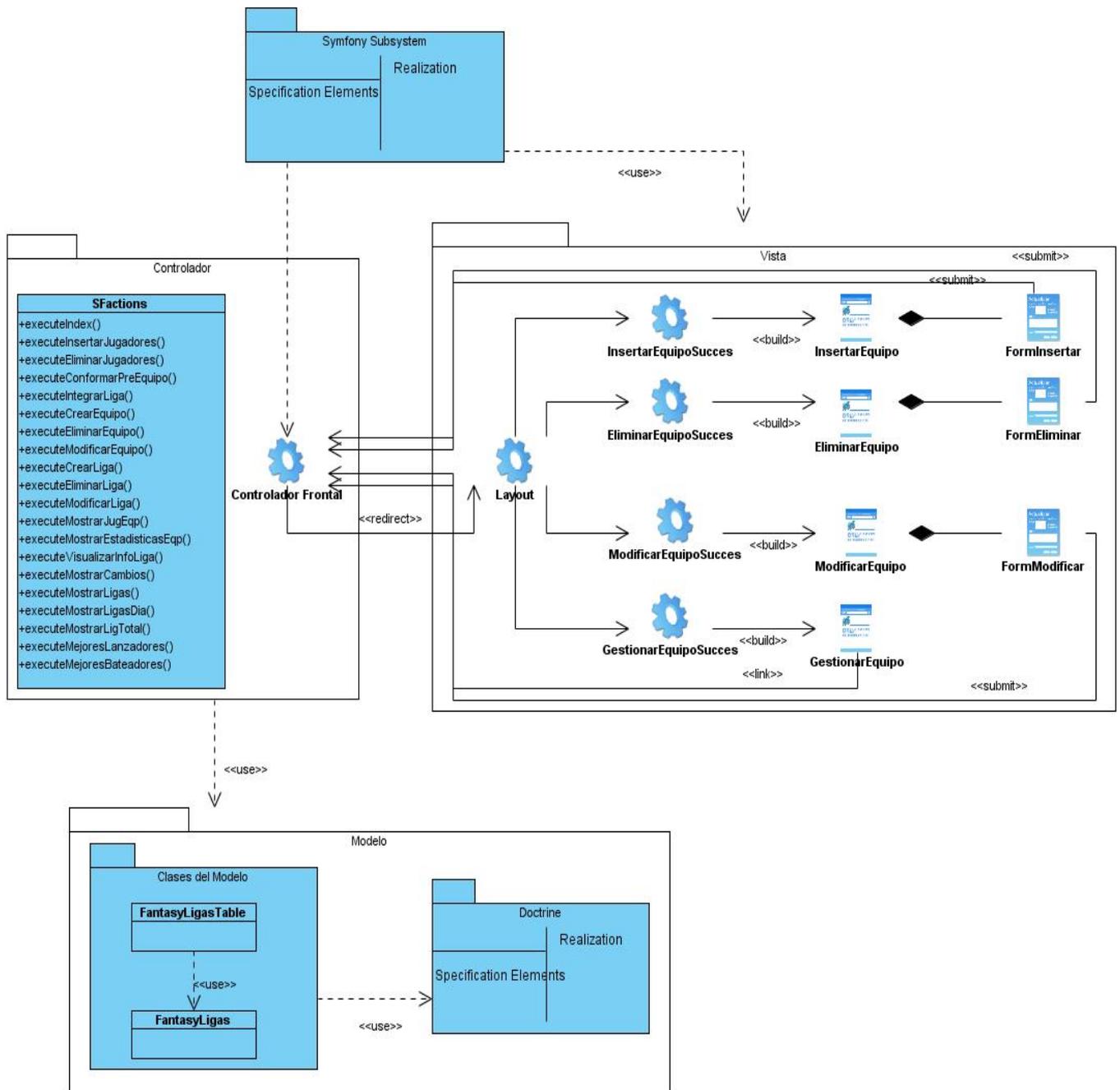


Fig41. DCD Gestionar Ligas

