



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

**“COLISEO VIRTUAL”,  
SISTEMA DE APOYO AL PROCESO DE ENSEÑANZA-  
APRENDIZAJE DESDE JUEGOS DIDÁCTICOS.**

**Tesis propuesta para optar por el título de  
Ingeniero en Ciencias Informáticas**

**Autores:** Ismarai Núñez Viltres

Abdiel Matos Nieto

**Tutor:** Lic. Rolan R. Bullain Dieguez

Ciudad de la Habana  
2009

## Declaración de autoría

Declaramos ser los únicos autores de este trabajo y autorizamos al Departamento de Especialidades [Facultad 3], de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 20 días del mes de Junio del año 2009.

---

Ismarai Núñez Viltres

---

Abdiel Matos Nieto

---

Lic. Rolan R. Bullain Diéguez.

## Datos de contacto

Ismarai Núñez Viltres

Email: [iviltres@estudiantes.uci.cu](mailto:iviltres@estudiantes.uci.cu)

Grupo: 3507

Sexo: F

Abdiel Matos Nieto

Email: [anieto@estudiantes.uci.cu](mailto:anieto@estudiantes.uci.cu)

Grupo: 3506

Sexo: M

## **Dedicatoria**

Ismarai

Dedico este trabajo de diploma a mis padres, especialmente a mi mamá por su ejemplo, guía e incondicional apoyo. A mi hermano, por estar siempre conmigo y apoyarme en los momentos difíciles, a mi abuelita y el resto de mi familia. A los amigos con los que he compartido tantos años y experiencias inolvidables.

Abdiel

Dedico esta tesis además a mi padre Alberto, mi madre Magalis, y mi otra madre Mayita, que siempre han sabido darme el consejo en el momento oportuno, han sabido estar a mi lado cada vez que lo necesito, por el esfuerzo que han realizado para que yo llegue a ser quien soy. A mis hermanos porque a pesar de ser menores que yo siempre he podido contar con ellos, y han estado a mi lado. A mi padrastro por sus locuras conmigo y sus consejos. A toda mi familia en general porque he podido contar con su apoyo en todo momento.

## Agradecimientos

Agradecemos a todas las personas que nos han brindado su apoyo para la realización de este trabajo de diploma: A nuestro tutor Rolan por su constante empeño de hacernos mejores. A la Cátedra de Programación Avanzada y en especial a su administrador Anglada por compartir con nosotros sus experiencias. A los compañeros de proyecto Armando Robert Lobo y Alex González Paz, amigos y demás personas que de algún modo han hecho posible la confección de esta tesis.

Ismarai

En primer lugar le doy gracias a mi Dios, Señor y Salvador Jesucristo que ha sido el principal causante de que hasta aquí haya llegado. Él ha estado dándome fuerzas y ánimo cada día, haciéndome entender que está y estará conmigo siempre a pesar de cualquier cosa que esté pasando a mí alrededor.

Le doy gracias a mis compañeros de grupo que me han soportado y compartido conmigo estos 5 años de universidad. A mis hermanos del alma especialmente a Omar (La Geisha), Héctor Miguel (El Yoga), Armando (El Lobo), que de no ser por el mí este trabajo no hubiera tenido la misma calidad, y Lázaro Ahmed (El Lashy) que siempre han estado cuando más lo he necesitado, en cada jodedera, en las buenas y en las malas, apoyándome y ayudándome en todo momento. También agradecer a Tico, Julio, Daniel el cid, y a todos aquellos compañeros, amigos, hermanos que de cierta forma han contribuido a ser quien soy. Agradecer especialmente a Elsyta por ser quien ha sido conmigo, por estar siempre cuando se lo he pedido, y ser esa personita tan importante que siempre ha sido para mí desde primer año, que a ella le debo muchas cosas de las que hoy disfruto y soy.

Abdiel

## **Resumen**

La presente investigación da continuidad al trabajo de diploma “Análisis del Sistema Coliseo Virtual” defendido en el curso 2007-2008 por Janet Carreño Cáceres. Tuvo como entrada el análisis del Modelo del Sistema “Coliseo Virtual” obtenido a partir de la captura de requisitos. Este trabajo tiene como objetivo el desarrollo del sistema antes mencionado. El cual constituye una solución de software que permitirá al usuario participar en varios estilos de competencias, buscando potenciar el desarrollo de habilidades en las técnicas de programación, el diseño y el análisis de algoritmos, el trabajo en equipo, entre otras. Esta aplicación contará con un juez en línea que evaluará las soluciones, un generador de pruebas configurable y un servicio de mensajería para los concursos centralizados. Las opciones serán dadas al usuario mediante una interfaz gráfica amigable y de fácil acceso.

# Contenido

---

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....</b>	<b>4</b>
1.1-SISTEMAS PARA FORTALECER EL DESARROLLO DE HABILIDADES Y JUECES EN LÍNEA. ....	4
1.2 METODOLOGÍAS DE DESARROLLO .....	6
2.1 Extreme Programming .....	7
1.2.2 Microsoft Solution Framework .....	8
1.2.3 Proceso Unificado de Desarrollo .....	8
1.3 PARADIGMAS DE PROGRAMACIÓN .....	10
1.3.1-Programación estructurada .....	10
1.3.2-Programación modular .....	12
1.3.3-Programación orientada a objetos.....	12
1.3.4-Programación concurrente .....	13
1.3.5-Programación funcional.....	13
1.3.6-Programación lógica .....	14
1.3.7-Programación Orientada a aspectos.....	14
1.4 LENGUAJES.....	15
1.4.1 PHP .....	15
1.4.2 Javascript.....	16
1.4.3 UML.....	17
1.4.4 SQL .....	18
1.5 SISTEMA GESTOR DE CONTENIDOS .....	18
1.5.1 Drupal .....	19
1.5.2 Joomla! .....	20
1.6 HERRAMIENTAS CASE.....	22
1.6.1 Visual Paradigm.....	22
1.6.2 Rational Rose .....	23
1.7 SISTEMAS GESTORES DE BASE DE DATOS.....	24
1.7.1 MySQL.....	25
1.7.2 SQL Server .....	26
1.8 ENTORNO DE DESARROLLO INTEGRADO PARA APLICACIONES WEB .....	27
1.8.1 Aptana Studio .....	27
1.8.2 Zend Studio.....	28
1.9 EXT. JS.....	29
CONCLUSIONES .....	30
<b>CAPÍTULO2.DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.....</b>	<b>31</b>
2.1 VALORACIÓN CRÍTICA DEL DISEÑO PROPUESTO POR EL ANALISTA .....	31
2.1.1-Módulos del Sistema Coliseo Virtual.....	33
2.2-ANÁLISIS DE POSIBLES IMPLEMENTACIONES, COMPONENTES O MÓDULOS YA EXISTENTES.....	35
2.2.1-Juez en Línea .....	35
2.2.2-Remository.....	35
2.2.3-UddelM. Servicio de mensajería.....	36
2.3- ESTRATEGIA DE INTEGRACIÓN DE LOS COMPONENTES A UTILIZAR .....	38
2.4-ANÁLISIS DE LOS ARTEFACTOS OBTENIDOS.....	39
2.5- PATRONES DE DISEÑO .....	42
2.5.1-Modelo Vista Controlador .....	43
2.6- DESCRIPCIÓN DE LAS NUEVAS CLASES. ....	44
CONCLUSIONES .....	51
<b>CAPÍTULO3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....</b>	<b>52</b>

3.1- BÚSQUEDA O DISEÑO DE LOS TEST QUE PERMITAN VALIDAR LA SOLUCIÓN PROPUESTA .....	52
3.2-DESCRIPCIÓN DE LOS TEST DE UNIDADES .....	53
3.2.1-Módulo Prueba En línea.....	54
3.2.2- Módulo Competencia. ....	55
3.2.3-Módulo de Retos.....	57
3.2.4-Componente Enviar Solución .....	58
3.2.5-Módulo Estilo Libre.....	59
3.2.6-Módulo Test.....	60
3.2.7-Módulo Concurso.....	61
3.3-EVALUACIÓN DE LA EJECUCIÓN DE LOS TEST Y DE LOS RESULTADOS OBTENIDOS. ....	62
3.3.1-Prueba de Aceptación .....	65
<b>CONCLUSIONES GENERALES.....</b>	<b>68</b>
<b>RECOMENDACIONES .....</b>	<b>69</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>70</b>
<b>BIBLIOGRAFÍA CONSULTADA .....</b>	<b>74</b>
<b>ANEXOS.....</b>	<b>76</b>
<b>APÉNDICES .....</b>	<b>90</b>
GLOSARIO DE TÉRMINOS .....	90
ÍNDICE DE FIGURAS Y TABLAS.....	91



## Introducción

El constante desarrollo de la informática ha hecho de la misma una ciencia de múltiple empleo en la mayoría de las esferas de la vida actual, a tal sentido que constituye un elemento fundamental para la superación y desarrollo de un país. Ante el nuevo entorno económico mundial las naciones se plantean como una línea estratégica, la preparación de profesionales en áreas de la informática y las telecomunicaciones, capaces de enfrentar los retos que se tienen hoy en día. Cuba apuesta hoy al desarrollo de la informática, tomando como partida el proceso de capacitación y desarrollo de los Recursos Humanos asociados a todos los sectores, buscando el logro de una cultura generalizada en el marco de una adecuada promoción del proceso de informatización.

Se ha creado una infraestructura de universidades y politécnicos a lo largo de todo el país con el fin de fortalecer los conocimientos sobre la informática, provocar un impacto positivo en la informatización de la sociedad y contribuir a la formación de profesionales en esta esfera. Como parte de este propósito surge en el año 2002, la Universidad de las Ciencias Informáticas (UCI, en lo adelante), centro donde se desarrolla esta investigación.

La UCI posee un plan de estudio que vincula de forma directa la formación y la producción con objetivo de formar profesionales altamente capacitados. Aun así, las herramientas con objetivo de fomentar mejores resultados docentes y productivos, potenciar el aprendizaje de las técnicas de programación, análisis y diseño de algoritmos, trabajo en equipo, así como en las diferentes materias docentes, no son suficientes.

Una de las herramientas existentes (<http://cpav.uci.cu>), se encuentra patrocinada por la Cátedra de Programación Avanzada. Brinda servicio de competencias y la posibilidad de resolver los ejercicios disponibles. La misma dispone de un Jurado que permite la realización de las Copas UCI, y otras. Esta herramienta posee la limitación, de que el estudiante no puede gestionar las competencias u otros servicios similares, solo acceder a las que se encuentren en el sistema.

Se posee además, como resultado del Análisis del Sistema “Coliseo Virtual”, el levantamiento de requisitos y el análisis del Sistema. El cual permitirá a los estudiantes el desarrollo de habilidades de programación mediante competencias, pruebas en línea entre otras modalidades. Pero no se dispone aún del diseño e implementación de cada uno de sus módulos.

Por lo cual, se define como **situación problémica**: la necesidad de implementar un sistema a partir de Análisis del Sistema “Coliseo Virtual”, que potencie el desarrollo de algunas de las habilidades requeridas para el futuro desempeño profesional del Ingeniero en Ciencias Informáticas.

Definiendo el **problema científico** como: la ausencia de una aplicación de apoyo al proceso de enseñanza-aprendizaje desde sus funcionalidades, a partir de la investigación precedente.

Para darle solución al problema planteado, este trabajo tiene como **objeto de estudio**: proceso de desarrollo de software.

El **objetivo de la investigación** es: el desarrollo de un sistema que potencie el proceso de enseñanza- aprendizaje a partir de los juegos didácticos.

Cuyos objetivos específicos son:

- Desarrollar la base de datos del sistema “Coliseo Virtual”.
- Desarrollar los distintos módulos del sistema:
  - Módulo de pruebas en línea.
  - Módulo de retos.
  - Módulo de concursos.
  - Módulo de Estilo Libre.
  - Módulo de Creación y Gestión de Competencias.
- Gestionar el sistema.

El presente trabajo se enmarca en: el software de gestión y los jueces en línea, como **campo de acción**.

**Idea a Defender:** la obtención de un sistema de apoyo al proceso de enseñanza-aprendizaje centrado en los juegos didácticos, potenciará en los estudiantes el desarrollo de habilidades en diferentes materias docentes.

Para llevar a cabo el presente trabajo se definieron las siguientes **tareas de investigación:**

- Realizar un estudio del estado del arte de las herramientas a utilizar en la construcción del sistema.
- Evaluar la información obtenida sobre el estado del arte y adaptarla a nuestro rol y requerimientos.
- Realizar el Modelo de Despliegue.
- Realizar el Modelo de Implementación.
- Modelar la Base de Datos.
- Implementar la Base de Datos.
- Definir la Arquitectura.
- Analizar de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados y estrategia de integración
- Describir los módulos.

El presente trabajo está organizado en tres capítulos, donde se abordan los siguientes temas: *Capítulo 1*, Incluye un estado del arte de los jueces en línea existentes. Se muestra un análisis de las metodologías de desarrollo del software y los diferentes paradigmas de programación, así como las tendencias, tecnologías, herramientas, y plataformas de desarrollo que la soportan. *Capítulo 2*, Muestra la valoración crítica del diseño propuesto por el analista, el análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados y la estrategia de integración seguida. Se describen los algoritmos fundamentales y se determina la complejidad algorítmica de los mismos. *Capítulo 3*, Se realiza una validación de la solución propuesta y un análisis de los resultados y el trabajo realizado. Se define una estrategia de prueba para testear las funcionalidades del sistema. Se evalúa el cumplimiento de los requisitos definidos por el analista.

# Capítulo 1. Fundamentación Teórica

El presente capítulo incluye un estado del arte de las aplicaciones y jueces en línea existentes en el ámbito internacional y nacional, con funcionalidades similares al sistema a desarrollar. Se muestra un estudio de las metodologías de desarrollo de software existentes y los diferentes paradigmas de programación actuales. Se realiza un análisis de las tendencias, tecnologías y herramientas, así como de las plataformas de desarrollo que la soportan.

## 1.1- Sistemas para fortalecer el desarrollo de habilidades y Jueces en línea.

Existen a nivel mundial muchas aplicaciones web desarrolladas con el objetivo de fomentar habilidades en la búsqueda de soluciones mediante la programación, que poseen la implementación de un Juez en línea. Se encuentran generalmente patrocinadas por universidades, aunque la mayor patrocinadora de estos eventos es la Asociación para la Maquinaria de Computación (ACM, por sus siglas en inglés). En algunos casos, responden a los intereses de las grandes compañías, en su afán de poseer a su servicio los mejores profesionales. Dada sus características funcionales similares al sistema informático a desarrollar, se ha considerado oportuno realizar un breve análisis de las mismas.

- <http://acm.zju.edu.cn>: Se encuentra ubicado en China, patrocinado por la Universidad de Zhejiang y por la ACM. El usuario debe registrarse en el sitio. Se llevará una estadística de su desempeño en la solución de problemas (cantidad de problemas resueltos, lenguaje en el que ha sido implementado, tiempo de ejecución, tamaño del código). Para enviar la solución es necesario: el identificador del usuario, el identificador del problema, contraseña del usuario y seleccionar el lenguaje (C, C++), no se puede adjuntar el fichero fuente. De cada problema también se lleva una estadística y un ranking de los participantes (Zhejiang, 2009).
- <http://acm.timus.ru>: Ubicado en Rusia, patrocinado por la Universidad Estatal Ural. El usuario debe registrarse en el sitio, donde se llevará una estadística de

- su desempeño en la solución de problemas (cantidad de problemas, lenguaje en el que ha sido implementado, tiempo de ejecución, memoria usada). Para enviar la solución es necesario: el identificador del usuario, el identificador del problema, donde recibirá la respuesta y seleccionar el lenguaje (C, C++, Pascal, C#, Java), se puede adjuntar el fichero fuente o se puede copiar (Timus, 2009).
- *<http://acm.sgu.ru>*: Ubicado en Rusia, patrocinado por la Universidad de Sarátov. El usuario debe registrarse en el sitio, donde se llevará una estadística de su desempeño en la solución de problemas (cantidad de problemas aceptados, lenguaje en el que ha sido implementado, tiempo de ejecución, memoria usada). Para enviar la solución es necesario: el identificador del usuario, el identificador del problema, contraseña del usuario y seleccionar el lenguaje (Delphi 6.0, GNU C, GNU C++, Visual Studio C 7.1, Visual Studio C++ 7.1, Java 1.5), no se puede adjuntar el fichero fuente (Sarátov, 2008).
  - *<http://acm.pku.edu.cn/JudgeOnline>*: Ubicado en China, patrocinado por la Universidad de Pekín y ACM. El usuario debe registrarse en el sitio, donde se llevará una estadística de su desempeño en la solución de problemas (cantidad de problemas resueltos, lenguaje en el que ha sido implementado, tiempo de ejecución, memoria usada). Para poder enviar una solución es necesario autenticarse primero. Los lenguajes aceptados son: (C, C++, Java, C#, Delphi 6.0, GNU C, GNU C++). Como característica relevante tiene un juez en línea que puede ser descargado (PKU, 2007).
  - *<http://imaginecup.com>*: Concurso patrocinado por Microsoft, que se desarrolla a nivel mundial. Su objetivo fundamental es la casa de talentos. Actualmente cuenta con más de de 10 estilos de competencias, entre ellos se destacan: diseño de software, desarrollo incorporado, desarrollo web, algoritmo, fotografía en arte digital, cortos en arte digital y diseño de interfaz. Las competencias se realizan por equipos, cuya cantidad de integrantes varía dependiendo del estilo de competencia, se realiza a nivel internacional en varias etapas (Microsoft, 2009).
  - *Copa Pascal*: Es un evento dirigido por la Cátedra de Programación Avanzada y patrocinado por la dirección de la UCI. Solo está disponible en el momento del

concurso. Las veces que se ha usado un juez en línea, ha variado la cantidad de lenguajes que acepta. En ocasiones se lleva un ranking sobre los problemas solucionados por los usuarios o equipo (UCI, 2009).

A pesar de que la aplicación a desarrollar con este trabajo, no coincide en fines con algunas de las herramientas antes mencionadas, las mismas aportan ideas y experiencias que con un adecuado tratamiento pueden proporcionar visibles beneficios en el trabajo docente.

## **1.2 Metodologías de desarrollo**

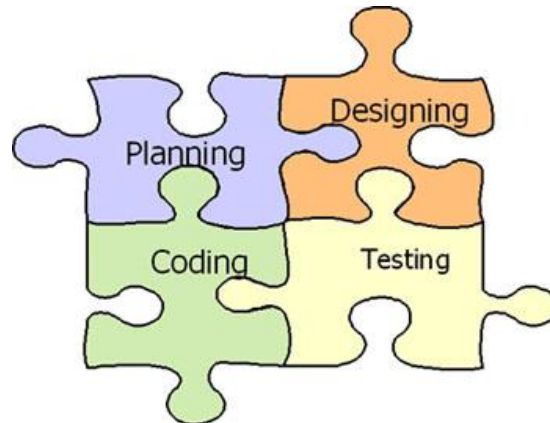
Una metodología de desarrollo es un medio de estandarización que cubre completamente el proceso de desarrollo de un software. Posibilita el movimiento en doble sentido entre las fases del ciclo de desarrollo para verificar trabajos realizados y corrección de los errores detectados. Provee un entendimiento efectivo entre los analistas, programadores, clientes, usuarios y gestores para una correcta toma de decisiones (Mendoza, 2004).

Una metodología de desarrollo especifica cuáles son las tareas a realizar y quiénes son los responsables de las mismas. Resulta necesario tener en cuenta que no es lo mismo hacer uso de un modelo de estandarización para un sitio Web que para el desarrollo de una aplicación Desktop. Existe una gran diversificación de metodologías de desarrollo, útiles según las características del proyecto que se desee desarrollar.

Existen dos tipos de metodologías: las Robustas y las Ágiles. Entre las metodologías Robustas se encuentran Rational Unified Process (RUP), Microsoft Solutions Framework (MSF), Métrica 3.0, entre otras. Dentro de las metodologías Ágiles se encuentran: Extreme Programming, Scrum y Feature Driven Development, de estos dos tipos de metodologías las más consolidadas dentro de su categoría son RUP y XP.

En (Mendoza, 2004).nos resalta como las metodologías más usadas: XP, MSF y RUP; razón por la cual se decide hacer un análisis más profundo de estas metodologías de desarrollo.

## 2.1 Extreme Programming



**Fig. 1** Metodología XP

La Programación Extrema XP (Extreme Programming, en inglés), es una metodología ágil nacida en el verano de 1996 de la mano de Kent Beck cuando éste se incorpora a Chrysler Corporation para desarrollar una aplicación de nóminas, donde las condiciones existentes conducían a este tipo de metodologías no convencionales que él propuso aplicar. La misma se encuentra centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Le da prioridad a las tareas que dan un resultado directo y que reducen la burocracia que hay alrededor de la programación (Beck, 1999).

XP, se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y responsabilidad a la hora de afrontar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Beck, 1999).

El sistema a desarrollar cuenta con requisitos bien definidos y estables. Necesita ser documentado de forma eficiente para los usuarios finales y posterior uso. Por tales motivos no se considera óptimo el uso de esta metodología en el presente trabajo.

## 1.2.2 Microsoft Solution Framework

Microsoft Solution Framework (MSF, en lo adelante) es una flexible e interrelacionada serie de conceptos, modelos y prácticas de uso que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos (Microsoft y Rational, 1998) .MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.



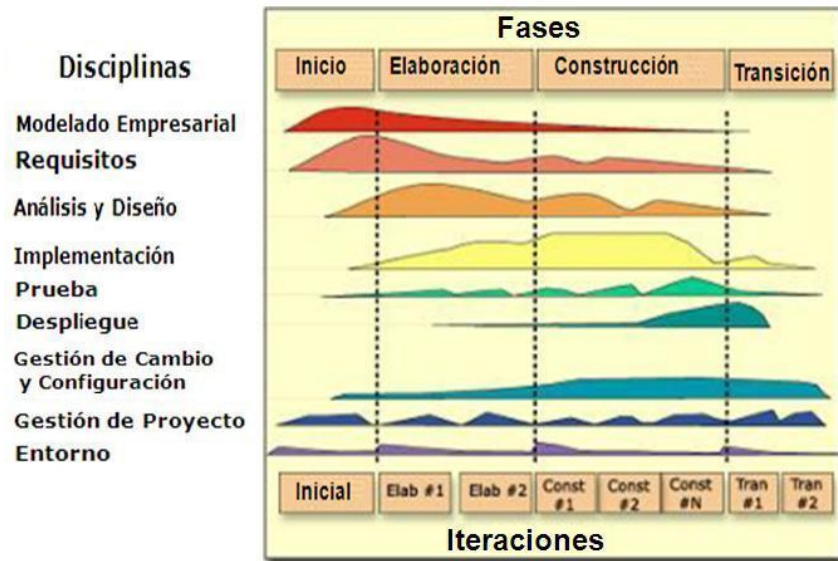
Fig. 2 Metodología MSF

Originalmente creado en 1994 para resolver los problemas a los que se enfrentaban las empresas en sus respectivos proyectos. Se ha convertido posteriormente en un modelo práctico que facilita el éxito de los proyectos tecnológico. MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación (Mendoza, 2004).

## 1.2.3 Proceso Unificado de Desarrollo

El Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés) constituyó el resultado de tres décadas de desarrollo y trabajo práctico. Transitó desde el Proceso “Objectory” con su primera publicación en 1987 hasta el “Proceso Objectory del Rational” en 1997, para luego en 1998, finalmente tomar el nombre de “Proceso Unificado de Rational”. Es un proceso de desarrollo de software que ofrece un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. El Proceso Unificado es más que un simple proceso: es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos (Jacobson, y otros, 2000).





**Fig. 3** Metodología RUP

Este proceso está basado en componentes y como aspectos fundamentales, se destacan que es dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental.

Entre las características de RUP definidas por Rational Software Corporation, en el año 2003 se encuentran las siguientes:

- *El desarrollo iterativo del software:* el software es desarrollado bajo un ciclo de iteraciones. Una iteración incorpora una serie de actividades secuenciales en el modelado de negocios, requerimientos, análisis y diseño, implementación, prueba, despliegue y, en varias proporciones dependiendo en qué parte del ciclo de desarrollo se encuentre la iteración.
- *La gestión de los requerimientos:* es un enfoque sistemático para encontrar, documentar, organizar requisitos, además de un sistema de seguimiento de la evolución de las necesidades.
- *Una arquitectura basada en componentes:* se basan en la independencia, componentes modulares, ayuda a administrar la complejidad y fomentar la reutilización. La arquitectura permite obtener y conservar el control intelectual sobre el proyecto, es una base efectiva para la reutilización a gran escala y proporciona una base para la gestión de proyectos. El sistema estará constituido

de discretos componentes ejecutables, que se desarrollan independientemente unos de otros, posiblemente creados por diferentes equipos.

