



ANÁLISIS DEL SISTEMA "COLISEO VIRTUAL".

**por
Janet Carreño Cáceres.**

**Tesis propuesta para optar por el título de: Ingeniero
Informático**

Tutor: Lic. Rolan R. Bullain Diéguez.

Universidad de las Ciencias Informáticas.

2007

DECLARACIÓN DE AUTORÍA.

Declaro que soy la única autora de este trabajo y autorizo 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 firmo la presente a los ____ días del mes de _____ del año _____.

Janet Carreño Cáceres.

Lic. Rolan R. Bullain Diéguez.

AGRADECIMIENTOS.

A la fortaleza y dedicación de mis padres.

Gracias por enseñarme con el ejemplo, la responsabilidad y el respeto hacia el trabajo, a forjar el carácter para enfrentar mis temores y tragedias. Por su cariño, apoyo, esfuerzo y valentía para formar un ingeniero.

Para ambos, muchas gracias.

A mi hermanito que es mi vida.

A mi abuelita que la quise tanto y se que si estuviera aquí, se sentiría orgullosa de mi.

A Jaz por estar siempre a mi lado, en los momentos buenos y en los malos, aconsejándome y apoyándome en toda decisión.

A mi tutor Rolan, gracias, por apoyarme, aconsejarme y confiar en mí desde el inicio.

A todos nuestros compañeros y compañeras de aula (y de año) por hacer mas fáciles todos estos años.

A mi familia por su apoyo y por confiar en todo momento en mí.

A nuestra Revolución y nuestro Comandante por darnos todo para poder ser alguien en la vida.

A todos muchas gracias.

RESUMEN.

Este trabajo tiene su motivación en el concurso universitario Copa Pascal, su objetivo principal es realizar el análisis para el desarrollo de un sistema que potencie el desarrollo del pensamiento lógico, deductivo y de habilidades en la programación mediante una forma didáctica. El estado del arte sobre el concurso Copa Pascal, jueces online y sitios web dedicados a la organización y gestión de concursos nos brindan los elementos necesarios para la creación de una aplicación más general, la cual dará la opción de: publicación de artículos, realizar competencias, definir retos, descargar cursos. La misma estará apoyada por un juez online que evaluará las soluciones enviadas en los distintos lenguajes de programación definidos; con un generador de pruebas configurable mediante un criterio especificado por el usuario (año(s), materia(s), tema(s)); un calificador de test y un chat para los concursos centralizados. Todo esto bajo una interfaz grafica amigable y de fácil acceso.

TABLA DE CONTENIDO.

Introducción	1
Conclusiones.....	3
Capítulo 1: Fundamentación Teórica	4
1.1 Jueces Online y aplicaciones Web dedicadas al desarrollo de habilidades.....	4
1.2 Metodologías, Herramientas, Lenguajes de Modelación.....	6
Metodologías.....	7
Herramientas.....	10
UML como lenguaje de modelación.....	11
1.3 Ingeniería de Requisitos.....	12
1.4 Gestión de Requisitos.....	15
1.5 Tareas principales de la Gestión de Requisitos.....	15
1.6 Patrones de Caso de Uso.....	17
1.7 Analista de Sistemas.....	18
1.8 Conclusiones.....	19
Capítulo 2: Modelo del Dominio y Requerimientos	20
2.1 Objeto de Estudio.....	20
2.1.1 Problema.....	20
2.1.2 Objeto de Automatización.....	21
2.1.3 Propuesta de Sistema.....	21
2.2 Modelo del Dominio.....	22
2.2.1 Descripción general del proceso.....	22
Modelamiento:.....	22
2.3 Especificaciones de los Requerimientos del Software.....	24
2.3.1 Requisitos funcionales.....	24
2.3.2. Requisitos no funcionales.....	26
2.4 Conclusiones.....	28
Capítulo 3: Sistema	29
3.1 Casos de uso del sistema.....	29
3.1.1 Definición de los actores.....	29
3.1.2 Descripción de casos de uso.....	30
3.1.3 Diagrama de casos de uso y paquetes.....	37

3.1.4 Casos de uso expandidos.....	42
3.2 Análisis del Resultado.....	57
3.3 Conclusiones.....	61
Conclusiones Generales.....	62
Recomendaciones.....	63
Referencias Bibliograficas.....	64
Bibliografias.....	65
Glosario de Términos.....	68

INTRODUCCIÓN.

La informática es una de las ciencias más jóvenes que está en constante desarrollo, esta se encuentra presente en prácticamente todas las esferas del mundo actual. Nuestro país, desde hace algunos años, se ha dedicado a intensificar dicha área. Para cumplir este objetivo se ha creado toda una infraestructura de universidades, dentro de las cuales se encuentra la Universidad de las Ciencias Informáticas (U.C.I), con el fin de potenciar el desarrollo de profesionales altamente capacitados.

En nuestra universidad existe un plan de estudio flexible ajustado a las características de centro docente y empresa productiva, donde se realiza un trabajo intenso con los estudiantes, para que puedan obtener las diferentes habilidades que necesita un especialista del área. La mayor parte de este trabajo se ejecuta mediante las actividades docentes curriculares, también se planifican consultas, planes remediales y actividades docentes didáctico-recreativas con el fin de vincular y estimular desde la recreación, el desarrollo del conocimiento; sin embargo, existe un número insuficiente de herramientas en explotación que nos ayuden a cumplir los objetivos planteados por nuestra universidad. Actualmente en la universidad se realiza la Copa Pascal como uno de los eventos docentes más importantes de cada curso, para la realización de la misma se han usado jueces online ubicados en Internet o la adaptación de un juez online descargado desde algunos de los sitios proveedores de esta herramienta.

La Copa Pascal se realiza una vez por curso, a nivel de facultad y luego a nivel universitario, limitando a este tiempo las posibilidades de competir y medir sus conocimientos de una forma didáctica. Otros jueces online han sido publicados en la facultad 4 y 8, funcionando, en ocasiones, de manera intermitente. Ambos con el inconveniente que solo incluyen lo que hemos denominamos en este trabajo, el estilo libre de competencia. Por lo que los estudiantes están limitados a: un solo estilo de competencia durante el período de la Copa Pascal, no pueden crear sus propias competencias y no tienen herramientas que le permita desarrollar otras habilidades y que a la vez les ayude a consolidar sus conocimientos docentes; por lo que surge la necesidad de definir las fronteras de un nuevo sistema, los requisitos pertinentes para el mismo y las interfaces enfocadas a necesidades y objetivos del usuario. Considerando todo lo antes planteado la **situación problémica** de este trabajo.

Como **problema científico** a partir de la situación anteriormente descrita se define: el pobre entendimiento entre los clientes y el equipo de desarrollo acerca de que funcionalidades debe proveer el sistema, debido a la ausencia de los artefactos que deben ser generados por el analista a partir de una correcta captura de requisitos.

Para darle solución al problema planteado, este trabajo se enmarca en la Ingeniería de requisitos para la informatización de una aplicación enfocada al desarrollo de habilidades de programación como **objeto de estudio**.

Para lo cual nuestro **objetivo de la investigación** es: obtener un modelo del sistema "Coliseo Virtual", a partir de una correcta captura de requisitos, con el fin de automatizar un sistema que valga de apoyo al usuario para desarrollar sus habilidades en la programación.

Cuyos **objetivos específicos** son:

- Realizar el modelo del entorno en el que se encuentra el sistema (Modelo del Dominio).
- Efectuar la captura de requisitos tanto funcionales como no funcionales.
- Describir el modelo y los casos de uso del sistema que darán la funcionalidad previa al mismo.
- Obtener prototipos no funcionales de las páginas de autenticación del sistema "Coliseo Virtual".

El **campo de acción** se encuentra constituido por el desarrollo y gestión de requisitos y la obtención de artefactos propios del análisis.

Hipótesis: Si se generan todos los artefactos correspondientes al análisis, a partir de una correcta captura de requisitos, se logrará, por parte del equipo de desarrollo, un sistema eficiente que logre satisfacer las expectativas y necesidades de los usuarios.

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

- Realizar el estado del arte sobre el concurso Copa Pascal, jueces online y sitios web dedicados a la organización y gestión de concursos.
- Evaluar la información obtenida sobre el estado del arte y adaptarla a nuestros requerimientos y expectativas.
- Realizar el Modelo del dominio.

- Realizar la captura, definición y validación de requisitos tanto funcionales como no funcionales mediante técnicas definidas (Entrevistas, Tormenta de Ideas, etc.).
- Definir actores del sistema.
- Definir casos de uso del sistema.
- Realizar diagramas de casos de uso del sistema y agruparlos en paquetes.
- Realizar una breve descripción de cada uno de los casos de uso definidos.
- Realizar las especificaciones¹ correspondientes a cada uno de los casos de uso definidos.
- Realizar una definición de los prototipos no funcionales de las páginas de autenticación del sistema.
- Definir el patrón de casos de uso a utilizar y argumentar su selección.
- Definir los artefactos que se obtendrán como resultado de la metodología seleccionada.

Conclusiones.

A lo largo del presente trabajo se muestran 3 capítulos fundamentales. En el *Capítulo 1* se aborda sobre el estado del arte realizado a varios sitios de desarrollo de habilidades; se exponen las características de las herramientas y tecnologías seleccionadas para darle solución al problema antes descrito, se define la Ingeniería de requisitos y la labor del analista de sistemas. En el *Capítulo 2* se define: objeto de estudio, de automatización, sistema propuesto, modelo del dominio, especificación de los requerimientos del sistema. En el *Capítulo 3* se definen los actores y los diagramas de casos de uso del sistema, concluyendo con un análisis de resultado del trabajo realizado.

¹ Especificaciones: Colección de escenarios de uso con descripciones en lenguaje natural [17].

Capítulo 
Fundamentación Teórica.

En el siguiente capítulo se realiza: un estudio de sistemas con características similares al sistema a desarrollar, se exponen las características de las herramientas y tecnologías seleccionadas para darle solución al problema, se argumenta la elección de las mismas, y se aborda sobre la Ingeniería de requisitos y la labor del analista de sistemas.

1.1 Jueces Online y aplicaciones Web dedicadas al desarrollo de habilidades.

En Internet se pueden encontrar varios jueces online dedicados a la calificación de concursos de programación. Estos servidores se encuentran ubicados en diferentes países, de forma general son patrocinados por universidades y por la Asociación para la Maquinaria de Computación (ACM). El objetivo primordial de estos concursos es la casa de talentos para las grandes compañías. Objetivo que dista mucho de las metas de este trabajo, pero algo si tienen positivo estos concursos, su influencia en el desarrollo del pensamiento lógico-algorítmico y el trabajo en equipo. Otorgándole el tratamiento adecuado pueden obtenerse beneficios que complementen el trabajo docente. Es por ello que se ha estudiado las características de algunos de estos jueces online:

- <http://acm.zju.edu.cn>: Se encuentra ubicado en China, patrocinado por la Universidad de Zhejiang y por la ACM. Los problemas se encuentran agrupados en volúmenes (18 en total), cada uno con 99 problemas. 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, 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.
- <http://acm.timus.ru>: Ubicado en Rusia, patrocinado por la Universidad Estatal Ural. Los problemas se encuentran agrupados en volúmenes (6 en total), cada uno con 99 problemas. 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.

- <http://acm.sgu.ru>: Ubicado en Rusia, patrocinado por la Universidad de Saratov. Los problemas se encuentran agrupados en volúmenes (3 en total), cada uno con 99 problemas. 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.
- <http://acm.pku.edu.cn/JudgeOnline>: Ubicado en China, patrocinado por la Universidad de Pekín y ACM. Los problemas se encuentran agrupados en volúmenes (19 en total), cada uno con 99 problemas. 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 online que puede ser descargado.
- Copa Pascal: No es exactamente un juez online, está dirigido por el Departamento de Técnica de Programación y patrocinado por la dirección de la UCI. No dispone de un url fijo y solo se puede encontrar disponible el momento del concurso. Las veces que se ha usado un juez online, ha variado la cantidad de lenguajes que acepta en ocasiones se lleva una estadística de los problemas solucionados por los equipos, así como un ranking.
- <http://imaginecup.com>: Concurso mundial, patrocinado por Microsoft, su objetivo fundamental es la casa de talentos. Inicialmente comenzó con no más de tres estilos de competencias, en la actualidad tiene más de 10, los más populares son: 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.
- <http://www.mslatam.com/latam/msdn/comunidad/dce2005/>: Popularmente conocido como "Desarrollador Cinco Estrellas", permite la certificación online. Una vez registrado se le suministran materiales de estudio a los usuarios y luego se le aplica un examen que va subiendo de nivel, las estrellas. En dichos exámenes se deben responder cierta cantidad de preguntas.

Hay que destacar todas las preguntas son de selección simple o múltiple pero no le resta calidad a los exámenes.

Como se puede observar la mayoría de estos sitios están dotados de herramientas para desempeñar su objetivo, jueces online. Unos dedicados solos a lo que se le ha llamado “Estilo Libre”, otros al desarrollo de exámenes como el “Desarrollador Cinco Estrellas” y otros con estilos más variados como el “Imagine Cup”; pero todos con fines lucrativos, aunque enmascarados.

1.2 Metodologías, Herramientas, Lenguajes de Modelación.

"La Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos" [1]

La ingeniería del Software es una tecnología multicapa constituida por: Procesos, métodos y herramientas.



Fig. 1.1: La Ingeniería de Software, una tecnología multicapa.

El proceso de la ingeniería del software es la unión de las capas de tecnologías que permite un desarrollo racional y oportuno de la ingeniería del software; los métodos de la ingeniería del software indican como construir técnicamente el software, estos abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento; las herramientas de la ingeniería del software proporcionan un enfoque automático o semi-automático

para el proceso y para los métodos. Cuando se integran herramientas para que la información creada por una herramienta la pueda utilizar otra, se establece un sistema de soporte para el desarrollo del software llamado Ingeniería del Software Asistida por Computador (CASE) [2]

Metodologías.

Una metodología es la que se encarga de elaborar estrategias de desarrollo de software que promuevan prácticas adoptativas en vez de predictivas; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente.[3]

El **Proceso Unificado de Rational** o RUP (siglas en inglés), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado. [4]

Sus principales características son:

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

RUP reemplaza el tradicional esquema de desarrollo “en cascada”, donde cada etapa se iba desarrollando en forma secuencial (modelo del negocio, análisis de requerimientos, diseño, construcción, prueba e implementación) por un proceso:

- **Iterativo e incremental:** consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante

la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

- **Centrado en la arquitectura:** la arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.
- **Guiado por casos de uso:** Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido, representan los requisitos funcionales del sistema.

RUP incluye artefactos y roles, divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

- Inicio: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- Transición: se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados. [5]

Métrica 3.

Es una Metodología de Planificación, Desarrollo y Mantenimiento de Sistemas de Información. Puede ser utilizada libremente con la restricción de citar la fuente de su propiedad intelectual: el Ministerio de Administraciones Públicas de España. Este Ministerio, desde el Consejo Superior de Informática,

ofrece así un instrumento para la sistematización de las actividades que dan soporte al ciclo de vida del software en el desarrollo de Sistemas de Información.