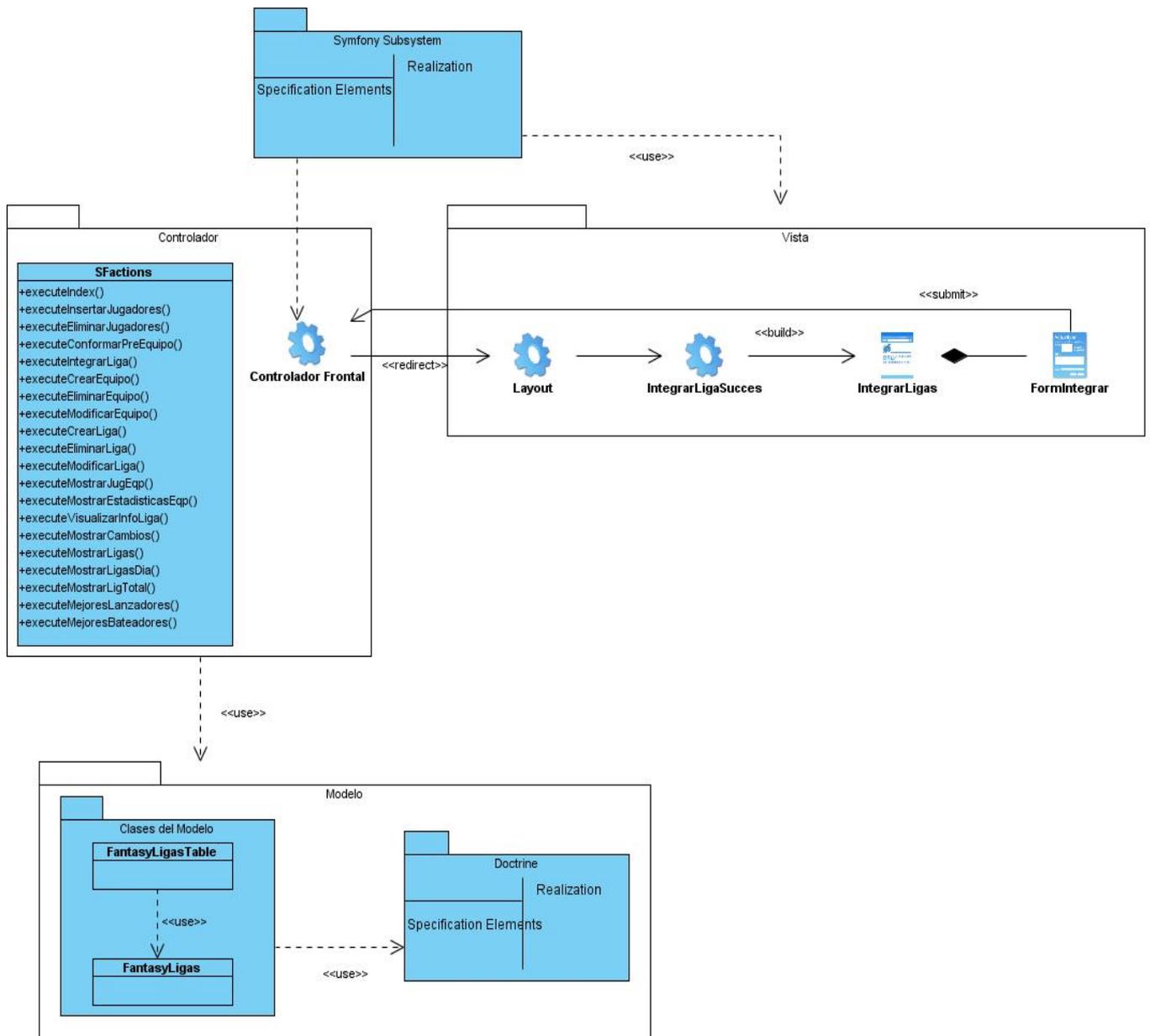


Fig42. DCD Integrar Ligas

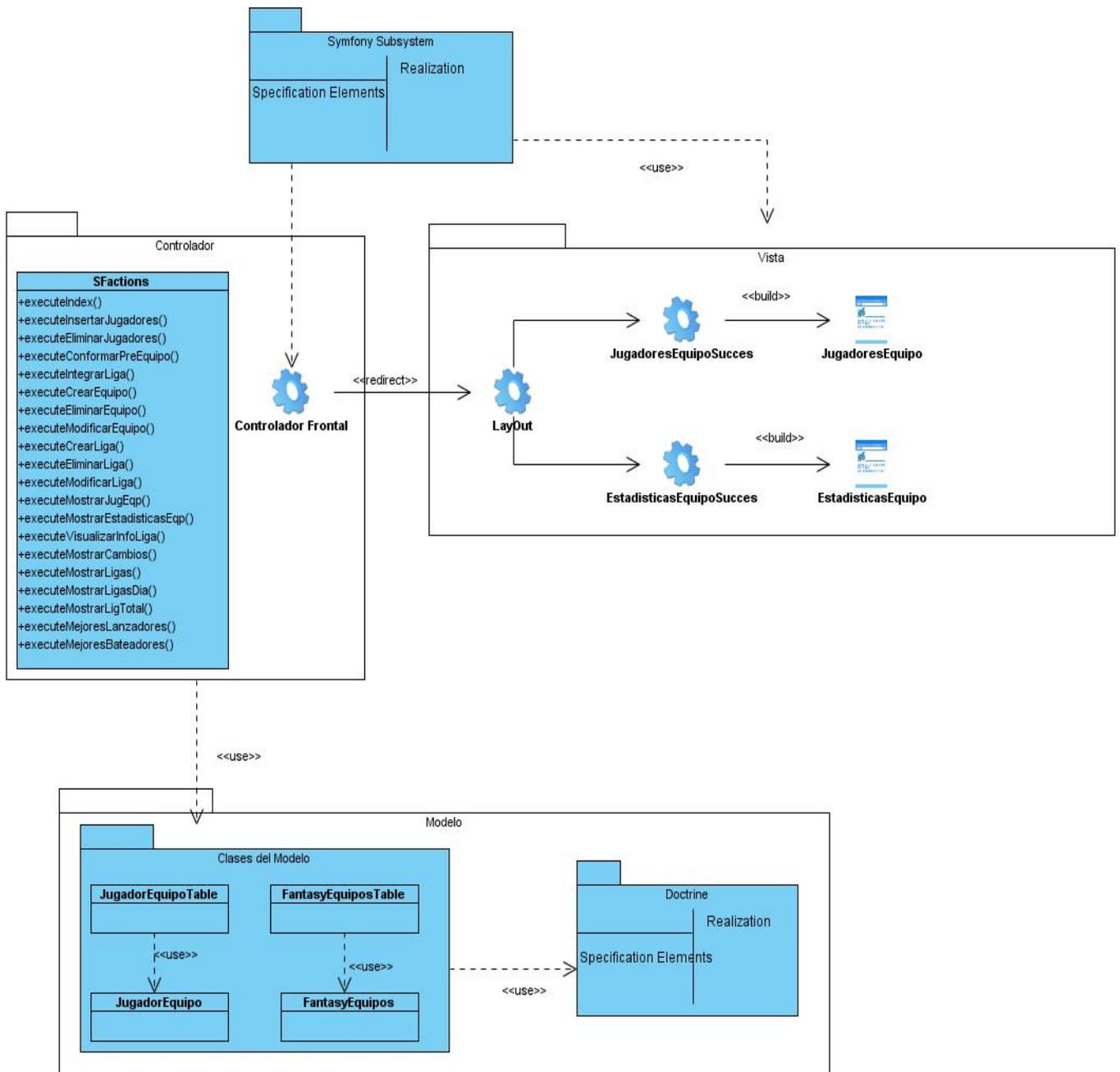