- *El uso de software de modelado visual:* el modelado visual es el uso de notaciones gráficas y textuales de diseño semánticamente ricas para capturar el diseño de software. Un lenguaje, como por ejemplo la UML, permite plantear un nivel de abstracción, manteniendo una rigurosa sintaxis.
- *La verificación de la calidad del software:* es importante que la calidad de todos los artefactos sea evaluada en varios momentos del ciclo de vida del proyecto a medida que madura el mismo. Deben ser comprobados tanto los artefactos como las actividades que los producen y se completan al término de cada iteración.
- *El control de cambios del software:* describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a utilización y actualización concurrente de elementos y control de versiones.

Las prácticas anteriormente mencionadas hacen de RUP, la metodología estándar más eficaz para la implementación y documentación del sistema. Además permite la generación de artefactos y documentación correcta por cada fase y flujo, que tributan a un buen producto final.

### **1.3 Paradigmas de programación**

Un paradigma de programación es un estilo fundamental definido por la forma de dar solución a problemas. Un paradigma proporciona y determina la visión que el programador tiene acerca de la ejecución de un programa (Miranda, 2008).

En este epígrafe se muestra un breve análisis de algunos de los paradigmas existentes. Se hará énfasis en el paradigma que incluye el lenguaje seleccionado para el desarrollo del sistema.

#### **1.3.1-Programación estructurada**

La programación estructurada (PE, en lo adelante), es un paradigma de programación mediante el cual se elaboran programas haciendo uso de tres estructuras básicas de control lógico:

- *Secuencia:* Indica la ejecución ordenada de las instrucciones, una tras otra, siguiendo el orden en que aparecen en el programa. Se representa gráficamente como una caja después de otra, ambas con una sola entrada y una única salida.
- *Selección:* Conocida además como la estructura SI-CIERTO-FALSO, consiste en la selección entre dos alternativas de acuerdo a la evaluación de una condición o predicado preestablecido. Es equivalente a la instrucción IF de todos los lenguajes de programación.
- *Iteración:* También nombrada HACER-MIENTRAS-QUE, corresponde a la ejecución repetida de una instrucción mientras que se cumple una determinada condición predefinida.

Un programa estructurado se compone de: funciones, segmentos, módulos y/o subrutinas; cada una con una sola entrada y una sola salida. La PE plantea según su teorema fundamental que cualquier programa, sin importar el trabajo ejecutado por el mismo se puede elaborar mediante el uso de las tres estructuras básicas: secuencia, selección, iteración.

Elaborar programas de mediana y alta complejidad haciendo uso de este paradigma requiere mayor esfuerzo, creatividad y dominio de la misma. Aun así con este estilo podemos obtener las siguientes ventajas:

- *Los programas son más fáciles de entender:* pueden leerse de forma secuencial, sin tener que realizar saltos según la lógica del programa, lo cual resulta común en otros estilos de programación. La claridad en la estructura del programa debido a la relación entre las instrucciones, hace más fácil de comprender cada función.
- *Reducción del esfuerzo en las pruebas:* el programa se puede tener listo en poco tiempo. El proceso de depuración se facilita debido a la lógica simple. Los errores se pueden detectar y corregir más fácilmente.

### **1.3.2-Programación modular**

La programación modular es un paradigma de programación que consiste en dividir un programa en módulos ó subprogramas con el fin de hacerlo más legible y manejable. Consta de varias secciones divididas de forma que interactúan a través de llamadas a procedimientos, que integran el programa en su totalidad. En la programación modular, el programa principal coordina las llamadas a los módulos secundarios y pasa los datos necesarios en forma de parámetros. A su vez cada módulo puede contener sus propios datos y llamar a otros módulos o funciones (Pérez, 2009).

Un módulo es cada uno de las partes en que se divide un programa con objetivo de hacer más fácil la solución del problema inicial. Estos módulos poseen tareas y funciones bien definidas. Además poseen una interfaz definida que le permitan comunicarse con otros módulos en caso de que lo necesite.

La programación modular es reconocida como una evolución de la programación estructurada, ya que da solución a problemas de programación más grandes y complejos de lo que ésta puede resolver.

### **1.3.3-Programación orientada a objetos**

La Programación Orientada a Objetos (POO, en lo adelante) es un paradigma de programación muy utilizado en la actualidad. Posee una base teórica robusta y es reconocible el número de herramientas que permiten crear código teniendo como partida diseños desarrollados sobre este paradigma. Las características anteriores la convierten en la alternativa más adecuada para el desarrollo de aplicaciones (Winblad, 1993).

El uso de esta técnica aumenta de modo considerable la velocidad de desarrollo de los programas gracias a la reutilización de los objetos, que constituyen el elemento fundamental, estos últimos son un conjunto complejo de datos que poseen estructura y forman parte de una organización.

Los métodos (comportamiento) y atributos (estado) están estrechamente relacionados por la propiedad de conjunto. Esta propiedad destaca que una clase requiere de métodos para poder tratar los atributos con los que cuenta. (Winblad, 1993)

La POO se diferencia de la programación estructurada tradicional, en cuanto a la organización y relación entre los datos.

Debido a las ventajas que posee este paradigma en cuanto a que: describe el problema en términos similares a los que utiliza la mente humana, permite construir soluciones más seguras y con un mantenimiento más seguro, será usado para el desarrollo del presente trabajo.

#### **1.3.4-Programación concurrente**

La computación concurrente consiste en la ejecución de múltiples tareas interactivas de forma simultánea. Estas tareas pueden ser un conjunto de procesos o hilos de ejecución creados por un único programa. Las tareas se pueden ejecutar en una sola unidad central de procesos (multiprogramación), en varios procesadores o en una red de computadoras distribuidas (Giordani, 2009).

La programación concurrente difiere de la programación paralela en cuanto a su énfasis en la interacción entre tareas, a pesar de que se encuentran relacionadas. La clave de esta disciplina radica en su correcta interacción o comunicación entre procesos y el acceso coordinado de recursos compartidos.

Este paradigma se utiliza mayormente cuando se necesita realizar varias acciones de forma simultánea. Que pudieran ser el control el acceso de usuarios y programas a un recurso de forma concurrente.

#### **1.3.5-Programación funcional**

La Programación funcional es un paradigma de programación declarativa basado en la utilización de funciones matemáticas. (Chávez, 2007)

Entre sus características propias se encuentran: no posee construcciones estructuradas como la secuencia o la iteración, hace obligatorio el uso de funciones recursivas y que no posee asignación de variables.

Los programas escritos con este paradigma, están compuestos solamente por funciones matemáticas en las que se evidencian características como la transparencia referencial, evidenciada en que el significado de cada función depende únicamente del

significado de sus subexpresiones y por tanto, existe una carencia total de efectos colaterales.

Las categorías de lenguajes funcionales existentes: puros e híbridos, se diferencian en que los lenguajes funcionales híbridos, son menos dogmáticos que los puros, ya que admiten conceptos pertenecientes a los lenguajes imperativos como la asignación de variables y la secuencia de instrucciones. Sin embargo los lenguajes funcionales puros poseen una mayor potencia expresiva, conservando a la vez su transparencia referencial, algo que no se cumple siempre con un lenguaje funcional híbrido.

### **1.3.6-Programación lógica**

La programación lógica, junto con la funcional, forma parte de lo que se conoce como programación declarativa. En los lenguajes tradicionales, la programación consiste en indicar cómo resolver un problema mediante sentencias; en la programación lógica, se trabaja de una forma descriptiva, estableciendo relaciones entre entidades, indicando no cómo, sino qué hacer (Rossel, 2004).

Permite formalizar hechos del mundo real mediante reglas y restricciones. Ante las cuales es posible establecer hipótesis o deducciones que el ordenador se encarga de comprobar mediante un motor de inferencia basado en la lógica de primer orden.

La ecuación propuesta por Robert Kowalski (Universidad de Edimburgo) muestra la idea central de la programación lógica: *algoritmos = lógica + control*. Evidenciando que un algoritmo se construye mediante un lenguaje formal y el problema se resuelve mediante un mecanismo de inferencia (Rossel, 2004).

La programación lógica es usada mayormente en la inteligencia artificial y otros programas que posean características similares. Se basa en una teoría matemática que permite lograr que mediante hechos y reglas, un ordenador pueda dar soluciones inteligentes. Está basada en el cálculo de predicados.

### **1.3.7-Programación Orientada a aspectos**

La Programación Orientada a Aspectos (POA, en lo adelante) es un paradigma de la programación relativamente reciente cuya intención es permitir una adecuada

modularización de las aplicaciones y posibilitar una mejor separación de conceptos. (Quintero, 2000)

Este paradigma permite encapsular los diferentes conceptos que componen una aplicación, eliminando las dependencias entre cada uno de los módulos. Lo cual posibilita eliminar la dispersión de código, razonar mejor sobre conceptos y las implementaciones resultan más comprensibles, adaptables y reusables. El término POA es usado para referirse a varias tecnologías relacionadas como los métodos adaptativos, los filtros de composición, la programación orientada a sujetos o la separación multidimensional de competencias.

A pesar de ser uno de los paradigmas más avanzados, no será usado en el desarrollo de esta investigación dada la siguiente característica que imposibilita el óptimo desarrollo y mantenimiento de la aplicación a desarrollar: utiliza datos compartidos que quedan esparcidos por todo el código, con lo que integrar un nuevo tipo de datos implica modificar varias funciones (Quintero, 2000).

## **1.4 Lenguajes**

Un lenguaje de programación es un conjunto de instrucciones entendibles directamente o traducibles al lenguaje del ordenador con el que se trabaje; combinando estas instrucciones realizamos programas (Eidos, 2000).

### **1.4.1 PHP**

El lenguaje PHP, ha sido diseñado especialmente para crear páginas web dinámicas y usado en la interpretación del lado del servidor (Server-side scripting). Aunque actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas (Thomson, 2003).

PHP, es un lenguaje de propósito general usado para el desarrollo de aplicaciones a web y puede ser embebido dentro de código HTML. Es ejecutado en un web, creando páginas web a partir del código PHP que toma como entrada. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno (Southwell, 2005).

Entre las ventajas fundamentales que este lenguaje posee, se encuentran:

- Alto rendimiento (Thomson, 2003).
- Es un lenguaje multiplataforma (Thomson, 2003).
- Posee capacidad de conexión con gran parte de los manejadores de datos existentes, en especial con MySQL.
- Brinda la posibilidad de expandir su potencial utilizando los módulos (llamados ext's o extensiones).
- Posee una amplia documentación donde se explican las funciones del sistema.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Posee una Biblioteca nativa de funciones sumamente amplia.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde php5).

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación (Rodas, 2005).

La aparición de soluciones más adecuadas y sencillas hace que PHP se convierta en la mejor opción actual para multitud de necesidades. Es uno de los paquetes para programación de internet más utilizados: PHP4 está en el 77.72% de los sitios, PHP5 en el 21.91% según encuestas actuales (php.net, 2007). Su condición de lenguaje para el diseño de aplicaciones web, junto a sus facilidades de uso, seguridad y su excelente combinación con herramientas como la base de datos MySQL justifican su selección para el desarrollo del presente trabajo.

### **1.4.2 Javascript**

Este es un lenguaje interpretado creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web. El mismo, a pesar de su semejanza con Java, se diferencia de este en que no dispone de herencia. La mayoría de los navegadores en sus últimas versiones interpretan código Javascript (Eguíluz, 2009).



El código Javascript puede ser integrado dentro de las páginas web. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseño un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento) (Pérez, 2007).

Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, con funciones y estructuras de datos complejas. Además, pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente. Estas características permiten que se considere su uso en el desarrollo del presente trabajo.

### **1.4.3 UML**

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (Booch, 2007).

Resulta importante destacar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Es utilizado para definir un sistema de software, detallar los artefactos en el sistema, documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar. (Booch, 2007).

El lenguaje de modelado UML cuenta con la participación de metodólogos influyentes, de importantes empresas y un estándar del *Object Management Group* (OMG), siendo utilizado diariamente por grandes organizaciones como: Microsoft, Oracle y Rational. Sus características y funcionalidades hacen propicia su selección como lenguaje de modelado para el marco de este trabajo.

#### 1.4.4 SQL

SQL (Structured Query Language, en inglés), es un lenguaje de consulta estructurado y declarativo de acceso a bases de datos relacionales que permite realizar diversas operaciones sobre las mismas. Posee manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre la misma.

El SQL es un lenguaje que explota la flexibilidad y permite gran número operaciones sobre los sistemas relacionales (Casares, 2004). Se hace uso de este lenguaje para acceder a la base de datos del sistema, dotando el acceso de las ventajas que el mismo posee.

### 1.5 Sistema Gestor de Contenidos

Un Sistema de gestión de contenidos (CMS, por sus siglas en inglés) es un software que permite crear una estructura de soporte (framework) para la creación y administración de contenidos por parte de los participantes principalmente en páginas web. (Cuerda, 2004)

Consiste en una interfaz, que posee el control sobre una o varias bases de datos donde se almacena el contenido del sitio. Permite manejar de manera independiente el contenido y el diseño. De esta forma, es posible durante el trabajo con el contenido, darle si se prefiere, un diseño distinto al original, sin tener que cambiar el formato del contenido existente. Además de permitir la publicación de varios editores, de forma fácil y controlada.

Actualmente existen muchos CMS según sus funciones o finalidad: noticias, blog, comercio electrónico, entre otras. Entre ellos:

- **Joomla:** Joomla está programado en PHP para Bases de Datos MySQL.
- **Mambo:** Programado en PHP para Bases de Datos MySQL.
- **Drupal.** Drupal está programado en PHP para Bases de Datos MySQL y PostgreSQL.
- **Plone:** Plone sobre Zope, programado en Python para todo tipo de BD.
- **Muravey-Web:** Programado en Ruby sobre Rails para todo tipo de BD.

- **Typo3:** Programado en PHP4 para Bases de Datos MySQL (y otras)

La aplicación a desarrollar requiere de tecnología multiplataforma y de código abierto para garantizar su utilidad posterior. Por el anterior motivo se ha considerado solo el análisis de aquellos que brindan soporte para el desarrollo de portales, que sean de código abierto, programados en el lenguaje anteriormente seleccionado PHP 5. Resultaron los CMS a analizar: Drupal y Joomla.

### **1.5.1 Drupal**

Drupal es un sistema de gestión de contenido para sitios Web. Permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos (HandBook, 2005).

*Drupal es un sistema dinámico:* no almacena de forma estática sus archivos en el sistema, el contenido textual de las páginas y otras configuraciones, son almacenados en una base de datos y se editan utilizando un entorno Web incluido en el producto. Se distribuye bajo la licencia GNU GPL, y por lo tanto es software libre (drupal.org, 2008).

*Drupal es un sistema de gestión de contenido modular y muy configurable:* Entre sus principales facilidades se destacan, la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema. El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar diferentes sitios y aplicaciones web (UXI, 2009).

A continuación se ofrecen algunas de las características fundamentales que posee este CMS ofrecidas en el sitio oficial de la comunidad en español de Drupal (Drupal, 2005).

*Módulos:* La comunidad de Drupal ha contribuido muchos módulos que proporcionan funcionalidades como “página de categorías” autenticación mediante *jabber*, mensajes privados, bookmarks, y otros (Mellado, 2005).

*Personalización:* Posee un robusto entorno de personalización implementado en su núcleo. Tanto el contenido como la presentación pueden ser individualizados de acuerdo las preferencias definidas por el usuario (Alejandro, 2008).

*Autenticación de usuarios:* Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo como Jabber, Blogger, LiveJournal u otro sitio Drupal.

*Permisos basados en roles:* Los administradores de Drupal pueden asignar permisos a un “rol” y agrupar los usuarios por roles., evitando tener que configurar permisos para cada usuario.

*Plantillas (Platillas):* El contenido y la presentación se encuentran separados, lo que permite controlar o cambiar fácilmente el aspecto del sitio web.

*Agregador de noticias:* Incluye un potente Agregador de Noticias para leer y publicar enlaces a noticias de otros sitios web. Incorpora un sistema de cache en la base de datos, con temporización configurable.

*Independencia de la base de datos:* Drupal incorpora una 'capa de abstracción de base de datos que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

*Multiplataforma:* Drupal ha sido diseñado desde el principio para ser multi-plataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solari, Windows y Mac OS X. Además es un CMS portable, debido al lenguaje en que está desarrollado: PHP. Por tales motivos se considera óptimo para el desarrollo del sistema.

### **1.5.2 Joomla!**

Es un sistema de administración de contenidos de código abierto construido con PHP bajo una licencia GPL. Es usado para publicar en Internet e intranets utilizando una base de datos MySQL. Posee un número considerables de facilidades, entre ellas: hacer caché de páginas para mejorar el rendimiento, indexar web, feed RSS, versiones

imprimibles de páginas, flash con noticias, blogs, foros, polls (encuestas), calendarios, búsqueda en el sitio web, e internacionalización del lenguaje.

Su nombre es una pronunciación fonética para anglófonos de la palabra swahili jumla que significa "todos juntos" o "como un todo". Se escogió como una reflexión del compromiso del grupo de desarrolladores y la comunidad del proyecto. (Trevejo, 2009)

Joomla corresponde al grupo de soluciones de código abierto, es un producto de software libre. Para el desarrollo de sus múltiples frentes, usa diferentes formas de comunicación como son: los encuentros por medio de IRC, foros, listas de correo, "wikis" y blogs.

A continuación se ofrecen algunas de las características fundamentales que posee este CMS.

*Organización del sitio web:* Joomla permite organizar eficientemente los contenidos de un sitio en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla para los administradores. Esto también forma parte de la optimización para buscadores (codenova.net, 2009).

*Publicación de Contenidos:* Con su uso, se pueden crear páginas ilimitadas. Permitiendo su edición desde un sencillo editor (similar al editor de MS Word) que permite darle a los textos el formato e imágenes deseadas. Los contenidos son totalmente editables y modificables.

*Escalabilidad e implementación de nuevas funcionalidades:* Joomla ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos. Lo que permitirá agregar servicios de valor para los usuarios finales, por ejemplo: galerías de imágenes, foros, newsletters, clasificados, entre otros (codenova.net, 2009).

*Administración de usuarios:* Joomla permite almacenar datos de usuarios registrados y también te da la posibilidad de enviar E-mails masivos a todos los usuarios. La administración de usuarios es jerárquica, y los distintos grupos de usuarios poseen diferentes niveles de facultades/permisos dentro de la gestión y administración del sitio (White y otros, 2006).

*Disposición de módulos modificable:* En un sitio creado con Joomla, la posición de módulos puede acomodarse como se prefiera. Se conoce por módulos los bloques laterales del sitio Web, por ejemplo: banners y menús.

*Plataformas compatibles:* Es compatible con las plataformas Linux, Windows y Mac OSX. (LeBlanc, 2007)

Tras un análisis de los CMS Drupal y Joomla y las ventajas que ambos poseen en cuanto a su portabilidad, gestión de usuarios, de contenidos y disposición de módulos, se concluye que el uso de cualquiera de los mismos, permite el desarrollo de un sistema a la altura de los requerimientos definidos para el “Coliseo Virtual” (Ver Tabla 1). Se seleccionó Joomla debido a su gran interfaz gráfica configurable, facilidad de aprendizaje y personalización, además de ser del que mayor experiencia poseen los autores de este trabajo.

## **1.6 Herramientas CASE**

Según el Instituto Nacional de Estadística e Informática de Perú, se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. (INEI, 1999)

### **1.6.1 Visual Paradigm**

Visual Paradigm para UML es una herramienta UML profesional que da soporte al análisis y diseño orientados a objetos, construcción, pruebas y despliegue del ciclo de desarrollo de un software. Permite la rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Brinda la posibilidad de crear todos los tipos de diagramas de clases, código inverso.

Brinda la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir de código. Esta es

precisamente una gran ventaja puesto que el sistema será desarrollado en PHP. Visual Paradigm soporta la captura de los requisitos del sistema y la identificación de las clases candidatas, mediante el análisis textual.

Esta herramienta se encuentra disponible en múltiples plataformas y en múltiples versiones. Sus características lo hacen una opción viable ante herramientas como el Rational Rose, que es una herramienta muy recomendada y además profesional, pero obliga al usuario a desarrollar en máquinas con el sistema operativo Windows, mientras que el Visual Paradigm está disponible para varios sistemas operativos como Windows, Linux, Unix. Por tales motivos resulta ser Visual Paradigm la herramienta utilizada para el desarrollo del presente trabajo.

### **1.6.2 Rational Rose**

Es una herramienta CASE que cubre todo el ciclo de ayuda de un proyecto: concepción y formalización del modelo, construcción de los componentes y certificación de las distintas fases. Permite trazabilidad real entre modelo (análisis y diseño) y el código ejecutable.

Sus principales características son:

- Mantiene la consistencia de los modelos del sistema software.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Chequeo de la sintaxis UML.
- Genera documentación automáticamente.
- Generación de código a partir de los modelos.
- Disponibilidad en múltiples plataformas.
- Capacidades de Ingeniería Inversa.

Rational Rose es reconocida como una herramienta de las más técnicas debido a que se encarga de llevar a cabo la automatización de los sistemas para la posterior generación de código. Permitiendo a los desarrolladores y arquitectos de software utilizar un lenguaje común para poder visualizar el sistema completo (Boggs, 2002). También pueden los diseñadores modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. Posee el inconveniente

de que no es multiplataforma motivo por el cual no fue seleccionada para el presente trabajo.

## **1.7 Sistemas Gestores de Base de Datos**

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos (Ramos, 2008). Entre las características fundamentales que deben poseer los SGBD, según Gary W.Hansen (Hansen, 1997) se encuentran las analizadas a continuación.

*Abstracción de la información.* Los usuario no interaccionan directamente con la base de datos y por consiguiente con el almacenamiento físico de la información. Con lo que pueden definirse varios niveles de abstracción

*Independencia.* Consiste en mantener la independencia entre el esquema (físico o lógico) de una base de datos y las aplicaciones que se sirven de ella.

*Consistencia.* Se han de tomar precauciones con la actualización de la información almacenada en caso de que no se haya logrado eliminar la redundancia. Con el objetivo de que sea actualizada de forma coherente y los datos repetidos se actualicen de forma simultánea.

*Seguridad.* Deben garantizar la seguridad de la información almacenada en el mismo ante el uso no autorizado de usuarios o ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Es común que los SGBD dispongan de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

*Integridad.* Se refiere a las medidas que ha de adoptar el gestor para garantizar la validez de los datos almacenados, o proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada

*Respaldo.* Deben realizarse copias de respaldo de la información en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.



*Control de la concurrencia.* Debe controlar el acceso concurrente a la información para evitar inconsistencias

*Manejo de Transacciones.* Una transacción se ejecuta como un solo programa. Lo cual indica que el estado luego de una ejecución en la que se produce una falla es el mismo que se obtendría si el programa no se hubiera ejecutado. Los SGBD deben proporcionar mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.

*Tiempo de respuesta.* Debe minimizar el tiempo que tarda en ofrecer información solicitada y en almacenar los cambios realizados a los mismos.

### **1.7.1 MySQL**

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca (Gallego, 2003).

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. (Welling, 2003)

Entre las características que hacen posible su gran aceptación se encuentra la existencia de infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Entre las características principales definidas en el Manual de Referencia MySQL 5.0 por Sun Microsystems, Inc. Se encuentran:

- Alto rendimiento.
- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP).

- Gran portabilidad entre sistemas.
- Facilidad de configuración y aprendizaje.
- Gestión de usuarios y contraseñas manteniendo un muy buen nivel de seguridad en los datos.

Se considera que dadas las potencialidades de este gestor, y las características del sistema, podrá usarse para su desarrollo.

### **1.7.2 SQL Server**

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) capaz brindar a muchos usuarios, grandes cantidades de datos de manera simultánea

Microsoft SQL Server se basa en el lenguaje Transac SQL. Fue creado por la Microsoft y resulta competente ante otros potentes sistemas gestores de bases de datos como son Oracle, Sybase ASE, PostgreSQL, Interbase, Firebird, o MySQL.

Entre sus características fundamentales se encuentran:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Se decide no hacer uso de este gestor por su condición de ser propietario, debido además a que el sistema a desarrollar requiere de un gestor sobre plataforma libre.

Una vez analizadas las características y facilidades de los SGBD presentados, y las de la herramienta a desarrollar se decide usar el MySQL como SGBD, por las siguientes razones:

- El PHP maneja más fácil al MySQL que al SQL Server, debido a la gran cantidad de funciones que tiene explícitas.
- El MySQL es multiplataforma. (Welling, 2003)
- El MySQL no tiene precio en el mercado, se adquiere libremente.