Su aplicación pretende los siguientes objetivos:

Proporcionar o definir Sistemas de Información que ayuden a conseguir los fines de la organización mediante la definición de un marco estratégico para el desarrollo de los mismos.

- Dotar a la Organización de productos software que satisfagan las necesidades de los usuarios dando una mayor importancia al análisis de requisitos.
- Facilitar la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos.
- Facilitar la operación, mantenimiento y uso de los productos software obtenido.

Esta metodología no es orientada a objetos. [6]

Extreme Programming.

Extreme Programming es una metodología para equipos de desarrollo de software pequeños o medianos. Es una metodología que se utiliza en organizaciones en donde los requisitos son vagos y variantes. Sus elementos claves son: poca documentación, simplicidad, análisis como una actividad constante, diseño evolutivo, integraciones y testeos diarios. Elimina la gran documentación del sistema como un todo y ataca mínimas partes del sistema que son rápidamente implementadas y testeadas. De esta forma el sistema va creciendo junto con el nuevo conocimiento del cliente y sus nuevas necesidades. [7].

La metodología se basa en:

- Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- Re fabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma

estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

Herramientas.

Las herramientas constituyen productos de software que apoyan al desarrollo de software, las mismas permiten: automatizar procesos repetitivos, mantener la información bien estructurada, gestionar grandes cantidades de información, además de permitir un aumento de la productividad y la calidad.

Rational Rose.

Es la herramienta CASE desarrollada por los creadores de UML, que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables; ofrece la capacidad de modelar y visualizar sus procesos de negocios y destacar oportunidades para aumentar la eficiencia. El navegador UML de Rational Rose nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción.[8]

El Rational Rose unifica a los analistas de negocios, sistemas y datos al permitirles crear y administrar modelos en una herramienta con un solo lenguaje de modelado. Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas.

Visual Paradigm.

Es una herramienta con un diseño centrado en casos de uso y enfocado al negocio que genera un software de calidad, tiene la particularidad de ser un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación. Posee capacidades de ingeniería directa (versión profesional) e inversa, modelo y código que permanece sincronizado en todo el ciclo de desarrollo, disponibilidad de múltiples versiones, disponibilidad en múltiples plataformas.

Existen varias versiones de Visual Paradigm para UML. La gratuita no permite realizar ingeniería inversa, pero permite crear diagramas y generar código a partir de ellos. Las versiones comerciales difieren entre ellas por su funcionalidad y su capacidad de integración con otras herramientas; también hay disponible un programa para acceder a las herramientas a un precio bajo si se desea emplear con fines académicos.

UML como lenguaje de modelación.

El lenguaje Unificado de Modelado es ante todo un lenguaje que proporciona un vocabulario y reglas para permitir una comunicación; en este caso este lenguaje se centra en la representación gráfica de un sistema.

UML surge a raíz de la necesidad de comunicar las ideas a otros desarrolladores además para servir de apoyo en los procesos de análisis de un problema. UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y diseño; nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: a partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Aunque este lenguaje está pensado para modelar sistemas complejos, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (workflow) en una empresa.

Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- Relaciones: relacionan los elementos entre sí.
- Diagramas: son colecciones de elementos con sus relaciones. [9]

El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue; sirve para el modelado completo de sistemas complejos, es además un método formal de modelado.

Esto aporta las siguientes ventajas:

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos.
- Es un lenguaje de uso universal.
- Prácticamente todas las herramientas CASE y de desarrollo la han adaptado como lenguaje de modelado.

1.3 Ingeniería de Requisitos.

“La Ingeniería de Requisitos trata con actividades en la cual intenta comprender las necesidades exactas de los usuarios del sistema software, para traducir tales necesidades en instrucciones precisas y no ambiguas las cuales podrían ser posteriormente utilizadas en el desarrollo del sistema”

[10]

La Ingeniería de Requisitos, disciplina de la Ingeniería de Software, es donde se identifica el propósito del sistema, dirección y alcance. Consiste en un conjunto de actividades y transformaciones que pretenden comprender las necesidades de un sistema software y convertir la declaración de estas necesidades en una descripción completa, precisa y documentada de los requerimientos del sistema siguiendo un determinado estándar. Los requisitos constituyen el enlace entre las necesidades reales de los clientes, usuarios y otros participantes vinculados al sistema.

En el proceso de desarrollo de un sistema, sea o no para la web, el equipo de desarrollo se enfrenta al problema de la identificación de requisitos. La definición de las necesidades del sistema es un proceso complejo, pues en él hay que identificar los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y de los clientes. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen. Por la complejidad que todo esto puede implicar, la ingeniería de requisitos ha trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa:

- Entrevistas: resultan una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural.
- JAD (Joint Application Development/Desarrollo conjunto de aplicaciones): esta técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes. Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que ve es lo que obtiene), es decir, durante la aplicación de la técnica se trabajará sobre lo que se generará.
- Tormenta de ideas: es también una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas.
- Casos de Uso: aunque inicialmente se desarrollaron como técnica para la definición de requisitos algunos autores proponen casos de uso como técnica para la captura de requisitos. Los casos de uso permiten mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema. Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una determinada función.
- Cuestionarios y Checklists: esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario (Checklist). Este cuestionario será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista.

Se debe tener en cuenta que la selección de las técnicas y el éxito de los resultados que se obtengan, depende en gran medida tanto del equipo de análisis y desarrollo, como de los propios clientes o usuarios que en ella participen.

El proceso de especificación de requisitos se puede dividir en tres grandes actividades:

1. captura de requisitos.
2. definición de requisitos.
3. validación de requisitos.

En la siguiente figura se presenta el proceso de ingeniería de requisitos que incluye estas tres actividades:

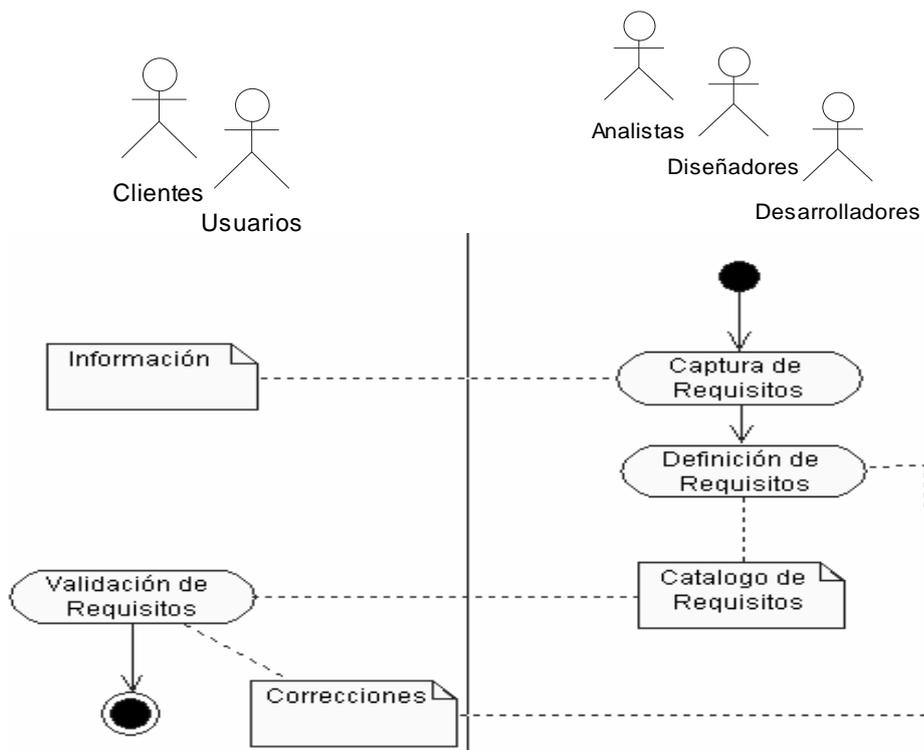


Fig. 1.2: Proceso de Ingeniería de Requisitos

El proceso comienza con la realización de la captura de requisitos, el grupo de analistas toma la información suministrada por los usuarios y clientes. Esta información puede provenir de fuentes muy diversas: documentos, aplicaciones existentes, a través de entrevistas, etc. En base a esta

información, el equipo de desarrollo elabora el catálogo de requisitos. Finalmente con la validación de requisitos se realiza la valoración de los mismos, comprobando si existen inconsistencias, errores o si faltan requisitos por definir. El proceso de definición-validación es iterativo y en algunos proyectos complejos resulta necesario ejecutarlo varias veces.

1.4 Gestión de Requisitos.

Los requisitos se inician cuando empieza un proyecto en las etapas de análisis y especificación de requisitos, posteriormente, dichos requisitos en el ciclo de vida de un proyecto pueden ser modificados por lo que se establece el concepto de Gestión de Requisitos, que es el tratamiento y control de las actualizaciones y cambios a los mismos.

La Gestión de Requisitos en Ingeniería de Sistemas, es el proceso encargado de la identificación, asignación y seguimiento de los requisitos, incluyendo el interfaz, verificación, modificación y control el estatus a lo largo del ciclo de vida. Es el conjunto de actividades que se concentra en el aseguramiento de las especificaciones, por ejemplo, los requisitos que son reunidos para la satisfacción del cliente. Es el proceso que inicia con la concepción de un proyecto y continúa hasta el resultado final del producto, después, ya no es necesario.

Debido a que un proyecto informático es susceptible de cambios, habría que proceder a su actualización o a la incorporación de nuevas funcionalidades o eliminar otras, esto obliga a mantener controlado y documentado el producto. Los cambios de requisitos deben ser gestionados para asegurar que la calidad de los mismos se mantenga, los problemas suscitados por los cambios de requisitos podrían incurrir en altos costos, siendo el requisito factor crítico de riesgo.

Tareas principales de la Gestión de Requisitos.

- **Recolección de requisitos:** es una actividad de comunicación iterativa entre clientes, gerentes y practicantes, para descubrir, definir, refinar y registrar una representación precisa de los requisitos del producto.
- **Documentación:** después que los requisitos han sido recolectados, hay que analizarlos a detalle y documentarlos en una especificación de requisitos. El resultado de la especificación de requisitos

y de cualquier especificación de requisitos de componentes hardware/software derivado sirve como registro de convenio con el cliente y compromiso con el proveedor.

- **Verificación:** una vez que la especificación de requisitos ha sido desarrollada, los requisitos son verificados. La verificación de requisitos es un proceso para asegurar que la especificación de requisito del producto es una representación exacta de las necesidades del cliente. Este proceso también asegura que los requisitos sean trazados y verificados a través de varias fases del ciclo de vida; particularmente en el diseño, implementación y pruebas.
- **Gestión de Cambios:** es un proceso formal para identificar, evaluar y reportar cambios propuestos y aprobados a la especificación del producto. Como el proyecto va evolucionando, los requerimientos pueden cambiar o expandirse para ajustar algunas modificaciones en el alcance o diseño del proyecto. Un proceso de gestión de cambios proporciona un rastreo completo y preciso de todos los cambios que son pertinentes al proyecto.

1.5 Tratamiento de Requisitos en propuestas para la Web.

El desarrollo de sistemas web agrupa una serie de características que lo hacen diferente del desarrollo de otros sistemas [11]. Por un lado, hay que tener en cuenta que roles muy diferentes de desarrolladores participan en el proceso: analistas, clientes, usuarios, diseñadores gráficos, expertos en multimedia y seguridad, etc. Por otro lado, la existencia en estos sistemas de una importante estructura de navegación obliga a un desarrollo preciso de este aspecto que le permita al usuario ir de una página a otra dentro del sistema con gran facilidad.

Algunas de estas características también hay que tenerlas en cuenta en la fase de especificación de requisitos:

- **Requisitos de datos,** también denominados requisitos de contenido, requisitos conceptuales o requisitos de almacenamiento de información. Éstos requisitos responden a la pregunta de qué información debe almacenar y administrar el sistema.
- **Requisitos de interfaz (al usuario),** también llamados en algunas propuestas requisitos de interacción o de usuario. Responden a la pregunta de cómo va a interactuar el usuario con el sistema.
- **Requisitos navegacionales,** recogen las necesidades de navegación del usuario.
- **Requisitos de personalización,** describen cómo debe adaptarse el sistema en función de qué usuario interactúe con él y de la descripción actual de dicho usuario.

- Requisitos no funcionales, son por ejemplo los requisitos de portabilidad, de usabilidad, de interfaz etc. [12]

1.6 Patrones de Caso de Uso.

Un patrón es un modelo que podemos seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos al tratar de lograr ciertos objetivos, estos capturan la experiencia existente y probada para promover buenas prácticas.

“Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma”.

Los patrones de casos de uso modelan reglas que hacen más comprensibles y correctos los modelos. Algunos de estos patrones y los que se seleccionaron a la hora de modelar los casos de uso fueron los siguientes:

- Extensión o Inclusión Concreta: Extensión: este patrón consiste en dos casos de uso y la relación de extensión entre ellos; es aplicable cuando un flujo puede extender el flujo de otro caso de uso. Tipo: Patrón de estructura.
- Extensión o Inclusión Concreta: Inclusión: este patrón consiste en la relación de incluido entre el caso de uso base y el caso de uso incluido. Es aplicable cuando un flujo puede incluir el flujo de otro caso de uso al igual que la realización del mismo. Tipo: Patrón de estructura.
- CRUD: Completo: Este patrón consiste en un caso de uso base llamado Información CRUD (o Administrador de información), el cual modela los distintos tipos de operaciones que pueden realizarse en un segmento de información de cierto tipo como: Crear, Eliminar, Actualizar, Modificar etc. Tipo: Patrón de estructura.
- Múltiples actores: Distintos Roles: este patrón consiste en un caso de uso y dos actores, es aplicable cuando dos actores interactúan con un mismo caso de uso pero con roles distintos e interacciones diferentes. Tipo: Patrón de estructura.
- Múltiples actores: Roles comunes: este patrón es un patrón alternativo en el que dos actores juegan el mismo rol en un caso de uso. Este patrón es aplicable cuando desde el punto de vista del caso de uso existe una entidad externa interactuando con cada instancia del caso de uso. Tipo: Patrón de estructura.[13]

1.7 Analista de Sistemas.

“Arqueólogo, innovador, mediador, líder del proyecto, investigador, consejero y diplomático.”

El analista de sistemas es más que un especificador, debe tener la habilidad: de comunicación aumentada, de mediar en los desacuerdos, necesita entender y apreciar los asuntos de los usuarios, mente lógica y organizada. [14]

El analista de sistemas generalmente valora la manera que funcionan los negocios examinando la entrada, el procesamiento de datos y la salida de información con el propósito de mejorar los procesos organizacionales. Muchas mejoras involucran mejor apoyo para las funciones de los negocios por medio del uso de sistemas de información computarizados. Esta definición enfatiza un enfoque sistemático y metódico para analizar, y posiblemente mejorar, lo que está sucediendo con el contexto específico creado por un negocio. Ellos poseen un amplio rango de habilidades. La primera y principal es que el analista soluciona problemas, le gusta el reto de analizar un problema y encontrar una respuesta funcional. Los analistas de sistemas requieren habilidades de comunicación que les permitan relacionarse en forma significativa con diferentes tipos de personas diariamente, así como habilidades de computación. Para su éxito es necesario que se involucre el usuario final.