Fig43. DCD Visualizar Informaciones Equipo

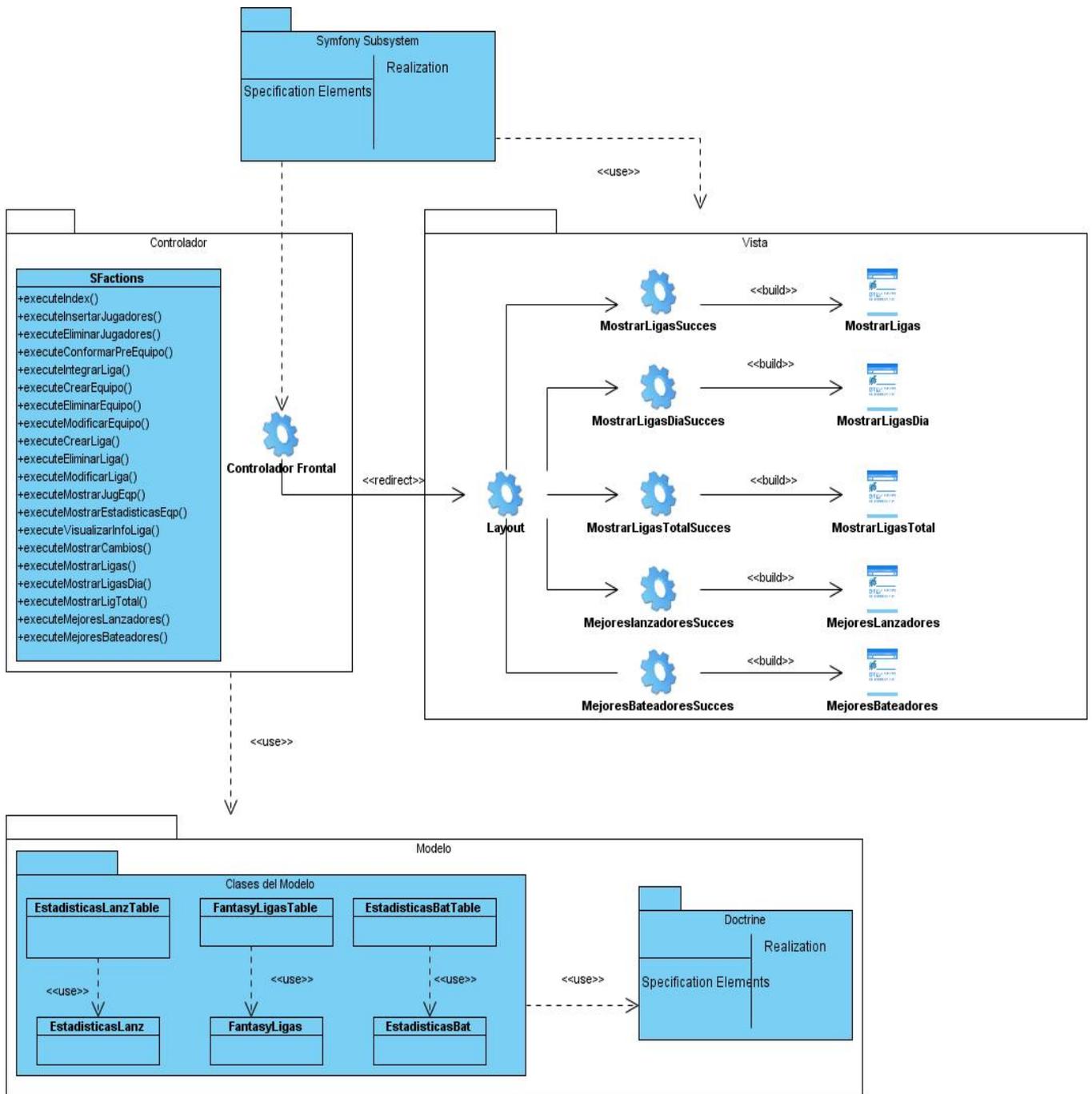