## **1.8 Entorno de Desarrollo Integrado para Aplicaciones Web**

Un IDE (*Integrated Development Environment*) es un entorno de programación que se compone un editor de código, un compilador, un depurador y en ocasiones un constructor de interfaz gráfica GUI, que han sido empaquetados como un programa de aplicación.

Los IDEs pueden ser aplicaciones que brindan funcionalidades propias o que sirven de apoyo a aplicaciones existentes. Disponen de un editor de código que además, puede servir para depurar y facilitar las diferentes tareas necesarias en el desarrollo de cualquier tipo de aplicación.

### **1.8.1 Aptana Studio**

Aptana Studio es un entorno de desarrollo integrado para la elaboración de aplicaciones web dinámicas que empleen PHP, Ruby, Ruby on Rails y Python (Aptana, Inc., 2009).

Es multiplataforma y de código abierto, enfocado al desarrollo de interfaces de usuario Web dinámicas ideal para maquetar páginas Web. Soporta las librerías más populares: Prototype, Scriptaculous, Dojo, MochiKit, Yahoo UI, Aflax, JQuery y Rico, HTML, CSS y JavaScript. (Pérez, 2008)

Posee un asistente de código que ayuda al programador en la escritura de los diferentes lenguajes, CSS y Javascript entre ellos. Aptana Studio contiene también información de soporte para los principales navegadores web: IE, Firefox, Opera, Netscape y Safari. (Aptana, Inc., 2009).

Otras características interesantes de Aptana son las siguientes:

- Explorador de código en forma de árbol.
- Librerías populares AJAX/Javascript.
- Extensión de funcionalidad mediante macros y acciones.

- Visor de errores y advertencias.
- Servidor local para probar el código.

Una característica considerada muy útil para la presente investigación es que brinda la posibilidad de publicar y descargar archivos directamente en un servidor de producción asociando el proyecto a una conexión FTP (Aptana, Inc., 2009).

Su condición de software libre y multiplataforma junto a sus facilidades de uso, lo hacen útil para el desarrollo del presente trabajo según las funcionalidades definidas para el mismo.

### **1.8.2 Zend Studio**

Zend Studio o Zend Development Environment es un entorno de desarrollo integrado para el trabajo con el lenguaje de programación PHP. Se encuentra implementado en Java y disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux. (Álvarez, 2009)

Junto a Zend Platform, constituye una solución de Zend Technologies para el desarrollo de aplicaciones Web utilizando el lenguaje PHP. Dispone de Zend Studio como la parte cliente y Zend Platform brinda servicio de servidor. Resulta curioso que ambos casos sean software comercial en contraste con PHP que es de Software. Aun así, Zend Studio, es uno de los IDEs, más completos para desarrollar con el lenguaje PHP (The PHP Company, 2009).

Entre las características más relevantes que posee se encuentran las siguientes:

- No es necesario realizar la instalación del lenguaje PHP ni el entorno de ejecución Java de forma previa y brinda soporte para PHP 4 y PHP 5.
- Posee autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
- PHPDoc integrado.
- Detección de errores de sintaxis en tiempo real.
- Funciones de depuración: Botón de ejecución y traza, marcadores, puntos de parada (breakpoints), seguimiento de variables y mensajes de error del intérprete de PHP.

- Soporte para gestión de grandes proyectos de desarrollo.
- Manual de PHP integrado.
- Soporte para control de versiones usando CVS o Subversion (a elección del desarrollador).
- Cliente FTP integrado.
- Soporte para navegación en bases de datos y ejecución de consultas SQL.
- Aunque fue diseñado para usarse con el lenguaje PHP; pero ofrece soporte básico para otros lenguajes Web, como HTML, Javascript y XML.

A pesar de las potencialidades y facilidades que brinda, no se ajusta a las características definidas para el sistema a desarrollar por su condición de software propietario.

## **1.9 Ext. JS**

Ext.JS es una biblioteca de Javascript para la construcción de aplicaciones web interactivas utilizando técnicas tales como AJAX, DHTML y DOM (Document Object Model). Fue construido originalmente como un complemento de extensión de la biblioteca de YUI (The Yahoo! UI Library). Posee interoperabilidad con jQuery y Prototipo.

Ext. JS incluye un conjunto diverso de GUI basados en los controles (o "widgets") para su uso en aplicaciones web, algunos de estos son:

- Campo de texto y controles de entrada de texto.
- Campos de fecha con un pop-up-selector de fecha.
- Los campos numéricos.
- Editor HTML de control.
- Barras de herramientas.
- Aplicación de escritorio de estilo menú.

Muchos de estos controles son capaces de comunicarse con un servidor web utilizando AJAX. Otras de sus características es la inclusión de un selector DOM de clase que permite realizar operaciones sobre los elementos dentro de la página, crear clases y

administrar datos en formatos XML y JSON. Se hará uso de Ext. JS para el trabajo con Javascript por las facilidades que ofrece esta librería al lenguaje antes mencionado.

## **Conclusiones**

Luego del análisis de las metodologías de desarrollo de software, paradigmas de programación, lenguajes y herramientas anteriormente expuestas, se propone la siguiente solución.

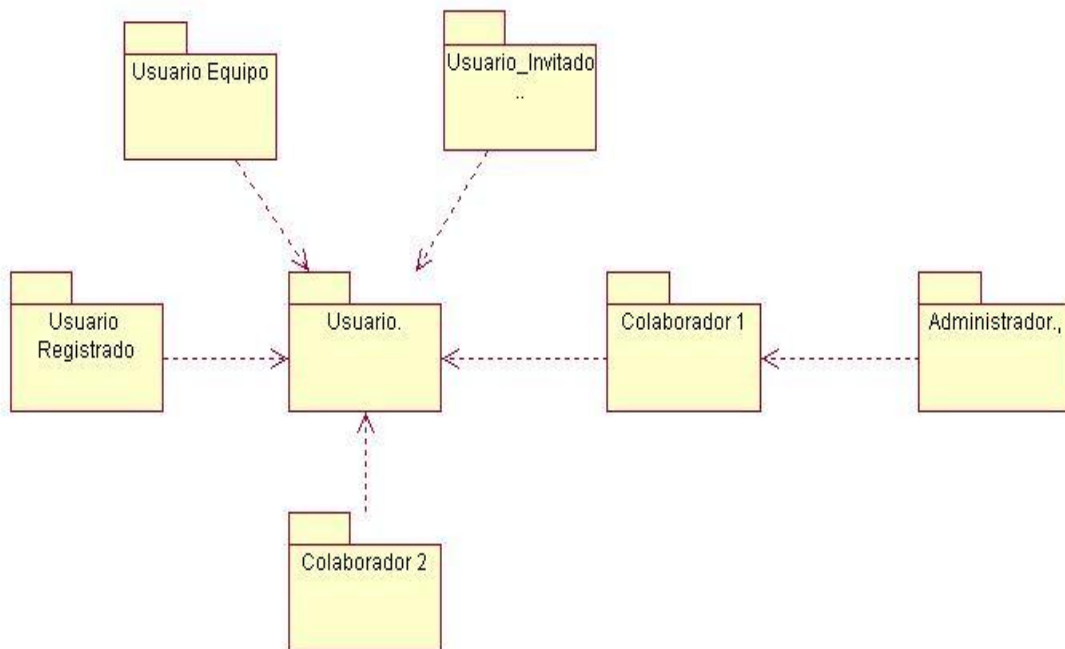
El uso de RUP como metodología de desarrollo, debido a las potencialidades que brinda en cuanto a la documentación, desarrollo iterativo y su arquitectura basada en componentes. Se selecciona la Programación Orientada a Objetos como paradigma para llevar a cabo el desarrollo del sistema por la amplia gama características que posee y por incluir las bondades de otros paradigmas de programación. De los lenguajes de programación existentes que soportan este paradigma se selecciona el lenguaje PHP5, por su seguridad, alto rendimiento y compatibilidad con el gestor de base de datos MySQL. Este último, gestor multiplataforma y de software libre, que vienen a hacer en conjunto características claves del CMS seleccionado: Joomla. Se hará uso del entorno de desarrollo Aptana Studio por ser multiplataforma y de código abierto.

## Capítulo 2. Descripción y análisis de la solución propuesta.

Se realiza una valoración crítica del diseño propuesto por el analista y se presenta el análisis de posibles implementaciones, componentes o módulos ya existentes que puedan ser rehusados. Se muestra un análisis de los artefactos más importantes obtenidos como parte del diseño e implementación y la descripción de las clases utilizadas.

### 2.1 Valoración crítica del diseño propuesto por el analista

El diseño propuesto por el analista está compuesto por un conjunto de paquetes, que se organizan de acuerdo a los diferentes actores del sistema en orden jerárquico.



**Fig. 4** Diseño propuesto por el analista

El modelo anterior proporciona una vista detallada sobre cada uno de los casos de uso con que se relaciona el actor de acuerdo a sus funcionalidades y privilegios. Resulta adecuado para concebir la jerarquía de actores y definir los módulos con que se

relacionan de forma individual. No obstante, tras el objetivo de tener una vista central sobre el funcionamiento del sistema en su totalidad y lograr una arquitectura sólida que garantice un buen diseño de la base de datos, se considera más óptimo realizar un modelo general que evidencie la interacción del usuario con los módulos del sistema. Con este fin, se realiza el siguiente diagrama de clases del diseño.

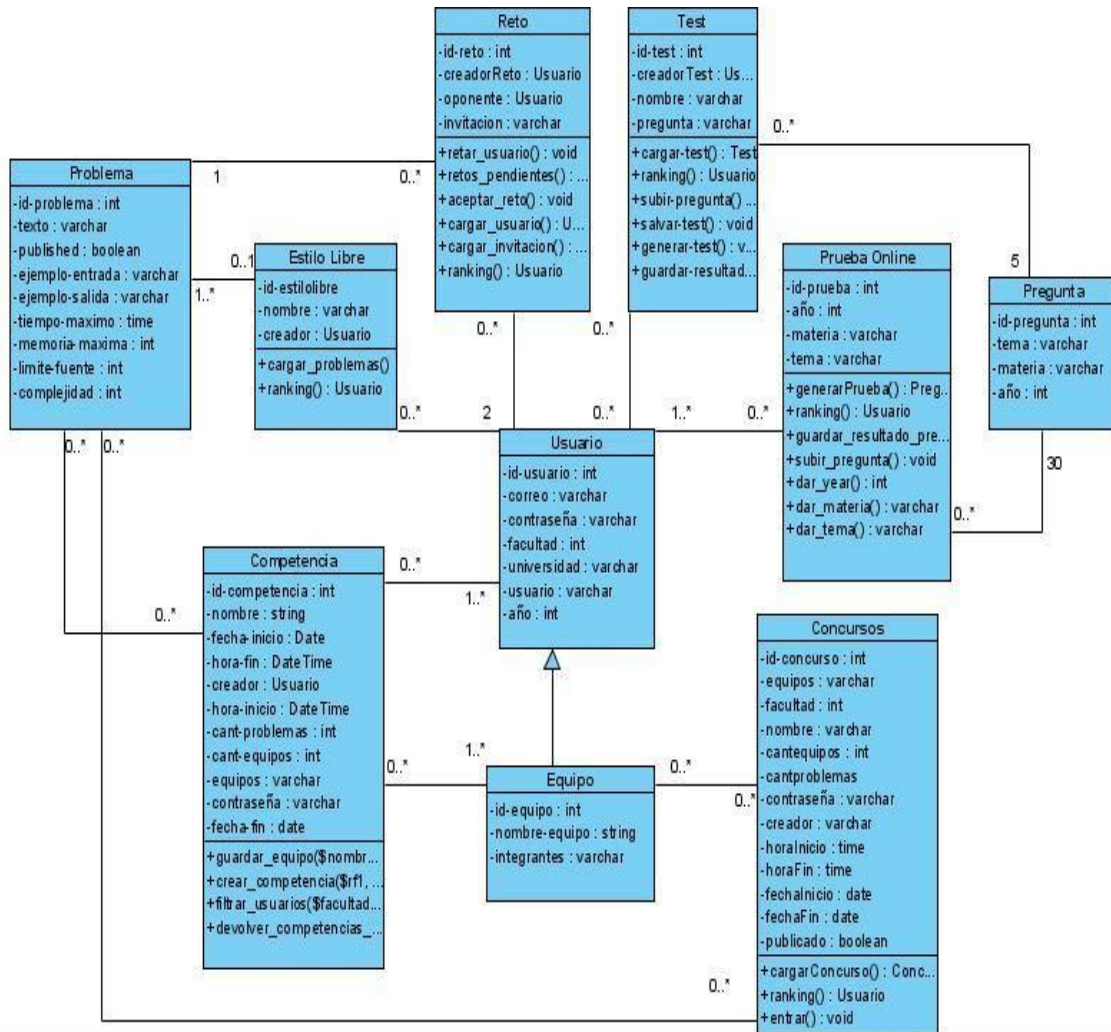


Fig. 5 Diseño de clases del sistema.



### **2.1.1-Módulos del Sistema Coliseo Virtual**

*Módulo Pruebas en Línea:* este módulo contiene un generador de pruebas configurable según el año(s), la materia(s) y el tema(s) que el usuario desee. Una vez seleccionados los parámetros anteriores se generará una prueba de treinta preguntas en correspondencia a los mismos, con un tiempo límite definido. Las calificaciones se sumaran al ranking del usuario en este estilo de competencia y en el ranking general. El usuario tendrá las opciones de guardar la prueba, abandonar la misma o pasar a la pregunta siguiente. Se gestionará toda la información relacionada con la publicación y desarrollo de las pruebas en línea, buscando ayudar a la consolidación y apropiación del conocimiento en diferentes áreas.

*Módulo Retos:* es el encargado de gestionar la creación y desarrollo de los Retos por parte de los usuarios que acceden al sistema. El usuario tendrá la opción de seleccionar su compañero de reto del listado de usuarios registrados en el sistema. El sistema enviará un mensaje de solicitud al usuario seleccionado. Una vez aceptada la solicitud se le mostrará el problema asignado. El usuario podrá enviar la solución mediante el componente “Enviar Solución”, además de “Retar usuario”, “Aceptar solicitud de reto”, consultar el “Ranking” o “Salir” del módulo si es de su interés. Tiene como objetivo potenciar el desarrollo de habilidades en el análisis y diseño de algoritmos, rapidez en la codificación de programas correctos y robustos.

*Módulo Concursos:* en este módulo se encuentra toda la información relacionada con la publicación y desarrollo de los concursos en línea. El administrador será el encargado de su publicación, para lo cual contará con una clave de acceso. Este módulo posee un servicio de mensajería para contactar al juez durante el desarrollo de concursos centralizados. Los problemas pertenecientes al concurso disponible estarán publicados en el enlace Recursos del Menú del Sistema en la portada del sitio. El usuario podrá enviar la solución mediante el componente “Enviar Solución”, además de “Participar en un concurso” o “Salir” del módulo. Tiene como objetivo potenciar el desarrollo de las habilidades antes mencionadas al explicar el módulo: “Retos”, además habilidades organizativas, trabajo en equipo, liderazgo y aprendizaje colaborativo.

*Módulo Test:* en este módulo se encuentran los test disponibles en el sistema. El usuario podrá seleccionar el que desea resolver. Cada test cuenta con 5 preguntas a responder sin tiempo establecido. Cada pregunta será calificada y sumada al ranking individual del usuario. El mismo tendrá la posibilidad de “Abandonar el test”, pasar a la pregunta “Siguiente” del mismo, consultar el “Ranking” o “Salir” del módulo. Tiene como objetivo potenciar el desarrollo de agilidad mental, mediante el análisis y el pensamiento lógico - deductivo.

*Módulo de Estilo Libre:* es el encargado de gestionar el desarrollo de los problemas en que participan los usuarios que acceden al sistema. Las pruebas de este estilo estarán publicadas y a disposición de los usuarios. Cuentan con la información del tiempo, memoria máxima y límite de fuente permisible para su resolución. El usuario podrá enviar la solución mediante el componente “Enviar Solución”, consultar los “Envíos realizados” con anterioridad, consultar el “Ranking” o “Salir” del módulo. Este estilo de competencia contribuye al desarrollo de las habilidades antes mencionadas en los módulos de “Pruebas en línea”, “Retos” y “Concurso” con la diferencia que va más encaminado a la consolidación, complementación y apropiación del conocimiento; pues el estudiante no tiene restricción de tiempo para la búsqueda de la solución de un problema y puede intentar hasta que su solución propuesta sea correcta.

*Módulo de Creación y Gestión de Competencias:* es el encargado de gestionar todo el proceso de creación y desarrollo de las competencias disponibles en el sistema. El usuario creará su propia competencia ingresando los datos de entrada requeridos y el equipo al que quiera pertenecer en caso de que lo desee. Podrá crear su propio equipo, seleccionando sus integrantes del listado de usuarios registrados disponible en el sistema. El usuario podrá enviar la solución mediante el componente “Enviar Solución”. Tiene como objetivo potenciar el desarrollo de las habilidades antes mencionadas al explicar el módulo de “Concursos”, la diferencia radica en la flexibilidad de que cada usuario pueda crear su propia competencia.

*Módulo Gestión del sistema:* es este un subsistema rector, encargado de gestionar la información relacionada con los módulos anteriores y la administración del sistema. Brinda posibilidad de gestión de los estilos de competencia: Test, Concurso y Competencia. Permite asignar preguntas a los test, problemas a las competencias, equipos a los concursos, así como adicionar o eliminar cada uno de ellos.

## **2.2-Análisis de posibles implementaciones, componentes o módulos ya existentes.**

Se realizó una búsqueda de posibles implementaciones, componentes o módulos ya existentes que brindaran funcionalidades al sistema, con objetivo de obtener piezas claves para su funcionamiento sin pasar por el proceso de construcción. A partir de este análisis se decide utilizar el componente “Juez en Línea” para el proceso de calificación de las respuestas a los diferentes estilos de competencia enviadas por los usuarios y el componente “Remository” para la publicación y descarga de recursos como tutoriales y herramientas que estarán disponibles al usuario. Además, ante la necesidad de ofrecer un modo de “Contactar al juez en línea” en los concursos centralizados, se decide usar el componente “UddelM”, para ofrecer el servicio de mensajería.

### **2.2.1-Juez en Línea**

Constituye una solución de software que tiene como entrada un segmento de código y los parámetros de tiempo y memoria definidos para el desarrollo del mismo. Llevará a cabo el proceso de compilación y evaluación, además de emitir un resultado. Trabaja integrado al sistema en el proceso de calificación de las soluciones emitidas por el usuario.

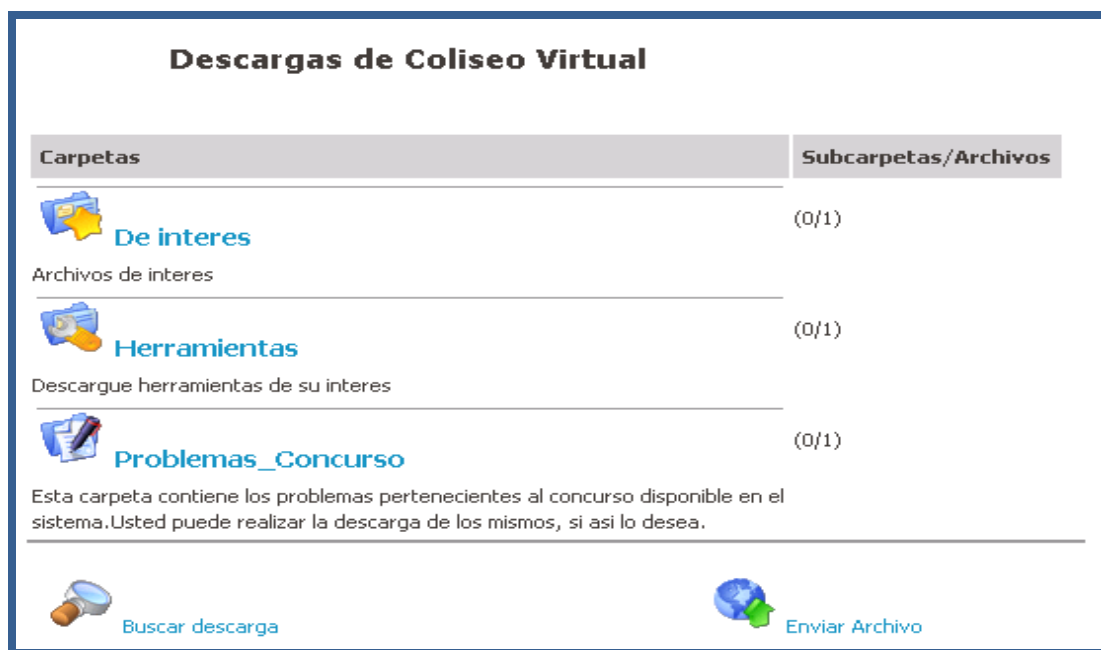
Una vez enviada la respuesta, el Juez en línea se encargará de tomar esa solución, evaluarla y emitir un resultado de la calificación que se le hará llegar el usuario en términos de: aceptado, incorrecto, error de compilación, tiempo excedido o memoria excedida. El usuario no tendrá interacción directa con el Juez, por tal motivo no se incluye en la interfaz del sistema. Este proceso se realizará de forma interna.

### **2.2.2-Remository**

Remository es un gestor de descargas para el CMS Joomla. Permite subir archivos al sitio, que estarán disponibles a la descarga para los usuarios. Posee un funcionamiento muy correcto, permitiendo crear categorías, dar permisos a usuarios entre otras facilidades.

Este componente posibilita la publicación de recursos en el sistema y una fácil administración y descarga de los mismos. El administrador define las secciones que se presentarán al usuario: De interés, Herramientas y Problemas Concurso. Publicará en las mismas cuantos tutoriales, compactados, video tutoriales desee con tamaño de archivo máximo definidos en la configuración del componente.

El usuario tendrá acceso a este componente a través del enlace “Recursos”, del Menú del “Sistema” disponible en la portada. En este lugar dispondrá del manual de usuarios y los problemas pertenecientes a los concursos que estén activos en el sistema.



**Fig. 6** Componente Remository

Este componente resulta de notable importancia para el sistema, facilita el cumplimiento del requisito funcional: “Gestionar Recursos” y “Descargar Problema Concurso” y los requisitos no funcionales: “Apariencia profesional” y “Simplicidad de uso”, definidos por el analista.

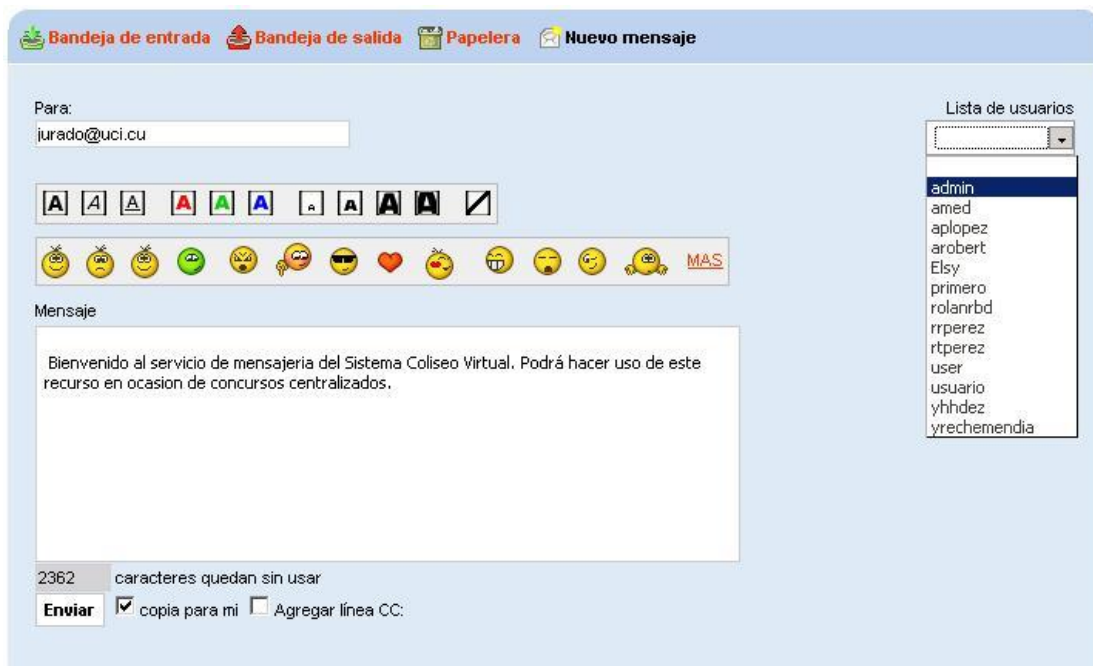
### 2.2.3-UddeIM. Servicio de mensajería

UddeIM es un componente que ofrece todas las características que necesita un sistema de mensajería privada (SPM) para Joomla 1.0, Joomla 1.5 y Mambo 4.5, disponible en 28 idiomas. Es altamente configurable: puede definirse el tamaño límite de los mensajes, recibir notificaciones, enviar mensajes para todos los usuarios, mostrar

mensajes de alerta, entre disímiles opciones más.