Los analistas proceden sistemáticamente. El marco de referencia para su enfoque sistemático es proporcionado por lo que es llamado el ciclo de vida del desarrollo de sistemas (SDLC). Este puede ser dividido en siete fases secuenciales, aunque en realidad las fases están interrelacionadas y frecuentemente se llevan a cabo simultáneamente:

- Identificación de problemas.
- Oportunidades y objetivos.
- Determinación de los requerimientos de información.
- Análisis de las necesidades de sistemas.
- Diseño del sistema recomendado.
- Desarrollo y documentación del software.
- Prueba y mantenimiento del sistema e implementación del mismo.

Los paquetes de software basados en microcomputadora automatizado para el análisis y diseño de sistemas son llamados herramientas CASE. Las cuatro razones para la adopción de herramientas

CASE son: El incremento de la productividad del analista, la mejora de la comunicación entre analistas y usuarios, la integración de actividades del ciclo de vida y el análisis, la valoración del impacto de los cambios por mantenimiento.

1.8 Conclusiones.

En el mundo actual existe una gran diversidad de metodologías, herramientas y lenguajes de modelado, cada una con sus propiedades que las hacen más o menos adecuadas a la hora de desarrollar un proyecto, utilizar debidamente estas, depende en gran medida de los requisitos que debe cumplir el sistema a desarrollar, de los objetivos que se persigue, así como las necesidades del cliente.

Una vez que fueron estudiadas y analizadas las metodologías, herramientas y lenguajes de modelado a lo largo del presente capítulo, se definió que para la realización de este proyecto el uso de RUP como proceso de desarrollo por su capacidad organizativa, documentación generada y las dimensiones del proyecto, el ser una metodología iterativa e incremental permite definir mini proyectos, permitiendo presentar a nuestro cliente un avance del mismo dándole la posibilidad de evaluar los resultados parciales del trabajo. Como herramienta de análisis y modelado el Rational Rose y UML como lenguaje de modelación por su potencialidades y profesionalismo.

También podemos concluir diciendo que la ingeniería de requerimientos permite establecer y mantener un acuerdo en qué el sistema debe hacer, proporcionar al equipo de desarrollo un entendimiento de los requisitos, definir los límites del sistema, ofrecer una base para la estimación de los costes y cronogramas, definir las necesidades y metas de los usuarios, controlar cambios a los requisitos.

Capítulo

Modelo del Dominio y Requerimientos

En el siguiente capítulo se define: el planteamiento del objeto de estudio, objeto de automatización, propuesta de sistema, modelo de dominio, además se hará referencia a los requisitos funcionales y no funcionales los cuales nos darán una visión del futuro sistema, resultado de las entrevistas con el cliente.

2.1 Objeto de Estudio.

2.1.1 Problema.

En la Universidad de las Ciencias Informáticas se realiza la Copa Pascal como uno de los eventos docentes más importantes de cada curso, para la realización de la misma se han usado jueces online ubicados en Internet o la adaptación de un juez online descargado desde algunos de los sitios proveedores de esta herramienta.

Este concurso se efectúa una vez por curso, limitando a este margen de tiempo las posibilidades de competir y medir los conocimientos de una forma didáctica de todo aquel que participe; además incluye un solo estilo de competencia, donde los concursantes no pueden crear sus propias competencias y no tienen herramientas que le permita desarrollar otras habilidades.

Surge entonces la necesidad de realizar el análisis de una aplicación la cual no pretende sustituir al evento Copa Pascal sino que lo incluye como una funcionalidad más de entre las distintas formas de competencias que se definen, que permita la fusión y complementación con los contenidos docentes, que le permita al usuario seleccionar el estilo en el cual desea participar, crear su propia competencia y un ranking. El sistema debe estar provisto de varios estilos de competencias, de forma tal que sean cumplidas las expectativas del usuario y logrando por parte del equipo de desarrolladores un proyecto con éxito.

2.1.2 Objeto de Automatización.

Se requiere automatizar los procesos de creación tanto de competencias², retos³, pruebas online⁴, estilo libre⁵ y test⁶, de igual forma se calificará mediante jueces online⁷ todo los estilos en los cuales el usuario incursione que requiera una calificación automatizando el proceso enviar solución además de la activación del evento Copa Pascal cuando este se muestre habilitado.

2.1.3 Propuesta de Sistema.

Se propone un sistema que permita automatizar las funcionalidades descritas anteriormente. Debe permitir un fácil manejo, accesibilidad y tiempo de respuesta aceptable; a través del cual se pretende estimular la interacción de los usuarios con la aplicación y el desarrollo de sus habilidades y la consolidación de sus conocimientos.

² Crear competencia: permite al usuario registrado configurar una competencia, seleccionando cantidad de equipos, integrantes por equipo, fecha, hora y otros parámetros. **[Ver Glosario de Términos]**

³ Reto: el usuario registrado invita a otro usuario conectado a un reto, gana el que más rápido resuelva de forma correcta el problema definido por el sistema. **[Ver Glosario de Términos]**

⁴ Prueba Online: permite al usuario registrado definir un criterio (año(s), materia(s), tema(s)) para que se le genere un examen de cuarenta y cinco preguntas que deberá resolverse en treinta minutos. **[Ver Glosario de Términos]**

⁵ Estilo libre: el usuario descarga un problema y envía la solución al juez software especificando un lenguaje de programación. **[Ver Glosario de Términos]**

⁶ Test: los usuarios registrados podrán realizar cinco preguntas de un test mental. **[Ver Glosario de Términos]**

⁷ Juez Online: programa que recibe como entrada el fichero con el código fuente de un programa, el lenguaje en el cual fue implementado y realiza la calificación del mismo, en el presente trabajo existen 3 tipos de jueces: juez software, juez prueba, juez test. **[ver sección 3.1.1. Definición de los actores]**

2.2 Modelo del Dominio.

2.2.1 Descripción general del proceso.

Existe poco entendimiento entre las expectativas de los clientes y las ideas del equipo de desarrollo ya que el evento Copa Pascal, descrito anteriormente presenta varias deficiencias, entre ellas vale destacar que:

- Se realiza una vez por curso, a nivel de facultad y luego a nivel universitario, limitando a este tiempo las posibilidades de competir y medir los conocimientos de una forma didáctica.
- Jueces online ubicados en Internet o la adaptación de un juez online descargado desde algunos de los sitios proveedores de esta herramienta
- Los estudiantes que participan en este evento están limitados a: un solo estilo de competencia, no pueden crear sus propias competencias y no tienen herramientas que le permita desarrollar otras habilidades y que a la vez les consolide sus conocimientos docentes adquiridos.

El usuario necesita una aplicación que le ayude a potenciar el perfeccionamiento del pensamiento lógico, deductivo y otras habilidades, y que además consolide conocimientos docentes adquiridos de una forma sencilla y didáctica, dándole la posibilidad de seleccionar en que estilo de competencia desea participar, configurarlo y hasta crear su propia competencia.

Una vez descritos los problemas, se hace necesario realizar el análisis para posteriormente automatizar el sistema "Coliseo Virtual" el cual solucione estas dificultades, y se hace uso de la Web precisamente por la facilidad de múltiple acceso que facilita a todos sus usuarios.

Modelamiento:

El negocio es el conjunto de servicios que una entidad, organización o empresa brinda a un conjunto de clientes o usuarios con el propósito de satisfacer las necesidades de estos. Al emplear la metodología basada en RUP, el modelamiento del negocio plantea la identificación de los procesos del negocio y su completo análisis, el cual servirá como base para la identificación de probables candidatos a sistemas informáticos que soporten el negocio total o parcialmente. [15]

Dependiendo de la situación o escenario que se presente, se determina la alternativa adecuada para desarrollar este proceso si Modelo del negocio o Modelo del dominio.

En el presente trabajo se considero realizar un modelamiento del dominio o sea un acercamiento visual sobre los conceptos que se manejan, ya que el entorno en el que se enmarca el proyecto estará altamente centrado en las tecnologías informáticas por lo que no es necesario un modelo de negocio completo, considerándose estos los conceptos con los que interactúan los usuarios a la hora de ejercitar sus conocimientos y habilidades en la programación.

Los conceptos (objetos) relacionados con el entorno son:

- **Usuario:** cualquier tipo de persona que interactúe con la aplicación (estudiante, profesor).
- **Servidor de Aplicaciones Web:** elemento encargado de visualizar la aplicación Teleformación.
- **Evento Copa Pascal:** concurso de programación de la UCI.
- **Teleformación:** elemento que permitirá visualizar Conferencias, Clases Practicas, Ejercicios, Pruebas de programación.
- **Servidor ftp:** elemento encargado de visualizar la aplicación Ucistore.
- **Ucistore:** elemento encargado de visualizar y proveer documentación y herramientas suficientes a los usuarios para su estudio.
- **Herramientas:** programas que podrán ser utilizados por los usuarios en la consolidación de sus conocimientos y en el desempeño de sus habilidades de programación.
- **Biblioteca Virtual:** elemento encargado de visualizar y proveer documentación suficiente a los usuarios para su estudio.
- **Documentación:** toda la información recogida en forma de artículo, manual, libro, revista, vínculos, entre otras para su estudio.
- **Departamento de especialidad:** elemento encargado de visualizar todas las actividades de programación que se realizan en pos de apoyar al aprendizaje del usuario: conferencias, clase practicas, ejercicios, entre otras funcionalidades.
- **Servidor de Recursos:** elemento generalizador, visualizador de las tecnologías anteriormente descritas.

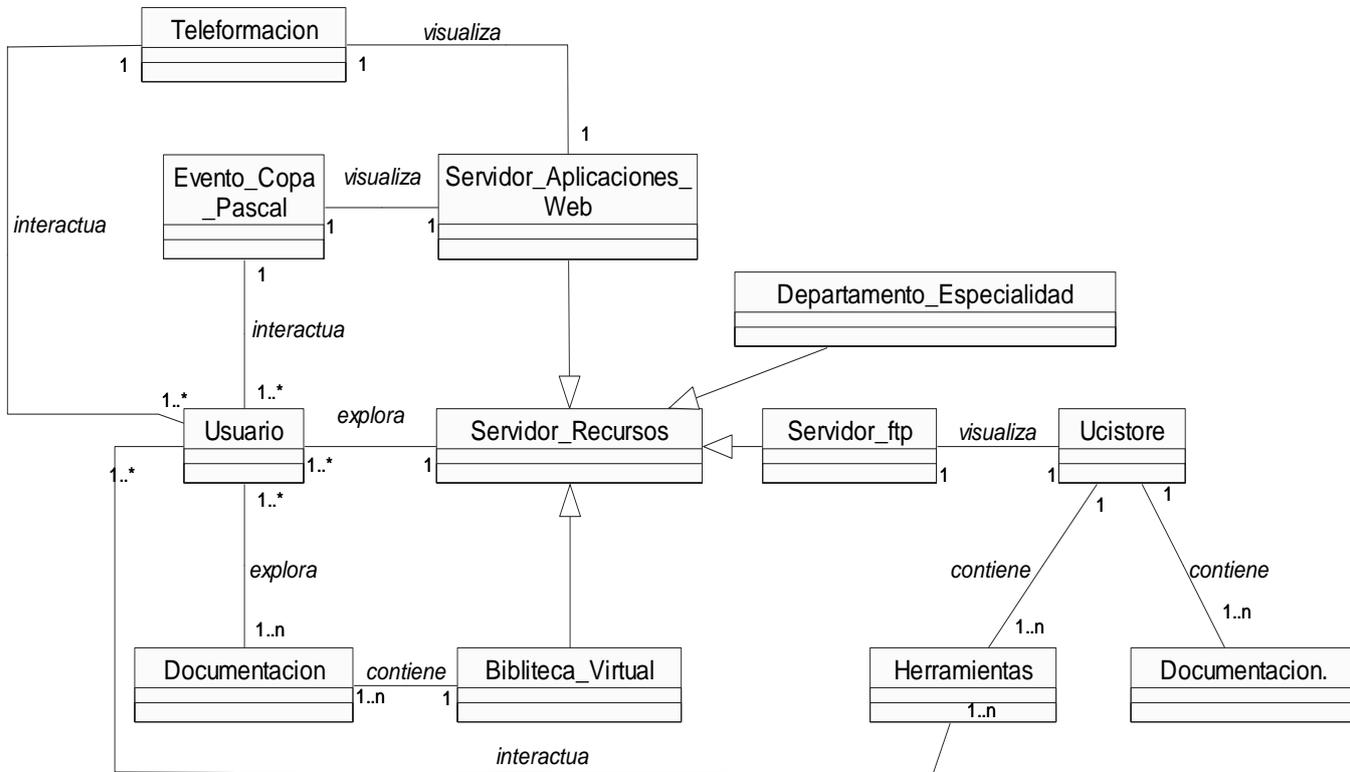


Fig. 2.1: Modelo de Dominio.

2.3 Especificaciones de los Requerimientos del Software

Requerimientos: Condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal. Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales. [16]

2.3.1 Requisitos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Los requerimientos funcionales de un sistema se corresponden con:

- Futuras opciones del software.
- Acciones ocultas del software.
- Condiciones extremas a determinar por el software.

Para darle cumplimiento a uno de los objetivos específicos, se plantean los siguientes requerimientos funcionales:

1. Registrar Usuario.
2. Autenticar Usuario.
3. Recuperar Contraseña.
4. Modificar Datos Usuario.
5. Gestionar Usuarios.
6. Gestionar Estilo Libre.
7. Gestionar Pruebas Online.
8. Gestionar Test.
9. Gestionar Retos.
10. Gestionar Crear Competencia.
11. Gestionar Concurso.
12. Gestionar Noticias.
13. Gestionar Cursos.
14. Gestionar Recursos.
15. Gestionar Link.
16. Gestionar Equipo.
17. Resolver Estilos Libres.
18. Resolver Pruebas Online.
19. Resolver Test.
20. Crear Reto.
21. Participar Reto.
22. Crear Competencia.
23. Modificar Datos Competencia.
24. Participar Competencia.
25. Enviar Solución.
26. Participar Concurso Copa Pascal.
27. Descargar Problemas Concurso.
28. Contactar Juez.
29. Mostrar Información. **[Ver sección 2.3 Especificaciones de Casos de uso]**

2.3.2. Requisitos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Los requerimientos no funcionales incluyen:

1. Conjunto de facilidades.
2. Capacidades.
3. Seguridad.

Requisitos de apariencia o interfaz externa.

El Producto tendrá una apariencia Profesional teniendo en cuenta el fin con el cual esta creada la universidad de potenciar el desarrollo de profesionales altamente capacitados. Debe ser ágil, muy legible y simple de usar.

Los conceptos que serán manejados, son conocidos por los usuarios para que les sea fácil su uso. La interfaz estará diseñada de modo tal que el usuario pueda tener en todo momento el control de la aplicación, lo que le permitirá ir de una página a otra dentro del sistema con gran facilidad.

Requisitos de Usabilidad.

El sistema podrá ser usado por cualquier tipo de usuario desde un estudiante hasta cualquier trabajador de la UCI que desee medir sus conocimientos y habilidades en la programación.