Fig44. DCD Visualizar Informaciones Generales

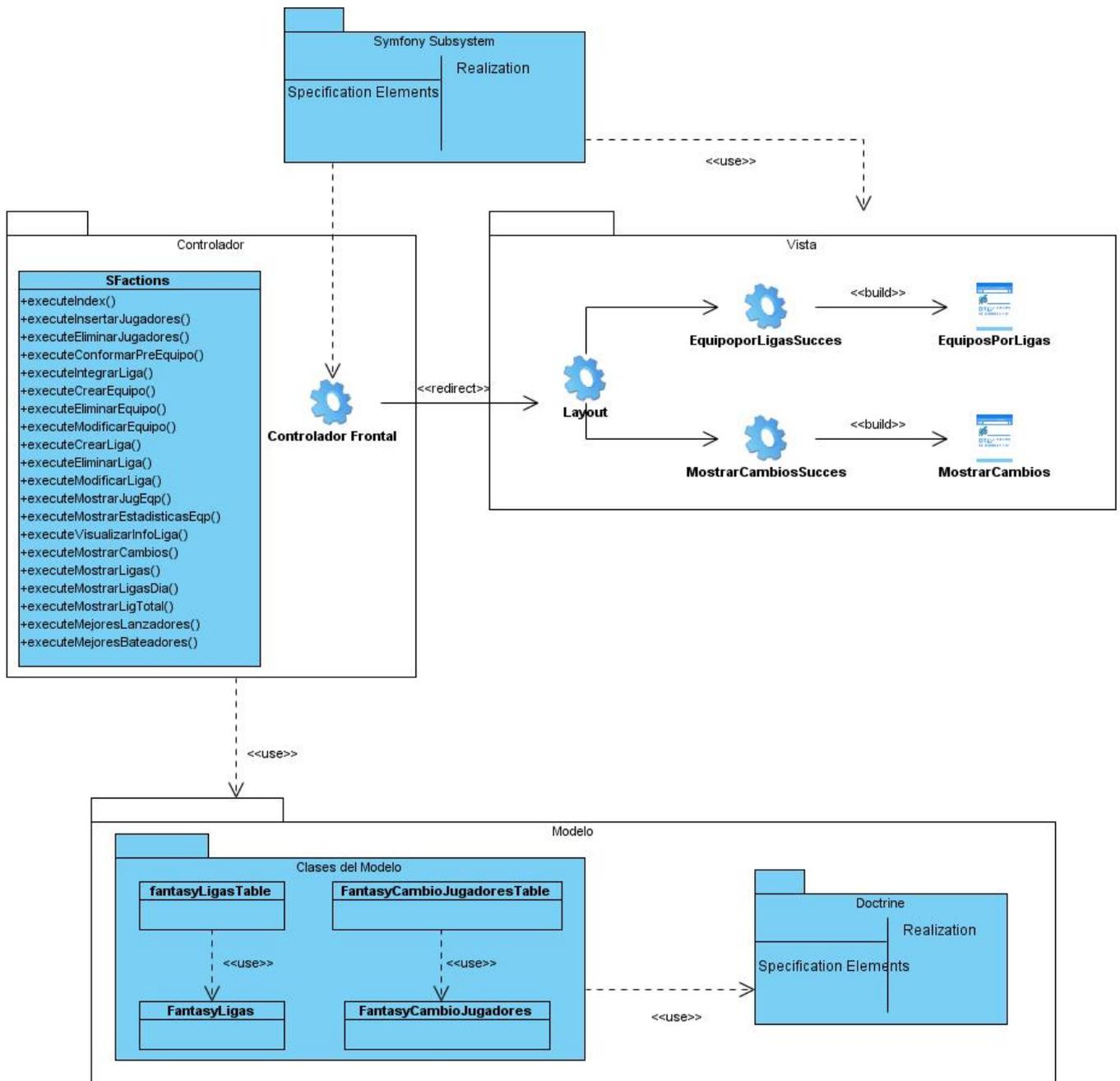