La lista de características es muy amplia, entre ellas:

- Listas de distribución restringida.
- Mensaje RSS.
- Mensaje cifrado.
- Archivo de mensajes.
- Varios destinatarios.
- Mensaje de transmisión.
- PhpBB códigos.
- Smileys estáticos y animados.
- Tres módulos, dos extensiones, cuatro CB Plugins.



**Fig. 7** Componente UddeIM

Los administradores podrán habilitar la Protección Captcha contra spam y Protección CSRF que protege de todas las formas de ataque del tipo "Cross-Site Request Forgery (falsas solicitudes).

Con el uso de este componente, se brinda bajo una interfaz sencilla y de forma amena la posibilidad tanto a usuarios como a administradores de poseer un medio de contacto durante el desarrollo de concursos centralizados.

## **2.3- Estrategia de integración de los componentes a utilizar**

Para llevar a cabo la integración de los componentes ya existentes: “Juez en Línea”, “UddeIM” y “Remository”, que se usan en el sistema “Coliseo Virtual” se probaron de forma independiente, lo cual se conoce como pruebas unitarias o pruebas individuales, verificando el comportamiento de los mismos de forma externa y analizando la implementación interna de la unidad.

Cumplido satisfactoriamente este nivel de prueba, se realizó la integración de los mismos al sistema. Los componentes UddeIM y Remository fueron adicionados por la administración al conjunto de componentes de Joomla. Se comprobó su funcionamiento integrado al resto de las funcionalidades del sistema desde los enlaces “Contactar Juez en Línea” y “Recursos” respectivamente.

El Juez en Línea fue integrado a la aplicación mediante un socket. Lo cual se define como un método para la comunicación entre un programa del cliente y un programa del servidor en una red mediante una dirección IP, un protocolo y un número de puerto. Una vez realizada esta conexión se realizan casos de pruebas que evidencien el funcionamiento del componente integrado al sistema.

Teniendo en cuenta los requisitos capturado en la investigación precedente: Análisis del Sistema Coliseo Virtual, el Juez en Línea permitirá el buen funcionamiento de los referidos a: Enviar Solución, Gestionar Competencia, gestionar Reto, Gestionar Estilo Libre y demás estilos que requieran el proceso de calificación. No aparecerá directamente en la interfaz de la aplicación así que no guarda relación directa con ningún requisito no funcional concebido anteriormente.

El gestor Remository permitirá dar cumplimiento al requisito funcional Gestionar Recursos definido por el analista. Estará disponible en una interfaz al usuario accesible desde el menú Sistema, disponible en la portada del sitio. Facilitará la gestión de los recursos disponibles en el sistema.

El componente UddeIM posibilitará el envío de mensajes entre usuario y administradores en ocasión de concursos centralizados. De este modo garantiza el

cumplimiento del requisito funcional Contactar Juez En línea, definido por el analista entre las funcionalidades para el estilo de competencias: Concurso.

El siguiente diagrama muestra la interacción entre los componentes que forman el sistema. Entre ellos, los antes mencionados a reutilizar en la aplicación, así como el componente de control de acceso propuesto por el CMS Joomla para tal propósito.

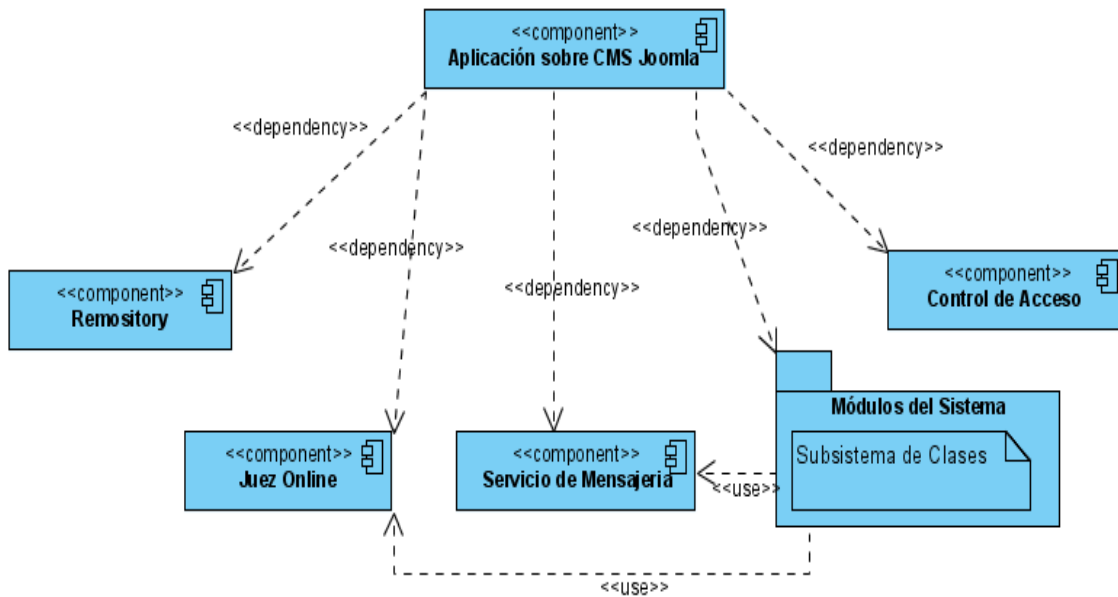
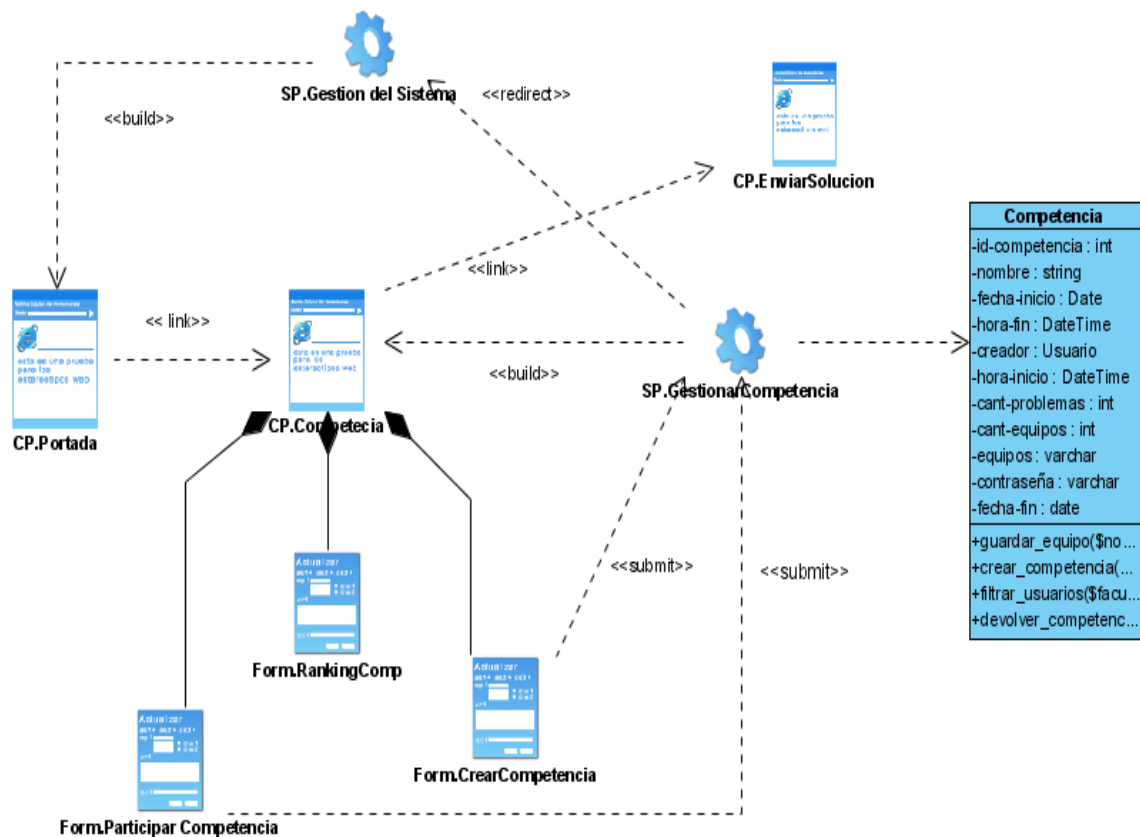


Fig. 8. Diagrama de componentes del Sistema.

## 2.4-Análisis de los artefactos obtenidos

Como parte del proceso de diseño e implementación se obtuvieron artefactos que sirvieron de apoyo al proceso de codificación. Entre los más relevantes se encuentran los diagramas de clases de diseño para cada módulo del sistema, diagrama de componentes y de implementación.

Los diagramas de clases del diseño muestran las clases fundamentales que dan funcionalidad al sistema según su arquitectura Web (paginas clientes, paginas servidoras y formularios) y la interacción entre ellas. A continuación se muestra el diagrama de clases del diseño perteneciente al Módulo Competencia, los restantes pueden consultarse en los Anexos.

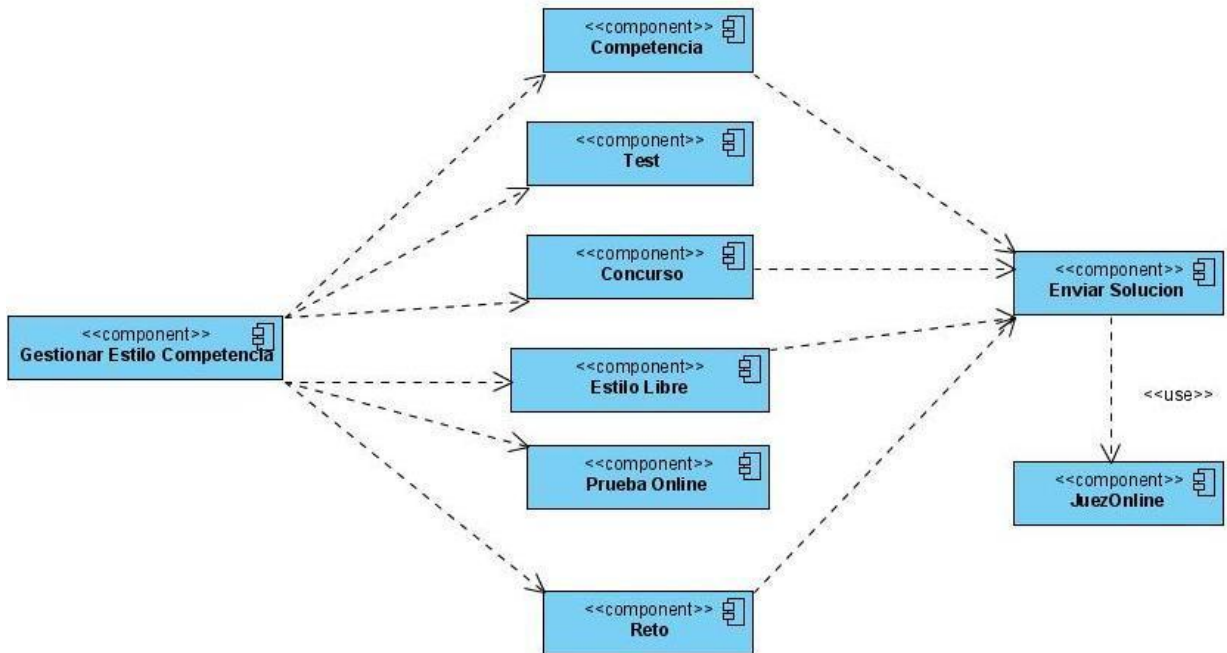


**Fig. 9.** Diagrama de Clases del Diseño. Módulo Competencia

El diagrama anterior muestra el enlace directo entre la página cliente de la portada del sistema y la que pertenece al módulo Competencia. La cual posee tres formularios: “Participar Competencia”, con funcionalidad de mostrar las competencias disponibles en el sistema; el formulario “Ranking” y “Crear Competencia”, formulario donde el usuario entrará los datos pertinentes para crear su competencia. La página cliente del Módulo posee además un enlace al componente enviar solución, representado por la página cliente “Enviar Solución”

Estas clases fueron implementadas junto a otros elementos lógicos, dando lugar a elementos físicos llamados componentes. A continuación se muestra uno de los diagramas obtenidos, los restantes están disponibles en los Anexos.

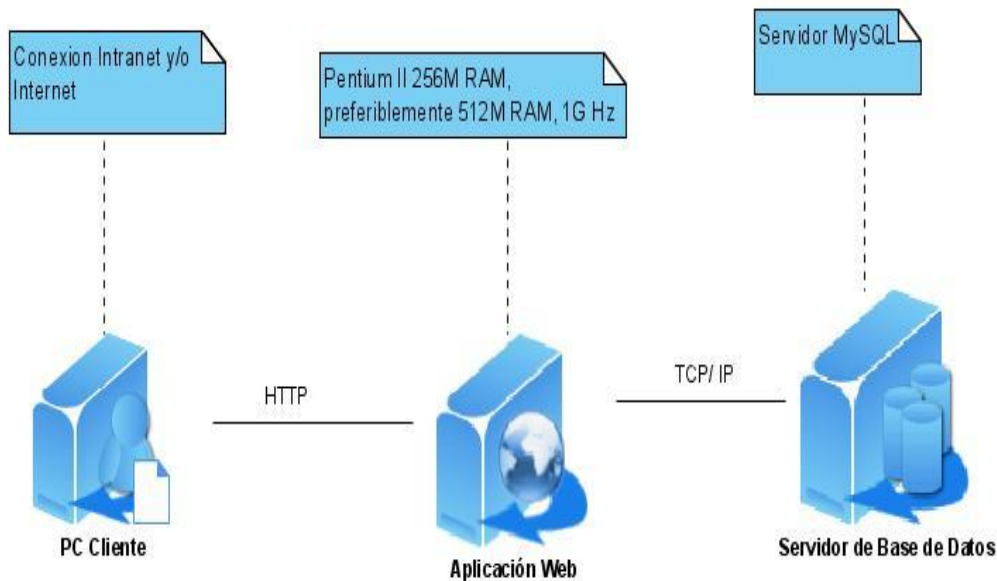




**Fig. 10.** Diagrama de componentes. Estilo de Competencia

El anterior diagrama muestra la relación de los módulos: “Competencia”, “Concurso”, “Estilo Libre” y “Reto” con el componente “Enviar Solución” y a su vez con el componente ya existente a reutilizar: “Juez Online”.

Con el objetivo de concebir la arquitectura física del sistema para su ejecución, mediante procesadores, dispositivos y componentes de software se elabora el modelo de despliegue mostrado a continuación concibiendo así la topología del sistema.



**Fig. 11.** Diagrama de componentes. Usuario

Para el uso de la aplicación se necesita una computadora cliente con conexión intranet o a internet, que se conectara mediante el protocolo HTTP al servidor de la aplicación web. Dicho servidor debe ser como mínimo Pentium II de 256 de RAM, para una concurrencia máxima de 50 usuarios, preferiblemente de 512 de RAM o 1G Hz si se estima una concurrencia mayor. El servidor de la aplicación estará conectado mediante el protocolo TCP/IP al gestor de base datos MySQL

## 2.5- Patrones de Diseño

Los patrones de diseño son un buen método para resolver pequeños problemas que pueden adaptarse a una variedad mucho más amplia de problemas específicos. En el contexto de los sistemas y aplicaciones basados en Web, los patrones de diseño pueden aplicarse en el nivel jerárquico, nivel de componentes y nivel de hipertexto (de navegación). (Pressman, 2005)

Christopher Alexander define un patrón como “un *patrón* describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución a ese problema de tal modo que se puede aplicar esta solución un millón de veces, sin hacer lo mismo dos veces” (Gamma, 2006).

### 2.5.1-Modelo Vista Controlador

Modelo Vista Controlador (MVC, en lo adelante) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

El **Modelo** contiene las funciones llamadas "lógica de negocio", o sea datos y reglas de negocio. Lleva un registro de las vistas y controladores del sistema. Cada acceso a datos se pone en su función individual porque, de esta forma, si se cambia de gestor de bases de datos este cambio sólo afecta a estas funciones, no al resto de la aplicación. Tener el modelo bien delimitado permite la existencia de varias aplicaciones que compartan el mismo modelo. (ANDRAM, 2008)

La **Vista**, en una aplicación web, muestra la información del modelo al usuario. Tienen un registro de su controlador asociado (normalmente porque además lo instancia). Tener la vista separada del controlador permite cambiar la aplicación para que genere, en lugar de HTML, algo distinto (por ejemplo, WML), sin tener que tocar más que una parte completamente delimitada del código. (ANDRAM, 2008)

El **Controlador** une la vista y el modelo. Por ejemplo, son las funciones que toman los valores de un formulario, consultan la base de datos (a través del modelo) y producen valores, que la vista tomará y convertirá en HTML. En resumen, gestiona las respuestas del sistema ante entradas del usuario. El MVC es un patrón ampliamente utilizado en múltiples plataformas y lenguajes.

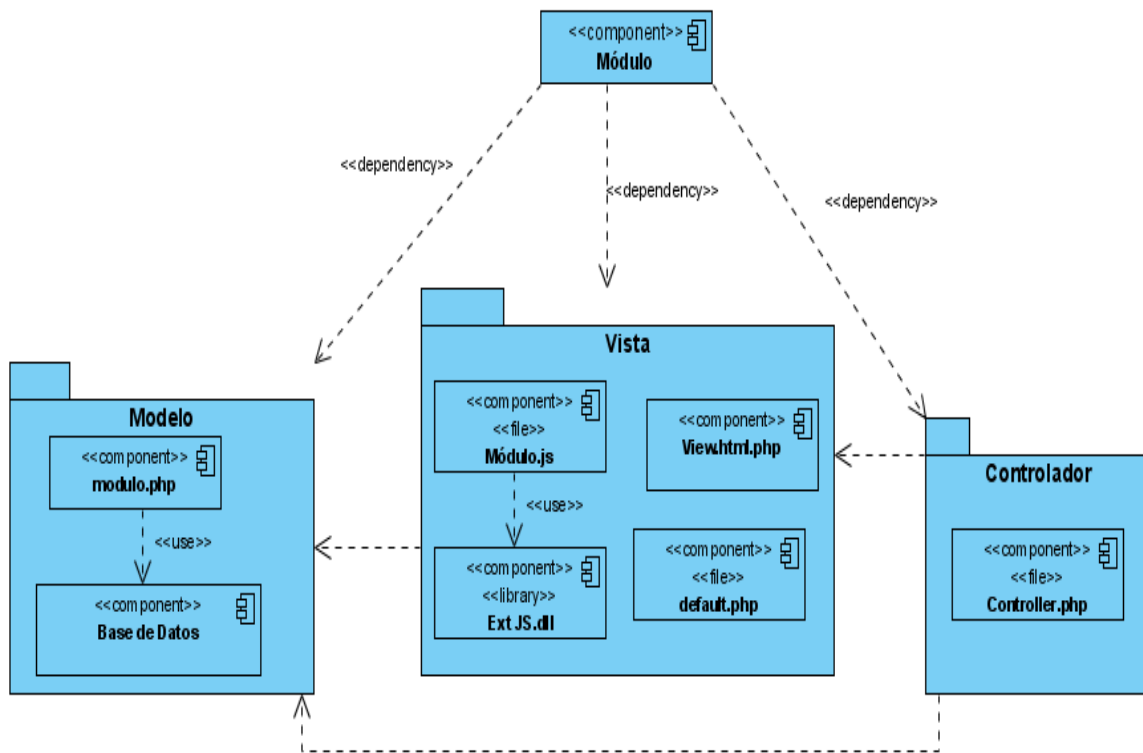
Entre las ventajas del estilo Modelo-Vista-Controlador están las siguientes:

- *Soporte de múltiples vistas:* dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación web pueden utilizar el mismo modelo de objetos mostrado de maneras diferentes.
- *Adaptación al cambio:* los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos

dispositivos. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Una desventaja que tiene este modelo es el costo de actualizaciones frecuentes: Si el modelo experimenta cambios frecuentes, por ejemplo, podría desbordar las vistas con una lluvia de requerimientos de actualización.

Para el desarrollo del sistema Coliseo Virtual, se hace uso de este patrón de diseño, dadas las ventajas anteriores que el mismo proporciona y las facilidades de que dota al CMS Joomla. En el siguiente diagrama de componentes se muestra la distribución física de su uso en los módulos existentes en la aplicación.



**Fig. 9** Diagrama de componentes. Módulos del sistema.

## 2.6- Descripción de las nuevas clases.

El CMS Joomla propone el uso de tres clases base: JView, JModel y JController que responden a su arquitectura Modelo-Vista-Controlador.

La clase JView dispone del código que provee de datos dinámicos a la página y los métodos que permiten gestionar la presentación al usuario. La clase JModel se encarga del tratamiento de la información contenida en la base de datos, representa la información del mundo real que el sistema debe procesar. La clase JController se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él.

<b>Nombre: JView</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$basePath \$ defaultModel \$ capa \$ capaExt \$ models \$ name \$ path	string string string string string string string
<b>Responsabilidades</b>	
<b>Nombre</b>	<b>Descripción</b>
JView construct ([ <b>\$config</b> = array()])	Este método permite crear una nueva instancia en las clases descendientes.
void addHelperPath (string/array <b>\$path</b> )	Este método permite adicionar un script de ayuda a la pila en orden LIFO.
void addPlatillaPath (string/array <b>\$path</b> )	Este método permite adicionar una platilla a la pila en orden LIFO.
<b>bool</b> assign ()	Este método permite asignar valores a una variable.
bool assignRef (string <b>\$key</b> , mixed <b>&amp;\$val</b> )	Este método permite asignar valor a una variable por referencia.
void display ([string <b>\$tpl</b> = null])	Este método ejecuta y muestra una platilla.
Mixed escape (mixed <b>\$var</b> )	Este método devuelve un valor de salida de una variable.
Mixed get (string <b>\$property</b> , [string <b>\$default</b> = null])	Este método devuelve un dato contenido en el modelo o una propiedad de la vista.
string getCapa ()	Este método devuelve el nombre de la capa.
mixed getModel ([string <b>\$name</b> = null])	Este método devuelve el nombre del modelo.
String getName ()	Este método devuelve el nombre de la vista.
boolean loadHelper ([ <b>\$hlp</b> = null], string <b>\$tpl</b> )	Este método carga un archivo de ayuda.

string loadPlatilla ([string \$tpl = null])	Este método carga una platilla.
void setEscape (mixed \$spec)	Este método cambia la función escape()
string setCapa ( \$capa, string \$platilla)	Este método cambia el nombre de la capa.
string setCapaExt (string \$value)	Este método permite el uso de una extensión diferente para la capa.
object setModel ( &\$model, [boolean \$default = false], object \$model)	Este método adiciona un modelo a la vista.
void addPath ( \$type, string/array \$path)	Este método adiciona un camino de búsqueda para las platillas y recursos.
void setPath (string \$type, string/array \$path)	Este método cambia el arreglo de caminos de búsqueda para platillas o recursos.

**Tabla 2** Clase Interfaz: JView

<b>Nombre: JModel</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$ bd \$ name \$ state	Object string string
<b>Responsabilidades</b>	
<b>Nombre</b>	<b>Descripción</b>
static void addTablePath (string/array \$path)	Este método adiciona una tabla al modelo.
JModel construct ([ \$config = array()])	Este método define el constructor de la clase.
array addIncludePath ([string \$path = "])	Este método adiciona un directorio de búsqueda.
Object getDBO ()	Este método devuelve el objeto de conexión de la base de datos.
mixed getInstance (string \$type, [string \$prefix = "], [array \$config = array()])	Este método devuelve una referencia del objeto del modelo.
String getName ()	Este método devuelve el nombre del modelo.
object getState ([string \$property = null])	Este método devuelve la variable de estado del modelo.
object getTable ([string \$name = "], [string \$prefix = 'Table'], [array \$options = array()])	Este método devuelve la instancia de una tabla.
Void setDBO ( &\$db, object \$db)	Este método cambia el objeto de conexión de la base de datos.

Mixed setState (string <b>\$property</b> , [mixed <b>\$value</b> = null])	Este método cambia la variable de estado del modelo.
Array getList (string <b>\$query</b> , [int <b>\$limitstart</b> = 0], [int <b>\$limit</b> = 0])	Este método devuelve la instancia de una lista
int getListCount (string <b>\$query</b> )	Este método devuelve el número de campos obtenidos de ejecutar una consulta.