Requisitos de Rendimiento.

El sistema estará diseñado sobre la arquitectura cliente/servidor. Este requerirá de una capacidad de procesamiento alta para ejecutar algoritmos complejos a la hora de un usuario enviar la solución en cualquier modalidad. Se hace necesario que los tiempos de respuesta sean cortos tanto para el envío de respuesta como para la calificación.

La aplicación deberá estar disponible las 24 horas del día.

Requisitos de Soporte.

Se usarán dos servidores, uno para la base de datos y otro para el servidor de aplicaciones. Las pruebas del sistema se realizarán en la misma universidad, las mismas permitirán evaluar en la práctica las funcionalidades, errores, rendimiento y ventajas de este nuevo producto.

Requisitos de Portabilidad.

Una de las mayores ventajas que tendrá el Sistema es su portabilidad ya que al estar desarrollado sobre Web el mismo podrá correr en cualquier plataforma, Windows, Linux etc.

Requisitos de Seguridad.

No se permitirá la navegación por el sistema sin autorización, en esto juega un papel importante el método de autenticación. También es de suma importancia garantizar la integridad de los datos que se almacenen en el servidor. La información deberá estar disponible a los usuarios en todo momento, limitada solamente por las restricciones que estos tengan de acuerdo a los privilegios.

Requisitos Políticos culturales.

Nuestro producto brindará soporte en español, garantizando con esto su fácil aplicación por personas de habla hispana, aunque algunos de sus problemas en determinados estilos estarán en inglés con el objetivo de incentivar al usuario por el estudio del idioma inglés.

Requisitos Legales.

Este producto no violará ninguna Ley.

Requisitos de Confiabilidad.

Es importante que el sistema presente un mecanismo de respuesta rápida ante fallos, garantizando con esto una rápida protección y recuperación ante un problema dado.

Requisitos de Ayuda y Documentación en Línea.

Debe disponerse de una ayuda detallada sobre las principales opciones del sistema.

Requisitos de Software.

1. Sistema Operativo: Linux/Windows para el cliente.
2. Servidor: Windows 2003 Server, IIS, .Net Framework v2.0, SQLServer 2000.

Requisitos de Hardware.

1. Pentium II 256M RAM, preferiblemente 512M RAM, a 1GHz para el servidor.

2.4 Conclusiones.

Este Capítulo 2 ha constituido el paso más importante a la hora de realizar el sistema ya que la correcta especificación de requisitos determina en gran medida el éxito de un proyecto de desarrollo de software. Además que el modelo del dominio nos dirige hacia una mejor forma de entender el entorno en el que estamos trabajando, también esta dado por la determinación del objeto de estudio, de automatización, sistema propuesto.

Capítulo 
Sistema

Las temáticas de este capítulo están dadas por: la definición de los actores que interactuarán con el sistema, breves descripciones de los casos de uso que nos darán una retrospectiva sobre las opciones que brindará el sistema, se definirán los diagramas de casos de uso, se harán especificaciones de los mismos en los cuales se expondrán secuencias del flujo eventos.

3.1 Casos de uso del sistema "Coliseo Virtual".

3.1.1 Definición de los actores.

Un actor es una entidad externa al sistema que se modela y que puede interactuar con él; un ejemplo de actor podría ser un usuario o cualquier otro sistema. Los actores identificados en este proyecto, son los siguientes:

Actores del sistema	Justificación
Administrador del sistema	El administrador del sistema es el encargado de gestionar cada cambio (insertar, eliminar, actualizar, etc.) realizado sobre el sistema en cualquiera de sus funcionalidades (Retos, Competencias, Estilos Libres, Pruebas Online, Cursos, Usuarios, Noticias, Recursos, Links).
Colaborador nivel 1	Es el encargado de colaborar con el administrador en cuanto a la gestión de funcionalidades del sistema (Retos, Competencias, Estilos Libres, Pruebas Online, Cursos, Noticias, Recursos, Links)
Colaborador nivel 2	Es el encargado de gestionar cada cambio en los equipos que se registren para las distintas competencias. Puede realizar acciones de Insertar, Eliminar y Actualizar cada uno de ellos cuando sea necesario.
Usuario invitado	El usuario invitado solo tiene acceso a la información encontrada en

	la página principal y a alguna otra información básica.
Usuario registrado	El usuario registrado es el que puede acceder a cualquier funcionalidad en el sistema hasta donde lleguen sus privilegios, o sea, puede acceder a los distintos estilos de competencias o simplemente leer información y descargarla.
Juez Concurso	Permanece en contacto con los usuarios de manera online a través del Chat o por e-mail.
Juez Software	Sistema que realiza la calificación de las competencias, estilos libres, y concurso de los usuarios registrados y equipos
Juez Test	Sistema que realiza la calificación de los Test de los usuarios registrados
Juez Prueba	Sistema que realiza la calificación de las pruebas online de los usuarios registrados.

Tabla 3.1: Actores involucrados en el sistema

3.1.2 Descripción de casos de uso

En las tablas siguientes se hace referencia a los casos de uso del sistema, se enuncian, se describen y se referencian con los requerimientos funcionales planteados.

CU-1	Registrar Usuario
Actores	Usuario Invitado
Descripción: El usuario invitado desea registrarse en el sistema para poder acceder a los privilegios que les son asignados a los usuarios registrados para ello deberá completar algunos datos de interés (nombre, apellidos, universidad, facultad, correo, alias etc.) y de la operación que se puede realizar con ellas que viene dada por el CU extendido Recuperar Contraseña.	
Referencias	1,3 (extendido)

CU-3	Recuperar Contraseña
Actores	Usuario Registrado
Descripción:	

Este Caso de Uso es un Caso de Uso extendido del Caso de Uso Registrar Usuario, en el cual el usuario luego de haberse registrado puede recuperar su contraseña.

Referencias	3
--------------------	---

CU-2	Autenticar Usuario
-------------	--------------------

Actores	Usuario Invitado
----------------	------------------

Descripción:
El usuario solicita al sistema acceder a la aplicación, este verifica sus datos y da los permisos pertinentes para este tipo de actor.

Referencias	2, 4 (extendido).
--------------------	-------------------

CU-4	Modificar Datos Usuario
-------------	-------------------------

Actores	Usuario Registrado
----------------	--------------------

Descripción:
Este Caso de Uso es un Caso de Uso extendido del Caso de Uso Autenticar Usuario, en la cual el usuario después de registrarse podrá modificar los datos correspondientes a cuando se registro por primera vez.

Referencias	4.
--------------------	----

CU-5	Gestionar Usuario
-------------	-------------------

Actores	Administrador
----------------	---------------

Descripción:
El administrador es el encargado de realizar todas las operaciones correspondientes a los usuarios del sistema (Buscar, Insertar, Eliminar, Actualizar sus datos, Listar).

Referencias	5.
--------------------	----

CU-6	Gestionar Estilo Libre
-------------	------------------------

Actores	Administrador, Colaborador 1
----------------	------------------------------

Descripción:
El Administrador/ Colaborador 1 es el encargado de realizar todas las operaciones correspondientes a los problemas que se encuentran dentro de la Arena de competencias/Estilos Libres en el sistema (Buscar, Insertar, Eliminar, Actualizar, Listar).

Referencias	6.
--------------------	----

CU-7	Gestionar Pruebas Online
Actores	Administrador, Colaborador 1
<p>Descripción:</p> <p>El Administrador/ Colaborador 1 es el encargado de realizar todas las operaciones correspondientes a las pruebas online que se encuentren dentro de la Arena de competencias en el sistema (Buscar, Insertar, Eliminar, Actualizar, Listar). También será el encargado de Crear pruebas online en las que podrán realizar operaciones con las preguntas que componen estas pruebas (Buscar, Insertar, Eliminar, Actualizar, Listar).</p>	
Referencias	7.

CU-8	Gestionar Test
Actores	Administrador, Colaborador 1
<p>Descripción:</p> <p>El Administrador/ Colaborador 1 es el encargado de realizar todas las operaciones correspondientes a los test que se encuentren dentro de la Arena de competencias en el sistema (Buscar, Insertar, Eliminar, Actualizar, Listar).</p>	
Referencias	8.

CU-9	Gestionar Reto
Actores	Administrador, Colaborador 1
<p>Descripción:</p> <p>El Administrador/ Colaborador 1 es el encargado de realizar todas las operaciones correspondientes a los retos que se encuentren dentro de la Arena de competencias en el sistema (Buscar, Insertar, Eliminar, Actualizar, Listar).</p>	
Referencias	9.

CU-10	Gestionar Competencias
Actores	Administrador, Colaborador 1
<p>Descripción:</p> <p>El Administrador/ Colaborador 1 es el encargado de realizar todas las operaciones correspondientes a las competencias que se encuentren dentro de la Arena de competencias en el sistema (Buscar, Insertar, Eliminar, Actualizar, Listar).</p>	

Referencias	10.
--------------------	-----

CU-11	Gestionar Concursos
--------------	---------------------

Actores	Administrador, Colaborador 1
----------------	------------------------------

Descripción:
 El Administrador/Colaborador 1 es el encargado de realizar todas las operaciones correspondientes a los Concursos que se encuentren dentro de la Arena de competencias en el sistema (Buscar, Insertar, Eliminar, Actualizar, Listar).

Referencias	11.
--------------------	-----

CU-12	Gestionar Noticias
--------------	--------------------

Actores	Administrador, Colaborador 1
----------------	------------------------------

Descripción:
 El Administrador/Colaborador 1 es el encargado de realizar todas las operaciones correspondientes a las Noticias que se encuentran dentro de información en el sistema (Buscar, Insertar, Eliminar, Actualizar, Listar).

Referencias	12.
--------------------	-----

CU-13	Gestionar Cursos
--------------	------------------

Actores	Administrador, Colaborador 1
----------------	------------------------------

Descripción:
 El Administrador/Colaborador 1 es el encargado de realizar todas las operaciones correspondientes a los Cursos que se encuentran dentro de información en el sistema (Buscar, Insertar, Eliminar, Actualizar, Listar).

Referencias	13.
--------------------	-----

CU-14	Gestionar Recursos
--------------	--------------------

Actores	Administrador, Colaborador 1
----------------	------------------------------

Descripción:
 El Administrador/Colaborador 1 es el encargado de realizar todas las operaciones correspondientes a los Recursos que se encuentran dentro de información en el sistema (Buscar, Insertar, Eliminar, Actualizar, Listar).

Referencias	14.
--------------------	-----

CU-15	Gestionar Link
Actores	Administrador, Colaborador 1
<p>Descripción:</p> <p>El Administrador/Colaborador 1 es el encargado de realizar todas las operaciones correspondientes a los Links que se encuentran dentro de información en el sistema (Buscar, Insertar, Eliminar, Actualizar, Listar).</p>	
Referencias	15.

CU-16	Gestionar Equipo
Actores	Colaborador 2
<p>Descripción:</p> <p>El Colaborador 2 es el encargado de realizar todas las operaciones correspondientes a los equipos del sistema (Buscar, Insertar, Eliminar, Actualizar sus datos además Listar).</p>	
Referencias	16.

CU-17	Resolver Estilo Libre
Actores	Usuario Registrado, Juez Software
<p>Descripción:</p> <p>El usuario registrado accede a los problemas (estilo libre) que están online, y a la operación que se puede realizar con ellos dada por el CU incluido Enviar Solución.</p>	
Referencias	17,25 (incluido)

CU-18	Resolver Pruebas Online
Actores	Usuario Registrado, Juez Prueba
<p>Descripción:</p> <p>El usuario registrado podrá resolver cualquier prueba online que este activa en el sistema y enviar la solución al Juez Prueba el cual le dará una calificación.</p>	
Referencias	18,25 (incluido)

CU-19	Resolver Test
Actores	Usuario Registrado, Juez Test
<p>Descripción:</p> <p>El usuario registrado podrá resolver cualquier test que este activo en el sistema y la operación que</p>	

se puede realizar dado por el CU incluido Enviar Solución hacia el Juez Test que será el que la califique.	
Referencias	19,25 (incluido)

CU-20	Crear Reto
Actores	Usuario Registrado
Descripción: El usuario registrado podrá crear un reto individual invitando a otro usuario registrado a participar.	
Referencias	20

CU-21	Participar Invitación Reto
Actores	Usuario Registrado
Descripción: El usuario registrado participa en la invitación hecha por otro usuario registrado.	
Referencias	21

CU-22	Crear Competencia
Actores	Usuario Registrado
Descripción: El usuario registrado podrá crear su propia competencia la cual va a formar parte de las funcionalidades del sistema donde los usuarios registrados podrán participar y la operación que se puede realizar con ella dada por el CU extendido Modificar Datos Competencia.	
Referencias	22,23 (extendido)

CU-23	Modificar Datos Competencia
Actores	Usuario Registrado
Descripción: Este Caso de Uso es un Caso de Uso extendido del Caso de Uso Crear Competencia, en la cual el usuario modifica los datos entrados inicialmente cuando la creó.	
Referencias	23

CU-24	Participar Competencia
Actores	Usuario Registrado, Usuario Equipo, Juez Software

<p>Descripción:</p> <p>El usuario registrado o el equipo participan en las competencias disponibles lo que trae consigo la operación para con el caso de uso Participar Competencias del CU incluido Enviar Solución al Juez Software</p>	
Referencias	24,25 (incluido)

CU-25	Enviar Solución
Actores	Usuario Registrado
<p>Descripción:</p> <p>Este Caso de Uso es un Caso de Uso incluido de los Caso de Uso en los cual el usuario resuelve o participa, enviando la respuesta al juez correspondiente.</p>	
Referencias	25

CU-26	Participar Concurso
Actores	Usuario Equipo
<p>Descripción:</p> <p>El Usuario Equipo participa en el evento Copa Pascal cuando este se muestre activo y las operaciones que se pueden realizar dadas por el CU extendido Descargar Problemas de Concurso y por el CU incluido Enviar Solución hacia el Juez Software que será el que la califique.</p>	
Referencias	26, 25 (incluido), 27 (extendido)

CU-27	Descargar Problemas Concurso
Actores	Usuario Equipo
<p>Descripción:</p> <p>Este Caso de Uso es un Caso de Uso extendido del Caso de Uso Participar Concurso, en la cual el usuario participa en el evento Copa Pascal y puede descargar los problemas.</p>	
Referencias	27

CU-28	Contactar Juez
Actores	Usuario Equipo
<p>Descripción:</p> <p>El Usuario Equipo durante el evento Copa Pascal podrá estar en contacto con el juez en caso de duda u otra interrogante.</p>	

Referencias	28
CU-29	Mostrar Información
Actores	Usuario Invitado, Administrador, Colaborador 1, Colaborador 2, Usuario Registrado, Usuario Equipo
Descripción: Todos los usuarios, independientemente del tipo que sean, podrán revisar la información publicada en el sistema: Cursos, Recursos, Noticias, Historial, Link, Ranking.	
Referencias	29

Tabla 3.2: Descripciones de los Casos de Uso del Sistema.

3.1.3 Diagrama de casos de uso y paquetes.

Los casos de uso constituyen una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas.

Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar como reacciona una respuesta a eventos que se producen en el mismo.

Los paquetes de casos de uso son la forma de agrupar a estos últimos respondiendo a algún criterio, en el presente proyecto se hizo un encapsulamiento de los casos de uso por actor en paquetes debido a que se hace más comprensible a la hora de ser analizado como se muestra en las siguientes figuras:

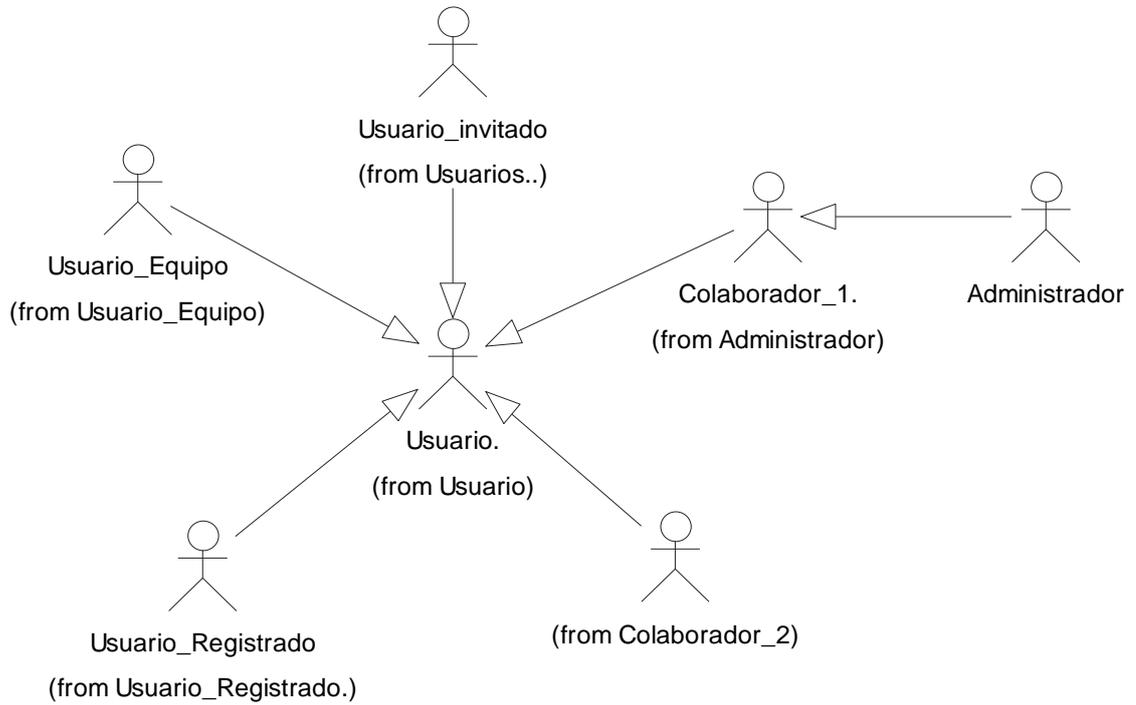


Fig. 3.1: Diagrama de Jerarquía de actores.



Fig. 3.2: Diagrama de casos de uso del Paquete Usuario Invitado.

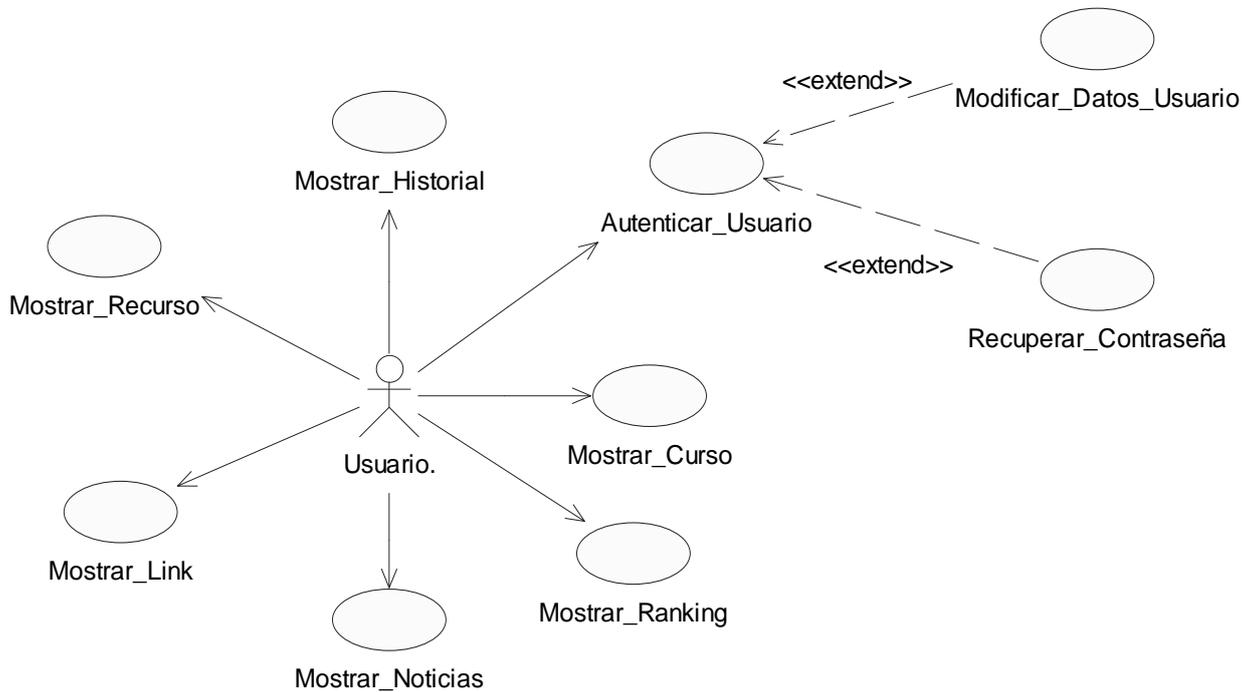


Fig. 3.3: Diagrama de casos de uso del Paquete Usuario.

En el diagrama de la figura 3.3 se hizo uso del patrón de caso de uso:

Extensión o Inclusión Concreta: Extensión.

Múltiples actores: Roles Comunes.



Fig. 3.4: Diagrama de casos de uso del Paquete Administrador.

En el diagrama de la figura 3.4 se utilizó el patrón CRUD.

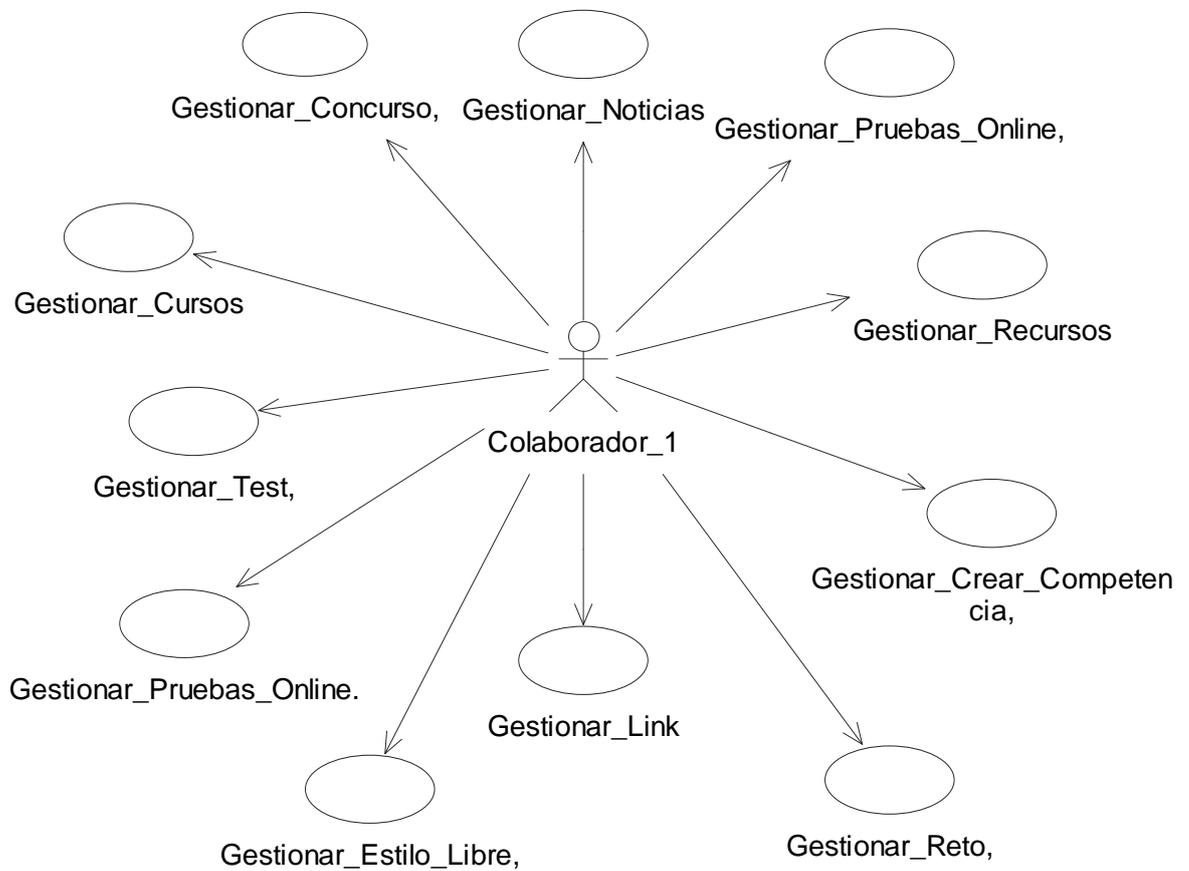


Fig. 3.5: Diagrama de casos de uso del Paquete Colaborador_1.

En el diagrama de la figura 3.5 se utilizo el patrón CRUD.



Fig. 3.6: Diagrama de casos de uso del Paquete Colaborador_2.

En el diagrama de la figura 3.6 se utilizó el patrón CRUD.

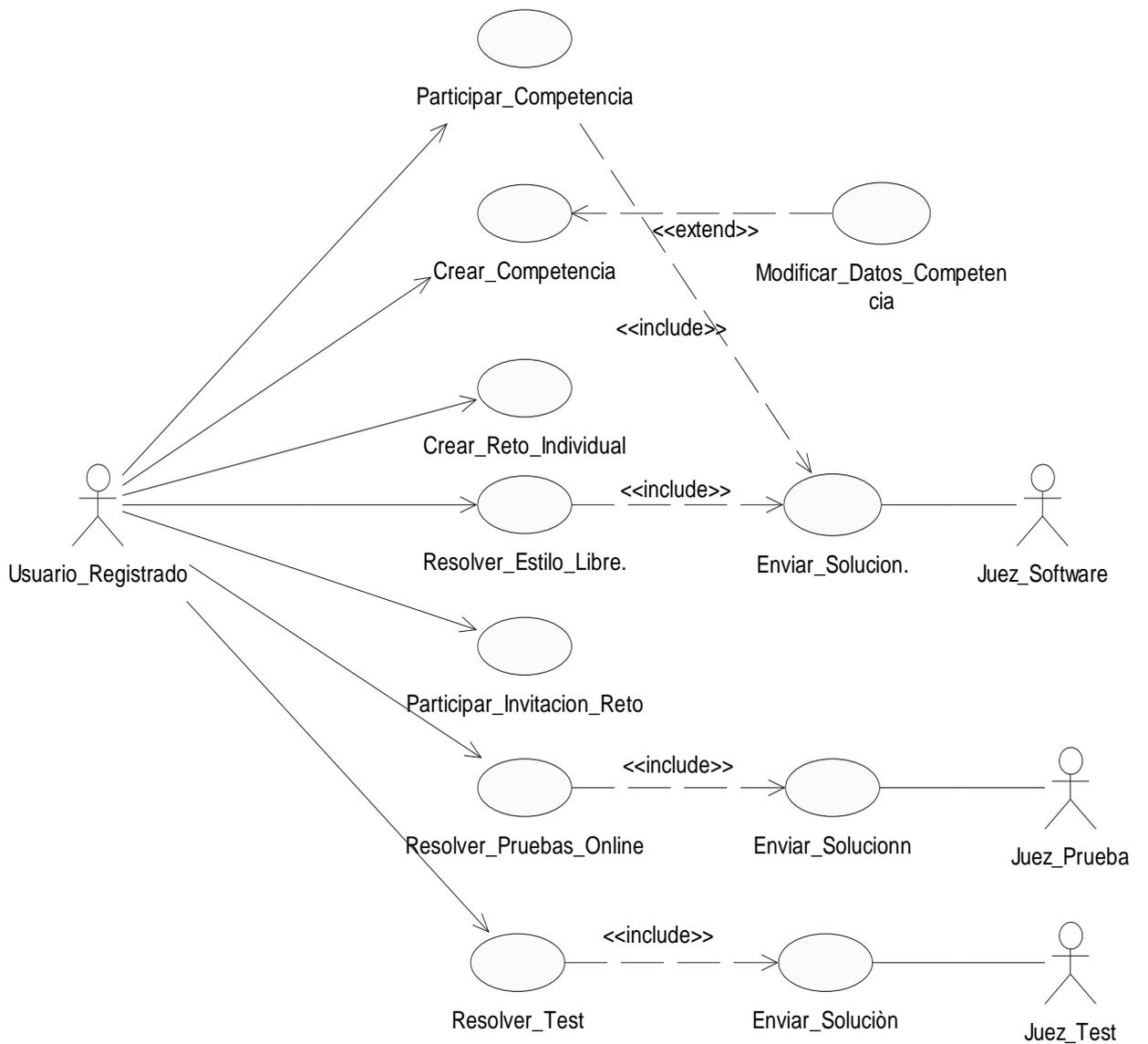


Fig. 3.7: Diagrama de casos de usos del Paquete Usuario Registrado.

En el diagrama de la figura 3.7 se utilizaron los patrones de casos de uso:

1. Extensión o Inclusión Concreta: Inclusión.

2. Extensión o Inclusión Concreta: Extensión.
2. Múltiples actores: Distintos Roles.

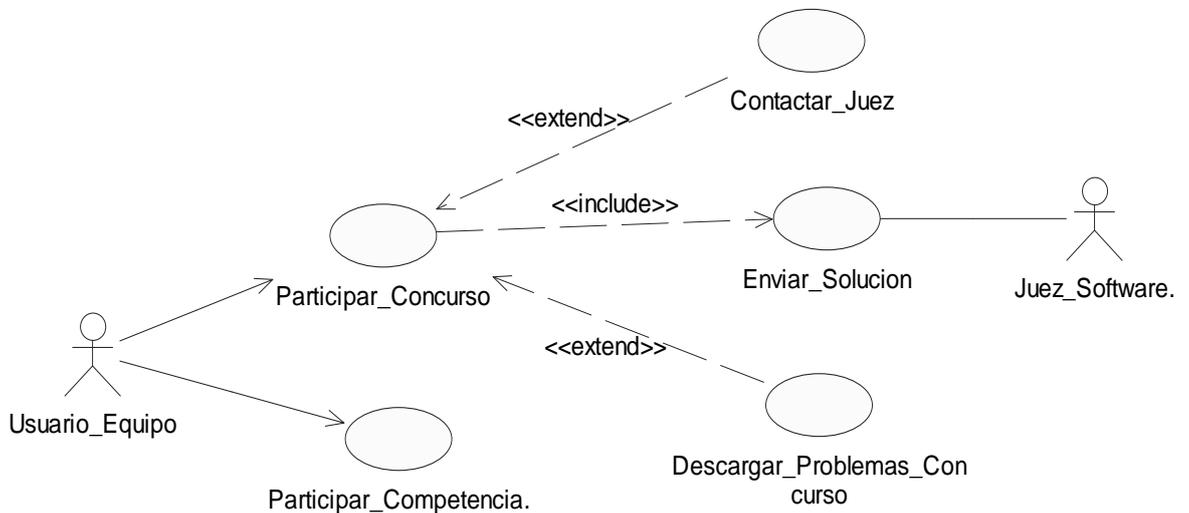


Fig. 3.8: Diagrama de casos de uso del Paquete Usuario Equipo.

En el diagrama de la figura 3.8 se utilizaron los patrones de caso de uso:

1. Extensión o Inclusión Concreta: Inclusión.
2. Extensión o Inclusión Concreta: Extensión.
3. Múltiples actores: Distintos Roles.

3.1.4 Casos de uso expandidos.

Mediante los casos de uso expandidos se describe paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema. A continuación se muestran las especificaciones de algunos de los casos de uso descritos en la sección 3.1.2. [Ver **Anexo 1**].

Caso de Uso	
CU 5	Gestionar Usuario.
Actores	Administrador.
Propósito	Gestionar usuario.
Resumen	El administrador es el encargado de realizar todas las operaciones

	correspondientes a los usuarios del sistema (Buscar, Insertar, Eliminar, Actualizar sus datos así como Listar).
Referencias	5.
Precondiciones	El Administrador del sistema debe estar autenticado.
Poscondiciones	Se buscan, listan, insertan, eliminan usuarios además de actualizar sus datos

Curso Normal de los Eventos

Acciones del Actor	Respuesta del Sistema
1. El Administrador del sistema necesita Insertar, Eliminar, Buscar, Listar y Actualizar los datos de un usuario.	2. El sistema ejecuta alguna de los siguientes escenarios: 2.1 Si decide registrar un usuario ir al escenario "Insertar". 2.2 Si decide eliminar un usuario ir al escenario "Eliminar". 2.3 Si decide listar mediante una búsqueda un grupo de usuarios ir al escenario "Listar". 2.4 Si decide buscar un usuario ir al escenario "Buscar". 2.5 Si decide actualizar los datos de un usuario ir a escenario "Actualizar".

Curso Normal de los Eventos

Escenario 1 Insertar

Acciones del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el administrador desea acceder en el menú Usuario a la interfaz Insertar.	2. El sistema visualiza la interfaz.
	3. El sistema muestra todos los datos correspondientes para insertar ese usuario en la base de datos.
4. El administrador entra los datos pertinentes: Nombre, Apellidos, Alias, Correo, Universidad, Facultad, Año etc. 4.1 Presiona el botón Insertar.	5. El sistema inserta el(los) usuario(s) en la Base de datos, culminando así el caso de uso.

Curso Alterno de los Eventos

Acciones del Actor	Respuesta del Sistema
4.2 El administrador presiona el botón Cancelar.	5.1. El sistema cancela la operación y cierra la interfaz, culminando así el caso de uso.
Escenario 2 Eliminar	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
6. El administrador selecciona Eliminar.	7. El sistema visualiza la interfaz
8. El administrador entra el(los) usuario(s) a eliminar. 8.1 Presiona el botón Eliminar.	9. El sistema elimina el(los) usuario(s) de la base de datos entrados por el administrador, culminando así el caso de uso.
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
6.1 El administrador decide en la misma interfaz de Eliminar, realizar una búsqueda del usuario a eliminar. 6.1.1 Presiona el botón Buscar.	7.1 El sistema visualiza la interfaz de la búsqueda.
8.2. El administrador entra los datos pertinentes a la búsqueda. 8.2.1 Presiona el botón Buscar.	9.1 El sistema muestra los resultados.
10. El administrador selecciona el(los) usuario(s) a eliminar. 10.1 Presiona el botón Eliminar.	11. El sistema los elimina de la base de datos, culminando así el caso de uso.
8.3 Presiona el botón Cancelar.	9.2 El sistema cancela la operación y cierra la interfaz, culminando así el caso de uso.
Escenario 3 Listar	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
12. El administrador selecciona Listar.	13. El sistema visualiza en la interfaz la opción de búsqueda para los usuarios que el administrador desea listar (por Nombre, por Apellidos, por Facultad etc.).
14. El administrador entra los datos pertinentes a la búsqueda.	15. El sistema lista los resultados, culminando así el caso de uso.

14.1. Presiona el botón Listar.	
Curso Alterno de los Eventos	
Acciones del Actor	Respuesta del Sistema
14.2. El administrador presiona el botón Cancelar.	15.1. El sistema cancela la operación y cierra la interfaz, culminando así el caso de uso.
Escenario 4 Actualizar Datos	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
16. El Administrador selecciona Actualizar Datos.	17. El sistema visualiza la interfaz.
18. El Administrador entra el usuario al cual desea actualizarle los datos.	19. El sistema visualiza una interfaz con los datos del usuario en cuestión.
20. El Administrador entra los datos a actualizar. 20.1. Presiona el botón Actualizar Datos.	21. El sistema actualiza los datos del usuario registrándolos en la base de datos, culminando así el caso de uso.
Curso Alterno de los Eventos	
Acciones del Actor	Respuesta del Sistema
16.1. El Administrador decide en la misma interfaz de Actualizar Datos, realizar una búsqueda del usuario del que va a actualizar sus datos. 16.1.1. Presiona el botón Búsqueda.	17.1. El sistema visualiza la interfaz de búsqueda.
18.2. El Administrador entra los datos pertinentes a la búsqueda: Nombre, Apellidos, Alias, Correo, Universidad, Facultad etc. 18.2.1 Presiona el botón Buscar.	19.1. El sistema muestra los resultados.
20.3. El Administrador selecciona el usuario al cual va a actualizarle los datos. 20.4 Presiona el botón Aceptar.	21.2. El sistema muestra los datos que pueden ser actualizados.
22. El Administrador entra los datos a actualizar. 22.1. Presiona el botón Actualizar Datos.	23. El sistema actualiza los datos del usuario registrándolos en la base de datos, culminando así el caso de uso.

20.2 El administrador presiona el botón Cancelar.	21.1. El sistema cancela la operación y cierra la interfaz, culminando así el caso de uso.
Escenario 5 Buscar	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
24. El Administrador solicita al sistema la interfaz Buscar.	25. El sistema visualiza la interfaz
26. El Administrador entra los datos del usuario a buscar. 26.1. Presiona el botón Buscar.	27. El sistema verifica que los datos entrados pertenezcan a algún usuario, lista los resultados y los muestra, culminando así el caso de uso.
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
	27.1. El sistema no encuentra resultados para esa búsqueda y emite un mensaje de error.
26.2. Presiona el botón Cancelar.	27.2. El sistema cancela la operación y cierra la interfaz, culminando así el caso de uso.
Prioridad	Critico.

Caso de Uso	
CU16	Gestionar Equipo.
Actores	Colaborador 2.
Propósito	Gestionar Equipo.
Resumen	El Colaborador 2 es el encargado de realizar todas las operaciones correspondientes a los equipos del sistema (Buscar, Insertar, Eliminar, Actualizar sus datos así como Listar).
Referencias	16.
Precondiciones	El Colaborador 2 debe estar autenticado.
Poscondiciones	Se buscan, insertan, eliminan equipos además de actualizar sus datos.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Colaborador 1 necesita Insertar, Eliminar, Buscar, Listar y Actualizar los datos de un equipo.	2. El sistema ejecuta alguna de los siguientes escenarios: 2.1. Si decide registrar un equipo ir al escenario.

	<p>“Insertar”.</p> <p>2.2 Si decide eliminar un equipo ir al escenario “Eliminar”.</p> <p>2.3 Si decide listar mediante una búsqueda un grupo de equipos ir al escenario “Listar”.</p> <p>2.4 Si decide buscar un equipo ir al escenario “Buscar”.</p> <p>2.5 Si decide actualizar los datos de un equipo ir a escenario “Actualizar”.</p>
Curso Normal de los Eventos	
Escenario 1 Insertar	
Acciones del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el Colaborador desea acceder en el menú Equipo a la interfaz Insertar.	2. El sistema visualiza la interfaz.
	3. El sistema muestra todos los datos correspondientes para insertar ese equipo en la base de datos.
4. El Colaborador entra los datos pertinentes: Nombre equipo, Líder, Integrantes, Correo, Universidad, Facultad, Curso etc. 4.1. Presiona el botón Insertar.	5. El sistema inserta el(los) equipo(s) en la Base de datos, culminando así el caso de uso
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
4.2. El Colaborador presiona el botón Cancelar.	5.1. El sistema cancela la operación y cierra la interfaz, culminando así el caso de uso.
Escenario 2 Eliminar	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
6. El Colaborador selecciona Eliminar.	7. El sistema visualiza la interfaz
8. El Colaborador entra el(los) equipo(s) a eliminar.	9. El sistema elimina el(los) equipo(s) de la base de datos entrados por el Colaborador, culminando así el

8.1. Presiona el botón Eliminar.	caso de uso.
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
6.1. El Colaborador decide en la misma interfaz de Eliminar, realizar una búsqueda del equipo a eliminar. 6.1.1. Presiona el botón Buscar.	7.1. El sistema visualiza la interfaz de la búsqueda
8.2. El Colaborador entra los datos pertinentes a la búsqueda. 8.2.1 Presiona el botón Buscar.	9.1. El sistema muestra los resultados.
10. El Colaborador selecciona el(los) equipo(s) a eliminar. 10.1. Presiona el botón Eliminar.	11. El sistema los elimina de la base de datos, culminando así el caso de uso.
8.3. Presiona el botón Cancelar.	9.2. El sistema cancela la operación y cierra la interfaz, culminando así el caso de uso.
Escenario 3 Listar	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
12. El Colaborador selecciona Listar.	13. El sistema visualiza en la interfaz la opción de búsqueda para los equipos que el Colaborador desea listar (por Integrantes, por Alias, por Facultad etc.).
14. El Colaborador entra los datos pertinentes a la búsqueda. 14.1. Presiona el botón Buscar.	15. El sistema lista los resultados, culminando así el caso de uso.
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
14.2. El Colaborador presiona el botón Cancelar.	15.1. El sistema cancela la operación y cierra la interfaz, culminando así el caso de uso.
Escenario 4 Actualizar Datos	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
16. El Colaborador selecciona Actualizar Datos.	17. El sistema visualiza en la interfaz

18. El Colaborador entra los datos del equipo a actualizar. 18.1. Presiona el botón Actualizar Datos.	19. El sistema actualiza los datos del equipo registrándolos en la base de datos, culminando así el caso de uso.
Curso Alterno de los Eventos	
Acciones del Actor	Respuesta del Sistema
16.1. El Colaborador decide en la misma interfaz de Actualizar Datos, realizar una búsqueda del equipo del que va a actualizar sus datos. 16.1.1. Presiona el botón Búsqueda.	17.1. el sistema visualiza la interfaz de búsqueda.
18.2. El Colaborador entra los datos pertinentes a la búsqueda: Alias, Integrantes, Facultad, etc. 18.2.1. Presiona el botón Buscar	19.1. El sistema muestra los resultados.
20. El Colaborador selecciona el equipo al cual va a actualizarle los datos. 20.1 Presiona el botón Aceptar.	21. El sistema muestra los datos que pueden ser actualizados.
22. El Administrador entra los datos del equipo a actualizar. 22.1. Presiona el botón Actualizar Datos.	23. El sistema actualiza los datos del equipo registrándolos en la base de datos, culminando así el caso de uso.
18.3. El Colaborador presiona el botón Cancelar.	19.2. El sistema cancela la operación y cierra la interfaz, culminando así el caso de uso.
Escenario 5 Buscar	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
24. El Colaborador solicita al sistema la interfaz Buscar.	25. El sistema visualiza la interfaz.
26. El Administrador entra los datos del equipo a buscar. 26.1. Presiona el botón Buscar.	27. El sistema verifica que los datos entrados pertenezcan a algún equipo, lista los resultados y los muestra, culminando así el caso de uso.
Curso Alterno de los Eventos	
Acciones del Actor	Respuesta del Sistema

	27.1. El sistema no encuentra resultados para esa búsqueda y emite un mensaje de error.
26.2. Presiona el botón Cancelar.	27.2. El sistema cancela la operación y cierra la interfaz, culminando así el caso de uso.
Prioridad	Critico.

Caso de uso	
CU 17	Resolver Estilo Libre.
Actores	Usuario Registrado, Juez Software.
Propósito	Resolver Estilo Libre.
Resumen: El usuario registrado accede a los problemas (estilo libre) que están online, y a la operación que se puede realizar con ellos dada por el CU incluido Enviar Solución.	
Referencias	17, 25 (incluido).
Precondiciones	El usuario debe estar registrado.
Poscondiciones	El usuario participa en un estilo libre.
Curso Normal de los Eventos	
Acciones del actor	Respuesta del Sistema.
1. El usuario solicita al sistema en el menú Arena de Competencias, Estilo Libre.	2. El sistema muestra la interfaz deseada.
3. El usuario selecciona el problema que desea resolver.	4. El sistema muestra una interfaz con el problema seleccionado.
5. El usuario accede al problema y plantea la solución. 5.1. Presiona el botón Enviar Solución.	6. El sistema envía la solución al Juez Software finalizando así el caso de uso.
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
3.1. El usuario presiona el botón Cancelar.	4.1. El sistema cancela la operación y cierra la interfaz.
Prioridad	Critico.

Caso de uso	
CU 18	Resolver Pruebas Online.
Actores	Usuario Registrado, Juez Prueba.

Propósito	Resolver Pruebas.
Resumen:	El usuario registrado podrá resolver cualquier prueba online que este activa en el sistema y enviar la solución al Juez Prueba el cual le dará una calificación.
Referencias	18,25 (incluido).
Precondiciones	El usuario debe estar registrado.
Poscondiciones	El usuario ejercita sus conocimientos a través de las pruebas online.
Curso Normal de los Eventos	
Acciones del actor	Respuesta del Sistema.
1. El usuario solicita al sistema en el menú Arena de Competencias, Pruebas Online.	2. El sistema muestra la interfaz correspondiente.
3. El usuario configura el criterio de selección de la Prueba Online.	4. El sistema muestra la prueba generada aleatoriamente por el según el criterio del usuario.
4. El usuario plantea la solución. 4.1 Presiona el botón Enviar Solución.	5. El sistema envía la solución al Juez Prueba finalizando así el caso de uso.
Curso Alterno de los Eventos	
Acciones del Actor	Respuesta del Sistema
4.2. El usuario presiona el botón Cancelar.	5.1. El sistema cancela la operación y cierra la interfaz.
Prioridad	Critico.

Caso de uso	
CU 19	Resolver Test.
Actores	Usuario Registrado, Juez Test.
Propósito	Resolver Test.
Resumen:	El usuario registrado podrá resolver cualquier test que este activo en el sistema y la operación que se puede realizar dado por el CU incluido Enviar Solución hacia el Juez Test que será el que la califique.
Referencias	19, 25 (incluido).
Precondiciones	El usuario debe estar registrado.
Poscondiciones	El usuario ejercita sus conocimientos a través de los Test.
Curso Normal de los Eventos	
Acciones del actor	Respuesta del Sistema.

1. El usuario solicita al sistema en el menú Arena de Competencias, Test.	2. El sistema muestra la interfaz (con el Test) solicitada por el usuario.
3. El usuario plantea la solución del test. 3.1. Presiona el botón Enviar Solución.	4. El sistema envía la solución al Juez Test finalizando así el caso de uso.
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
3.2. El usuario presiona el botón Cancelar.	4.1. El sistema cancela la operación y cierra la interfaz.
Prioridad	Critico.

Caso de uso	
CU 20	Crear Reto.
Actores	Usuario Registrado.
Propósito	Crear Reto.
Resumen: El usuario registrado podrá crear un reto individual invitando a otro usuario registrado a participar.	
Referencias	20.
Precondiciones	El usuario debe estar registrado.
Poscondiciones	El usuario crea su propio reto.
Curso Normal de los Eventos	
Acciones del actor	Respuesta del Sistema.
1. El usuario solicita al sistema en el menú Arena de Competencias, Retos.	2. El sistema muestra la interfaz deseada.
3. El usuario registrado selecciona el usuario el cual desea retar de la lista de usuarios conectados en su misma interfaz. 3.1. Presiona el botón Aceptar.	4. El sistema envía un mensaje de invitación al oponente culminando así el caso de uso (Ver CU 21 Participar Invitación Reto).
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
3.2. El usuario presiona el botón Cancelar.	4.1. El sistema cierra la interfaz.
Prioridad	Critico.

Caso de uso	
CU 21	Participar Invitación Reto.
Actores	Usuario Registrado.
Propósito	Participar Invitación Reto.
Resumen: El usuario registrado participa en la invitación hecha por otro usuario registrado.	
Referencias	21.
Precondiciones	El usuario debe estar registrado.
Poscondiciones	El usuario mide sus conocimientos a través de un reto.
Curso Normal de los Eventos	
Acciones del actor	Respuesta del Sistema.
1. El usuario accede a la invitación: Invitación Reto. 1.1. El usuario presiona el botón Aceptar forma inmediata.	2. El sistema muestra la interfaz del reto donde estarán el usuario y el que lo reta con un problema definido por el sistema.
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
1.1. El usuario presiona el botón Postergar reto y entra la fecha en la cual desea realizar el reto. 1.2. El usuario presiona el botón Cancelar de la invitación.	2.1. El sistema envía un correo al usuario que lo retó con estos datos y registra en Actividades Pendientes la invitación del reto del usuario retado, culminando así el caso de uso. 2.2 El sistema cierra la interfaz.
Prioridad	Critico.

Caso de uso	
CU 22	Crear Competencia.
Actores	Usuario Registrado.
Propósito	Crear Competencia.
Resumen: El usuario registrado podrá crear su propia competencia la cual va a formar parte de las funcionalidades del sistema donde los usuarios registrados podrán participar y la operación que se puede realizar con ella dada por el CU extendido Modificar Datos Competencia.	
Referencias	22,23 (extendido).
Precondiciones	El usuario debe estar registrado.
Poscondiciones	El usuario crea una competencia en la que otros usuarios pueden

	participar.
Curso Normal de los Eventos	
Acciones del actor	Respuesta del Sistema.
1. El usuario solicita al sistema en el menú Arena de Competencias, Crear Competencia.	2. El sistema muestra la interfaz deseada.
3. El usuario entra los datos para crear la nueva competencia (Nombre, Fecha, Hora inicio, Hora fin, Tiempo duración, Cantidad de problemas, Cantidad de equipos, Integrantes de los equipos, Contraseña de los equipos). 3.1. Presiona el botón Aceptar.	4. El sistema crea la competencia y la registra culminando así el caso de uso.
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
3.2. El usuario presiona el botón Cancelar.	4.1. El sistema cancela la operación y cierra la interfaz.
Prioridad	Critico.

Caso de uso	
CU 24	Participar Competencias.
Actores	Usuario Registrado, Usuario Equipo, Juez Software.
Propósito	Participar Competencias.
Resumen: El usuario registrado o el equipo participan en las competencias disponibles lo que trae consigo la operación para con el caso de uso Participar Competencias del CU incluido Enviar Solución al Juez Software.	
Referencias	24,25 (incluido).
Precondiciones	El usuario debe estar registrado.
Poscondiciones	El usuario participa en las competencias.
Curso Normal de los Eventos	
Acciones del actor	Respuesta del Sistema.
1. El usuario solicita al sistema en el menú Arena de Competencias, Participar Competencia.	2. El sistema muestra la interfaz deseada.
3. El usuario selecciona en Actividades Pendientes la competencia en la que desea	4. El sistema le pide usuario y contraseña para acceder a la competencia.

incursionar.	4.1. El sistema muestra una interfaz con la competencia seleccionada.
5. El usuario accede a la competencia y plantea la solución. 5.1. Presiona el botón Enviar Solución.	6. El sistema envía la solución al Juez Software finalizando así el caso de uso.
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
	4.2. El sistema no encuentra coincidencia con el usuario entrado o la contraseña entrada y cancela la operación.

Caso de uso	
CU25	Enviar Solución.
Actores	Usuario Registrado, Usuario Equipo, Juez Software, Juez Test, Juez Prueba.
Propósito	Enviar Solución.
Resumen: Este Caso de Uso es un Caso de Uso incluido de los Caso de Uso en los cual el usuario resuelve o participa, enviando la respuesta al juez correspondiente.	
Referencias	25.
Precondiciones	El usuario o equipo debe estar registrado.
Poscondiciones	El usuario es calificado.
Curso Normal de los Eventos	
Acciones del actor	Respuesta del Sistema.
1. El usuario adjunta el fichero con la solución del problema. 1.1. Presiona el botón Enviar Solución.	2. El sistema envía la solución del usuario al Juez el cual la califica. 2.1. El sistema envía automáticamente un correo al usuario con la calificación culminando así el caso de uso.
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
1.1. El usuario presiona el botón Cancelar.	2.2. El sistema cancela la operación y cierra la interfaz.

Caso de uso	
CU 26	Participar Concurso.
Actores	Usuario Equipo, Juez Software
Propósito	Participar Concurso
Resumen: El Usuario Equipo participa en el evento Copa Pascal y las operaciones que se pueden realizar dadas por los CU incluidos Descargar Problemas Concurso y Enviar Solución hacia el Juez Software que será el que la califique.	
Referencias	26, 27 (incluido).
Precondiciones	El equipo debe estar registrado
Poscondiciones	El equipo participa en el concurso
Curso Normal de los Eventos	
Acciones del actor	Respuesta del Sistema.
1. El Usuario Equipo solicita al sistema Participar Concurso.	2. El sistema muestra la interfaz deseada.
3. El Usuario Equipo accede a los problemas que le son asignados y plantea la solución. 3.1. Presiona el botón Enviar Solución.	4. El sistema envía la solución al Juez Software finalizando así el caso de uso.
Curso Alterno de los Eventos	
Acciones del Actor	Respuesta del Sistema
3.2. El Usuario Equipo presiona el botón Cancelar.	4.1. El sistema cancela la operación y cierra la interfaz.

Caso de uso	
CU28	Contactar Juez.
Actores	Usuario Equipo.
Propósito	Contactar Juez.
Resumen: El Usuario Equipo durante el evento Copa Pascal podrá estar en contacto con el juez en caso de duda u otra interrogante.	
Referencias	28.
Precondiciones	El usuario debe estar registrado.
Poscondiciones	El usuario contacta al juez.
Curso Normal de los Eventos	

Acciones del actor	Respuesta del Sistema.
1. El usuario solicita al sistema Contactar Juez.	2. El sistema muestra la interfaz deseada.
3. El usuario accede al Chat con el Juez. 3.1. El usuario presiona el botón Enviar mensaje.	
Curso Alternativo de los Eventos	
Acciones del Actor	Respuesta del Sistema
3.2. El usuario presiona el botón Cerrar.	4. El sistema cancela la operación y cierra la interfaz.
Prioridad	Secundario

Tabla 3.3: Especificaciones de los Casos de uso.

3.2 Análisis del Resultado.

Como resultado del análisis se generan especificaciones, modelos, que son producto de las interacciones con los usuarios que expresan ideas, decisiones, estrategias, procesos, reglas y relaciones, que constituyen información sobre el modo que la información nutrirá a la organización.

La comunicación entre usuarios y el equipo de desarrollo es un aspecto prioritario para el desarrollo de proyectos de software, donde el objetivo principal es satisfacer las necesidades de los primeros. En el presente proyecto esta comunicación ha constituido un proceso clave a lo largo de su vida de desarrollo, en el cual se han concertado entrevistas donde ambas partes involucradas trabajan juntos para identificar el problema, proponer soluciones, negociar diferentes enfoques y especificar un conjunto preliminar de requisitos de la solución hasta llegar al punto de éxito del producto, calidad.

La primera reunión estuvo principalmente enfocada para establecer el entorno en el cual se desarrollaba el negocio en este caso sería ver como los usuarios en su ambiente docente pueden obtener diferentes habilidades en la programación a través de clases prácticas, codificando, haciendo algoritmos en los distintos lenguajes, también en la Copa Pascal cuando esta se muestra activa en su período de competencia. Como resultado de esta se obtuvo el modelo de dominio (Fig. 2.1) en lo que los usuarios estuvieron 90% de acuerdo el otro 10% se fue aclarando mientras transcurría la entrevista. Luego se realizó la captura de requisitos como se muestra en la sección 2.3.1, en los que se concertaron los puntos principales: ¿Por qué se desarrolla el sistema?, ¿Cuáles son los objetivos a

cumplir?, ¿Quiénes serán los usuarios del sistema y cuales son sus objetivos?, ¿Cuáles deberán ser los privilegios de estos usuarios?, ¿Cuánto tiempo debe suponerles a los usuarios completar determinadas tareas del sistema?, entre otros. Durante la entrevista se fueron creando escenarios de cómo quedaría el sistema para ayudar a los usuarios a identificar mejor los requisitos fundamentales.

El resultado alcanzado como consecuencia de la identificación de requisitos estuvo sujeto a un procedimiento iterativo donde se eliminaron requisitos, se combinaron y/o se modificaron hasta conseguir satisfacer los objetivos planteados por los clientes los cuales estuvieron de acuerdo un 85%, el otro 15% se acordó creando otro conjunto de escenarios (en forma de casos de uso) que identificaban una línea de utilización para el sistema producto, al crear un acuerdo del 100% en cuanto a lo que el sistema debía hacer se realizaron esbozos de prototipos no funcionales de las páginas de autenticación del sistema (prototipos de interfaz de usuario) permitiendo definir la estructura visual de las pantallas, los registros de entrada y salida.

También se realizó un análisis de resultado basado en la opinión del equipo de desarrollo a la hora de desarrollar el producto encontraron base sólida en los requisitos capturados y los artefactos creados; también para ratificar la calidad del trabajo realizado se tuvo en cuenta un modelo de métricas el cual tiene por objetivo medir la calidad de los productos intermedios generados en un proyecto de software, este modelo define cuatro atributos genéricos de propiedades de calidad: consistencia, correctitud, completitud y complejidad:

Atributo	Nivel Conceptual	Nivel Especificación
Completitud	<p>Factor 1. ¿Han sido involucradas todas las áreas funcionales relevantes a las cuales apoyará el sistema?</p> <p>Factor 2. ¿Han sido definidos todos los roles relevantes de usuario encargados de generar/modificar o consultar información?</p> <p>Factor 3. ¿Han sido definidos todos los roles secundarios de usuario encargados de generar/modificar o consultar</p>	<p>Factor 8. ¿Están todas las acciones del flujo de eventos redactadas en función del responsable?</p> <p>Factor 9. ¿Se describen las condiciones de excepción relevantes que debe contemplar cada flujo de eventos?</p>

	<p>información?</p> <p>Factor 4. ¿Han sido considerados todos los sistemas externos con los cuáles interactuará el sistema?</p> <p>Factor 5. ¿Se presenta una descripción resumida de todos los conceptos del dominio?</p> <p>Factor 6. ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?</p> <p>Factor 7. ¿Existen requisitos que no han sido considerados en algún caso de uso?</p>	
Consistencia	<p>Factor 10. ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?</p> <p>Factor 11. ¿Representa el caso de uso una interacción observable por un actor?</p> <p>Factor 12. ¿No existe solapamiento en la funcionalidad que representan los diferentes casos de uso?</p> <p>Factor 13. ¿Existen acciones en el flujo de eventos asignadas a un responsable que no le corresponde?</p>	<p>Factor 14. ¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos?</p> <p>Factor 15. ¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o por una condición interna del sistema claramente identificable?</p> <p>Factor 16. Si en el caso de uso interviene mas de un actor, ¿existe claridad en cuál de ellos es el actor iniciador?</p> <p>Factor 17. ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados?</p>
Correctitud	Factor 18. ¿Representa el caso de uso	Factor 19. ¿Las interacciones definidas

	requisitos comprensibles por el usuario?	describen la funcionalidad requerida del sistema? Factor 20. ¿Las interacciones definidas introducen mejoras al proceso actual?
Complejidad	Factor 21. ¿En sistemas relativamente grandes se ha realizado una agrupación de los casos de uso en paquetes? Factor 22. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?	

Tabla 3.4: Factores de Calidad.

De lo cual se obtuvieron los siguientes resultados:

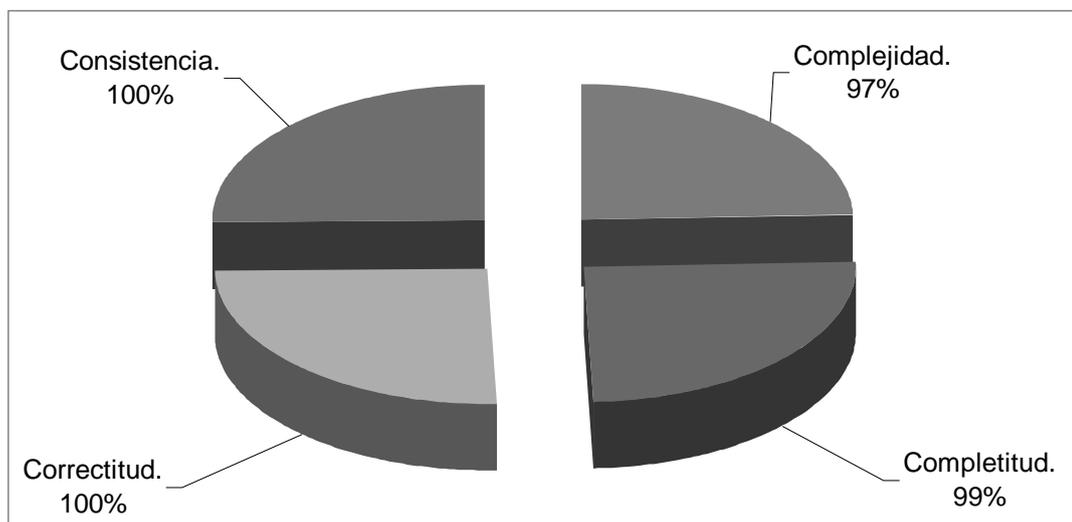


Fig. 3.9: Resultados obtenidos según los Factores de Calidad aplicados.

De acuerdo a los factores de calidad aplicados se pudieron percibir los errores cometidos: en la completitud a la hora de definir los roles secundarios que manejarían información así como en la captura de requisitos, se habían omitido algunas funcionalidades que tendría el sistema, en cuanto a la

consistencia no se tuvo errores ya que la descripción de los casos de uso estuvieron bien descritos, sin solapamientos, en un lenguaje natural, el nombre dado a estos es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario, existe también una adecuada separación en los flujos de eventos; en cuanto a correctitud tampoco existieron inconsistencias ya que todos los casos de uso representan requisitos comprensibles para el usuario, además las interacciones definidas describen la funcionalidad requerida del sistema, por otra parte se cometieron faltas en cuanto a la complejidad a la hora de agrupar los casos de uso por actores, también los elementos de los diagramas (casos de uso) algunos no estaban bien distribuidos, todos estos resultados sirvieron de apoyo y guía para corregir todos los errores cometidos y así lograr que a la hora de pasar a la fase de diseño no se cometieran desacuerdos sobre lo que el cliente quería realmente.

3.3 Conclusiones.

En el presente capítulo se ha establecido el primer paso en lenguaje propio de informáticos pues a partir de los casos de uso se realiza una descripción escrita del comportamiento del sistema al afrontar los requisitos, dado que es un mecanismo de organización. Un conjunto de casos de usos coherentes y consistentes, promueve una imagen fácil del comportamiento del sistema, un entendimiento común entre el cliente y el equipo de desarrollo. Además en el capítulo se aborda acerca de los resultados obtenidos a lo largo de la labor del analista apoyado por un sistema de factores.

CONCLUSIONES GENERALES.

A lo largo del presente trabajo se arribó a las siguientes conclusiones:

- Se realizó el modelo del dominio del sistema "Coliseo Virtual" mediante el cual se relacionaron todos los conceptos definidos por el usuario.
- Se realizó la captura de requisitos funcionales a través de los cuales se satisfacen las necesidades y expectativas del cliente. De igual forma se especificaron los requisitos no funcionales de acuerdo a las condiciones en las que se define el marco de estudio de este trabajo.
- Se definieron los actores y trabajadores del sistema, y los modelos de los casos de usos, estableciendo las interacciones entre estos; además se realizaron las especificaciones pertinentes mediante el flujo de eventos de cada caso de uso.
- Se realizó el análisis del sistema "Coliseo Virtual", a partir del cual los desarrolladores contarán con una base sólida para la continuación del ciclo de desarrollo del software.
- La propuesta del modelo de análisis realizado posibilitará que al ser desarrollado el sistema "Coliseo Virtual", los usuarios finales contarán con una herramienta que les permitirá:
 - Interactuar con el sistema en cualquier espacio de tiempo.
 - Consultar artículos, cursos, noticias, crear sus propias competencias, realizar test, realizar pruebas online y definir retos.
 - Consolidar y ejercitar los conocimientos y habilidades adquiridos a través del envío de soluciones, a los distintos jueces, de los problemas propuestos.
 - Generar pruebas online especificado un criterio determinado: año(s), materia(s), tema(s).

RECOMENDACIONES.

Se recomienda:

- Terminada la primera fase del ciclo de vida del software y generados todos los artefactos (análisis) se continúe hasta la fase final (implementación y prueba).
- Incluir en modelo de análisis las propuestas de otros estilos de competencias como el completamiento de algoritmos, dados algoritmos arribar a resultados.
- El estudio de otras herramientas tales como: RequisitePro, Microsoft Visio.
- Se recomienda que una vez desplegado el producto se extienda no solo a la universidad sino a otros centros de estudios.

REFERENCIAS BIBLIOGRÁFICAS.

- [1]. Cota, A.” Ingeniería de ría del Software .Un Enfoque Práctico”. Félix Varela. (2005).600.
- [3]. “Metodología (Ingeniería de Software)”. [Disponible en [http:// www.aibarra.org/investig/](http://www.aibarra.org/investig/)].
- [4]. “Proceso Unificado de Rational”. [Disponible en: <http://www.spinec.org/?p=48>.]
- [5]. “Proceso Unificado de Rational”. [Disponible en:
<https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducción%20a%20RUP.doc>.]
- [6]. “Métrica 3”. (2005). [Disponible en: <http://www.webintenta.com/metrica-3.html>.]
- [7]. Leonardi, Maria Carmen. “Un Proceso para XR basado en Reglas de Negocio”. [Disponible en:
http://www.willydev.net/descargas/WillyDev_Eli-Req-2.pdf]
- [8]. “Rational Rose: Procedimientos básicos para desarrollar un proyecto con UML”. [Disponible en:
<http://www.vico.org/TallerRationalRose.pdf>]
- [9]. Orallo, E. H. “El Lenguaje Unificado de Modelado (UML)”. Volumen, 6.[Disponible en:
<http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>]
- [10]. Landazuri, Barbara.” Definición de Perfiles en Herramientas de Gestión de Requisitos”. (2005)
- [11]. Kooch, N. “Software Engineering for Adaptative Hypermedia Applications”. Germany, Uni-Druck Publishing Company. Munich. (2001).
- [12]. Escalona, Maria José, Koch, Nora. “Ingeniería de Requisitos en Aplicaciones para la Web”. (2002).
- [13]. Övergaard, Gunnar, Palmkvist Karin. “Use Cases Patterns and Blueprints”. (2004).
- [14]. Rozas, Alejandro.”El Análisis de Sistemas y la Gestión de la Información en la organización”
- [15]. Fernando, A. “El Modelamiento del Negocio usando RUP”. (2006). [Disponible en:
<http://daguilar.evolutionperu.com/?p=7>.]
- [16]. Standards Coordinating Commitee IEEE “Standard Glossary of Software Engineering Terminology”. (1990).84.
- [17]. Pressman, Roger.” Ingeniería del software: Un enfoque práctico”. Félix Varela. (2005).

BIBLIOGRAFÍA.

1. “Análisis y Diseño: Métrica 3, UML y RUP”. (2007). [Disponible en:
<http://www.consultoriajava.com/umlrup.shtml>.]
2. “Productos Rational”. [Disponible en:
http://www.infosgroup.com/paginas/v4/publico/soluciones/soluciones_producto/rational/productosrational.asp?referal=/paginas/v4/publico/soluciones/soluciones_producto/rational.asp.]
3. “Rational Rose”. [Disponible en: http://www.indudata.com/1rational_rose.htm.]
4. “Rational Unified Process. Best Practices for Software. Development Teams”. [Disponible en: www.128.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_best_practices_TP026B.pdf]
5. Programa de Capacitación Online, “Desarrollador Cinco Estrellas”. (2005). [Disponible en: <http://www.mslatam.com/latam/msdn/comunidad/dce2005/>]
6. Caballero, I. “Una Herramienta CASE para ADOO: Visual Paradigm”. [Disponible en: http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf]
7. Díez, A. “IRqA y el desarrollo de proyectos: Experiencias Prácticas”. España. (2001).
8. Editores, S. H. T. (2007) InfoChannel. Oracle Business Process Analysis Suite.
[Disponible en:
http://www.infochannel.com.mx/mundos22.asp?id_notas=16405&tecnologia=2]
9. Gallón, A. R. “Desarrollo de Sistemas Informáticos”. (2004). [Disponible en:
<ftp://jano.unicauca.edu.co/cursos/Especializacion/ApliServicios/docs/UML-RUP-doc.pdf>.]
10. Grady Brooch, J. R., Ivar Jacobson. “UML”
11. . [Disponible en: <http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf>]
12. Zavala, J.” Ingeniería de Software”. (2000). [Disponible en:
<http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#Zavala>]

13. Sánchez, Maritza , Sinche, Franco, Gonzalez, Jimmy."RUP". (2006). [Disponible en:
<http://www.spinec.org/?m=200605>]
14. Marzo, J. V. "Taller de Modelado de Procesos de Negocio con UML. Herramienta CASE IBM Rational Rose" Volume, 15. (2004). [Disponible en:
<http://www.vico.org/learning/learning.php?op=recurso&id=217&e-learning=3525d2040feb7726668ac5de4e4d2359>]
15. Mendoza, M. A. "Metodologías de desarrollo de Software". (2004). [Disponible en:
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html]
16. Microsoft. Concurso de Habilidades. 2007[Disponible en: <http://imaginecup.com>.]
17. OMG. "OMG SysML".[Disponible en: from <http://www.omg-sysml.org/>.]
18. Vilorio, Orlando, T. R. "Herramientas CASE".[Disponible en:
http://www.lisi.usb.ve/publicaciones/01%20adopcion%20de%20nuevas%20tecnologias/adopcion_02.pdf]
19. Pekin, A., U. d. Jueces Online. 2007. [Disponible en: <http://acm.pku.edu.cn/JudgeOnline>.]
20. Saratov, U. Concurso Habilidades Programación. 2007.[Disponible en:<http://acm.sgu.ru>.]
21. Software, D. I. d. "Fase de Inicio. Modelo del Negocio". (2005-2006).
22. Ural, U. E. Concurso Habilidades Programación. [Disponible en: <http://acm.timus.ru>.]
23. Zhejiang, A. y. U. Concurso Habilidades Programación. 2007. [Disponible en: from
<http://acm.zju.edu.cn/>.]
24. Davis, A. "Identity and Measuring Quality in a Software Requirements Specification".
Baltimore. (1993)
25. Lorenzo T., Soulberto. El analista de sistemas y el paradigma estructurado [Disponible en:
<http://www.monografias.com/trabajos15/analista-sistem/analista-sistem.shtml#ANALISTA>]

26. Álvarez, S. y. H. A. "Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML". (2002)

GLOSARIO DE TÉRMINOS.

Actor: un actor es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio y el sistema interactúan.

Analista de Sistemas: un analista es aquel individuo que ejerce las tareas de análisis de los sistemas informáticos, con el fin de automatizarlos. La tarea de un analista es entender y describir un sistema de información hasta un grado suficiente como para ser automatizado mediante una computadora. Sus actividades se encuadran dentro de la fase de análisis del ciclo de vida del software.

Artefacto: un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software. Pueden ser artefactos un modelo, una descripción o un software. Los artefactos de UML se especifican en forma de diagramas, éstos, junto con la documentación sobre el sistema constituyen los artefactos principales que el modelador puede observar.

Caso de uso: es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

Copa Pascal: concurso de programación de la UCI, es uno de las actividades docentes más importante y se realiza una vez por curso. Involucra gran cantidad de estudiantes, potencia el desarrollo del conocimiento y el trabajo en equipo. Se realiza en dos etapas: primero, a nivel de facultad; segundo, a nivel universitario.

Crear competencia: estilo de competencia incluido en el diseño de la aplicación, permite al usuario registrado configurar una competencia, seleccionando cantidad de equipos, integrantes por equipo, fecha, hora y otros parámetros.

Estilo Libre: estilo de competencia incluido en el diseño de la aplicación, es el estilo que incluyen los juegos online que se encuentran en Internet. Todos compiten contra todos. El usuario descarga un

problema, envía la solución especificando un lenguaje de programación y el juez software le da una calificación si la respuesta resulta correcta.

FTP: es uno de los diversos protocolos de la red Internet, concretamente significa File Transfer Protocol (Protocolo de Transferencia de Ficheros) y es el ideal para transferir grandes bloques de datos por la red.

Herramientas CASE: 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 (Investigación Preliminar, Análisis, Diseño, Implementación e Instalación.).

Interfaz: la interfaz es lo que "media", lo que facilita la comunicación, la interacción, entre dos sistemas de diferente naturaleza, típicamente el ser humano y una máquina como el computador. Esto implica, además, que se trata de un sistema de traducción, ya que los dos "hablan" lenguajes diferentes: verbo-icónico en el caso del hombre y binario en el caso del procesador electrónico.

Juez online: programa que recibe como entrada el fichero con el código fuente de un programa, el lenguaje en el cual fue implementado y realiza la calificación del mismo.

Modelo del Dominio: descripción mediante un diagrama de clases UML donde se especifican las principales clases conceptuales que pueden intervenir en nuestro sistema, estos representarán los objetos que existen o eventos que suceden en el entorno en el que trabajará el sistema.

Online: en línea o sea que puede ser accedido por la Web.

Proceso: secuencia de actividades invocadas para producir un producto de software.

RAM: memoria RAM (Memoria de Acceso Aleatorio) es donde el computador guarda los datos que está utilizando en el momento presente. El almacenamiento es considerado temporal porque los datos y programas permanecen en ella mientras que la computadora este encendida o no sea reiniciada.

Reto individual: estilo de competencia incluido en el diseño de la aplicación, permite al usuario registrado enviarle a otro usuario conectado un reto para resolver un problema.

Requerimientos: condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal. Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales.

Rol: Papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso.

RUP: es un proceso de desarrollo de software, constituye una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo), asegura la producción de software de calidad dentro de plazos y presupuestos predecibles. Dirigido por casos de uso, centrado en la arquitectura, iterativo (mini-proyectos) e incremental (versiones). RUP apoya el desarrollo basado en componentes, tanto nuevos como preexistentes.

Servidor: un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos a la vez que permite a los usuarios almacenar y acceder a los archivos y a los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor.

Software: palabra en inglés utilizada para indicar a los programas de computadoras, a las aplicaciones.

Paquete: los paquetes son un mecanismo de organización de elementos que subdividen el modelo en otros más pequeños que colaboran entre sí.

Prototipo de interfaz usuario: presentación de la interfaz del producto que representa la funcionalidad contenida en los casos de uso; de manera que permita que el usuario verifique que el sistema va a satisfacer sus necesidades.

Pruebas Online: estilo de competencia incluido en el diseño de la aplicación, permite al usuario registrado definir un criterio (año(s), materia(s), tema(s)) para que se le genere un examen de cuarenta y cinco preguntas que deberá resolverse en treinta minutos.

Test mental: estilo de competencia incluido en el diseño de la aplicación, le permite a los usuarios registrados realizar cinco preguntas de un test mental.

UML: es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Usuario: es la persona que utiliza o trabaja con algún objeto.

Usuario final: el usuario final de un producto informático (bien sea hardware o software), es la persona a la que va destinada dicho producto una vez ha superado las fases de desarrollo correspondientes.

Usuario registrado: se denomina así a la persona que tiene derechos especiales en algún sistema por acreditarse en el mismo mediante un identificador y una clave de acceso.

Usuario invitado: el usuario invitado es aquel que navega en sitios web (o usa cualquier servicio de la red) sin autenticarse como usuario registrado.