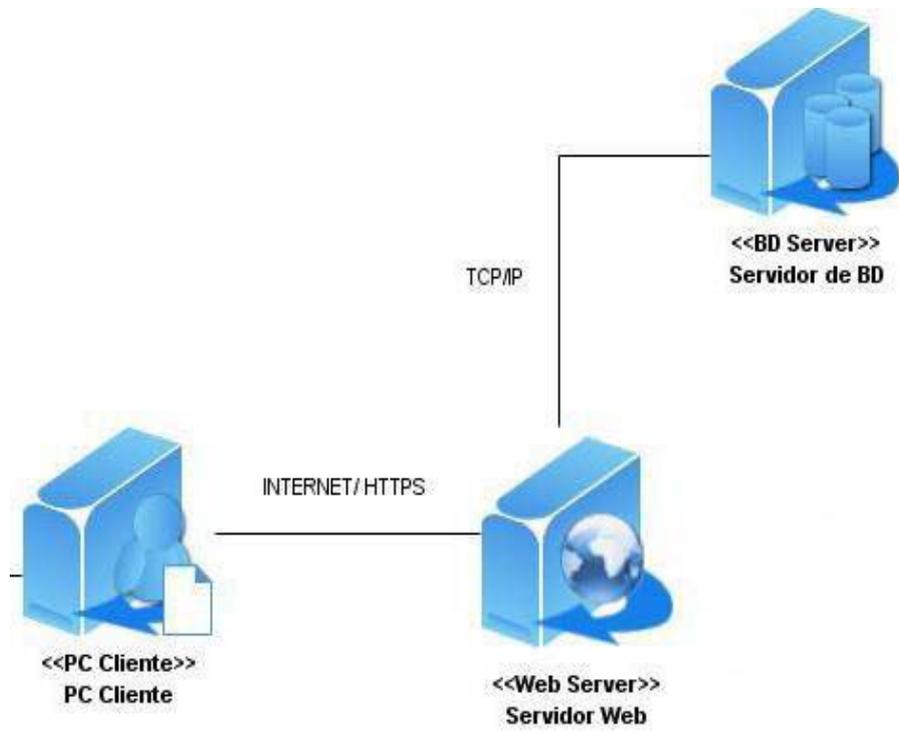
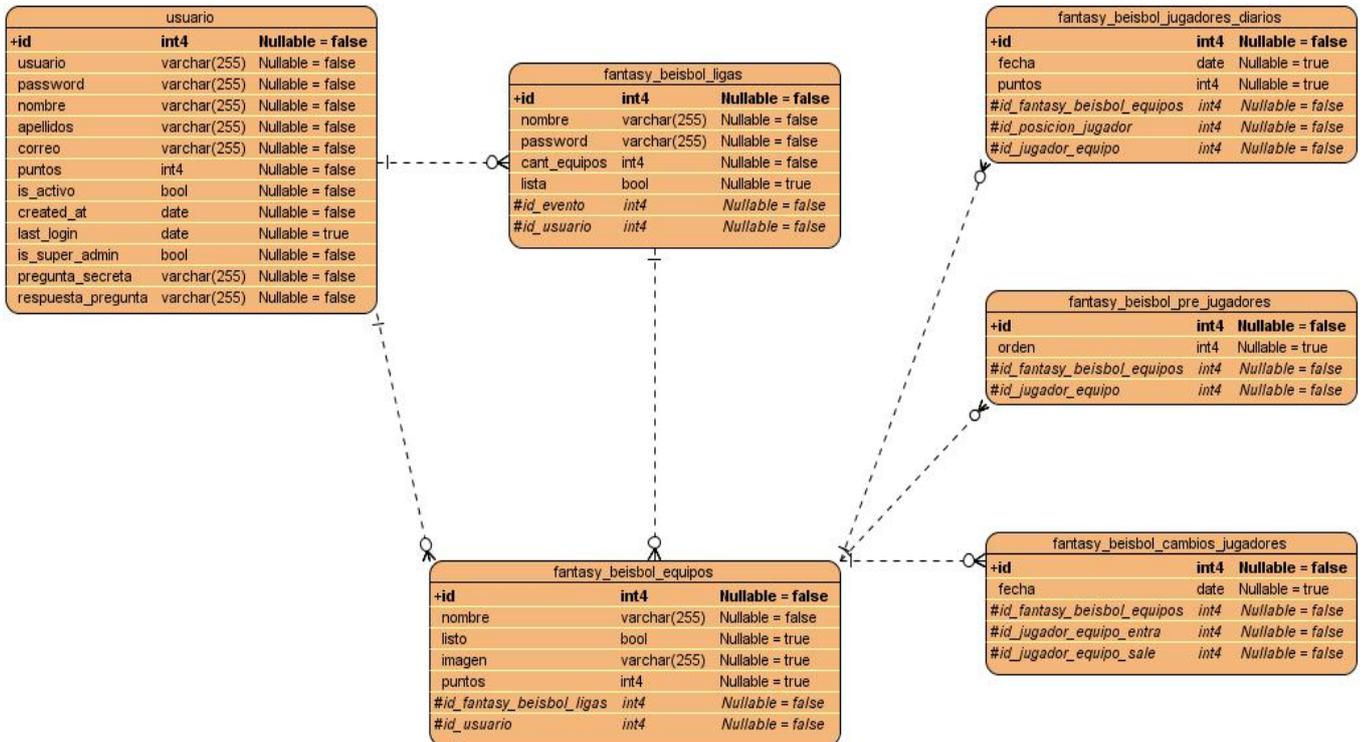