**Tabla 3** Clase Entidad: JModel

<b>Nombre: JController</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
\$ acoSection	String
\$ acoSectionValue	string
\$ basePath	string
\$ doTask	string
\$ message	string
\$ messageType	string
\$ methods	array
\$ name	array
\$ path	array
\$ redirect	string
\$ task	string
\$ taskMap	array
<b>Responsabilidades:</b>	
<b>Nombre</b>	<b>Descripción:</b>
static void addModelPath (string/array <b>\$path</b> )	Este método adiciona a la pila un modelo en orden LIFO.
static void addViewPath (string/array <b>\$path</b> )	Este método adiciona una o más vistas a la pila del controlador.
JController construct ([array <b>\$config</b> = array()])	Este método implementa el constructor de la clase.
Boolean authorize (string <b>\$task</b> )	Este método chequea la autorización.
Void display ([string <b>\$cacheable</b> = false])	Este método
Mixed execute (string <b>\$task</b> )	Este método ejecuta una tarea.
object getModel ([string <b>\$name</b> = ""], [string <b>\$prefix</b> = ""], [array <b>\$config</b> = array()])	Este método devuelve una instancia del modelo.
string getName ()	Este método devuelve el nombre del controlador.
String getTask ()	Este método devuelve la instancia de la última tarea.
array getTasks ()	Este método devuelve la tarea que se está

	ejecutando en el controlador.
object getView ([string <b>\$name</b> = ""], [string <b>\$type</b> = ""], [string <b>\$prefix</b> = ""], [array <b>\$config</b> = array()])	Este método devuelve una referencia de la vista actual.
Boolean redirect ()	Este método redirecciona la búsqueda.
Void registerDefaultTask (string <b>\$method</b> )	Este método registra una tarea definida por defecto.
Void registerTask (string <b>\$task</b> , string <b>\$method</b> )	Este método registra una tarea.
Void setAccessControl (string <b>\$section</b> , [string <b>\$value</b> = null])	Este método cambia el nivel de control de acceso.
String setMessage (string <b>\$text</b> )	Este método cambia un mensaje interno.
void setRedirect (string <b>\$url</b> , [string <b>\$msg</b> = null], [string <b>\$type</b> = 'message'])	Este método cambia la URL para la búsqueda.
Void addPath (string <b>\$type</b> , string/array <b>\$path</b> )	Este método adiciona
Void setPath (string <b>\$type</b> , string/array <b>\$path</b> )	Este método cambia el arreglo de caminos de búsqueda para platillas y recursos.

**Tabla 4** Clase Controladora: JController

<b>Nombre: CompetenciaModelcompetencia extend JModel</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
No tiene	No tiene
<b>Responsabilidades</b>	
<b>Nombre</b>	<b>Descripción</b>
guardar_equipo(\$nombreequipo,\$integrante1, \$integrante2, \$integrante3, \$idcompetencia)	Este método guarda el equipo creado en la base de datos.
crear_competencia(\$rf1, \$rf2, \$time1, \$time2, \$nombre, \$cprob, \$pass1)	Este método crea una nueva competencia con los parámetros de entrada.
filtrar_usuarios(\$facultad, \$year, \$universidad)	Muestra un listado de los usuarios que están registrados en la base de datos que cumplen con los parámetros de entrada.
devolver_competencias_involucradas()	Devuelve las competencias en las que ha participado un usuario.

**Tabla 5** Clase Entidad: CcompetenciaModelcompetencia

<b>Nombre: EstilolibreModelestilolibre extend JModel</b>
--



<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
No tiene	No tiene
<b>Responsabilidades</b>	
<b>Nombre</b>	<b>Descripción</b>
cargar_problemas()	Este método muestra los problemas disponibles.
ranking()	Este método devuelve un ranking de los usuarios registrados de acuerdo a sus resultados obtenidos al participar en la modalidad de competencia: Estilo Libre.

**Tabla 6** Clase Entidad: EstilolibreModeloestilolibre

<b>Nombre: PonlineController extend JController</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
No tiene	No tiene
<b>Responsabilidades</b>	
<b>Nombre</b>	<b>Descripción</b>
GenerarPrueba()	Este método genera una prueba teniendo en cuenta los parámetros de entrada: año, materia, tema.
ranking()	Este método devuelve un ranking de los usuarios registrados según sus resultados en la modalidad de competencia prueba-en línea.
Guardar_resultado_preguntas()	Este método guarda el resultado obtenido por el usuario en una competencia.
subir_pregunta()	Este método visualiza las preguntas en la prueba generada por el usuario.
dar_year()	Este método muestra el listado de años para que el usuario seleccione al que pertenece.
dar_materia()	Este método muestra el listado de las materias para que el usuario seleccione una para generar su prueba.
dar_tema()	Este método muestra el listado de los temas para que el usuario seleccione uno para generar su prueba.

**Tabla 7** Clase Controladora: PonlineController

<b>Nombre: retosController extend JController</b>
---

<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
No tiene	No tiene
<b>Responsabilidades</b>	
<b>Nombre</b>	<b>Descripción</b>
retar_usuario()	Este método envía una solicitud de reto al usuario seleccionado a retar.
retos_pendientes()	Muestra los retos que aun no se han realizado y el usuario tiene pendientes.
aceptar_reto()	Este método muestra el problema asignado por el sistema una vez que el usuario acepta la solicitud de reto.
cargar_usuario()	Este método muestra un listado de los usuarios registrados en el sistema, para que se seleccione el usuario a retar.
cargar_invitacion()	Muestra al usuario el listado de sus invitaciones pendientes.
ranking()	Este método devuelve un ranking de los usuarios registrados de acuerdo a sus resultados obtenidos al participar en la modalidad de competencia: Estilo Libre.

**Tabla 8** Clase Controladora: retosController

## **Conclusiones**

Mediante la realización de este capítulo se obtuvo como resultado, el diagrama de clases persistentes para el diseño de la base de datos del sistema, teniendo como partida el diseño propuesto por el analista. Se obtuvieron los artefactos propios diseño e implementación. Se definen las características específicas de cada uno de los módulos que componen el sistema. Se realiza además la integración de los componentes Juez en Línea, UddelM y Remository para el proceso de calificación de los distintos estilos de competencia, servicio de mensajería y gestión de archivos respectivamente, sometiéndolos inicialmente a pruebas de unidad y luego a la integración con el sistema.

El uso de las clases bases del CMS Joomla permitió consolidar la arquitectura: Modelo-Vista-Controlador definida para la aplicación. Con la propuesta hasta aquí concebida se pretende garantizar el óptimo funcionamiento del sistema dando cumplimiento a los requisitos definidos por el analista.

## Capítulo3. Validación de la solución propuesta

En el presente capítulo se realiza una validación de la solución propuesta mediante diferentes tipos de pruebas. Incluye una descripción de los test de unidad necesaria para asegurar el correcto funcionamiento de los módulos de forma independiente, haciendo énfasis en su objetivo, alcance, tipo y detalles. Se realiza una evaluación de su ejecución y los resultados obtenidos, escribiendo casos de prueba para cada función de los diferentes módulos, tras el objetivo de comprobar sus funcionalidades de forma independiente e integrada al sistema posteriormente.

### 3.1- Búsqueda o diseño de los test que permitan validar la solución propuesta

Las pruebas son aplicadas para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Se distinguen los siguientes niveles de prueba:

- Prueba de desarrollador
- Prueba Independiente
- Prueba de unidad
- Prueba de Integración
- Prueba de Aceptación
- Prueba de sistema

Entre los tipos de prueba comúnmente usados para validar el software se encuentran las de: unidad, integración, validación y sistema (Juristo, 2006). Con el objetivo de detectar errores de código que posibiliten la mejora de su estructura y lograr mayor independencia entre el código y la implementación se decide desarrollar pruebas de unidad para validar el funcionamiento de los módulos del sistema. Se desarrollan casos de prueba para cada módulo, de forma independiente. Los objetivos de cada uno de los tipos de pruebas elegido se describen a continuación:

*Pruebas de unidad:* están dirigidas a los diferentes módulos que forman el sistema. Se usa la descripción del diseño detallado como guía y se prueban los caminos de control más importantes, con el fin de descubrir errores dentro del ámbito del módulo. Haciendo un uso intensivo de las técnicas de prueba de caja blanca (Juristo, 2006).

*Pruebas de integración:* se realizan con el objetivo de tomar los módulos probados en las pruebas de unidad y construir una estructura para el programa, que esté acorde con lo especificado en el diseño. Prevalciendo en este tipo de prueba las técnicas de caja negra (Juristo, 2006).

*Pruebas de aceptación:* son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo del producto, pues sería impresentable al cliente; sino que se realizan sobre el producto terminado e integrado o sobre una versión del producto o una iteración funcional pactada previamente con el cliente. Son muy importantes, ya que definen el paso a nuevas fases del proyecto como el despliegue y mantenimiento (Juristo, 2006).

Se decide utilizar las pruebas de unidad, mediante las técnicas de Caja Blanca para el análisis del funcionamiento interno de los módulos de forma interna, las pruebas de integración mediante las técnicas de Caja Negra para detallar la interrelación entre los módulos y las interfaces. Finalmente una vez listo el sistema se realizara pruebas de aceptación, chequeando el cumplimiento de los requisitos definidos para el sistema.

### **3.2-Descripción de los test de unidades.**

Para la descripción a los test de unidades se tendrá en cuenta el objetivo del test, alcance y detalle del mismo. Se desarrollan casos de prueba para el análisis del camino de control perteneciente a cada funcionalidad importante de los módulos del sistema y matrices de prueba para el análisis de los diferentes escenarios de casos de uso.

Para cada módulo del sistema se desarrollaron más de 20 casos de prueba con diferentes valores de entrada para analizar la respuesta del sistema ante los mismos. Estuvieron orientados a comprobar la completitud del código y la respuesta del sistema a los datos insertados por el usuario. A continuación se muestra un caso de prueba por módulo y la matriz de casos de prueba correspondiente al análisis de sus escenarios. Se realiza además un resumen de las anomalías encontradas y corregidas en el período de prueba.

### 3.2.1-Módulo Prueba En línea

#### Caso de Prueba # 1

Este caso de prueba se hace para analizar el cumplimiento del requisito “Generar pruebas en línea”. El usuario deberá seleccionar el año(s), materia(s) y tema(s), de los cuales desea generar la prueba. Como resultado se generará una prueba que será mostrada al usuario.

Caso de Uso	Gestionar Prueba en Línea		
Caso de prueba	Generar_Prueba()		
Condiciones	Para que este caso de prueba se cumpla exitosamente, el usuario debe seleccionar el año(s), materia(s) y tema(s), en los campos correspondientes.		
Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba
<b>Años:</b> 1,4 <b>Materias:</b> IP, IA <b>Temas:</b> A, B	El usuario no selecciona los datos requeridos.	Con esta prueba se espera que el sistema genere una prueba de 45 preguntas que se correspondan con los parámetros entrados por el usuario.	El resultado de la prueba es el esperado, la prueba es generada inmediatamente.  Muestra mensaje “No más preguntas en esta prueba”  Muestra mensaje “Debe seleccionar los criterios.”

**Tabla 9** Caso de Prueba 1

Id de escenario	Escenario	Años Académicos	Materias	Temas	Respuesta del sistema
<b>EC1</b>	Prueba generada con éxito	1,5	Introducción a la Programación  Inteligencia Artificial	Arreglos y algoritmos básicos.  Programación Lógica.	El resultado es el esperado, la prueba es generada.
<b>EC2</b>	Faltan datos obligatorios	3	""	""	El resultado es el esperado, el sistema muestra el mensaje “Debe entrar los criterios”
<b>EC3</b>	Faltan datos obligatorios	4	Inteligencia Artificial	""	El resultado es el esperado, el sistema muestra el mensaje “Debe entrar los criterios”
<b>EC4</b>	Faltan datos	""	""	""	El resultado es el esperado,

	obligatorios				el sistema muestra el mensaje "Debe entrar los criterios"
<b>EC5</b>	No hay preguntas para esa prueba	1	Introducción a la Programación	Arreglos y algoritmos básicos	El resultado es el esperado, el sistema muestra el mensaje: "No hay más preguntas para esta prueba" y ofrece la puntuación obtenida.

**Tabla 10** Matriz de casos de Prueba 1

Las anomalías fundamentales encontradas en el período de prueba de este módulo fueron:

- Repeticiones en las preguntas de la prueba en línea. Evidenciando anomalías en el algoritmo de calificación.
- El reloj para medir el tiempo de ejecución de las preguntas sufría un proceso de aceleración. Evidenciando anomalías en el algoritmo para reiniciar y detener el reloj.
- El contenedor donde se muestran las preguntas crecía a medida que el usuario pulsaba el botón siguiente para ver la próxima pregunta.

Todos los errores encontrados fueron corregidos, para un 100% de efectividad de los casos de prueba realizados.

### 3.2.2- Módulo Competencia.

#### *Caso de Prueba # 2*

Este caso de prueba se hace para validar la funcionalidad que el método para verificar el cumplimiento del requisito gestionar las competencias. La competencia será creada dados los datos entrados por el usuario y guardada en la base de datos del sistema de forma automática.

<b>Caso de Uso</b>	<b>Gestionar Competencia</b>
<b>Caso de Prueba</b>	<b>Crear_Competencia()</b>
<b>Condiciones</b>	Para que este caso de prueba se cumpla exitosamente, el usuario debe entrar los datos referentes a la prueba que desea crear: nombre, cantidad de problemas, fecha de inicio, hora inicio, entre otros

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba
Nombre, 5, 19/4/09, 3:00, clave, 20/4/09, 5:00.	5, 19/4/09, 3:00, clave, 20/4/09, 5:00.	Con esta prueba se espera que el sistema cree una competencia, dados los datos de entrada.	El resultado de la prueba es el esperado, la competencia es creada y guardada en la base de datos.  El sistema muestra el mensaje: "Campo requerido"

**Tabla 11** Caso de Prueba 2

Id	Escenario	Nombre	Cant. de problemas	contraseña	Repetir contraseña	Hora inicio Hora fin	Fecha inicio Fecha fin	Respuesta del sistema
EC1	Competencia creada con éxito	Comp1	5	Cl2v4	Cl2v4	2:00/ 3:00	20/4/09 21/4/09	Resultado esperado la competencias es creada
EC2	Faltan datos de entrada	Comp2	2	Br67	Br67	2:00/ 3:00	""	El sistema muestra el mensaje: "Debe llenar los campos requeridos"
EC3	Faltan datos de entrada	Comp2	2	Br67	Br67	""	20/4/09 21/4/09	El sistema muestra el mensaje: "Debe llenar los campos requeridos"
EC4	Faltan datos de entrada	Comp2	2	Br67	""	2:00/ 3:00	20/4/09 21/4/09	El sistema muestra el mensaje: "Debe llenar los campos requeridos"
EC5	Faltan datos de entrada	Comp2	2	""	""	2:00/ 3:00	20/4/09 21/4/09	El sistema muestra el mensaje: "Debe llenar los campos requeridos"
EC6	Faltan datos de entrada	Comp2	""	Br67	Br67	2:00/ 3:00	20/4/09 21/4/09	El sistema muestra el mensaje: "Debe llenar los campos requeridos"
EC7	Faltan datos de entrada	""	2	Br67	""	2:00/ 3:00	20/4/09 21/4/09	El sistema muestra el mensaje: "Debe llenar los campos requeridos"



<b>EC8</b>	Las contraseñas no coinciden	Comp4	5	123	342	2:00/ 3:00	2:00/ 3:30	Resultado esperado el sistema muestra el mensaje: "Las contraseñas no coinciden"
------------	------------------------------	-------	---	-----	-----	---------------	---------------	--

**Tabla 12** Matriz de casos de Prueba 2

No se encontraron anomalías en el anterior módulo, durante el periodo de prueba del sistema. El mismo funciona de acuerdo a las descripciones anteriores. Posee una correcta respuesta del sistema a todas las entradas posibles del usuario.

### 3.2.3-Módulo de Retos

#### *Caso de Prueba # 3*

Este caso de prueba se hace para demostrar que el método que posee la responsabilidad de crear un reto funcione correctamente. El usuario seleccionará del listado de usuarios registrados el usuario a retar. El sistema enviará una solicitud de reto que adicionará en las invitaciones de retos del usuario retado y en el listado de restos pendientes con un problema asignado automáticamente.

<b>Caso de Uso</b>	<b>Gestionar Reto</b>		
<b>Caso de Prueba</b>	<b>Crear_Reto()</b>		
<b>Condiciones</b>	Para que este caso de prueba se cumpla exitosamente, se debe seleccionar el usuario a quien se desea enviar una solicitud de reto.		
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado Esperado</b>	<b>Resultado de la Prueba</b>
iviltres	No selecciona usuario.	Con esta prueba se espera que el sistema envíe una solicitud de reto al usuario seleccionado	El resultado de la prueba es el esperado, el sistema muestra el mensaje: "La solicitud de reto ha sido enviada".

**Tabla 13** Caso de Prueba 3

<b>ID escenario</b>	<b>Escenario</b>	<b>usuario</b>	<b>Respuesta del sistema</b>
<b>EC1</b>	Reto creado con éxito.	iviltres	El resultado es el esperado el sistema envía una solicitud.
<b>EC2</b>	Faltan datos de entrada	""	El resultado es el esperado el sistema envía un mensaje "Debe seleccionar un usuario."

<b>Id escenario</b>	<b>Escenario</b>	<b>Invitación de Reto</b>	<b>Respuesta del sistema</b>
<b>EC1</b>	Reto creado	1	Se muestra el problema asignado por el sistema al reto creado.
<b>EC2</b>	Faltan datos de entrada	""	El resultado es el esperado el sistema envía un mensaje "Debe seleccionar una invitación."

**Tabla 14** Matriz de Caso de Prueba 3

El módulo anterior funciona de un modo simple. Posee una correcta validación de las respuestas del sistema al usuario. La anomalía detectada en el periodo de prueba fue:

- Ausencia del usuario ganador del reto debido a la ausencia de conexión con el Juez en Línea, para comparar los resultados de ambos usuarios en el desarrollo del problema asignado.

En este momento el módulo funciona completamente, los puntos alcanzados por el usuario son sumados a su puntuación acumulada en el ranking de este estilo de competencia y en el ranking general.

### 3.2.4-Componente Enviar Solución

#### *Caso de Prueba # 4*

Este caso de prueba se realiza para comprobar el funcionamiento del componente enviar solución. El usuario ingresará el número identificador del problema y seleccionará el lenguaje de programación utilizado: C#, Java, C++, Python o Pascal del listado de lenguajes disponibles. Se podrá subir un archivo o escribir el código directamente en un campo disponible para editar.

<b>Caso de Uso</b>	<b>Gestionar Competencia</b>		
<b>Caso de Prueba</b>	<b>Enviar Solución()</b>		
<b>Condiciones</b>	Para que este caso de prueba se cumpla exitosamente, el usuario debe poner el número identificador del problema solucionado y el lenguaje de programación que utilizó		
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado Esperado</b>	<b>Resultado de la Prueba</b>
5, C#	No entra los datos	Con esta prueba se espera que el sistema	El resultado de la prueba es el esperado, la solución es

	requeridos.	envíe la solución emitida por el usuario al juez en línea y este devuelva el resultado de su calificación.	<p>enviada correctamente. El sistema muestra mensaje "Correcto."</p> <p>El sistema muestra mensaje "Debe llenar los campos requeridos."</p>
--	-------------	--	---

**Tabla 15.** Caso de Prueba 4

ID de escenario	Escenario	Id problema	Lenguaje	Respuesta del sistema.
<b>EC1</b>	Correcto envió de la solución	5	C#	La solución es enviada
<b>EC2</b>	Faltan datos de entrada	""	Python	El sistema muestra un mensaje: "Debe llenar los campos"
<b>EC3</b>	Faltan datos de entrada	5	""	El sistema muestra un mensaje: "Debe llenar los campos"

**Tabla 16.** Matriz de Caso de Prueba 4

La integración de este componente a los demás módulos del sistema funciona correctamente, al igual que su validación a las entradas del usuario. La conexión con el Juez En línea que proporciona este módulo posee las funcionalidades requeridas.

### 3.2.5-Módulo Estilo Libre

#### *Caso de Prueba # 5*

Este caso de prueba se hace con el objetivo de comprobar el funcionamiento del módulo Estilo Libre. El usuario que desee participar en este estilo de competencia, deberá seleccionar un problema de los publicados, en respuesta el sistema mostrará la descripción del mismo y permitirá enviar su solución.

<b>Caso de Uso</b>	<b>Gestionar Estilo Libre</b>		
<b>Caso de prueba</b>	<b>Participar Estilo Libre.</b>		
<b>Condiciones</b>	Para que este caso de prueba se cumpla exitosamente, el usuario debe seleccionar un problema publicado en la aplicación.		
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado Esperado</b>	<b>Resultado de la Prueba</b>

Selecciona un problema.	No selecciona problema alguno.	Con esta prueba se espera que el sistema muestre la descripción del problema seleccionado por el usuario.	El resultado de la prueba es el esperado, se le ofrece la descripción del problema seleccionado al usuario. El sistema muestra mensaje "Debe seleccionar un problema."
-------------------------	--------------------------------	---	---

**Tabla 17.** Caso de Prueba 5

<b>Id. del escenario</b>	<b>Escenario</b>	<b>ID problema</b>	<b>Respuesta del sistema.</b>
<b>EC1</b>	Correcta selección del problema	5	La descripción del problema es mostrada.
<b>EC2</b>	Problema no seleccionado.	""	No se muestra la descripción.

**Tabla 18.** Matriz de Caso de Prueba 5

En el módulo anterior no se le informaba al usuario como ver la descripción de los problemas disponibles. La anterior anomalía fue corregida con el mensaje: "Seleccione un problema para ver su descripción. "

### 3.2.6-Módulo Test

#### *Caso de Prueba # 6*

Este caso de prueba se hace con el objetivo de comprobar el funcionamiento del módulo Test. El usuario que desee participar en este estilo de competencia, deberá seleccionar un test de los publicados, en respuesta el sistema mostrará la descripción del mismo y permitirá enviar su solución.

<b>Caso de Uso</b>	<b>Gestionar Test</b>		
<b>Caso de prueba</b>	<b>Resolver test.</b>		
<b>Condiciones</b>	Para que este caso de prueba se cumpla exitosamente, el usuario debe seleccionar un test publicado en la aplicación.		
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado Esperado</b>	<b>Resultado de la Prueba</b>
Selecciona un test	No selecciona problema alguno.	Con esta prueba se espera que el sistema muestre la descripción del problema seleccionado	El resultado de la prueba es el esperado, se le ofrece la descripción del test seleccionado al usuario."

		por el usuario.	El sistema muestra mensaje "Debe seleccionar un test."
--	--	-----------------	--

**Tabla 19.** Caso de Prueba 6

<b>Id del escenario</b>	<b>Escenario</b>	<b>Id test</b>	<b>Respuesta del sistema.</b>
<b>EC1</b>	Correcta selección del test	5	Se muestran las preguntas asignadas al test
<b>EC2</b>	Test no seleccionado.	""	Se muestra el mensaje: "Debe seleccionar un test"

**Tabla 20.** Matriz de Caso de Prueba 6

El error detectado en el módulo Test fue:

- No crea los componentes definidos para las preguntas del test.

La anterior anomalía fue corregida. El módulo posee una adecuada validación de las respuestas del sistema y permite realizar las funcionalidades definidas por el analista para el Módulo Test.

### 3.2.7-Módulo Concurso

#### *Caso de Prueba # 7*

Este caso de prueba se hace con el objetivo de comprobar el funcionamiento del módulo Concurso. El usuario que desee participar en este estilo de competencia, deberá seleccionar un concurso de los publicados, en respuesta el sistema mostrará la descripción del mismo y permitirá enviar su solución.

<b>Caso de Uso</b>	<b>Gestionar Concurso</b>		
<b>Caso de Prueba</b>	<b>Participar concurso</b>		
<b>Condiciones</b>	Para que este caso de prueba se cumpla exitosamente, el usuario debe seleccionar un concurso publicado en la aplicación y entrar la contraseña de acceso al mismo, puesta por el administrador.		
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado Esperado</b>	<b>Resultado de la Prueba</b>

Selecciona un test Entra la contraseña	No selecciona problema alguno. La contraseña es incorrecta	Con esta prueba se espera que el sistema muestre la descripción del problema seleccionado por el usuario luego de probar que la contraseña de acceso es valida	El resultado de la prueba es el esperado, se le ofrece la descripción del test seleccionado al usuario.” El sistema muestra mensaje “Debe seleccionar un test.” El sistema muestra mensaje “Contraseña incorrecta.”
---	---	---	--

**Tabla 21.** Caso de Prueba 7

ID del escenario	Escenario	Id test	Respuesta del sistema.
EC1	Correcta selección del concurso	3	Muestra ventana con solicitud de contraseña.
EC2	Concurso no seleccionado.	”	Se muestra el mensaje: “Debe seleccionar un concurso”
ID del escenario	Escenario	Contraseña	Respuesta del sistema.
EC3	Contraseña Incorrecta	F5dt3r	Se muestra el mensaje: “Contraseña Incorrecta”
EC4	Contraseña correcta	coliseo	Muestra el problema del concurso.

**Tabla 22.** Matriz de Caso de Prueba 7

No se encontraron errores en el presente módulo. Posee un funcionamiento correcto según las funcionalidades definidas por el analista.

### 3.3-Evaluación de la ejecución de los test y de los resultados obtenidos.

Una vez realizadas las pruebas de unidad para comprobar el correcto funcionamiento de los módulos del sistema se procede a realizar pruebas de integración de los mismos para valorar su funcionamiento integrado como parte del sistema. Las pruebas de integración involucran varios módulos pudiendo ser:

- *Estructurales:* como las de caja blanca pero analizando llamadas entre módulos

- *Funcionales*: como las de caja negra pero comprobando funcionalidades conjuntas.

Las pruebas finales consideran todo el sistema cubriendo plenamente la especificación de requisitos del usuario (Juristo, 2006).

Las pruebas estructurales estuvieron centradas en evaluar las llamadas entre las interfaces de los módulos: Competencia, Estilo Libre, Concurso y Reto con el componente Enviar Solución que se encarga de enviar la solución al juez y devolver la calificación del envío emitido por el usuario. Las funcionalidades conjuntas fueron evaluadas mediante la integración ascendente de los diferentes módulos al sistema procediendo a la evaluación del resultado en las categorías bien, regular o mal (B, R, M).

<b>Módulos a Integrar</b>	<b>Llamadas entre módulos.</b>	<b>Funcionalidades conjuntas</b>	<b>Eval. de la interacción entre sus interfaces</b>
Competencia y Enviar Solución	x	x	B
Estilo Libre y Enviar Solución	x	x	B
Concurso y Enviar Solución	x	x	B
Retos y Enviar Solución	x	x	B

**Tabla 23.** Integración de los módulos.

Para considerar las funcionalidades del sistema cubriendo las especificaciones de los requisitos definidos por el analista se realizó una Lista de Chequeo del cumplimiento de los mismos. Asignándoles un peso de crítico si determinan directamente la funcionalidad del sistema y no crítico, si están relacionados con algún flujo alternativo o no determinan las funcionalidades del módulo al que pertenecen. Los indicadores se evalúan de '0' en caso de mal y 1 en caso de que no presente errores. La cantidad de elementos afectados indica el número de errores encontrados. Los comentarios están referidos a los señalamientos o sugerencias que se considera pertinente realizar.

**Requisitos Funcionales del sistema.**

<b>Peso</b>	<b>Requisito Funcional</b>	<b>Eval.</b>	<b>Elementos afectados</b>	<b>Comentarios</b>
Crítico	Registrar Usuario.	1	0	Dado por el componente acceso, disponible en la portada.
Crítico	Autenticar Usuario.	1	0	Dado por el componente acceso, disponible en la portada.
No Crítico	Recuperar Contraseña.	1	0	Dado por el componente acceso, disponible en la portada.
Crítico	Modificar Datos Usuario.	1	0	Dado por el componente acceso, disponible en la portada.
Crítico	Gestionar Usuarios.	1	0	Administración del sistema
Crítico	Gestionar Estilo Libre.	1	0	Administración del sistema
Crítico	Gestionar Pruebas En línea.	1	0	Administración del sistema
Crítico	Gestionar Test.	1	0	Administración del sistema
Crítico	Gestionar Retos.	1	0	Administración del sistema
Crítico	Gestionar Crear Competencia.	1	0	Administración del sistema
Crítico	Gestionar Concurso.	1	0	Administración del sistema
No Crítico	Gestionar Noticias.	1	0	Administración del sistema
Crítico	Gestionar Cursos.	1	0	Administración del sistema
No Crítico	Gestionar Recursos.	1	0	Administración del sistema
No Crítico	Gestionar Enlace.	1	0	Administración del sistema
Crítico	Gestionar Equipo.	1	0	Administración del sistema
Crítico	Resolver Estilos Libres.	1	0	Módulo Estilo Libre
Crítico	Resolver Pruebas en Línea.	1	0	Módulo Pruebas En línea
Crítico	Resolver Test.	1	0	Módulo test
Crítico	Crear Reto.	1	0	Módulo Retos
Crítico	Participar Reto.	1	0	Módulo Retos



Crítico	Crear Competencia.	1	0	Módulo Competencia
No Crítico	Modificar Datos Competencia.	1	0	Módulo Competencia
Crítico	Participar Competencia.	1	0	Módulo Competencia
Crítico	Enviar Solución.	1	0	Conexión con el Juez en Línea
Crítico	Participar Concurso Copa Pascal.	1	0	Módulo Concurso
No Crítico	Descargar Problemas Concurso.	1	0	Enlace Recursos. menú Sistema
No Crítico	Contactar Juez.	1	0	Módulo Concurso
No Crítico	Mostrar Información	1	0	Menú Sistema

**Tabla 24.** Lista de Chequeo

A través de los resultados anteriormente expuestos puede concluirse que el sistema posee una completitud de un 100% de funcionalidad teniendo en cuenta el resultado de 29 requisitos evaluados sin error, de un total de 29 definidos por el analista. Los casos de prueba realizados, según las anomalías o errores existentes con relación a los corregidos son satisfactorios y la integración de los módulos del sistema está evaluada de bien a un 100%.

### 3.3.1-Prueba de Aceptación

Para llevar a cabo las pruebas de aceptación se realizó un análisis de las *no conformidades* detectadas hasta el momento: que no es más que problemas detectados en un artefacto (casos de usos, sistema, etc.) según:

- Error con respecto a lo definido en artefactos anteriores y/o en lo pactado con el cliente.
- No concordancia con Normas internacionales que deben ser cumplidas por el artefacto.
- Insatisfacción del cliente con el resultado final de un Elemento de Configuración según lo pactado con anterioridad en el proyecto.

El grado de impacto posible en las no conformidades encontradas se encuentra en el rango de 1 a 3. Se califica de 3 (Alta), la no conformidad referida a los errores que impidan llegar al final del flujo básico o errores en la interpretación del proceso de negocio que impidan el correcto funcionamiento de la aplicación. Se asigna el valor de 2 (Media) a los errores relacionados con la completitud de algún flujo alterno y 1 (Baja) a los errores de validación.

No conformidad	Módulo	Importancia	Descripción	Observación
Ofrecer un Enviar Solución Local, en cada módulo del sistema.	Competencia Concurso	1	Baja	Mayor facilidad de uso
Realizar paginado	Administración	1	Baja	Mayor facilidad de uso.

**Tabla 25.** Análisis de No conformidades.

Resulta oportuno señalar que el componente Enviar Solución está disponible en el sistema, la no conformidad relacionada a este tema, se refiere a la comodidad de poseer un componente de este tipo para cada estilo de competencias. Con lo que se evitaría tener que redireccionar a otro componente para que el usuario realice el envío y sería más fácil tener el control del estilo de competencia donde se emite la respuesta.

Las no conformidades encontradas están referidas a funcionalidades que no se incluyen dentro de las definidas para el sistema, pero brindarán facilidad de uso, tanto a usuarios como administradores.

### 3.4-Resultados de la presentación en eventos

El sistema web “Coliseo Virtual”, fue presentado en el reconocido evento “Jornada Científica Estudiantil” en su edición del 2009, a los niveles de Facultad y Universidad. Obteniendo el resultado de relevante en ambos casos.

Se destacó por parte de los jurados, de ambos certámenes, su calidad y aporte educativo. Además de valorar de muy interesantes, características del sistema como: la cantidad de estilos de competencias disponibles, la gestión por parte de los usuarios de sus propias competencias, el generador de pruebas configurables en el módulo:

“Prueba en Línea”, y la conexión al juez en línea que califica los diferentes envíos realizados por los usuarios.

Se contribuye con lo anterior, a la validación de la calidad del sistema en cuestión, que arrojó el máximo resultado en ambas lides. Concluyéndose que tiene una buena aceptación, en general. Estos resultados están respaldados por documentos que acreditan su prueba de veracidad.

## **Conclusiones**

A través del presente capítulo se ha hecho un análisis del funcionamiento del sistema, teniendo como partida los tipos de prueba: unidad e integración. Se desarrollaron casos de pruebas para probar el curso principal de las funcionalidades de los módulos y matrices de casos de prueba para el análisis de los diferentes escenario posibles de los caso de uso. Los resultados de los casos de prueba coinciden con la respuesta esperada por el sistema.

El proceso de integración ascendente de los módulos al sistema, evidenció la correcta relación entre sus interfaces e integración entre los mismos. La Lista de Chequeo presentada muestra el grado de cumplimiento del sistema con los requisitos funcionales definidos por el analista con un 100% de completitud.

Las no conformidades existentes no poseen importancia alta y la efectividad de los casos de prueba realizados y la Lista de chequeo, teniendo en cuenta las Anomalías encontradas y corregidas es superior a un 90 %.

## Conclusiones generales

Al culminar el desarrollo del presente trabajo se concluye que:

- Se realizó el diseño de la base de datos y de todos los módulos previstos según las funcionalidades definidas: Test, Retos, Competencias, Concurso, Prueba En línea, Estilo Libre y Gestión del Sistema.
- Se generaron los artefactos propios del diseño e implementación: diagrama de clases del diseño, diagrama de componentes y diagrama de despliegue.
- Se realizaron pruebas de unidad e integración para comprobar el correcto acoplamiento y funcionalidad de los módulos.
- Se confecciono la documentación requerida: una Ayuda y un Manual de Usuario.
- Se desarrolló el sistema informático “Coliseo Virtual” a partir de los requisitos definidos por el analista en la investigación precedente “Análisis del Sistema Coliseo Virtual”, el mismo podrá ser usado como una herramienta didáctica por parte de los estudiantes o de apoyo al desarrollo de habilidades en las técnicas de algoritmo, trabajo en equipo, agilidad mental, entre otras, en el proceso de enseñanza - aprendizaje.

## Recomendaciones

- Implementar el componente del “Juez en Línea” sobre plataforma GNU/Linux, para tener un producto de código libre en su totalidad.
- Perfeccionar la interfaz de la portada del sistema, haciendo partícipes de su desarrollo a diseñadores y especialistas en el tema.
- Añadir nuevos estilos de competencias.
- Agregar al sistema las facilidades de: un componente “Enviar Solución” para cada módulo, permitiendo conocer con mayor facilidad a que estilo de competencia pertenece el envío realizado, hacer más fácil y rápido el acceso a la opción y realizar paginado para los componentes de administración del sistema para mayor facilidad de uso.

## Referencias bibliográficas

- Álvarez Miguel Ángel. (2009) Informe Zend Studio. Disponible en <http://www.desarrolloweb.com>]
- ANDRAM. CMS gratuitos/de código abierto.(2008).[Disponible en: <http://andram.nireblog.com/cat/cms>]
- Aptana Studio 1.0. (2009) [Disponible en <http://www.genbeta.com>]
- Beck Kent. (1999). “Extreme Programming Explained.” First Edition, ISBN: 0201616416
- Booch, Jacobson, Rumbaugh. (2000). El Proceso de Desarrollo Unificado del Software. Pearson Educacion.S.A. ISBN: 84-7829-036-2
- Booch, Rumbaugh, Jacobson. (2007) “The UML User Guide”. Addison Wesley. ISBN 0-201-57168-4
- Casares Claudio. (2004). Introducción a SQL. [Disponible en <http://www.maestrosdelweb.com>]
- Chaves Duran Grover Marcelo.(2007). Paradigma de la programación funcional y su aplicación en la inteligencia artificial
- Cuerda García Xavier y Minguillón Alfonso Julia.( 2004) Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto.[Disponible en <http://mosaic.uoc.edu/articulos/cms1204.html>]
- Dorado Andrés. (2009).Patrones de Diseño
- Eguíluz Pérez Javier. (2009) Introducción a JavaScript. [Disponible en <http://www.librosweb.es>]

- Fokker Jeroen. (1996). Programación Funcional. [Disponible en <http://people.cs.uu.nl/jeroen/courses/fp-sp.pdf>]
- Gallego Vázquez José Antonio. (2003.) Desarrollo Web con PHP y MySQL. Anaya Multimedia, ISBN 8441515255
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. (2006). Patrones de Diseño: Elementos de Software Orientado a Objetos Reutilizables. Addison-Wesley.
- Giordani Leonardo. (2009) Programación concurrente - Principios e introducción a procesos. [Disponible en <http://es.tldp.org/LinuxFocus>]
- Gómez Arguello Wilson Javier. (2007). "Metodología de Desarrollo de Software, un enfoque práctico y global". Orange Technologies. Versión 1.0.11 Beta (20071020)
- HandBook Drupal. (2005). Developing for Drupal.
- Hansen, Gary W. - Hansen, James V. (1999). Diseño y administración de bases de datos. PRENTICE HALL ISBN: 84-8322-002-4, 1997
- INEI (Instituto Nacional de Estadística e Informática). (1999) "Herramientas Case".
- Jurista Natalia, Moreno Ana M., Vegas Sira (2006). Técnicas de evaluación del software. Versión 12.
- LeBlanc Joseph. (2007). Learning Joomla Extension Development. ISBN: 1847191304
- Mellado Juan. (2005). Drupal: Creación de Módulos.
- Mendoza, M. A. (2004). "Metodologías de desarrollo de Software". [Disponible en: [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_softwa\\_re\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_softwa_re_07062004.html)]

- Microsoft y Rational. (1998). A White Paper on the Benefits of Integrating Microsoft Solutions Framework and The Rational Process. Disponible en [<http://www.rational.com/uml/papers.>]
- Miranda Perea Favio E.(2007). Lenguajes de programación y sus paradigmas. Facultad de Ciencias UNAM.
- Mosse Francisco. ( 2009). “Programación Estructurada”.[Disponible en <http://www.alipso.com/monografias> ]
- Pérez. Maritza. (2009). “Programación modular: procedimientos y funciones”.
- Pérez Valdés Damián. (2008). “Editores Web que facilitan tu trabajo.” [Disponible en <http://www.maestrosdelweb.com>]
- Pekin, A., U.d.Jueces En línea. (2009). [Disponible en: <http://acm.pku.edu.cn/JudgeEn línea.>]
- Pressman Roger. (2005).Ingeniería del Software: Un Enfoque Práctico. ISBN: 9701054733.
- Quintero Reina. (2000).“Visión General de la Programación Orientada a Aspectos”. [Disponible en <http://www.lsi.us.es/docs/informes/aopv3.pdf>]
- Ramos J, Ramos A, Montero F. (2008), SGBD. Sistemas gestores de bases de Datos.
- Reina Quintero Antonia M<sup>a</sup>.( 2000). Visión General de la Programación Orientada a Aspectos.
- Rossel Gerardo. (2004). Programación lógica.[Disponible en [http://www.amzi.com/articles/code07\\_whitepaper.pdf](http://www.amzi.com/articles/code07_whitepaper.pdf)]
- RM Alejandro José. (2008). Gestión de Contenidos Drupal Sistema Modular y Configurable



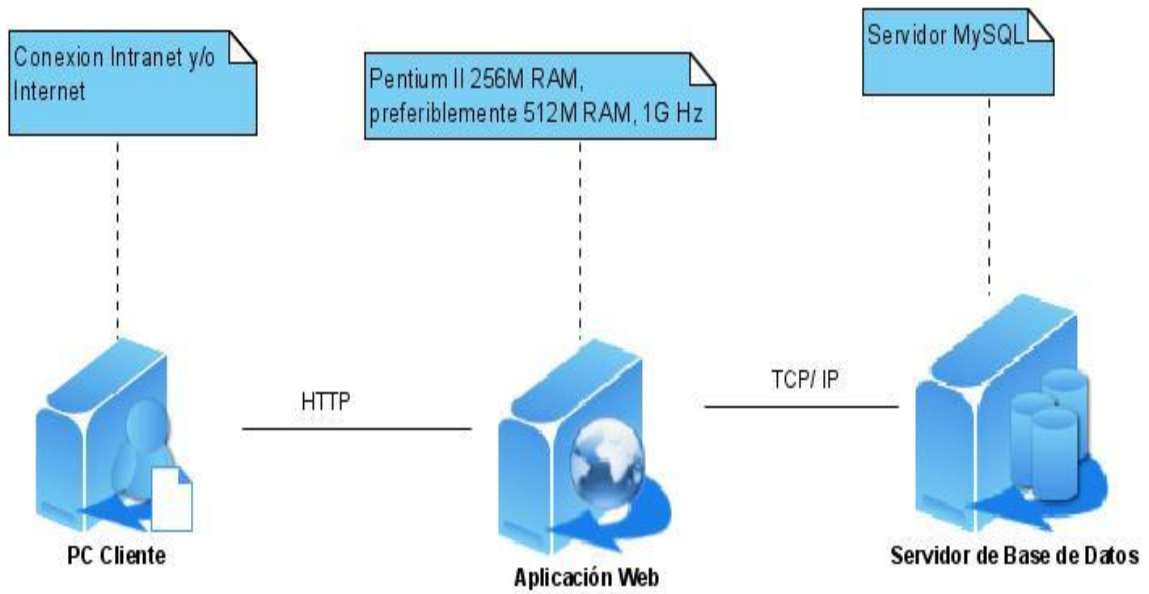
- Saratov, U. (2007). Concurso Habilidades Programación. [Disponible en: <http://acm.sgu.ru>.]
- Southwell, Michael. (2005). Pro PHP Security.
- Sun Microsoft Inc. ( 2009). MySQL 5.0 Reference Manual.
- Trevejo Alonso Juan Antonio. (2009). Joomla! para Principiantes. Aprendiendo a crear y mantener sitios web. [Disponible en <http://e-book.joomlaos.net/>]
- UCI. (2008). Cátedra de Programación Avanzada, [Disponible en <http://cpav.uci.cu>]
- Ural, U.E. Concurso Habilidades Programación. [Disponible en: <http://acm.timus.ru>.]
- Welling Luck, Thomson Laura. (2005). Desarrollo Web con PHP y MySQL. Anaya Multimedia. ISBN: 978-84-415-1818-6
- White Steve, Wallace Andy. (2006). Joomla! User Manual.
- Winblad. Ann L. (1993). "Software orientado a objetos".

## Bibliografía consultada

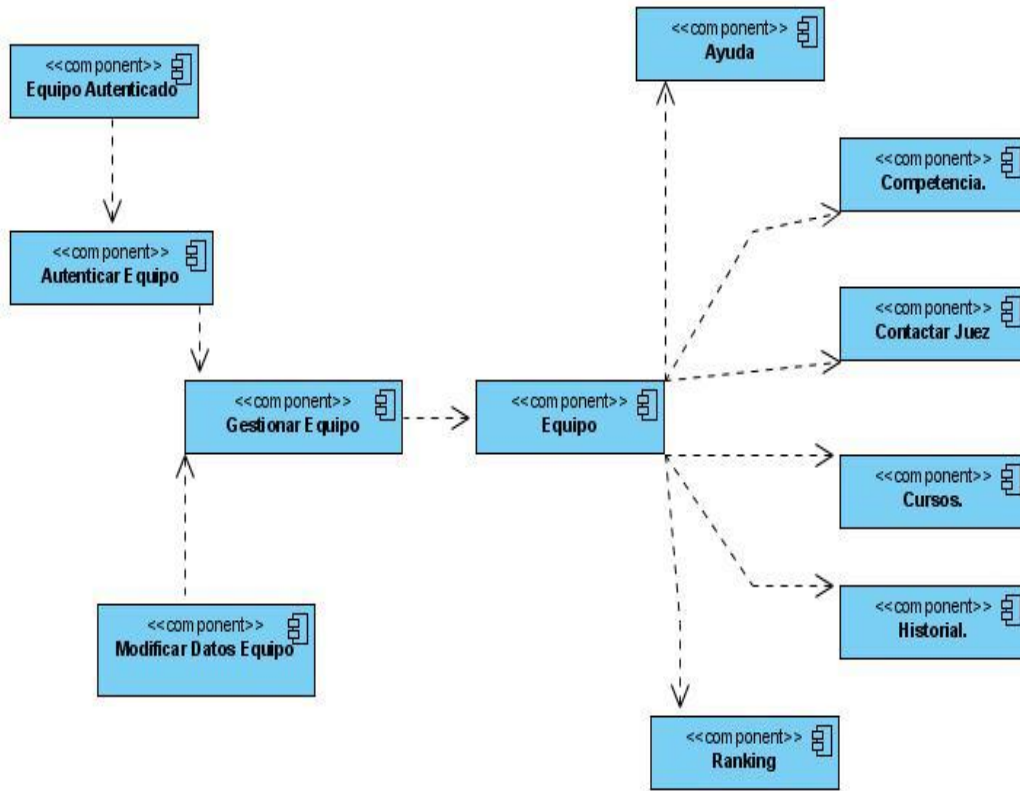
- Álvarez Ángel Miguel, Informe Zend Studio. Director de DesarrolloWeb.com  
[Disponible en <http://www.desarrolloweb.com>]
- Boggs Wendy y Boggs Michael (2002). UML with Rational Rose 2002.
- Booch, Jacobson, Rumbaugh. (2000). El Proceso de Desarrollo Unificado del Software.  
[ Disponible en <http://bibliodoc.uci.cu/pdf/reg00060.pdf> ]
- Gallego Vázquez José Antonio. (2003). Desarrollo Web con PHP y MySQL. Ediciones Anaya Multimedia S.A. ISBN 84-415-1525-5
- Welling Luck, Thompson Laura. (2005). Desarrollo Web con PHP y MySQL. Anaya Multimedia. ISBN: 978-84-415-1818-6
- El Proceso de Desarrollo Unificado del Software. Booch, Jacobson, Rumbaugh.  
[ Disponible en <http://bibliodoc.uci.cu/pdf/reg00060.pdf> ]
- Pekin, A., U. d. Jueces En línea. (2007).  
[Disponible en: <http://acm.pku.edu.cn/JudgeOnline> ]
- Pressman Roger. (2005).Ingeniería del Software: Un Enfoque Práctico. ISBN: 9701054733.
- Ural, U. E. Concurso Habilidades Programación.  
[Disponible en: <http://acm.timus.ru>.]
- Saratov, U. (2007). Concurso Habilidades Programación.  
[Disponible en: <http://acm.sgu.ru>.]
- UCI, Cátedra de Programación Avanzada, (2008).  
[Disponible en: <http://cpav.uci.cu>]

- Mendoza, M. A. (2004). "Metodologías de desarrollo de Software".  
[Disponible en:  
[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)]
- Gallego Vázquez José Antonio. (2003). Desarrollo Web con PHP y MySQL.  
[Disponible en <http://bibliodoc.uci.cu/pdf/reg02138.pdf> ]
- Pérez Valdés Damián. (2008). "Editores Web que facilitan tu trabajo."  
[Disponible en <http://www.maestrosdelweb.com>]
- Reina Quintero. (2000). "Visión General de la Programación Orientada a Aspectos".  
[Disponible en <http://www.lsi.us.es/docs/informes/aopv3.pdf>]
- Welling Luck, Thomson Laura. (2003). Desarrollo Web con PHP y MySQL.  
[Disponible en <http://bibliodoc.uci.cu/pdf/reg02819.pdf>]
- Estadísticas PHP para Agosto del 2007  
[Disponible en <http://php.net/> ]
- Razones para usar MySQL.  
[Disponible en <http://warp.es/mysql> ]
- Southwell, Michael. (2005). Pro PHP Security
- White Steve, Wallace Andy. (2006). Joomla! User Manual.
- LeBlanc Joseph. (2007). Learning Joomla Extension Development.
- Mellado Juan. (2005) Drupal: Creación de Módulos.