Fig45. DCD Visualizar Informaciones Ligas

Anexo 5 Diagrama de Despliegue



Anexo 6 Base de Datos



Anexo 7 Descripción Tablas BD

Nombre Fantasy_beisbol_jugadores_diarios		
Descripción Contiene las estadísticas de los jugadores en el día		
Atributo	Tipo	Descripción
Id	Serial	Identificador de la tabla
fecha	Date	Almacena la fecha del día
puntos	Integer	Puntos acumulados
Id_fantasy_beisbol_equipos	Integer	Id del equipo al que pertenece el jugador
Id_posición_jugador	Integer	Identificador de las posición del jugador
Id_jugador_equipo	Integer	Id del jugador dentro del equipo

Tabla 4 Fantasy_beisbol_jugadores_diarios

Nombre Fantasy_beisbol_ligas		
Descripción Contiene los datos de una liga de la fantasy		
Atributo	Tipo	Descripción
Id	Serial	Identificador de la tabla
nombre	Character varying (255)	Nombre de la liga
password	Character	Contraseña de la Liga

	varying (255)	
Cant_equipos	Integer	Cantidad de equipos que tendrá la liga
Listo	Boolean(false)	Chequeo para ver estado de liga
Id_evento	Integer	Identificador del evento usado
Id_usuario	Integer	Id del usuario creador de la liga

Tabla 5 Fantasy_beisbol_ligas

Nombre Fantasy_beisbol_pre_jugadores		
Descripción Contiene las estadísticas de los jugadores preseleccionados para el equipo		
Atributo	Tipo	Descripción
Id	Serial	Identificador de la tabla
orden	Integer	Almacena el orden de selección de los jugadores
Id_fantasy_beisbol_equipos	Integer	Id del equipo al que pertenece el jugador
Id_jugador_equipo	Integer	Id del jugador dentro del equipo

Tabla 6 Fantasy_pre_jugadores

Nombre Evento		
Descripción Tipo de competencia que se ejecuta la fantasy		
Atributo	Tipo	Descripción

Id	Serial	Identificador de la tabla
nombre	Character varying (50)	Nombre del evento
Notación_oficial	Character varying (50)	Notación oficial
Id_etapa	Integer	Identificador de la etapa
Id_tipoevento	Integer	Identificador del tipo de evento
activo	Boolean(false)	Chequeador del estado del evento

Tabla 7 Evento

Nombre Equipo		
Descripción Contiene las estadísticas de los equipos reales de béisbol		
Atributo	Tipo	Descripción
id	Serial	Identificador de la tabla
siglas	Character varying (50)	Siglas del equipo
nombre	Character varying (50)	Nombre del equipo
imagen	Character varying (50)	Imagen representativa del equipo

Tabla 8 Equipo

Nombre Equipo_evento		
Descripción Contiene las estadísticas de los equipos en los eventos		
Atributo	Tipo	Descripción
Id	Serial	Identificador de la tabla
Jgvs	Integer	Juegos ganados como visitador
Jpvs	Integer	Juegos perdidos como visitador
Jghc	Integer	Juegos ganados como home club
Jphc	Integer	Juegos perdidos como home club
Id_evento	Integer	Identificador del evento
Id_grupo	Integer	Identificador del grupo
Id_equipo	Integer	Identificador del equipo

Tabla 9 Equipo_evento

Nombre jugador		
Descripción Contiene la información sobre los jugadores		
Atributo	Tipo	Descripción
Id	Serial	Identificador de la tabla

nombre	Character varying (50)	Nombre del jugador
Apellidos	Character varying (50)	Apellidos del jugador
Peso	Real	Peso del jugador
Altura	Real	Altura del jugador
Descripción	text	Observaciones sobre el jugador
Activo	Boolean(true)	Si se encuentra en activo
Imagen	Character varying (255)	Primera imagen
Imagen2	Character varying (255)	Segunda Imagen
Id_mano_batear	Integer	Identificador de la mano de batear
Id_mano_lanzar	Integer	Identificador de la mano de lanzar
Id_equipo_comenzo	Integer	Identificador del equipo en que se inició
Id_equipo_actual	Integer	Identificador del equipo en que juega
Id_provincia	Integer	Identificador de su provincia de origen

Tabla 10 Jugador

Nombre jugador_equipo

Descripción Tipo de competencia que se ejecuta la fantasy		
Atributo	Tipo	Descripción
Id	Serial	Identificador de la tabla
Novato	Character varying (50)	Chequea si el jugador es novato
Número	Integer	Número del jugador en el equipo
Id_equipo_evento	Integer	Identificador de la relación equipo-evento
Id_jugador	Integer	Identificador del jugador
Id_posición_jugador	Integer	Identificador de la posición del jugador

Tabla 11 Jugador_Equipo

Nombre Estadísticas_juego_lanzadores		
Descripción Estadísticas de los lanzadores en un juego		
Atributo	Tipo	Descripción
Id	Serial	Identificador de la tabla
Id_jugador	Integer	Identificador del jugador
Id_evento	Integer	Identificador del evento
Id_juego	Integer	Identificador del juego
Id_posición_juego	Integer	Identificador de las posición en el juego

series	Integer	Series lanzadas
Jl	Integer	Juegos lanzados
Ji	Integer	Juegos iniciados
Jc	Integer	Juegos completos
Jr	Integer	Juegos relevados
Jp	Integer	Juegos perdidos
Jg	Integer	Juegos ganados
d	Integer	
Pro	Real	Promedio de victorias
l	Integer	Lechadas
Par	Integer	Participación en lechadas
Js	Integer	Juegos salvados
Inn	Real	Innings
Vb	Integer	Veces al bate
Be	Integer	Bateadores enfrentados
h	Integer	Hits
Ave	Integer	Average de los contrarios
c	Integer	Carreras

Pcl	Real	Promedio de carreras limpias
So	Integer	Ponches
bb	Integer	Bases por bolas
Bi	Integer	Bases intencionales
Dobles	Integer	Dobles
Triples	Integer	Triples
Hr	Integer	Jonrón
Sh	Integer	Hit de sacrificio
Sf	Integer	Fly de sacrificio
Db	Integer	Deathball
Bk	Integer	Balk
Wp	Integer	Wild pitch
lo	Integer	
ranking	Real	ranking

Tabla 12 Estadísticas_juego_lanzadores

Nombre Estadísticas_juego_bateadores
Descripción Estadísticas de los bateadores en un juego