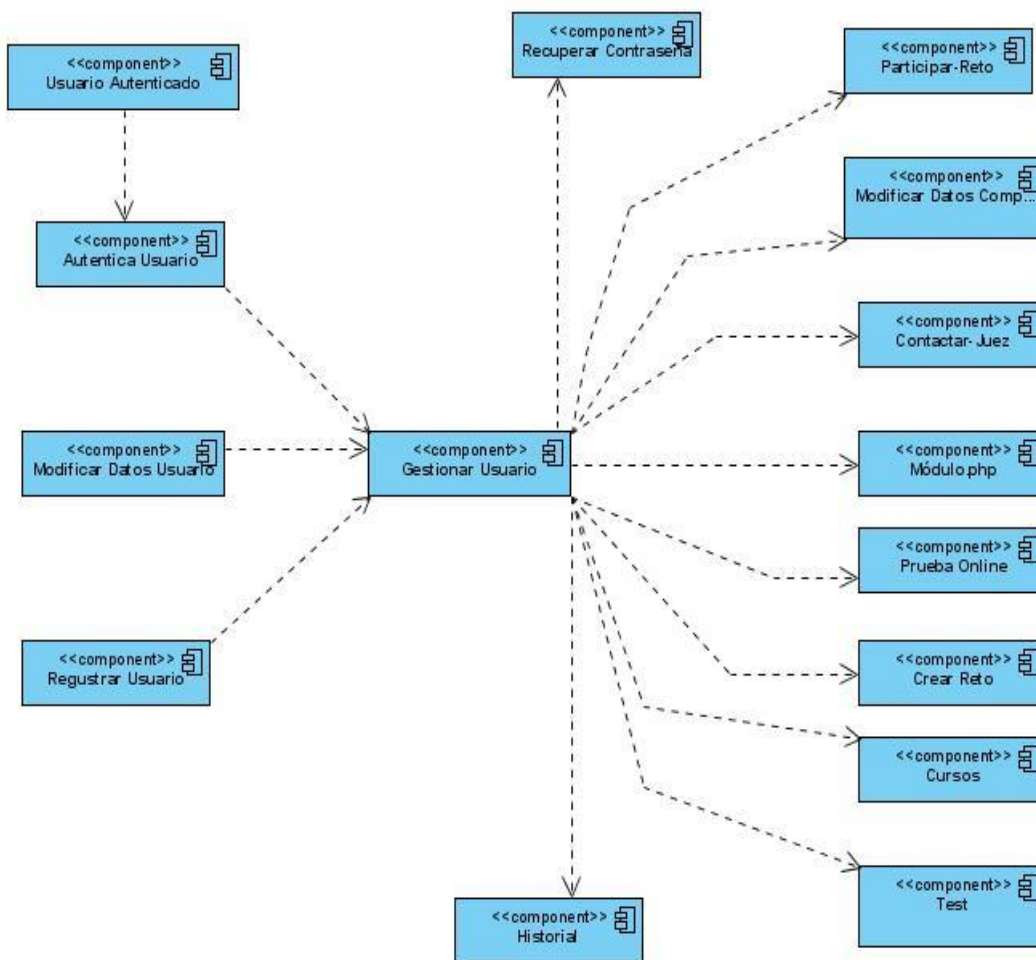
## Anexos



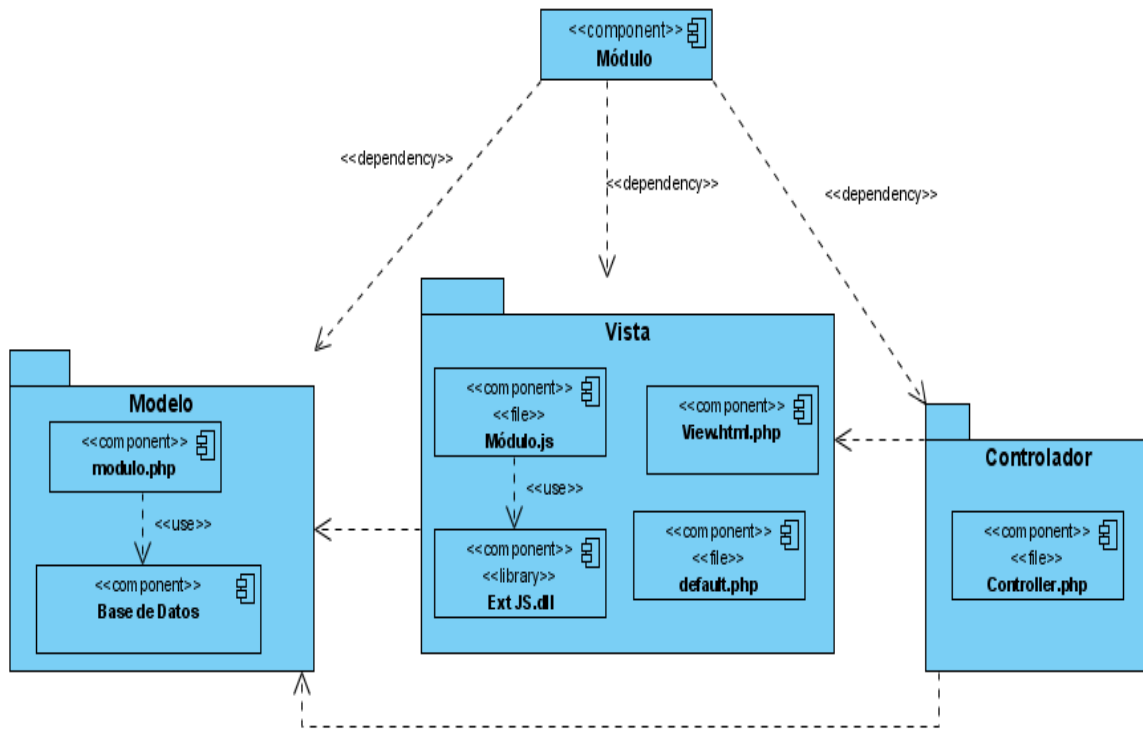
**Fig.** Diagrama de Despliegue del Sistema



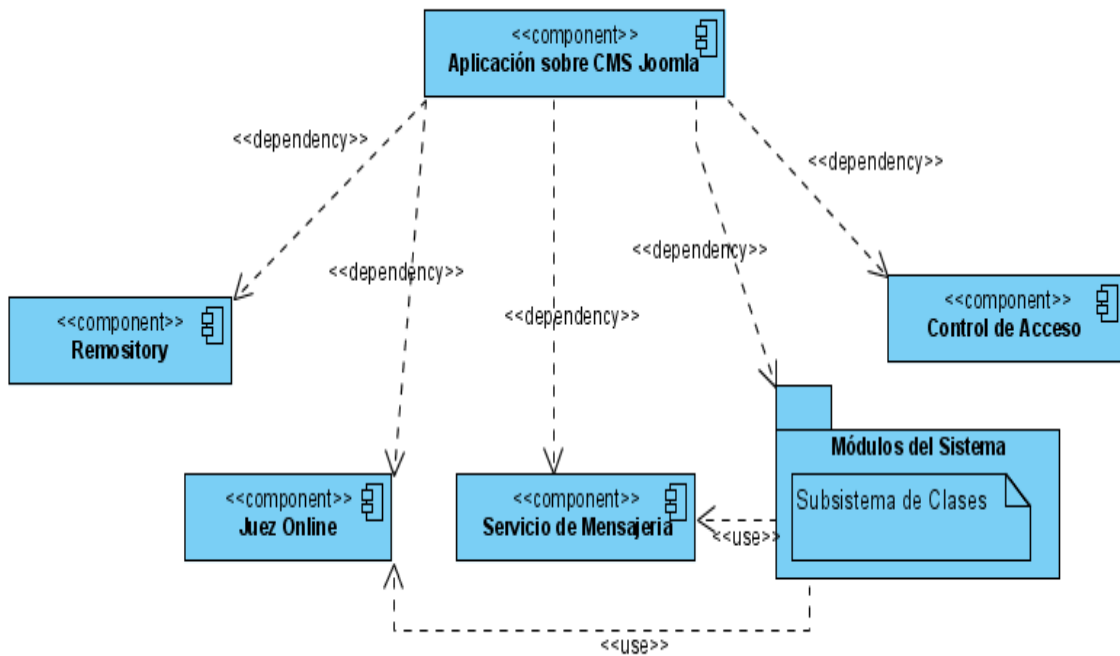
**Fig.** Diagrama de componentes equipo.



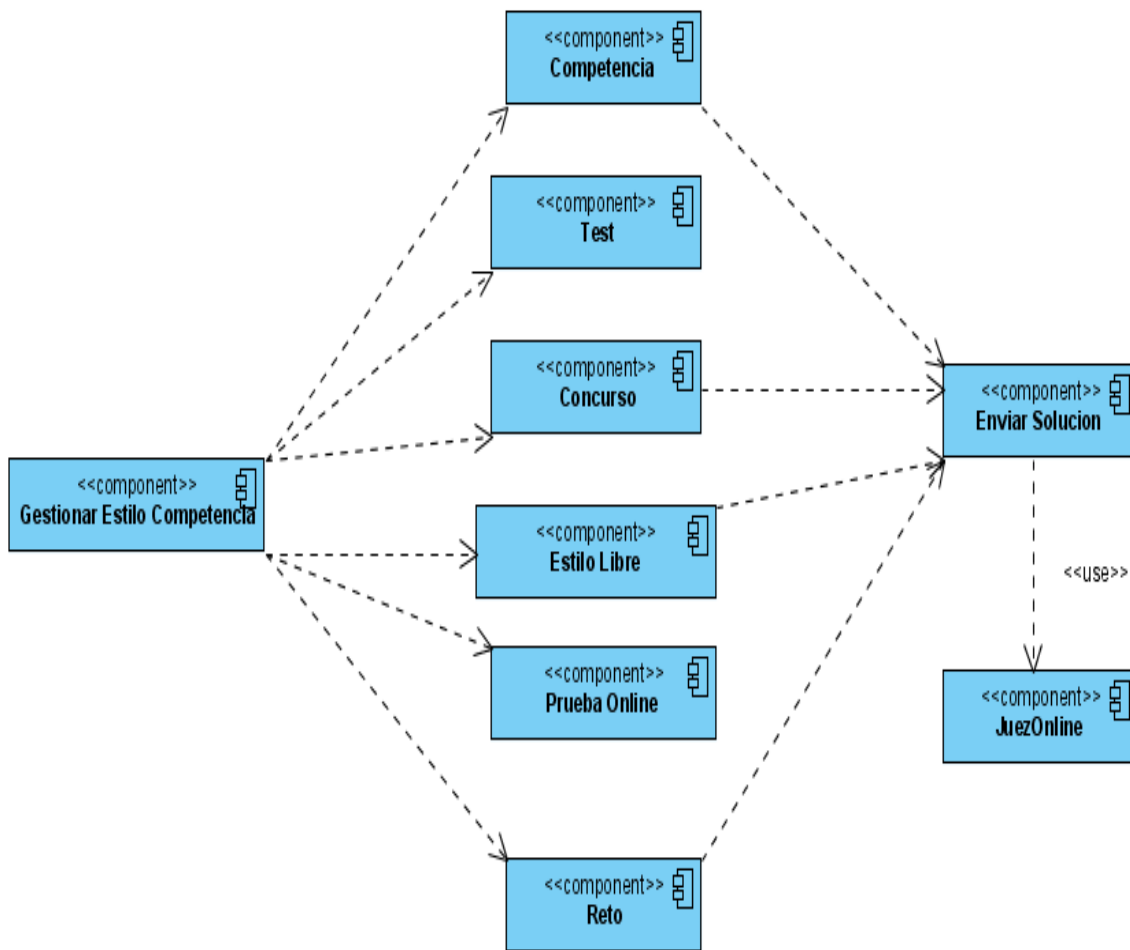
**Fig.** Diagrama de componentes Usuario



**Fig.** Diagrama de componentes del Sistema



**Fig.** Diagrama de componentes de los módulos del sistema.



**Fig.** Diagrama de componentes del Subsistema Estilos de Competencia.



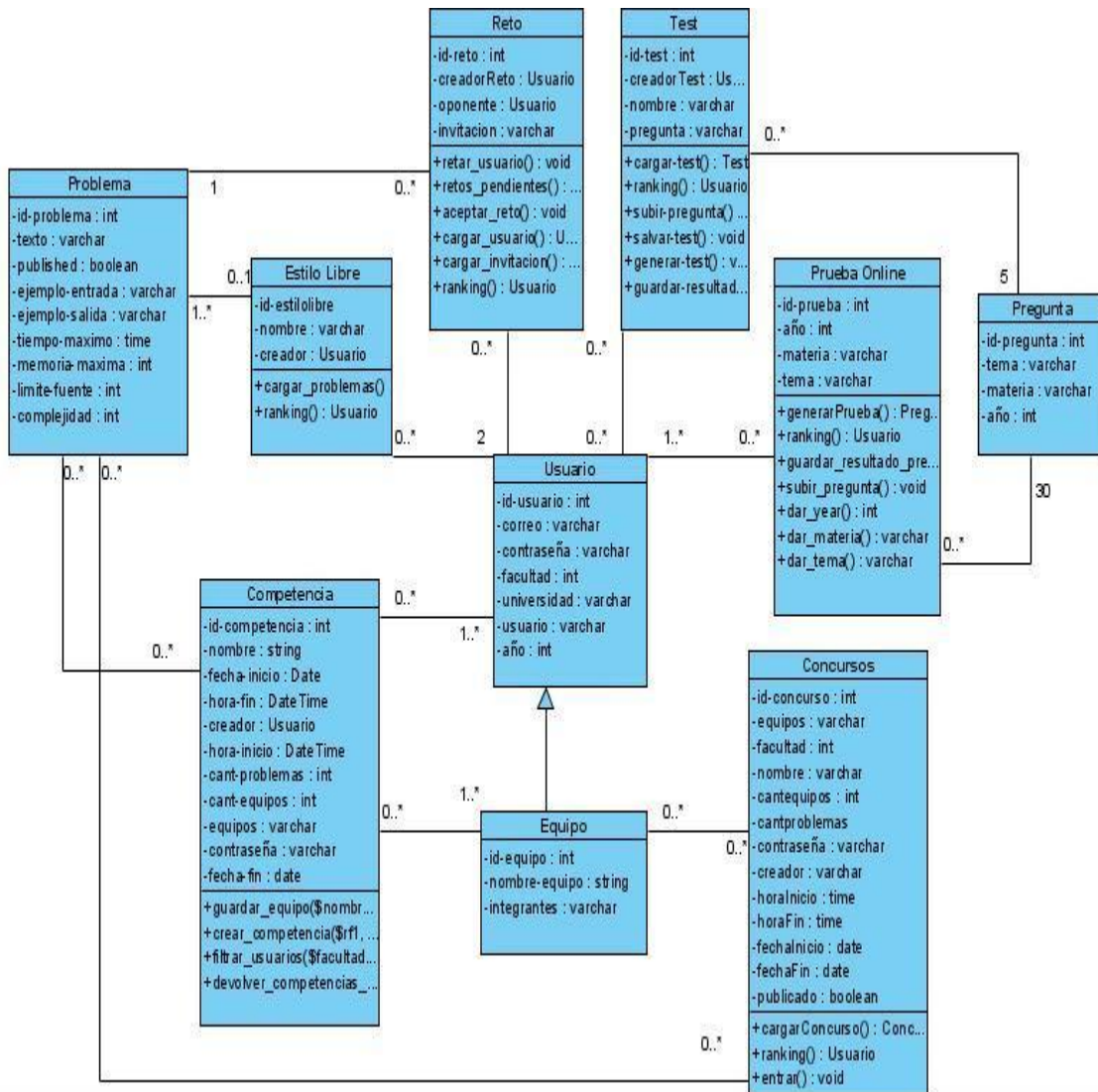
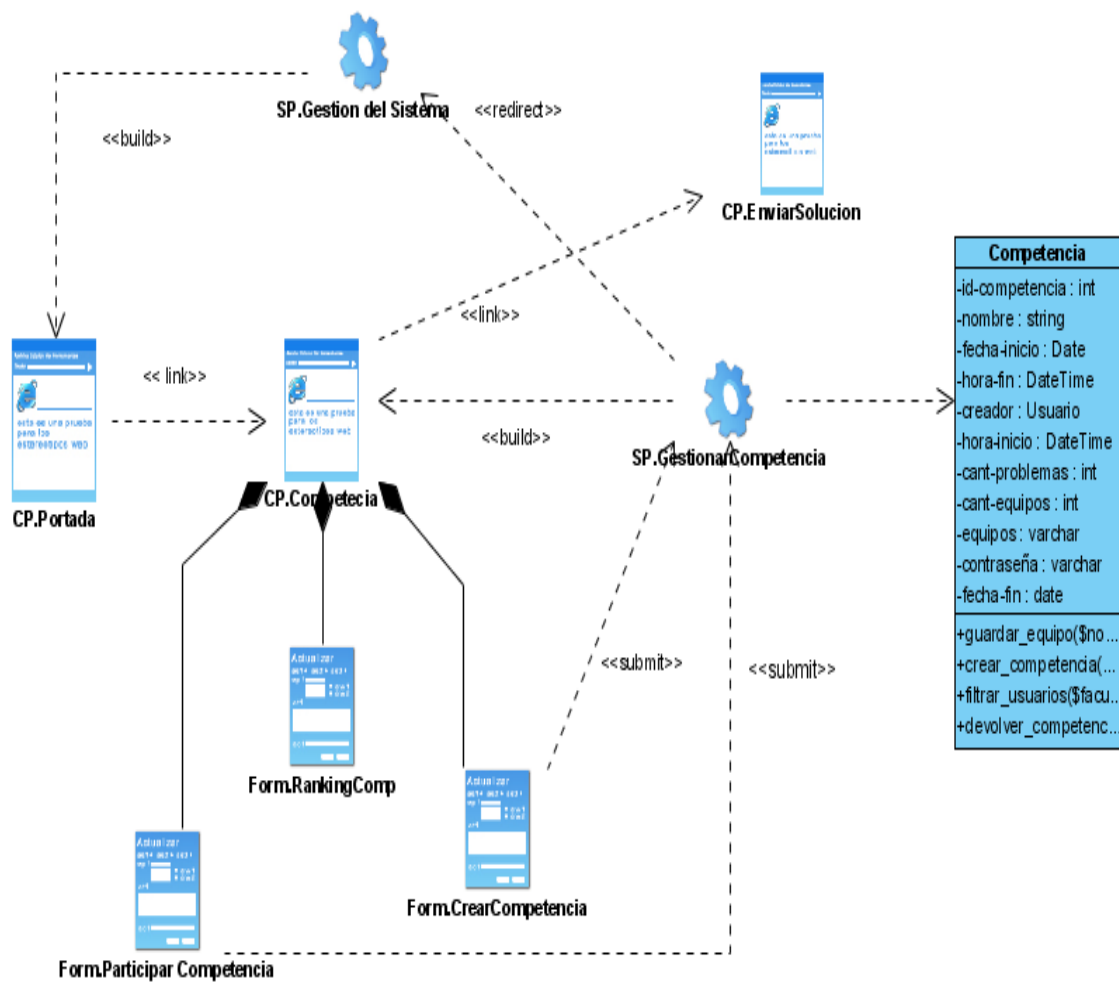
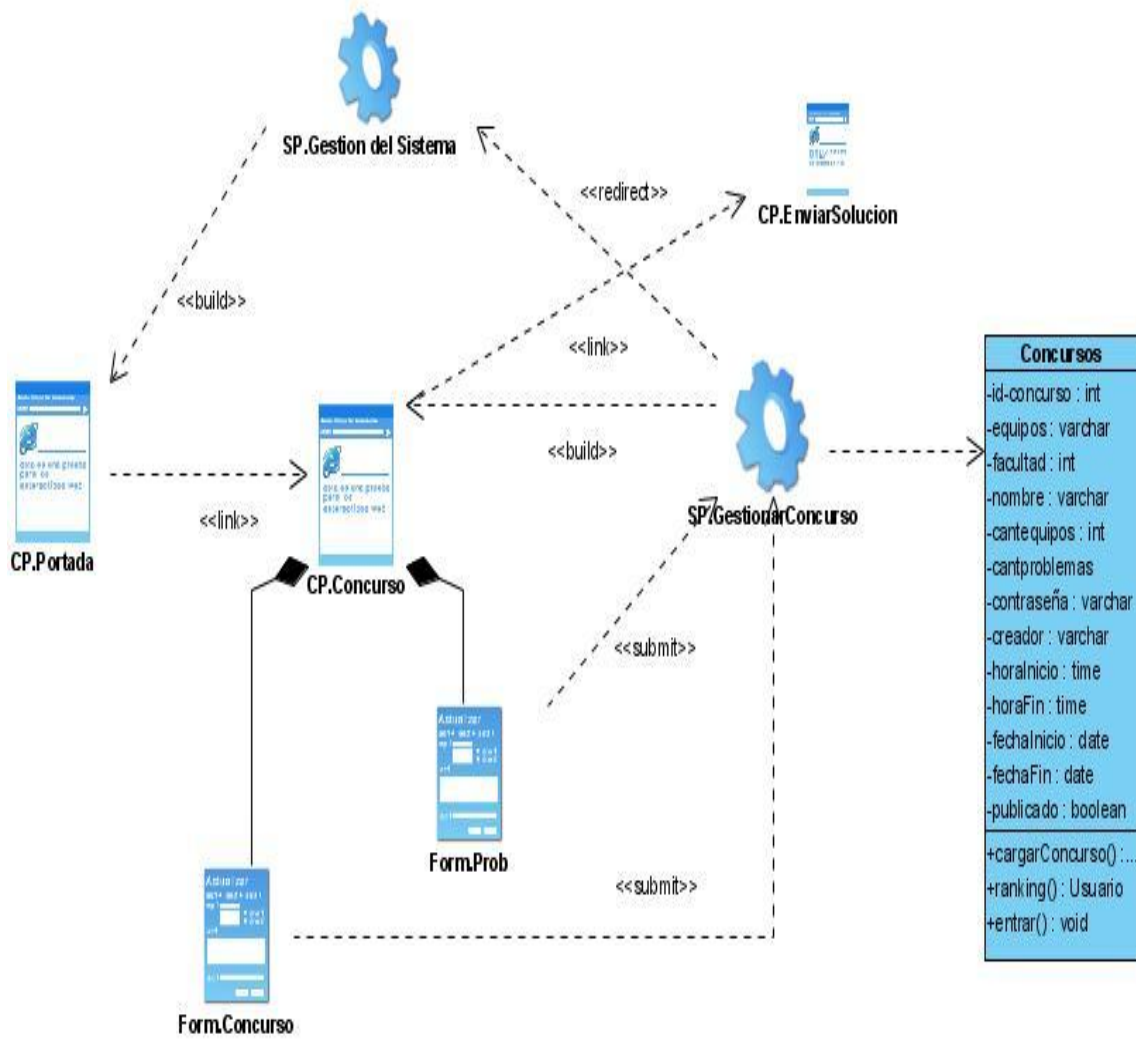


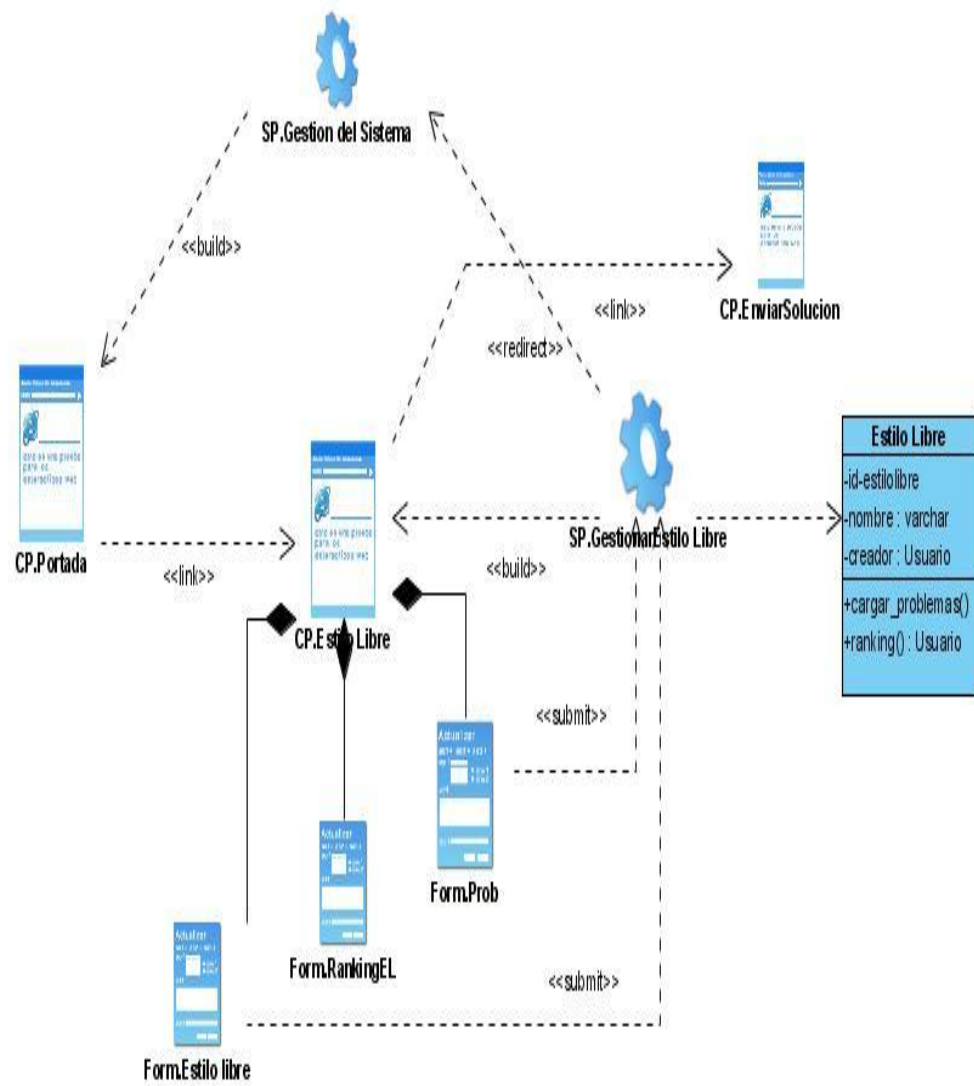
Fig. Diagrama de clases.



**Fig.** Diagrama de Clases del Diseño. Competencia



**Fig.** Diagrama de Clases del Diseño. Concurso



**Fig.** Diagrama de Clases del Diseño. Estilo Libre

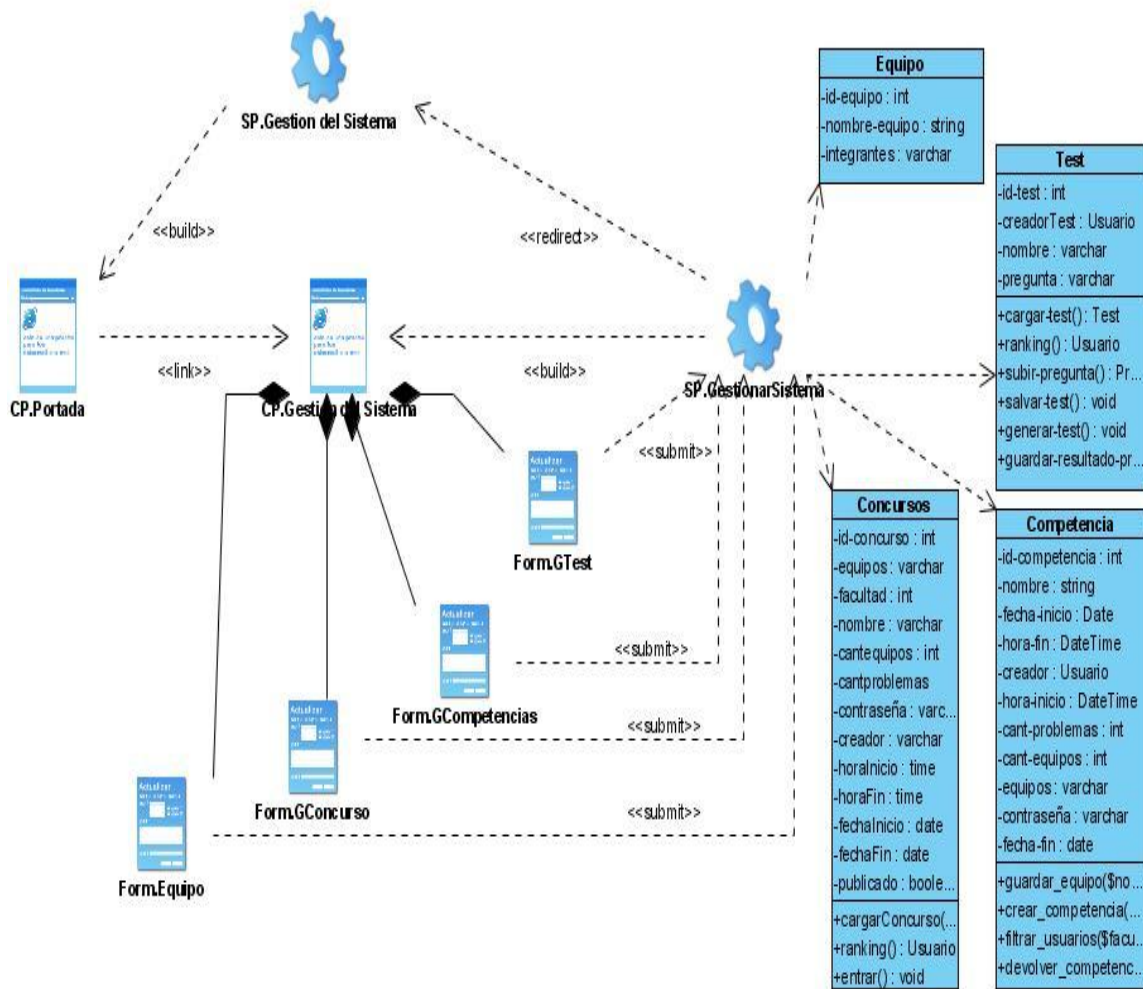
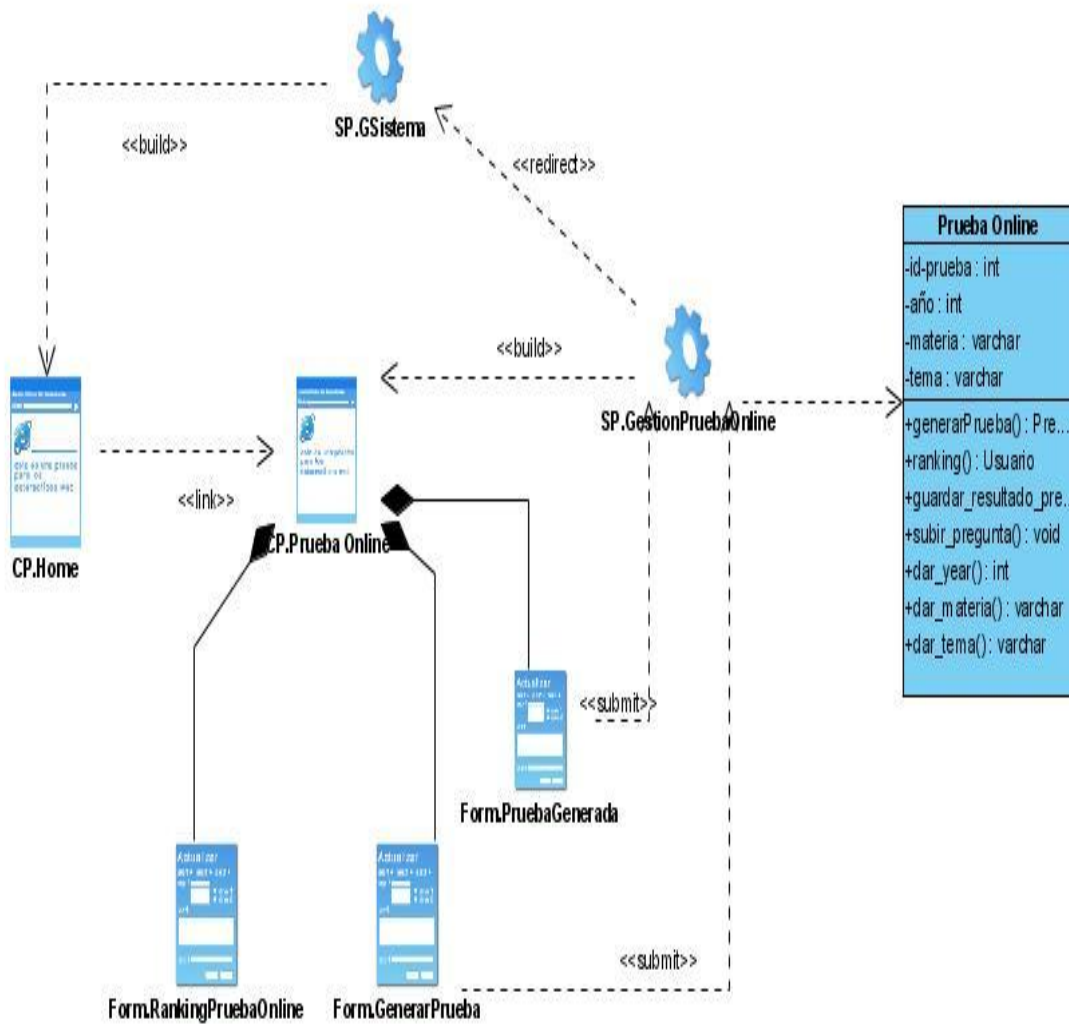


Fig. Diagrama de Clases del Diseño. Gestionar Sistema



**Fig.** Diagrama de Clases del Diseño. Prueba En línea

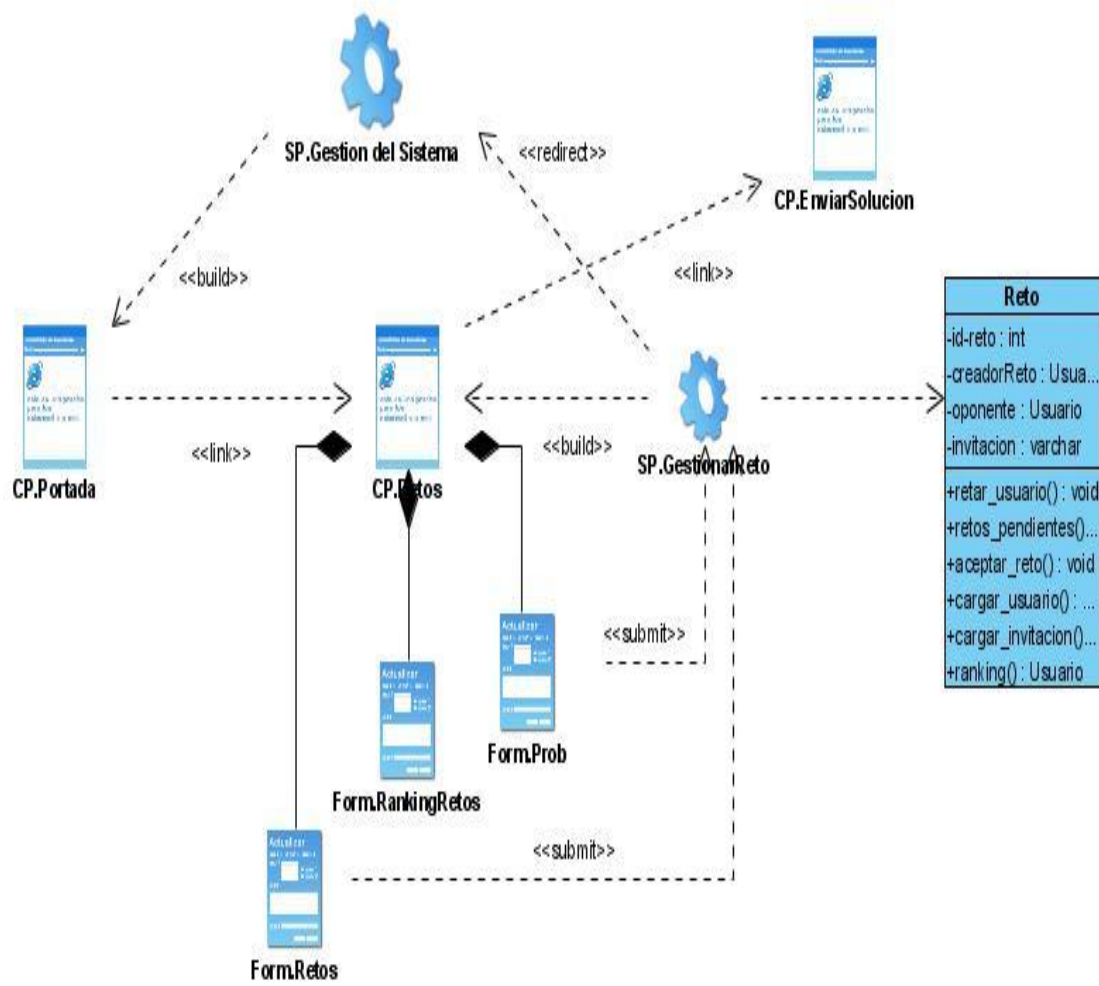
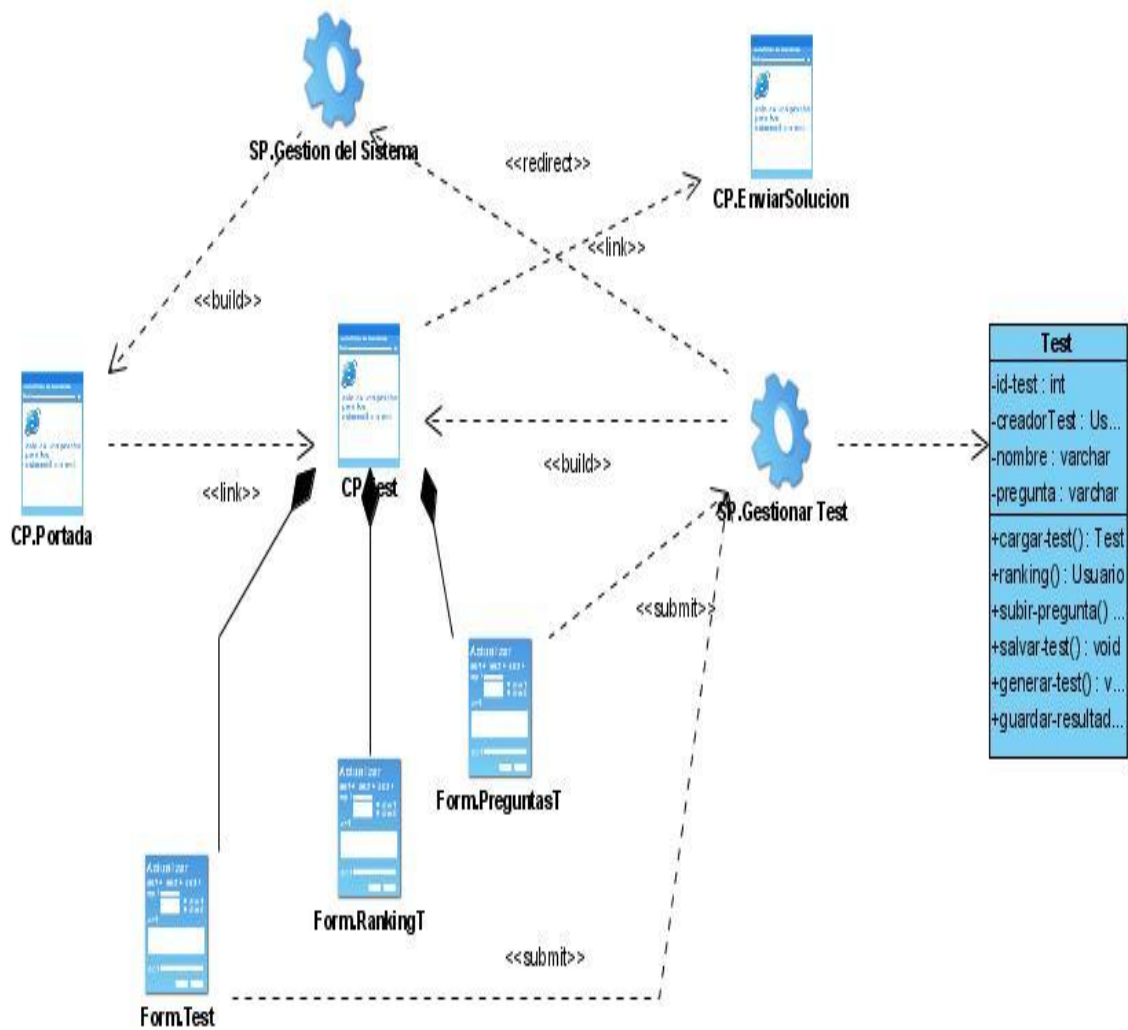


Fig. Diagrama de Clases del Diseño. Reto



**Fig.** Diagrama de Clases del Diseño. Test



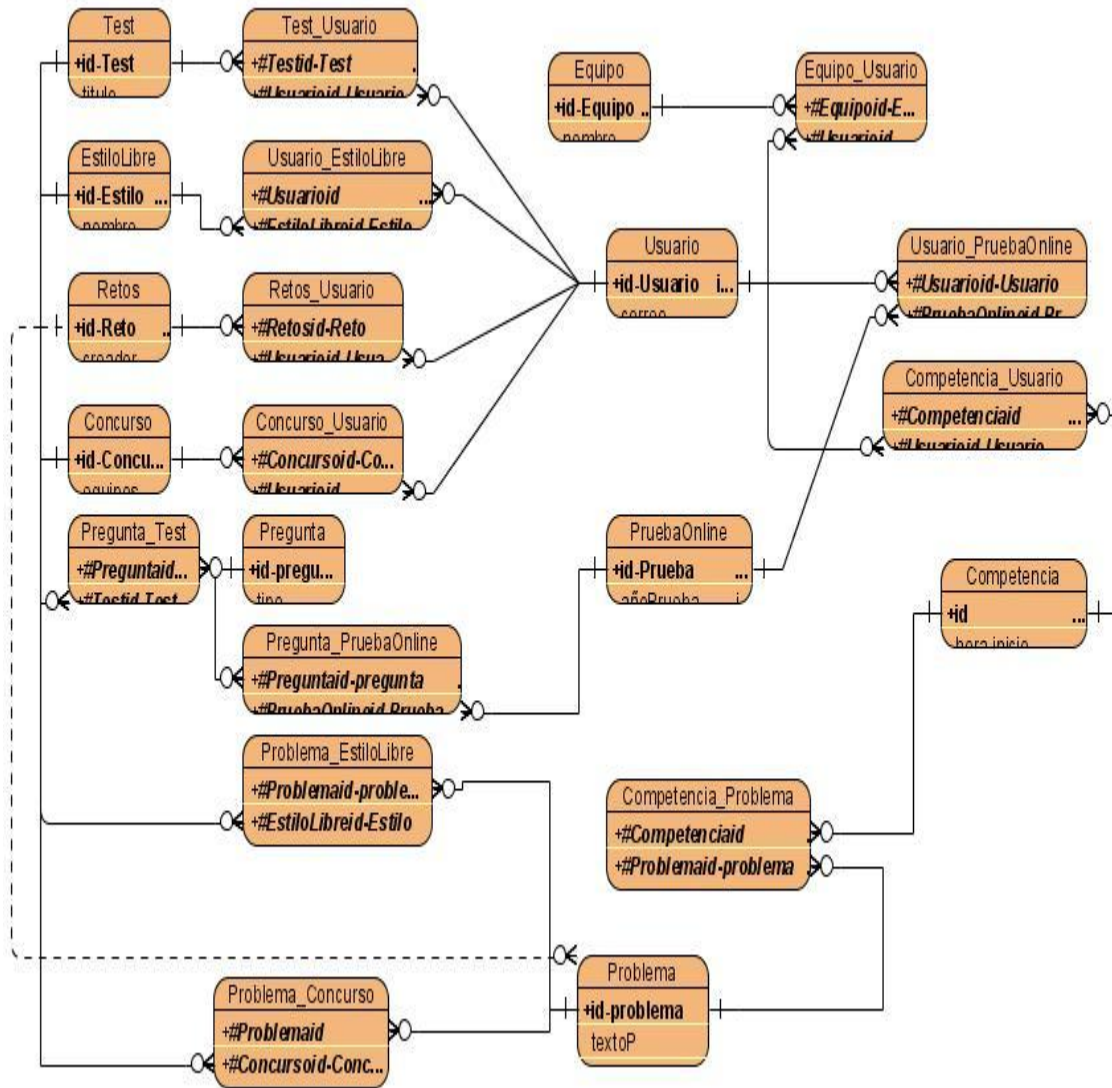


Fig. Diagrama Entidad Relación.

# Apéndices

## Glosario de términos

**Procesos:** conjunto de actividades secuenciales que realizan una transformación de una serie de inputs (material, mano de obra, capital, información, etc.) en los outputs deseados (bienes y/o servicios) añadiendo valor.

**Paradigma:** los paradigmas son un conjunto de conocimientos y creencias que forman una visión del mundo (cosmovisión), en torno a una teoría hegemónica en determinado periodo histórico.

**Hilos de ejecución:** es una característica que permite a una aplicación realizar varias tareas concurrentemente. Los distintos hilos de ejecución comparten una serie de recursos tales como el espacio de memoria, los archivos abiertos, situación de autenticación entre otras.

**Lista de Chequeo:** se entiende por lista de chequeo (cheks-list) a un listado de preguntas, en forma de cuestionario que sirve para verificar el grado de cumplimiento de determinadas reglas establecidas a prioridad con un fin determinado

**Caso de prueba:** “un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular, ejemplo: ejercitar un camino concreto de un programa o verificar el cumplimiento de un determinado requisito”.

**Aplicación Distribuida:** aplicación con distintos componentes que se ejecutan en entornos separados, normalmente en diferentes plataformas conectadas a través de una red.

**Socket:** método para la comunicación entre un programa del cliente y un programa del servidor en una red. Se define por una dirección IP, un protocolo de comunicación y un puerto.

# Índice de figuras y tablas

## Índice de figuras

- Fig .1 Metodología XP
- Fig .2 Metodología MSF
- Fig .3 Diseño propuesto por el analista
- Fig .4 Diseño de clases del sistema.
- Fig .5 Componente Remository
- Fig .6 Componente UddeIM.
- Fig .7 Diagrama de componentes del Sistema.
- Fig .8 Diagrama de componentes de código fuente.
- Fig .9 Diagrama de Clases del Diseño. Módulo Competencia
- Fig .10 Diagrama de componentes. Estilos de Competencia
- Fig .11 Diagrama de Despliegue.

## Índice de tablas

Características	Drupal	Joomla
<b>Código abierto:</b>	El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL.	Joomla corresponde al grupo de soluciones de código abierto, es un producto de software libre.
<b>Módulos:</b>	Posee muchos módulos que proporcionan funcionalidades como "página de categorías" autenticación mediante <i>jabber</i> , mensajes privados, <i>bookmarks</i> , entre otros.	En un sitio creado con Joomla, la posición de módulos puede acomodarse como se prefiera. Módulos son los bloques laterales del sitio
<b>Personalización:</b>	Un robusto entorno de personalización. Tanto el contenido como la presentación pueden ser personalizados.	Ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos, que agregarán servicios de valor a los visitantes de tu sitio web, por ejemplo: galerías de imágenes, foros, newsletters, clasificados, etc.
<b>Autenticación de usuarios:</b>	Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo. Para su uso en una intranet, Drupal se puede integrar con un servidor LDAP.	Joomla permite almacenar datos de usuarios registrados y también te da la posibilidad de enviar E-mails masivos a todos los usuarios.
<b>Permisos basados en roles:</b>	Los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un "rol" y agrupar los usuarios por roles.	La administración de usuarios es jerárquica, y los distintos grupos de usuarios poseen diferentes niveles de facultades/permisos dentro de la gestión y administración del sitio
<b>Plantillas (Platillas):</b>	El sistema de temas de Drupal separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web. Se pueden crear plantillas con HTML y/o con PHP.	Es posible cambiar todo el aspecto del sitio web tan solo con un par de clicks, gracias al sistema de platillas que utiliza Joomla.
<b>Agregador de noticias:</b>	Drupal incluye un potente Agregador de Noticias para leer y publicar enlaces a noticias de otros sitios web. Incorpora un sistema de cache en la base de datos, con temporización configurable.	Se pueden crear páginas ilimitadas y editarlas desde un sencillo editor (similar al editor de MS Word) que permite formatear los textos con los estilos e imágenes deseados. Los contenidos son totalmente editables y modificables.
<b>Independencia de la base de datos:</b>	Drupal incorpora una 'capa de abstracción de base de datos que actualmente está implementada y mantenida para MySQL.	Permite el trabajo con las bases de datos MySQL de forma independiente.
<b>Multi-plataforma:</b>	Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solari, Windows y Mac OS X.	Es compatible con las plataformas Linux, Windows y Mac OSX.

**Tabla:** Resumen comparativo entre los CMS Joomla y Drupal teniendo en cuenta los aspectos fundamentales que exige el sistema a desarrollar.

- Tabla 1. Clase Interfaz: JView
- Tabla 2. Clase Entidad: JModel
- Tabla 3. Clase Controladora: JController
- Tabla 4. Clase Entidad: CcompetenciaModelcompetencia
- Tabla 5. Clase Entidad: EstilolibreModelestilolibre
- Tabla 6. Clase Controladora: PruebaOnlineController
- Tabla 7. Clase Controladora: retosController
- Tabla 8. Caso de Prueba 1
- Tabla 9. Matriz de casos de Prueba 1
- Tabla 10. Caso de Prueba 2
- Tabla 11. Matriz de casos de Prueba 2
- Tabla 12. Caso de Prueba 3
- Tabla 13. Matriz de Caso de Prueba 3
- Tabla 14. Caso de Prueba 4
- Tabla 15. Matriz de Caso de Prueba 4
- Tabla 16. Caso de Prueba 5
- Tabla 17. Matriz de Caso de Prueba 5
- Tabla 18. Caso de Prueba 6
- Tabla 19. Matriz de Caso de Prueba 6

Tabla 20. Caso de Prueba 7

Tabla 21. Matriz de Caso de Prueba 7

Tabla 22. Integración de los módulos.

Tabla 23. Lista de Chequeo

Tabla 24. Análisis de No conformidades.