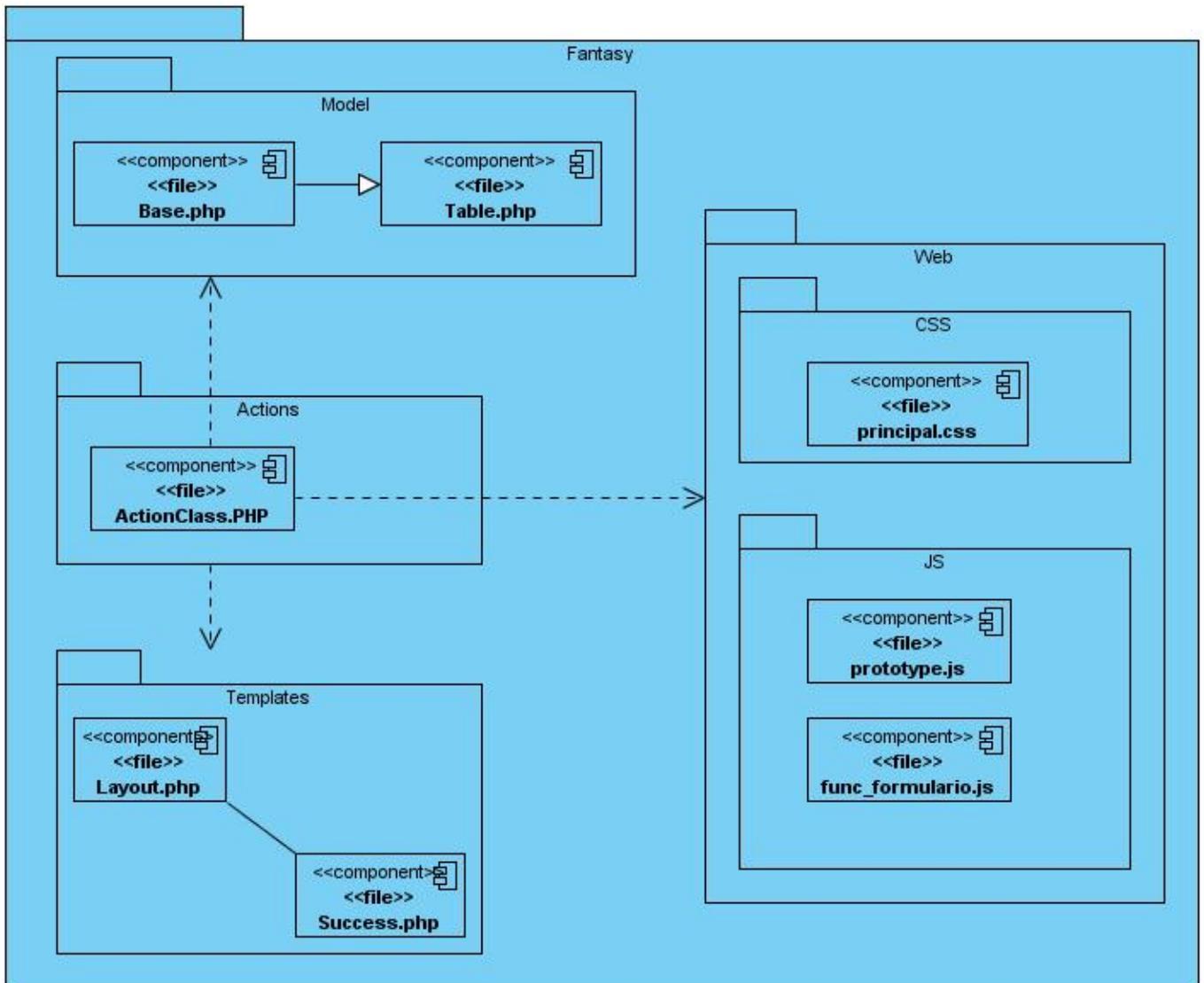
Atributo	Tipo	Descripción
Id	Serial	Identificador de la tabla
Id_jugador	Integer	Identificador del jugador
Id_evento	Integer	Identificador del evento
Id_juego	Integer	Identificador del juego
Id_posición_juego	Integer	Identificador de las posición en el juego
Series	Integer	Series jugadas
Cb	Integer	Comparecencias al bate
Vb	Integer	Veces al bate
H	Integer	Hits
Dobles	Integer	Dobles
Triples	Integer	Triples
Hr	Integer	Jonrón
Tb	Integer	Total de bases
Slg	real	Slugging
Br	Integer	Bases robadas
Cr	Integer	Cogido robando
Ci	Integer	Carreras impulsadas

Sh	Integer	Hit de sacrificio
Sf	Integer	Fly de sacrificio
Db	Integer	Deathball
Bb	Integer	Bases por bolas
Bi	Integer	Bases intencionales
So	Integer	Ponches
Bd	Integer	
Io	Integer	
Obp	real	Por ciento de embasado
Ops	real	Por ciento de embasado + slugging
Cp	Integer	
Emb	Integer	Embasado
Jj	Integer	Juegos jugados
Inn	real	Innings
O	Integer	Outs
A	Integer	Asistencias
E	Integer	Errores
Tl	Integer	Total de lances

Def	real	Promedio defensivo
Dp	Integer	Doble plays
Tp	Integer	
Pb	Integer	Pass ball
Br_d	Integer	Base robada, defensa
Cr_d	Integer	Cogido robando , defensa
ranking	real	Ranking

Tabla 13 Estadísticas_juego_bateadores

Anexo8 Diagrama de Componentes



Glosario de términos

API: (Application Program Interfase, o interfaz de comunicación entre programas de aplicación) representa un interfaz de comunicación entre componentes software.

ORM (Object-Relational Mapping, en español, Mapeo Relacional de Objetos) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

XHTML: (Lenguaje de Marcado de Hipertexto Extensible) es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de remplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos.