

**Universidad de Sancti Spíritus “José Martí Pérez”  
Facultad de Ingeniería**

**Portal web para contribuir al proceso de  
almacenamiento y recuperación de  
información en formato digital en el CEEPI de la  
Universidad de Sancti Spíritus “José Martí  
Pérez”**

**Trabajo de diploma para optar por el título de Ingeniería  
Informática**

**Autor: Yurisbel Brito Díaz**

**Tutores: Ing. Yainel García Alfonso**

**Ing. Orelvis López Jiménez**

**Junio de 2013**

## *Dedicatoria*

A toda mi familia, en especial a mis padres Bertha y Armando.

## *Agradecimientos*

A mis padres Bertha y Armando por su apoyo constante durante toda la carrera.

A mi hermana Chabely y a mi cuñado Leydel por su preocupación.

A Nane por su ayuda incondicional.

A Irina Machado y Yanetsy Jiménez por “correr” conmigo.

A Yadira Yera por su ayuda.

A todos los que contribuyeron con la realización de esta investigación.

## *Resumen*

Debido al volumen de información existente actualmente en el Centro de Estudios Energéticos y Procesos Industriales (CEEPI) de la Universidad de Sancti Spíritus “José Martí Pérez”, surge una necesidad real, orientada hacia el almacenamiento y recuperación de información en formato digital. Dicho proceso incluye además la necesidad de automatizar actividades que se realizan manualmente, tal es el caso de la búsqueda en repositorios de archivos académicos.

La presente investigación titulada: **“Portal web para contribuir al proceso de almacenamiento y recuperación de información en formato digital en el CEEPI de la Universidad de Sancti Spíritus “José Martí Pérez”**, tiene como objetivo desarrollar un portal web para contribuir al proceso de almacenamiento y recuperación de información en formato digital en el CEEPI de la Universidad de Sancti Spíritus “José Martí Pérez”.

Para el desarrollo de la aplicación, se utilizó la metodología RUP (Proceso Unificado de Desarrollo); y como lenguaje de modelado, el UML (Lenguaje Unificado de Modelado); lo cual permitió la adecuada documentación del análisis, diseño e implementación de la solución. Se implementó utilizando el framework Symfony 2.2, lenguaje PHP para la lógica del negocio y JavaScript para la validación, lo que permitió un código eficiente y un agradable diseño, adecuado a las especificaciones del cliente. Se utilizó además como herramienta de modelado, el Rational Rose 7.

## *Abstract*

Due to the volume of existent information at the moment in the Center for Energetic and Industrial Process Studies (CEEPI) of the University of Sancti Spíritus “José Martí Pérez”, a real necessity arises, guided toward the storage and recovery of information in digital format. This process also includes the necessity to automate activities that are carried out manually, such it is the case of the search in repositories of academic files.

The present titled investigation: "**Portal web to contribute to the storage process and recovery of information in digital format in the CEEPI of the University of Sancti Spíritus "José Martí Pérez"**", has as objective to develop a portal web to contribute to the storage process and recovery of information in digital format in the CEEPI of the University of Sancti Spíritus "José Martí Pérez".

For the development of the application, the methodology RUP was used (Rational Unified Process); and as language of modeling, the UML (Unified Model Language); that which facilitated the appropriate documentation of the analysis, design and implementation of the solution. It was implemented using the framework Symfony 2.2, language PHP for the logic of the business and JavaScript for the validation, what allowed an efficient code and a pleasant design, appropriate to the client's specifications. It was also used as tool of modeling, the Rational Rose 7.

## Sumario

Introducción.....	1
Capítulo 1: Fundamentación teórica.....	7
1.1 Introducción.....	7
1.2 Conceptos asociados al dominio del problema.....	7
1.2.1 Gestión.....	7
1.2.2 Gestión de la información.....	7
1.2.3 Portal web.....	8
1.3 Centro de Estudios Energéticos y Procesos Industriales (CEEPI).....	8
1.4 Metodologías utilizadas para el diseño.....	10
1.4.1 Lenguaje Unificado de Modelado (UML).....	10
1.4.2 Proceso Unificado de Desarrollo (RUP).....	11
1.5 Tendencias y tecnologías actuales.....	11
1.5.1 Software Libre.....	11
1.5.2 Arquitectura Cliente/Servidor. Arquitectura en N capas.....	13
1.6 Lenguajes de programación.....	15
1.6.1 Lenguajes de programación web.....	15
1.6.2 PHP.....	15
1.6.3 HTML.....	15
1.6.4 JavaScript.....	15
1.6.5 CSS.....	16
1.7 Servidores web.....	17
1.7.1 Apache.....	18
1.8 Herramientas de desarrollo.....	18
1.8.1 Rational Rose.....	18
1.8.2 Git.....	19
1.9 Framework.....	19
1.9.1 Symfony.....	20
1.9.2 Doctrine.....	21
1.10 Sistemas Gestores de Bases de Datos (SGBD).....	23
1.10.1 SQL (Structure Query Language).....	23
1.10.2 SQL Server.....	24

## Sumario

1.10.3 PostgreSQL.....	24
1.10.4 MySQL.....	24
1.11 ¿Por qué utilizar Apache-PHP-MySQL?.....	26
1.12 Conclusiones.....	27
Capítulo 2: Descripción de la solución propuesta.....	28
2.1 Introducción.....	28
2.2 Descripción del modelo del negocio.....	28
2.3 Reglas del negocio.....	29
2.4 Descripción del proceso de negocio.....	29
2.5 Modelo de casos de uso del negocio.....	30
2.5.1 Actores del negocio.....	30
2.5.2 Trabajadores del negocio.....	31
2.5.3 Diagrama de casos de uso del negocio.....	31
2.5.4 Descripción de los casos de uso del negocio.....	32
2.5.5 Diagrama de actividades del negocio.....	33
2.5.6 Diagrama del modelo de objetos.....	33
2.6 Requerimientos funcionales.....	34
2.7 Requerimientos no funcionales.....	35
2.8 Modelo de casos de uso del sistema.....	38
2.8.1 Actores del sistema.....	38
2.8.2 Diagramas de casos de uso del sistema.....	39
2.9 Conclusiones.....	43
Capítulo 3: Construcción de la solución propuesta.....	44
3.1 Introducción.....	44
3.2 Descripción de los casos de uso del sistema.....	44
3.3 Diagrama de clases web.....	51
3.4 Diseño de la base de datos.....	52
3.4.1 Diagrama de clases persistentes.....	52
3.4.2 Modelo de datos.....	53
3.5 Principios de diseño.....	54
3.5.1 Estándares en la interfaz de la aplicación.....	54

## *Sumario*

3.5.2 Formatos de reportes.....	54
3.5.3 Tratamiento de errores.....	55
3.5.4 Seguridad.....	55
3.6 Diagrama de despliegue.....	55
3.7 Diagrama de componentes.....	56
3.8 Conclusiones.....	57
Conclusiones generales.....	58
Recomendaciones.....	59
Bibliografía referenciada.....	60
Anexos.....	62



## *Tablas*

Tabla 1. Actores del negocio.....	31
Tabla 2. Trabajadores del negocio.....	31
Tabla 3. Descripción del caso de uso “Buscar información”.....	32
Tabla 4. Actores del sistema.....	39
Tabla 5. Descripción del caso de uso del sistema “Gestionar usuario”.....	44
Tabla 6. Descripción del caso de uso del sistema “Restaurar contraseña”.....	45
Tabla 7. Descripción del caso de uso del sistema “Publicar ponencia”.....	45
Tabla 8. Descripción del caso de uso del sistema “Gestionar datos ponencia”.....	46
Tabla 9. Descripción del caso de uso del sistema “Gestionar evento”.....	46
Tabla 10. Descripción del caso de uso del sistema “Enviar notificación”.....	46
Tabla 11. Descripción del caso de uso del sistema “Buscar en repositorio de archivos académicos”.....	47
Tabla 12. Descripción del caso de uso del sistema “Enviar ponencia”.....	47
Tabla 13. Descripción del caso de uso del sistema “Autenticarse”.....	48
Tabla 14. Descripción del caso de uso del sistema “Cambiar contraseña”.....	48
Tabla 15. Descripción del caso de uso del sistema “Mostrar ponencias”.....	48
Tabla 16. Descripción del caso de uso del sistema “Mostrar lista de usuarios”.....	49
Tabla 17. Descripción del caso de uso del sistema “Mostrar ranking de categorías con más ponencias”.....	49
Tabla 18. Descripción del caso de uso del sistema “Mostrar ranking de ponencias más revisadas”.....	50
Tabla 19. Descripción del caso de uso del sistema “Mostrar usuarios conectados”..	50
Tabla 20. Descripción del caso de uso del sistema “Mostrar resultados de las búsquedas”.....	50
Tabla 21. Diagrama de clases web.....	51
Tabla 22. Descripción de los paquetes de componentes.....	57

## *Diagramas*

Diagrama 1. Casos de uso del negocio.....	32
Diagrama 2. Diagrama de actividades “Buscar información”.....	33
Diagrama 3. Diagrama del modelo de objetos.....	34
Diagrama 4. Relaciones entre los paquetes.....	40
Diagrama 5. Paquete de seguridad.....	41
Diagrama 6. Paquete de administración.....	41
Diagrama 7. Paquete de gestión.....	42
Diagrama 8. Paquete de reportes.....	42
Diagrama 9. Diagrama de clases persistentes.....	53
Diagrama 10. Diagrama de despliegue.....	56
Diagrama 11. Diagrama de componentes.....	56
Diagrama 12. Diagrama de clases web “Gestionar usuario”.....	62
Diagrama 13. Diagrama de clases web “Restaurar contraseña”.....	62
Diagrama 14. Diagrama de clases web “Publicar ponencia”.....	63
Diagrama 15. Diagrama de clases web “Enviar notificación”.....	63
Diagrama 16. Diagrama de clases web “Gestionar datos ponencia”.....	64
Diagrama 17. Diagrama de clases web “Gestionar evento”.....	64
Diagrama 18. Diagrama de clases web “Buscar en repositorio de archivos académicos”.....	65
Diagrama 19. Diagrama de clases web “Enviar ponencia”.....	66
Diagrama 20. Diagrama de clases web “Autenticarse”.....	66
Diagrama 21. Diagrama de clases web “Cambiar contraseña”.....	67
Diagrama 22. Diagrama de clases web “Mostrar ponencias”.....	67
Diagrama 23. Diagrama de clases web “Mostrar lista de usuarios”.....	68
Diagrama 24. Diagrama de clases web “Mostrar ranking de usuarios con más ponencias”.....	69
Diagrama 25. Diagrama de clases web “Mostrar ranking de ponencias más revisadas”.....	70
Diagrama 26. Diagrama de clases web “Mostar usuarios conectados”.....	71
Diagrama 27. Diagrama de clases web “Mostar resultados de las búsquedas”.....	71
Diagrama 28. Diagrama del modelo de datos.....	72

Las nuevas tecnologías de la información y las comunicaciones (NTIC) han cambiado el curso de la historia porque han traído consigo otras formas de trabajar, de pensar, de relacionarse con las personas, etc. La automatización de su gestión se ha convertido en una herramienta imprescindible y clave para empresas e instituciones.

Las actuales generaciones transitan en una época matizada por la información donde se habla comúnmente de las NTIC, de ahí que se afirme que vivimos en “la era de la información”. Sin lugar a dudas, se está en presencia de una revolución tecnológica de alcance insospechado.

Muchas son las definiciones que se han referido a las NTIC, una de ellas sostiene que son “aquellas herramientas y métodos empleados para recabar, retener, manipular o distribuir información” (Bologna & Walsh, 1997). De modo que generalmente se asocian con las computadoras y las tecnologías afines.

Para muchas empresas, la información y la tecnología que la soporta, representan sus activos más valiosos. Es por ello que las mismas han incrementado sus expectativas relacionadas con la entrega de servicios de las Tecnologías de la Información (TI). Necesitándose que dichos servicios presenten incrementos en calidad, funcionalidad y facilidades de uso, así como una mejora continua y disminución de los tiempos de entrega.

En un inicio, las Tecnologías de la Información (TI) de una empresa se consideraban ajenas al resto del negocio. Actualmente, gran parte de las inversiones se dedican a la infraestructura y nuevas aplicaciones de las TI involucradas a la informatización del negocio, es por ello que la gestión y administración de servicios de las Tecnologías de la Información (TI) representan una parte sustancial de los procesos de negocio.

En Cuba, el campo de la Gestión de la Información (GI) y el conocimiento ha experimentado cambios ventajosos tras la utilización de las NTIC, para el desarrollo de grandes sistemas de gestión y control, los cuales han facilitado el viejo proceso de recogida y análisis de información, permitiendo una mejor centralización, accesibilidad y manipulación de la misma. Además, se han proporcionado nuevos resultados estadísticos debido al descubrimiento de nuevas interrelaciones entre los datos

manejados, permitiendo lograr mejoras en la toma de decisiones en el proceso de gestión de la información.

Nuestro país se ha ido insertando en el desarrollo y utilización de las NTIC en la sociedad y la economía, potenciando sobre todo la utilización de la informática como herramienta fundamental en la gestión empresarial, debido a la eficiencia que las mismas ofrecen en los procesos de las organizaciones.

Las NTIC han sido incorporadas al proceso educativo desde hace unos años. Aún no existen estudios concluyentes que permitan afirmar que la utilización de los medios informáticos en la educación ha servido para mejorar los resultados académicos, sin embargo a menudo se refieren a las transformaciones obtenidas en el modo de hacer.

Para los profesores las tecnologías informáticas han servido hasta ahora para facilitar la búsqueda de material didáctico, contribuir a la colaboración con otros medios de enseñanza e incitar a la planificación de las actividades de aprendizaje de acuerdo con las características de la tecnología utilizada.

Las NTIC pueden emplearse en el sistema educativo de tres maneras distintas: como objeto de aprendizaje, como medio para aprender y como apoyo al aprendizaje. El trabajo que se presenta clasifica entre las tecnologías como apoyo al aprendizaje que es en realidad donde las NTIC encuentran su verdadero sitio en la enseñanza.

En 1996 se creó en la Universidad de Sancti Spíritus “José Martí Pérez”, un proyecto de investigación llamado: Grupo de Energía, que años más tarde cambiaría su nombre por el de Centro de Estudios Energéticos y Procesos Industriales (CEEPI). Este está constituido por 11 trabajadores: 9 profesores, un especialista en informática y un adiestrado, todos en función de desarrollar investigaciones en los grupos de trabajo científico:

- ✓ Biomasa y Biogás
- ✓ Economía energética (Eficiencia energética y Energía renovable)
- ✓ Gestión y calidad de los procesos.

Cuenta además con un laboratorio: Biogás e Ingeniería Ambiental, para la realización de investigaciones particularmente a ciclo cerrado y prestación de servicios al territorio. Los profesores, adiestrados y colaboradores del Centro, provenientes de diferentes especialidades, están vinculados a las líneas de investigación definidas, las cuales se organizan por proyectos de investigación de acuerdo a las necesidades del territorio.

Dado que en la Universidad no existe un software que almacene todas las investigaciones publicadas en el exterior: trabajos, ponencias, artículos, eventos, y demás documentación perteneciente al CEEPI; se hace difícil acceder a la información relacionada con el mismo debido a su constante crecimiento investigativo, ya que cada trabajador guarda sus investigaciones en su computadora, y para que otro trabajador del centro tenga acceso a ella, tiene que ser por medio del envío de la misma a través de correo electrónico o memoria flash.

En consecuencia se determinó el siguiente **problema científico**:

¿Cómo contribuir al almacenamiento y recuperación de información en formato digital en el Centro de Estudios Energéticos y Procesos Industriales (CEEPI), de la Universidad de Sancti Spíritus “José Martí Pérez”?

Para llevar a cabo este proyecto de investigación se toma como **objeto de estudio** el proceso de almacenamiento y recuperación de información en formato digital, y como **campo de acción** se abarcará el proceso de gestión de la información relacionada con el almacenamiento y recuperación de información en formato digital del CEEPI.

Para solucionar el problema planteado anteriormente, esta investigación se trazó como **objetivo general**:

Desarrollar un portal web para contribuir al proceso de almacenamiento y recuperación de información en formato digital en el CEEPI de la Universidad de Sancti Spíritus “José Martí Pérez”.

Para dar cumplimiento al objetivo general se crean las siguientes **preguntas científicas**:

1. ¿Cuáles son los fundamentos teóricos y metodológicos que sustentan la elaboración de un portal web para contribuir al almacenamiento y recuperación de información en formato digital?
2. ¿Cómo diseñar un portal web para contribuir al almacenamiento y recuperación de información en formato digital en el Centro de Estudios Energéticos y Procesos Industriales (CEEPI) de la Universidad de Sancti Spíritus “José Martí Pérez”?
3. ¿Cómo implementar un portal web para contribuir al almacenamiento y recuperación de información en formato digital en el Centro de Estudios Energéticos y Procesos Industriales (CEEPI) de la Universidad de Sancti Spíritus “José Martí Pérez”?

Para el logro del objetivo propuesto y por ende, para la solución del problema científico se plantearon las siguientes **tareas de investigación**:

1. Determinar de los fundamentos teóricos y metodológicos que sustentan la elaboración de un portal web para contribuir al almacenamiento y recuperación de información en formato digital.
2. Diseñar un portal web para contribuir al almacenamiento y recuperación de información en formato digital en el Centro de Estudios Energéticos y Procesos Industriales (CEEPI) de la Universidad de Sancti Spíritus “José Martí Pérez”.
3. Implementar un portal web para contribuir al almacenamiento y recuperación de información en formato digital en el Centro de Estudios Energéticos y Procesos Industriales (CEEPI) de la Universidad de Sancti Spíritus “José Martí Pérez”.

El trabajo posee una introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y anexos. La introducción expresa los aspectos que justifican la realización del trabajo.

**Capítulo 1: Fundamentación teórica:** En este capítulo se exponen los fundamentos teóricos del presente trabajo. Además, se revisan las tecnologías, lenguajes y herramientas existentes; se describen sus características y el por qué se seleccionaron.

**Capítulo 2: Descripción de la solución propuesta:** Describe el modelo del negocio identificando los procesos involucrados en él, las reglas que lo caracterizan, se identifican actores y trabajadores que en él intervienen, el diagrama de casos de uso y

los diagramas de actividades correspondientes; también se muestra el diagrama de clases del modelo de objetos. Se definen los requerimientos funcionales, los no funcionales y los casos de uso del sistema así como los diagramas asociados a cada uno de ellos.

**Capítulo 3: Construcción de la solución propuesta:** Se describen los casos de uso del sistema; se define el diseño mediante los diagramas de clases de diseño; se representa el diseño de la base de datos a través del diagrama de clases persistentes y el modelo de datos, que incluye la descripción de cada una de las tablas; además se describe la implementación a través del diagrama de despliegue y del diagrama de componentes.





# 1

## Capítulo Fundamentación teórica

### 1.1 Introducción

En este capítulo se abordan los principales conceptos asociados al campo de acción y objeto de estudio vinculados al problema de esta investigación. Se realiza un análisis de las tendencias y tecnologías actuales; que pueden ser adecuados en el desarrollo de la aplicación y la obtención de un producto de software con la calidad y fiabilidad requeridas.

### 1.2 Conceptos asociados al dominio del problema.

#### 1.2.1 Gestión:

Gestionar es coordinar todos los recursos disponibles para conseguir determinados objetivos, implica amplias y fuertes interacciones fundamentalmente entre el entorno, las estructuras, el proceso y los productos que se deseen obtener. (Concepto de gestión, 1924)

#### 1.2.2 Gestión de la información:

La gestión de la información se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar, y posteriormente, recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades. (Concepto de gestión, 1924)

Gestión de información: Comprende las actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada. (Woodman, 1985)

### **1.2.3 Portal web:**

Es un sitio web cuya característica fundamental es la de servir de *Puerta de entrada* (única) para ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios relacionados a un mismo tema. Incluye: enlaces, buscadores, foros, documentos, aplicaciones, compra electrónica, etc. Principalmente un portal en Internet está dirigido a resolver necesidades de información específica de un tema en particular. (Wikipedia)

### **1.3 Centro de Estudios Energéticos y Procesos Industriales (CEEPI)**

En 1996 nace el Grupo de Energía, que se convierte en el año 2001 en Centro de Estudios de Energía y Procesos Industriales (CEEPI) de la Facultad de Ingeniería perteneciente a la Universidad de Sancti Spíritus “José Martí Pérez”, cuando fue aprobado en el mes de Enero del 2001 por Resolución Ministerial, del Ministerio de Educación Superior de la República de Cuba. Desarrolla investigaciones relacionadas con la energía y el medio ambiente y contribuye a la formación de profesionales de las carreras de Ingeniería Industrial, Ingeniería de Procesos Agroindustriales e Ingeniería Informática.

Desde su fundación el CEEPI tiene precisado los objetivos estratégicos para cada período de proyección y los objetivos tácticos definidos en cada año para alcanzar un crecimiento aceptable en su gestión, así como, los criterios de medidas que le permiten evaluar el resultado alcanzado. Toda la proyección estratégica está estructurada de forma tal que se cumpla en los proyectos y planes de trabajo que ejecuta cada integrante de la plantilla del Centro y sus colaboradores. La misión está definida a partir del objeto social y la visión expresa el estado deseado y se concretan en las metas

estratégicas de la institución que se establecen por cada uno de los procesos universitarios.

El CEEPI dispone de una estrategia de Ciencia e Innovación para el período 2011 – 2015 sobre las bases de la Estrategia de Investigación-Desarrollo y la Política Científica de la institución y el país, que está estructurada según la gestión integrada de proyectos de investigación y desarrollo con los procedimientos normativos correspondientes.

El trabajo desarrollado en los 11 años de creado el CEEPI, constituyen la base de la estrategia definida. A través de las principales variables de visibilidad, se presenta el comportamiento de la producción científica, cuya variabilidad está determinada por el nivel científico de sus integrantes en cada momento y el período particular de la investigación, que realiza relaciones intra-universitarias, extrauniversitarias y con otras entidades

Al interior de la UNISS, el Centro mantiene relaciones sistemáticas en la Facultad de Ingeniería con los departamentos de informática y matemática y mantiene un estrecho vínculo con el departamento de ingeniería industrial, donde los profesores cumplen la décima docente en dicha carrera y a su vez los profesores de ingeniería investigan en la línea científica del CEEPI distribuidos según grupo de trabajo científico (GTC) de interés. Además, mantiene una colaboración constante con los dos centros de estudios de la institución: el CETAD y el CECES y con las Facultades de Agronomía, Contabilidad y Finanzas y Humanidades, así como las Filiales Universitarias de Taguasco, Yaguajay, Cabaiguán, Trinidad y Jatibonico.

En la visión del trabajo científico del área, se establece: alcanzar resultados científicos de alto impacto en el desarrollo del territorio espirituano, logrando una elevada producción científica que ubique al Centro en una destacada posición dentro de la provincia, lograr además un protagonismo nacional en el campo del desarrollo energético e industrial y el uso de métodos matemáticos y la informática para la optimización y mejoramiento de los procesos y su gestión; alcanzando importantes premios provinciales y nacionales en el FORUM, la Academia de Ciencias, la

Innovación Tecnológica y otros eventos; todo ello unido a un desarrollo sostenido de las instalaciones y la tecnología, un destacado trabajo científico con los estudiantes y a una estrecha vinculación del trabajo científico con el docente.

### **1.4 Metodologías utilizadas para el diseño**

Con el fin de lograr un mejor desempeño y calidad para desarrollar un producto informático, se hace necesario la consulta de las nuevas tendencias y del uso de una metodología que indique los pasos correctos a seguir. Además, el desarrollo del software propuesto, exige de antemano una búsqueda, estudio y análisis de diversas tecnologías, lenguajes, gestores de base de datos y herramientas de desarrollo que hagan posible su adecuada realización y que garantice el cumplimiento de todas sus funcionalidades.

#### **1.4.1 Lenguaje Unificado de Modelado (UML)**

El Lenguaje Unificado de Modelado (*Unified Modeling Language*, UML) es un lenguaje estándar para escribir planos de software, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software (Jacobson, Booch, & Rumbaugh, 2006).

Es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental tal como el Proceso Unificado de Desarrollo (Jacobson, Booch & Rumbaugh, 2006).

El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas (Jacobson, Booch & Rumbaugh, 2006).

El vocabulario y las reglas de un lenguaje como UML indican cómo crear y leer modelos bien formados, pero no dicen qué modelos se deben crear ni cuándo se deberían crear. Esta es la tarea del proceso de desarrollo de software. Un proceso bien definido guiará a sus usuarios al decidir qué artefactos producir, qué actividades y qué personal se emplea para crearlos y gestionarlos, y cómo usar esos artefactos para medir y controlar el proyecto de forma global (Jacobson, Booch & Rumbaugh, 2006).

### **1.4.2 Proceso Unificado de Desarrollo (RUP)**

El Proceso Unificado de Desarrollo o RUP (*Rational Unified Process*), es un proceso de desarrollo de software que utilizando UML, constituye una metodología estándar muy utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. (Jacobson, Booch & Rumbaugh, 2006)

RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (productos tangibles del proceso) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al concluir cada ciclo (Jacobson, Booch & Rumbaugh, 2006)

## **1.5 Tendencias y tecnologías actuales.**

### **1.5.1 Software Libre.**

Entre los años 60 y 70 del Siglo XX, era común que los programadores y desarrolladores de software compartieran libremente sus programas unos con otros. Esto se debía en gran medida a que, por ese entonces, el software no era visto como un producto en sí mismo sino como un adjunto que los vendedores de las grandes computadoras de la época (las mainframes) aportaban a sus clientes para que éstos

podieran usarlas. A finales de los 70, las compañías iniciaron el hábito de imponer restricciones a los usuarios, con el uso de acuerdos de licencia. Todo esto fue empeorando con la llegada de las computadoras modernas que necesitaban de un sistema operativo (Free Software Foundation, 2008).

Dada esta situación, en 1985, Richard Stallman fundó la Free Software Foundation (FSF). Con el fin de otorgar libertad a los usuarios y restringir las posibilidades de apropiación del software, Stallman introdujo la definición de free software (software libre) que según él se refería a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo cual se puede ayudar a otros y de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (para la segunda y última libertad mencionadas, el acceso al código fuente es un requisito previo). (Free Software Foundation, 2008)

En la actualidad existen varios países en los cuales sus administraciones públicas muestran apoyo al software libre, tal es el caso de Alemania, Argentina, Brasil, Cuba, Chile, China, Ecuador, España, Francia, México, República Dominicana y Venezuela.

Principalmente para los países subdesarrollados, la aparición del software libre ha tenido un gran significado, pues en ocasiones el pago constante de software no libre es casi intolerable. Para Cuba, por sus características especiales de ser un país bloqueado, en ocasiones el pago de estas licencias no es tomado en cuenta y, por lo general, existe una tendencia a la piratería informática. Con el fin de ir extinguiendo esta tendencia y de apoyarse a los beneficios que trae consigo el software libre, nuestro país lleva a cabo un amplio programa de apoyo al software de fuentes abiertas y ha ido paulatinamente migrando total o parcialmente sus servidores y sistemas de escritorio.

### **1.5.2 Arquitectura Cliente/Servidor. Arquitectura en N capas**

La arquitectura Cliente\Servidor es un modelo para desarrollar sistemas de información en el que las transacciones se dividen en elementos independientes que tienen la facilidad de cooperar entre ellos para intercambiar información, recursos o servicios. Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. (Molina & Castro, 2009)

La arquitectura cliente/servidor genérica tiene dos tipos de nodos en la red: clientes y servidores. Consecuentemente, se refieren a veces como arquitecturas de dos niveles o dos capas. Algunas redes disponen de tres tipos de nodos. (Hurtado Jara, 2008)

La arquitectura de n capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica proveyendo una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada. (Pérez Sandoval, 2009)

#### **Arquitectura de 3 Capas**

Una buena arquitectura de software debe facilitar los requerimientos de mantenimiento, reusabilidad, escalabilidad, y robustez del mismo. Al concertar la solución de un problema como una serie de capas, cada capa debe ocuparse de un subconjunto de responsabilidades fuertemente acopladas y tener poca cohesión con las demás. (Peláez, 2009)

La primera capa se denomina capa de presentación y normalmente consiste en una interfaz gráfica de usuario de algún tipo. La capa intermedia, consiste en la aplicación o lógica del negocio, y la tercera capa, la capa de datos, contiene los datos necesarios para la aplicación (bases de datos y otros). Esta separación entre la lógica de aplicación de la interfaz de usuario añade una enorme flexibilidad al diseño de la aplicación. (Chappell & Kirk, 2006)

Dentro de la arquitectura de n capas se encuentra el patrón Modelo-Vista-Controlador (MVC), el cual es usado en este trabajo.

Fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

### **Definición de las partes**

**El Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. (Catalani, 2007)

**La Vista** es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. (Catalani, 2007)

**El Controlador** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Sirve como un intermediario entre el Modelo y la Vista. (Catalani, 2007)

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Ventajas de este patrón según (Martínez, 2007):

- ✓ Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado.
- ✓ Debido a que cada parte es independiente de la otra los cambios o actualizaciones pueden ser realizados sin afectar la aplicación como un todo.
- ✓ La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.



## **1.6 Lenguajes de programación**

### **1.6.1 Lenguajes de programación web**

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Algunos de estos son JSP (Java Server Page), Python, Perl, Ruby, HTML (HyperText Markup Language), JavaScript, PHP (Personal Home Page), ASP (Active Server Page), ASP.Net (Active Server Page.Net). A continuación se dará una introducción a algunos de los utilizados en esta investigación.

#### **1.6.2 PHP**

Es un lenguaje de programación utilizado para la creación de sitio web. PHP es un acrónimo que significa “PHP Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. Surgió en 1995, desarrollado por PHP Group. PHP no necesita ser compilado para ejecutarse. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. (Pérez Valdés, 2007)

#### **1.6.3 HTML**

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas de Hipertexto). Desarrollado por el World Wide Web Consortium (W3C). Este lenguaje se caracteriza por ser un lenguaje de fácil aprendizaje, estático, lo admiten todos los exploradores, pero la interpretación de cada navegador puede ser diferente. (Webmaster, 2009)

#### **1.6.4 JavaScript**

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de

herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Java Script. El código JavaScript puede ser integrado dentro de las páginas web. Los script tienen capacidades limitadas, por razones de seguridad, el código visible por cualquier usuario y el código deben descargarse completamente.

### **1.6.5 CSS**

CSS (Cascade Style Sheets) es una tecnología que permite controlar la presentación de los documentos en la Web. Las hojas de estilo en cascada o CSS constituyen un lenguaje sencillo que complementa el de HTML, suponiendo un apoyo fundamental a la hora de diseñar páginas Web, porque permiten una mayor precisión en el ajuste de los elementos de diseño.

Esta técnica consiste en separar el diseño del contenido, de manera que las indicaciones para conformar el diseño se agrupan en una hoja de estilo o archivo fuera del contenido del documento de la página HTML. Lo que hace fundamentalmente el código de las hojas de estilos es transformar las etiquetas del lenguaje HTML y conformarlas a las características que se quiera darle; pero también, con este código se pueden crear etiquetas nuevas, que se introducen dentro del documento. Una de las ventajas de las hojas de estilos es que se puede modificar algunas características de todos los documentos de un sitio Web desde un archivo, sin tener que modificarlas en cada uno de los documentos.

Ventajas:

- ✓ Es la mejor forma de separar los contenidos y su presentación.
- ✓ Es imprescindible para crear páginas Web complejas.
- ✓ Reduce la complejidad de su mantenimiento.
- ✓ Permite visualizar el mismo documento en infinidad de dispositivos diferentes.
- ✓ Permite especificar la apariencia de todo el sitio web en un único lugar, los estilos para fuentes, tamaños de la fuente, espaciado de letras, colores, y márgenes. Un cambio en un fichero genera cambios en todos los ficheros.

Para la realización del Portal Web se decidió utilizar como lenguaje de programación para la lógica del negocio PHP, y para la validación JavaScript debido a sus ventajas. Pérez Valdés (2007) enumera las siguientes ventajas que tiene el uso de PHP:

- ✓ Se caracteriza por ser un lenguaje muy rápido.
- ✓ Soporta en cierta medida la orientación a objeto, clases y herencia.
- ✓ Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- ✓ Tiene capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- ✓ Capacidad de expandir su potencial utilizando módulos.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Incluye gran cantidad de funciones.
- ✓ No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Pérez Valdés (2007) enuncia las siguientes ventajas que tiene el uso de JavaScript:

- ✓ Tiene muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas.
- ✓ Es un lenguaje interpretado (que no es compilado) por el cliente.
- ✓ Las referencias a objetos se comprueban en tiempo de ejecución.
- ✓ Permite crear páginas más dinámicas, haciéndolas más atractivas para el usuario.

### **1.7 Servidores Web**

Un servidor web es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo está diseñado para transferir lo que se llama hipertextos, páginas web o páginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de

sonidos. Entre los servidores más utilizados en Internet se encuentran Apache e Internet Information Services (IIS). (Cuenca, 2006)

Entre los servidores web más utilizados se encuentran Microsoft Internet Information Server, Microsoft Personal Web Server, Apache HTTP Server, Netscape Enterprise Server y Sun ONE Web Server.

### 1.7.1 Apache

El servidor Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. (Una introducción a Apache, 2006)

Ventajas de su uso son (Una introducción a Apache, 2006):

- ✓ Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- ✓ Apache es una tecnología gratuita de código fuente libre.
- ✓ Apache trabaja con gran cantidad de Perl, PHP y otros lenguajes de script.
- ✓ Es un servidor altamente configurable de diseño modular.
- ✓ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.

## 1.8 Herramientas de desarrollo

### 1.8.1 Rational Rose

Para apoyar el trabajo de la metodología RUP, ha sido desarrollada por la Compañía norteamericana Rational Corporation la herramienta CASE (*Computer Assisted Software Engineering*) Rational Rose desde el año 2000. Esta herramienta integra todos los elementos que propone la metodología para cubrir el ciclo de vida de un proyecto.

Esta herramienta CASE propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. (Rodríguez Febles, 2005)

Rational Rose proporciona también mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño. (Rodríguez Febles, 2005)

### **1.8.2 Git**

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o frontend como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

### **1.9 Framework**

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones web. En general, con el término framework, se está refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. (Gutiérrez, 2006)

Algunas de las características presentes en casi todos son: abstracción de URL y sesiones (no es necesario manipular directamente las URLs ni las sesiones, el framework se encarga de esto); acceso datos (incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en bases de datos, etc.); controladores (la mayoría de los frameworks que existen implementa una serie de controladores para gestionar eventos). (Gutiérrez, 2006)

Son diseñados con la intención de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional.

En el mundo del software es común observar como la mayoría de los productos utilizan de alguna u otra forma una base de datos que permita el almacenamiento de la información. En este proceso de creación de software, el empleo de frameworks se ha convertido para muchos programadores en una potente herramienta.

Actualmente existen en el mundo una gran variedad de frameworks para PHP y JavaScript. Algunos de estos son Symfony, Akelos Framework, PHP Framework, Biscuit, Cake, CodeIgniter, PhpMVC, Prado, Seagull, Zend Framework, etc.

### **1.9.1 Symfony**

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

Symfony ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de gestores de

bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows.

Las características más comunes para el desarrollo de proyectos web están automatizadas en Symfony, tales como:

- ✓ Permite la internacionalización para la traducción del texto de la interfaz, los datos y el contenido de localización.
- ✓ La presentación usa templates y layouts que pueden ser construidos por diseñadores de HTML que no posean conocimientos del framework.
- ✓ Los formularios soportan la validación automática, lo cual asegura mejor calidad de los datos en las base de datos y una mejor experiencia para el usuario.
- ✓ El manejo de caché reduce el uso de banda ancha y la carga del servidor.
- ✓ La facilidad de soportar autenticación y credenciales facilita la creación de áreas restringidas y manejo de seguridad de los usuarios.
- ✓ El enrutamiento y las URLs inteligentes hacen amigable las direcciones de las páginas de la aplicación.
- ✓ Las listas son más amigables, ya que permite la paginación, clasificación y filtraje automáticos.
- ✓ Los plugins proveen un alto nivel de extensibilidad.

### **1.9.2 Doctrine**

Doctrine es un mapeador de objetos-relacional (ORM) escrito en PHP que proporciona una capa de persistencia para objetos PHP. Es una capa de abstracción que se sitúa justo encima de un Sistema Gestor de Bases de Datos.

Doctrine tiene influencias de docenas de proyectos de personas muy diferentes. Las mayores influencias son de Hibernate (el ORM de Java) y de ActiveRecord (de Ruby on

Rails). Ambos tienen una implementación completa tanto en Java como en Ruby. El propósito de Doctrine es construir una solución igual de potente para PHP.

Una característica de Doctrine es el bajo nivel de configuración que necesita para empezar un proyecto. Doctrine puede generar clases a partir de una base de datos existente y después el programador puede especificar relaciones y añadir funcionalidad extra a las clases autogeneradas. No es necesario generar o mantener complejos esquemas XML de base de datos como en otros frameworks.

Otra característica importante de Doctrine es la posibilidad de escribir consultas de base de datos utilizando un dialecto de SQL denominado DQL (Doctrine Query Language) que está inspirado en Hibernate (Java).

Otras características notables de Doctrine son:

- ✓ Soporte para datos jerárquicos.
- ✓ Soporte para hooks (métodos que pueden validar o modificar las escrituras y lecturas de la base de datos) y eventos para manejar la lógica de negocio relacionada.
- ✓ Herencia.
- ✓ Un framework de caché que utiliza diversos motores como memcached, SQLite o APC.
- ✓ Transacciones ACID.
- ✓ Diversos comportamientos del modelo (conjuntos anidados, internacionalización, log, índice de búsqueda).
- ✓ Una función "compilar" que combina varios archivos PHP del framework en uno solo para evitar el descenso de rendimiento que provoca incluir varios archivos PHP.



## **1.10 Sistemas Gestores de Base de Datos (SGBD)**

Los sistemas gestores de base de datos (SGBD); (en inglés: Database Management System, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Su propósito general es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos. (Sandoval Méndez, 2009)

El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado. (Mato García, 2006)

En el mundo existen varios tipos de SGBD clasificados en licenciamiento comercial como Oracle, SQL SERVER, etc. y los de licenciamiento libre entre los que se encuentra MySQL, PostgreSQL, FireBird, SQLite, etc.

### **1.10.1 SQL (Structure Query Language)**

SQL (Structure Query Language) o Lenguaje de Consulta Estructurado es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Posibilita lanzar consultas con el fin de recuperar información de interés de una base de datos de una forma sencilla.

SQL permite la concesión y denegación de permisos, la implementación de restricciones de integridad y controles de transacción, y la alteración de esquemas. Debido a que es un lenguaje declarativo, especifica qué es lo que se quiere y no como conseguirlo, por lo que una sentencia no establece explícitamente un orden de ejecución.

### **1.10.2 SQL Server**

Diseñada desde su inicio para trabajar en entornos Internet e Intranet, Microsoft SQL Server es capaz de integrar los nuevos desarrollos para estos entornos específicos con los desarrollos heredados de aplicaciones tradicionales. (Sandoval Méndez, 2009)

SQL Server soporta la configuración automática y la auto-optimización, además tiene una administración multiservidor para un gran número de servidores, pero es mayormente utilizada para trabajar en plataformas de Windows. (Sandoval Méndez, 2009)

### **1.10.3 PostgreSQL**

Es un sistema de gestión de base de datos relacional orientada a objetos y libre. PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma sin necesidad de bloqueos. Además de una amplia variedad de tipos nativos, provee soporte para, números de precisión arbitraria, texto de largo ilimitado, figuras geométricas, y arrays. Soporta transacciones, tiene mejor soporte para subselects, triggers, vistas y procedimientos almacenados en el servidor, además de ciertas características orientadas a objetos. (Johnson, 2007)

### **1.10.4 MySQL**

MySQL es el sistema gestor de bases de datos Open Source más popular en la comunidad de programadores. Este puede ser descargado de Internet y usado de forma gratuita. Su código se puede estudiar y adecuar a cualquier requerimiento establecido. Es muy rápido, fiable, fácil de usar y surge para manipular bases de datos muy grandes con gran eficiencia.

Es un sistema multiplataforma de base de datos, aspecto que le aporta la característica de ser fiable y veloz. Cuenta con un sistema de privilegios y contraseñas muy seguro que permite la autenticación básica para el acceso al servidor; elemento muy importante que contribuye a garantizar en gran medida la seguridad de la aplicación. El lenguaje PHP posee un amplio conjunto de funciones definidas para el tratamiento de este

gestor. Este SGBD es considerado (en su propia documentación así lo reseña) como la más rápida y robusta herramienta para la administración y gestión de bases de datos, tanto para volúmenes de datos grandes como pequeños.

Sus principales características son:

- ✓ Cuenta con tecnología InnoDB Engine, que proporciona transacciones, claves externas, actualización y borrado en cascada y bloqueo a nivel de fila, más rápido, con caché de consultas, mejoras en inserciones, búsqueda en índices compuestos y creación de índices sobre texto completo, un servidor embebido y compatibilidad con otras bases de datos.
- ✓ Es posible definir diversos tipos de columnas como enteros de 1, 2, 3, 4, y 8 bytes, coma flotante, doble precisión, carácter, fechas, enumerados, etc.
- ✓ El servidor soporta mensajes de error en distintas lenguas.
- ✓ Todas las columnas pueden tener valores por defecto.
- ✓ Admite registros de longitud fija y variable.
- ✓ Todas las claves viajan cifradas en la red.
- ✓ Cuenta con un sistema de contraseñas y privilegios muy flexibles y seguros.
- ✓ Puede trabajar en distintas plataformas y sistemas operativos.
- ✓ Implementa el multiprocesamiento, es decir, puede usar varias CPU si estas están disponibles.

Ventajas: (Riveros, 2008)

- ✓ Está disponible para la mayoría de las plataformas de sistemas operativos.
- ✓ Su bajo consumo lo hace apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- ✓ Es software libre. Cualquier persona puede utilizar y modificar el código ajustándolo a sus necesidades sin pagar por hacerlo.

- ✓ Alto rendimiento.
- ✓ Elevada velocidad tanto al conectar con el servidor como al servir consultas y demás funciones.
- ✓ Excelentes utilidades de administración (backup, recuperación de errores, etc).
- ✓ Aunque se bloquee, no suele perder información ni corromper los datos.
- ✓ Excelente integración con PHP.
- ✓ No tiene límites en el tamaño de los registros.
- ✓ Significativo control de acceso, es decir, qué usuarios tienen acceso a qué tablas y con qué permisos.
- ✓ MySQL se comporta mejor que otros motores de Base de Datos como SQL Server y Oracle a la hora de modificar ó añadir campos a una tabla.
- ✓ Es multi-hilo, o sea, maneja muchas conexiones al mismo tiempo, donde cada una tiene su propio hilo, de modo que ningún hilo tiene que esperar por otro, a menos que uno esté modificando una tabla que otro quiera acceder.

### **1.11 ¿Por qué utilizar Apache-PHP-MySQL?**

La selección de estas tecnologías se basó principalmente en los criterios de rapidez, sencillez y rendimiento, unidos a otros dos mucho más importantes: la tecnología Open Source y la característica de integrarse para funcionar en cualquier plataforma. Estas características aportan un gran beneficio gracias al entorno de software libre de la actualidad y cumplen con una política trazada por el país de proyectarse con visión al futuro y comenzar a desarrollar este tipo de aplicaciones.

Como SGBD se seleccionó MySQL, por su velocidad y sencillez. La selección estuvo motivada además por la utilización de PHP, ya que esta tecnología exhibe un magnífico soporte para el trabajo con MySQL. Finalmente como servidor web se determinó que la mejor opción era el Apache, por mostrar un excelente comportamiento y estabilidad.

Este servidor es el más utilizado en el mundo por su magnífico desempeño, seguridad y las posibilidades de gestionar de una forma eficiente muchos de los aspectos que definen en gran medida la calidad de los servidores web.

La selección de Apache añade una integración perfecta con las tecnologías antes mencionadas. Apache-PHP-MySQL forman en su conjunto una vía muy eficaz a la hora de tener en cuenta la implementación de sitios web de corto, mediano o gran alcance.

### **1.12 Conclusiones**

En este capítulo se analizaron las características de las distintas metodologías y herramientas de desarrollo, así como las ventajas de las mismas y se determinó seleccionar:

- ✓ RUP (Proceso Unificado de Desarrollo) como la metodología más apropiada para el desarrollo del proyecto.
- ✓ UML como el lenguaje de modelación utilizado.
- ✓ La arquitectura 3 Capas.
- ✓ Para la implementación el lenguaje PHP.
- ✓ MySQL como sistema gestor de bases de datos
- ✓ Rational Rose como herramienta de modelado UML.

La selección de estas tecnologías se basó principalmente en la tecnología Open Source y la característica de integrarse para funcionar en cualquier plataforma.

# 2

## Capítulo

### Descripción de la solución propuesta

#### 2.1 Introducción

En este capítulo se describirán las mejoras propuestas ante la situación problemática planteada en el capítulo anterior, se destacarán las reglas establecidas en el negocio, se identificarán los actores y trabajadores del mismo, así como el diagrama de casos de uso y su formato expandido.

Se plantea el análisis del sistema utilizando algunos de los artefactos que propone la metodología RUP: los requisitos funcionales y no funcionales. Se definen los actores, se representa el diagrama de casos de uso del sistema, y se realiza además la descripción de cada caso de uso.

#### 2.2 Descripción del modelo del negocio

Modelar e identificar el flujo de los procesos que serán objeto de automatización de un sistema informático, es un elemento clave para lograr un desarrollo exitoso del producto y una buena comunicación entre los desarrolladores, los clientes y el usuario final. A este flujo de trabajo se le denomina: Modelación del Negocio. (Rumbaugh, Booch, & Jacobson, 2006)

El modelado del negocio es una técnica que permite comprender los procesos de negocio de la organización y se desarrolla en dos pasos: (Rumbaugh, Booch, & Jacobson, 2006)

- 1 Confección de un modelo de casos de uso del negocio que identifique los actores y casos de uso del negocio que utilicen los actores.
- 2 Desarrollo de un modelo de objetos del negocio compuesto por trabajadores y entidades del negocio que juntos realizan los casos de uso del negocio.

Uno de los objetivos fundamentales de este capítulo es modelar los procesos de negocio que van a tener lugar en el Centro de Estudios Energéticos y Procesos Industriales de la Universidad de Sancti Spíritus “José Martí Pérez”, a través del uso de los artefactos que propone la metodología RUP y el lenguaje UML.

### **2.3 Reglas del negocio**

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. (Jacobson, 2006)

Partiendo de lo planteado anteriormente fueron identificadas las siguientes reglas:

- ✓ Se debe poder acceder al sistema desde cualquier computadora personal conectada a la red de la Universidad de Sancti Spíritus “José Martí Pérez”.
- ✓ El sistema contará con un usuario administrador y tantos usuarios de bajo nivel como sea necesario.

### **2.4 Descripción del proceso de negocio**

El primer paso del modelado del negocio consiste en capturar y definir los procesos de negocio, lo cual constituye la base fundamental para el posterior modelado. Cuando se hable de procesos de negocio se puede decir que son un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y forma, y que emplean los recursos de la organización para dar resultados que apoyen sus objetivos. (Jacobson, 2006)

A partir de este concepto fueron identificados los siguientes procesos del negocio:

- ✓ Almacenamiento de la información

- ✓ Búsqueda de la información

### **Almacenar información:**

En el Centro de Estudios Energéticos y Procesos Industriales (CEEPI), después que un trabajador realiza una investigación, la envía para su posterior publicación en sitios web y revistas científicas de universidades en el exterior. Si el trabajo investigativo es aceptado, el trabajador lo guarda en su computadora, dada la no existencia de un software que almacene todos los proyectos investigativos, ya sea por autor, categoría, año, etc.

### **Buscar información**

A la hora de buscar información relacionada con cualquier tema de investigación publicado por los trabajadores del Centro de Estudios Energéticos y Procesos Industriales (CEEPI), ya sea para leer, tener una referencia bibliográfica, etcétera, el interesado tiene que ir con el autor del trabajo y solicitárselo, este último se lo facilita a través de una memoria flash o se lo envía por correo electrónico.

## **2.5 Modelo de casos de uso del negocio**

La modelación del negocio se lleva a cabo a través del Modelo de Casos de Uso del Negocio que describe los procesos de una empresa en términos de casos de uso y actores del negocio en correspondencia con los procesos del negocio y los clientes, respectivamente. El modelo de casos de uso presenta un sistema desde la perspectiva de su uso y esquematiza cómo proporciona valor a sus usuarios. (Jacobson, 2006)

### **2.5.1 Actores del negocio**

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa pero que no son parte de él y se benefician del negocio. (Jacobson, 2006)

Los actores del negocio se listan a continuación:



Tabla 1. Actores del negocio

Actor del negocio	Justificación
Trabajador del CEEPI	Es un trabajador que se beneficia de la búsqueda y almacenamiento de información.

### 2.5.2 Trabajadores del negocio

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol. (Jacobson, 2006)

Los trabajadores del negocio se listan a continuación.

Tabla 2. Trabajadores del negocio

Trabajador del negocio	Justificación
Investigador	Es el encargado de actualizar toda la información que se genera en el CEEPI.

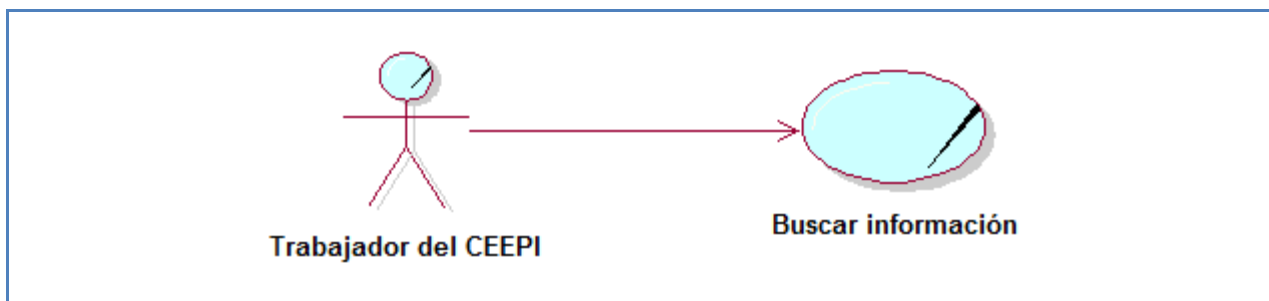
### 2.5.3 Diagrama de casos de uso del negocio

Los diagramas de casos de uso sirven para especificar las funcionalidades y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. (Jacobson, 2006)

Un diagrama de casos de uso del negocio representa gráficamente la interacción de los procesos del negocio y los actores y/o trabajadores del negocio. Permitted mostrar los límites y el entorno de la organización bajo estudio. (Jacobson, 2006)

A continuación se muestra el diagrama de casos de uso del negocio:

Diagrama 1. Casos de uso del negocio



### 2.5.4 Descripción de los casos de uso del negocio

Luego de identificar los procesos del negocio y realizar el diagrama de caso de usos del negocio, se hace necesario describirlos en detalle. A continuación tiene lugar dicha descripción representadas en forma de tablas:

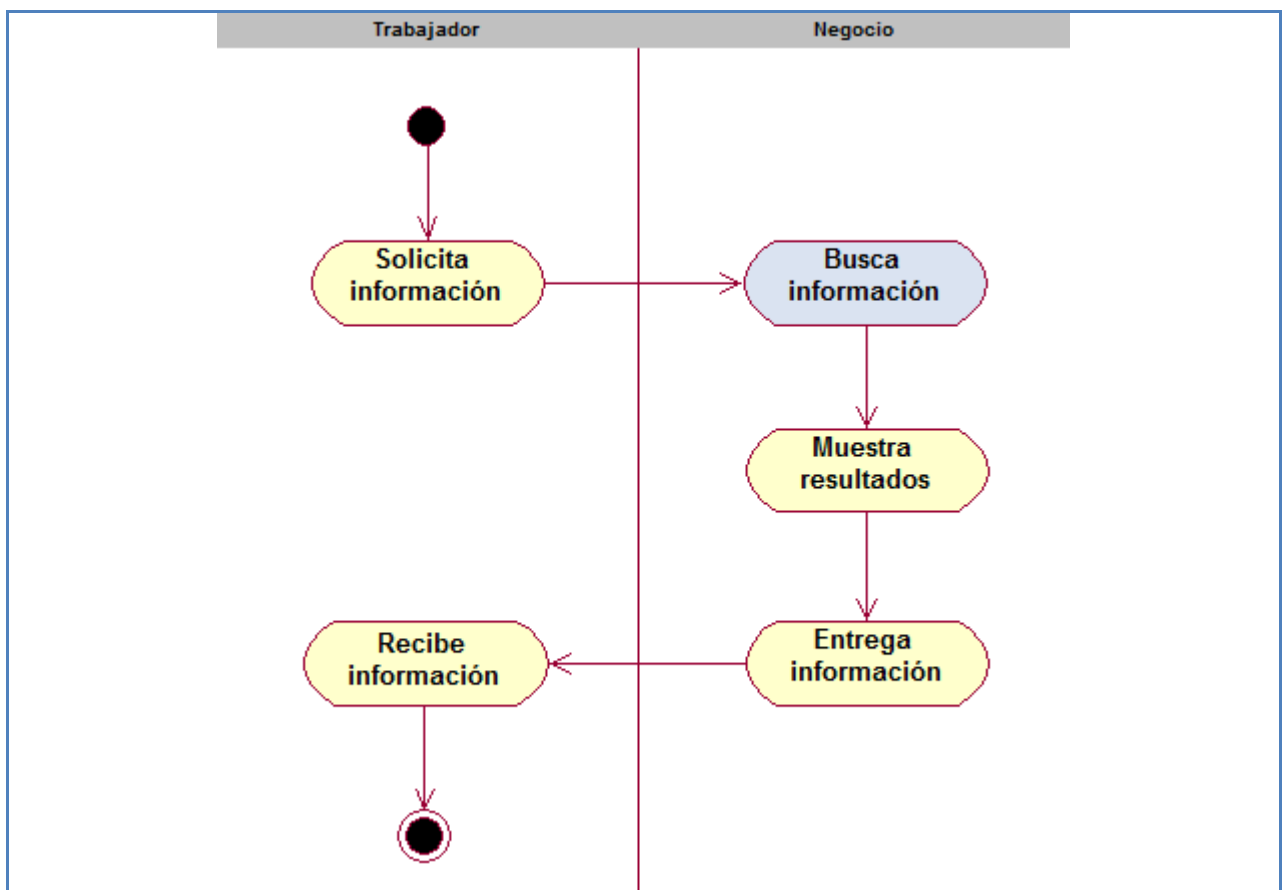
Tabla 3. Descripción del caso de uso “Buscar información”

<b>Caso de uso del negocio</b>	Buscar información	
<b>Actor</b>	Trabajador	
<b>Propósito</b>	Buscar información.	
<b>Resumen</b>	Se inicia el caso de uso cuando el trabajador solicita al autor de la investigación, la información correspondiente a la misma. Este último, se la entrega por medio de una memoria flash o a través del e-mail.	
<b>Flujo de trabajo</b>		
<b>Acción del actor</b>	<b>Respuesta del negocio</b>	
1. El trabajador desea obtener información.	2. El autor de la investigación realiza una búsqueda de la misma. 3. El autor entrega la información.	
4. El trabajador recibe la información.		
<b>Prioridad</b>	Alta	

### 2.5.5 Diagrama de actividades del negocio

Un diagrama de actividades es un diagrama que muestra el flujo de una actividad a otra; los diagramas de actividad tratan la vista dinámica de un sistema. Un caso especial de diagrama de estados (aquellos diagramas que tratan la vista dinámica de un sistema) en el cual todos o casi todos los estados son estados de acción y en el cual todas o casi todas las transiciones son disparadas por la terminación de las acciones en los estados origen. (Rumbaugh, Booch, & Jacobson, 2006)

Diagrama 2. Diagrama de actividades “Buscar información”

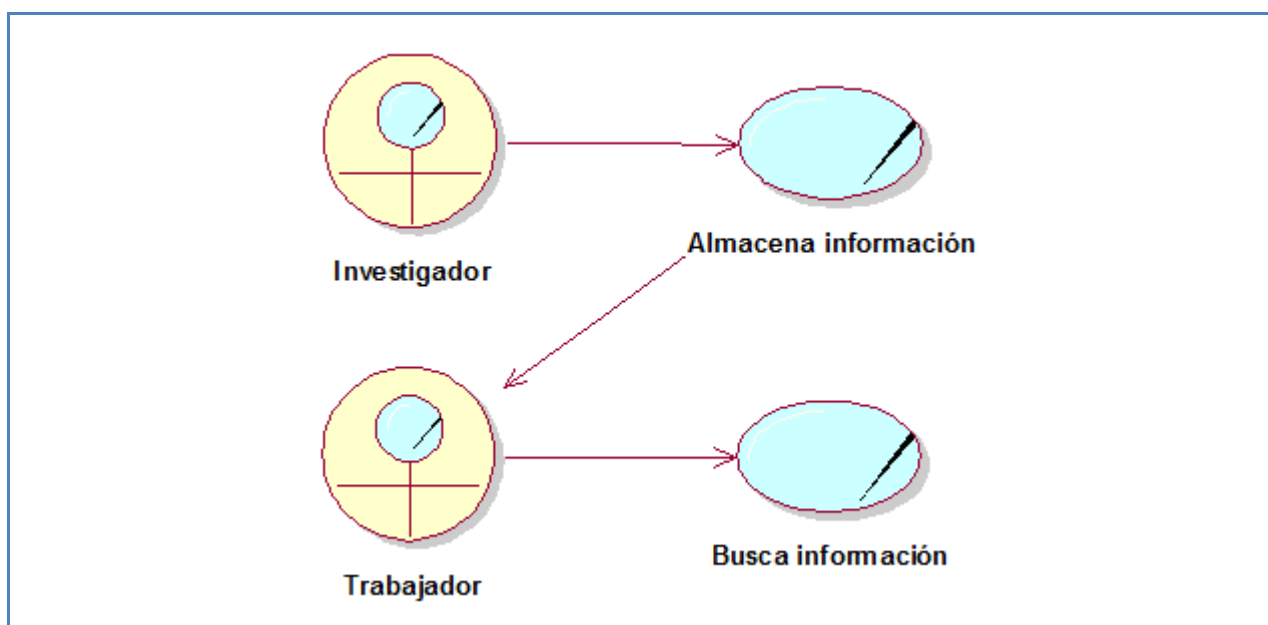


### 2.5.6 Diagrama del modelo de objetos

Un diagrama de objetos muestra un conjunto de objetos y sus relaciones en un momento determinado; los diagramas de objetos muestran el diseño o los procesos de un sistema desde el punto de vista estático. (Jacobson, 2006)

Un modelo de objetos del negocio es un modelo interno del negocio. Describe como cada caso de uso de negocio es llevado a cabo por un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y de unidades de trabajo. (Jacobson, 2006)

Diagrama 3. Diagrama del modelo de objetos



### 2.6 Requerimientos funcionales

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo. (Rumbaugh, Booch, & Jacobson, 2006) Todo esto basándose en las necesidades de usuarios y clientes.

Los requerimientos funcionales del sistema propuesto son los siguientes:

#### R1. Autenticarse

- 1.1 Validar el nombre de usuario y su clave de acceso.
- 1.2 Validar que el usuario que se autentifique tenga permisos para introducir u obtener información del sistema.

#### R2. Cambiar contraseña

- 2.1 Modificar la clave de acceso del usuario.

### **R3.** Restaurar contraseña

3.1 Cambiar la contraseña de usuario por la contraseña por defecto

### **R4.** Gestionar datos ponencia.

4.1 Crear nueva ponencia.

4.2 Modificar ponencia.

4.3 Eliminar ponencia.

### **R5.** Gestionar evento.

5.1. Crear nuevo evento.

5.2. Modificar evento.

5.3. Eliminar evento.

### **R6.** Gestionar Usuario.

6.1 Crear nuevo usuario.

6.2 Modificar usuario.

6.3 Eliminar usuario.

### **R7.** Enviar ponencia.

### **R8.** Publicar ponencia.

### **R9.** Buscar en repositorio de archivos académicos.

### **R10.** Enviar notificación.

### **R11.** Mostrar ponencia.

### **R12.** Mostrar lista de usuarios.

### **R13.** Mostrar ranking de categorías con más ponencias.

### **R14.** Mostrar ranking de ponencias más revisadas.

### **R15.** Mostrar usuarios conectados.

### **R16.** Mostrar resultados de las búsquedas.

## **2.7 Requerimientos no funcionales**

Los requerimientos no funcionales describen las restricciones del sistema o del proceso de desarrollo; no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las

restricciones del sistema como la capacidad de los dispositivos de entrada/salida, en cuanto a prestaciones, atributos de calidad y la representación de datos que se utiliza en la interfaz del sistema. Estos son:

### **Apariencia o interfaz externa.**

- ✓ La interfaz del sistema debe ser a través de una página web dinámica y personalizada de acuerdo al tipo de usuario que acceda al sistema.
- ✓ 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 un punto a otro dentro de ella con gran facilidad. Se cuidará porque la aplicación sea lo más interactiva posible. La interfaz será uniforme logrando identificarla como un mismo sistema.

### **Usabilidad.**

- ✓ Los usuarios del sistema quedan definidos por los trabajadores del CEEPI, dependiendo de su funcionalidad.
- ✓ Serán los trabajadores del propio CEEPI quienes utilicen más este software.

### **Rendimiento.**

- ✓ Se concibe un sistema diseñado sobre la arquitectura de tres capas.

### **Soporte.**

- ✓ Se requiere un servidor de bases de datos con soporte para grandes volúmenes de información, velocidad de procesamiento y tiempo de respuesta rápido en accesos concurrentes.
- ✓ El administrador tendrá la responsabilidad de mantener actualizada la aplicación.
- ✓ Las pruebas del sistema se realizarán en el CEEPI. Dichas pruebas permitirán evaluar en la práctica la funcionalidad y las ventajas de este nuevo producto.

### **Portabilidad.**

- ✓ La plataforma seleccionada para desarrollar la aplicación fue Windows, pero puede ser ejecutada desde otras plataformas como Linux, entre otras que soporten el lenguaje PHP, JavaScript y MySQL.

### **Requisitos de seguridad:**

- ✓ Debe garantizar la conectividad e integridad de los datos almacenados a través de la red. Esto está garantizado por el Sistema Operativo.
- ✓ Debe garantizar la confidencialidad para proteger la información de acceso no autorizado. Esto estará garantizado por el Sistema Gestor de Base de Datos.
- ✓ El sistema impondrá un estricto control de acceso que permitirá a cada usuario tener disponible solamente las opciones relacionadas con su actividad.
- ✓ La información deberá estar disponible a los usuarios en todo momento, limitada solamente por las restricciones que estos tengan de acuerdo con la política de seguridad del sistema.

### **Software.**

- ✓ Del lado del servidor:

Se debe disponer de un sistema operativo compatible, para la instalación de la aplicación y debe ser instalado el Apache como servidor web, el PHP (versión 5.3 o superior) como lenguaje de programación del lado del servidor y MySQL como Sistema Gestor de Base de Datos para garantizar la integridad referencial.

- ✓ Del lado del cliente:

Se requiere un navegador que interprete las funciones básicas de Java Script, recomendado Mozilla Firefox.

### **Hardware.**

#### ✓ Servidor:

La máquina servidora debe tener como mínimo las siguientes características de hardware: Procesador Pentium III 1 GHz de frecuencia o superior, 512 Mb de memoria RAM y 40 Gb de capacidad de disco duro.

#### ✓ Cliente:

Las computadoras situadas en los puestos de trabajo de los usuarios requerirán como mínimo un procesador Pentium III, 256 Mb de memoria RAM. Estas máquinas deben estar conectadas en red con el servidor a través de una tarjeta de red de 100 Mbps.

## **2.8 Modelo de casos de uso del sistema**

El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. Describe lo que hace el sistema para cada tipo de usuario y proporciona la entrada fundamental para el análisis, el diseño y las pruebas. (Rumbaugh, Booch, & Jacobson, 2006)

### **2.8.1 Actores del sistema**

Un actor es aquel que interactúa con el sistema, sin ser parte de él y puede asumir el rol que juega una o varias personas, un equipo o un sistema automatizado. (Pressman, 2007)

A continuación se definen y se muestran en la tabla siguiente los actores identificados en el sistema propuesto:



Tabla 4. Actores del sistema

Actor del sistema	Justificación
<b>Administrador</b>	Prepara el sistema para su puesta en marcha y lo actualiza. Tiene todos los permisos para trabajar con el sistema por tanto está relacionado con todos los casos de uso.
<b>Trabajador</b>	Son aquellos que podrán solicitar los servicios de la aplicación, además de realizar las búsquedas y enviar las ponencias para su posterior publicación.

### 2.8.2 Diagramas de casos de uso del sistema

Los actores interactúan y usan el sistema a través de casos de uso. Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. (Pressman, 2007)

Subdividir los casos de uso en paquetes resulta de mucha ayuda en la modelación de cualquier sistema informático. Los paquetes son un mecanismo de organización de elementos que subdividen el modelo en otros más pequeños que colaboran entre sí. (Pressman, 2007)

#### 1 Paquete Administración

- 1.1 Gestionar usuario
- 1.2 Restaurar contraseña
- 1.3 Publicar ponencia

#### 2 Paquete Gestión

- 2.1 Enviar notificación
- 2.2 Gestionar datos ponencia
- 2.3 Gestionar evento
- 2.4 Buscar en repositorio de archivos académicos
- 2.5 Enviar ponencia

#### 3 Paquete Seguridad

- 3.1 Autenticarse

## 3.2 Cambiar contraseña

### 4 Paquete Reportes

4.1 Mostrar ponencias

4.2 Mostrar lista de usuarios

4.3 Mostrar ranking de categorías con más ponencias

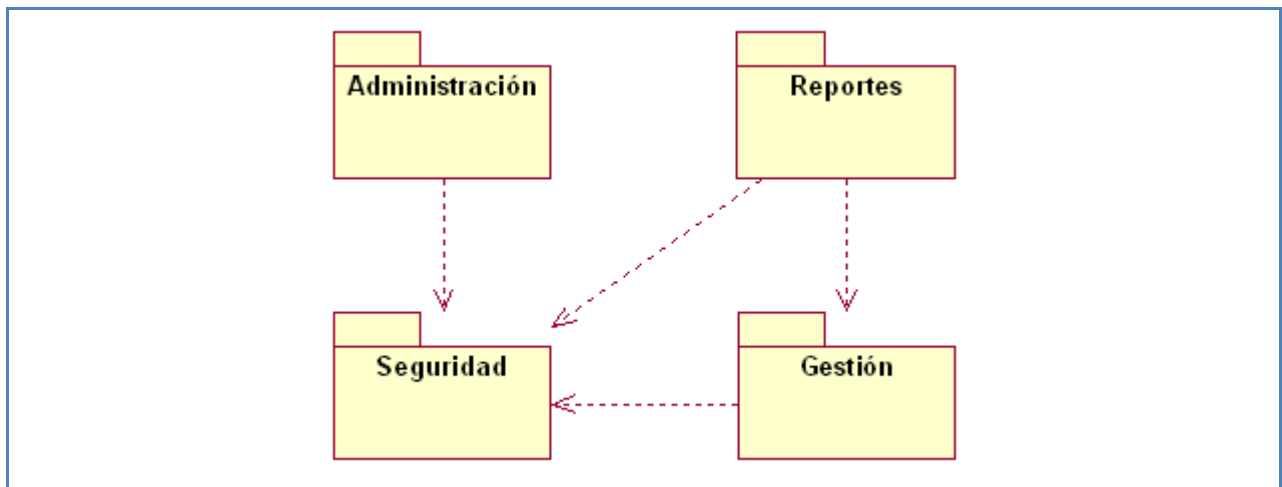
4.4 Mostrar ranking de ponencias más revisadas

4.5 Mostrar usuarios conectados

4.6 Mostrar resultados de las búsquedas

El siguiente diagrama muestra las relaciones existentes entre los paquetes.

Diagrama 4. Relaciones entre los paquetes



A continuación se muestran los diferentes diagramas de casos de uso del sistema asociado o perteneciente a cada paquete.

Diagrama 5. Paquete de seguridad.

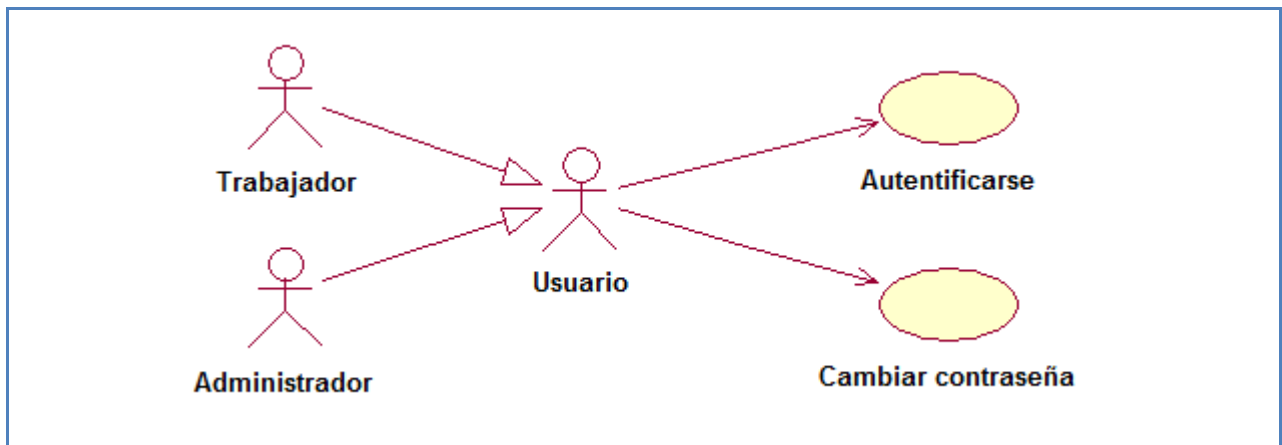


Diagrama 6. Paquete de administración

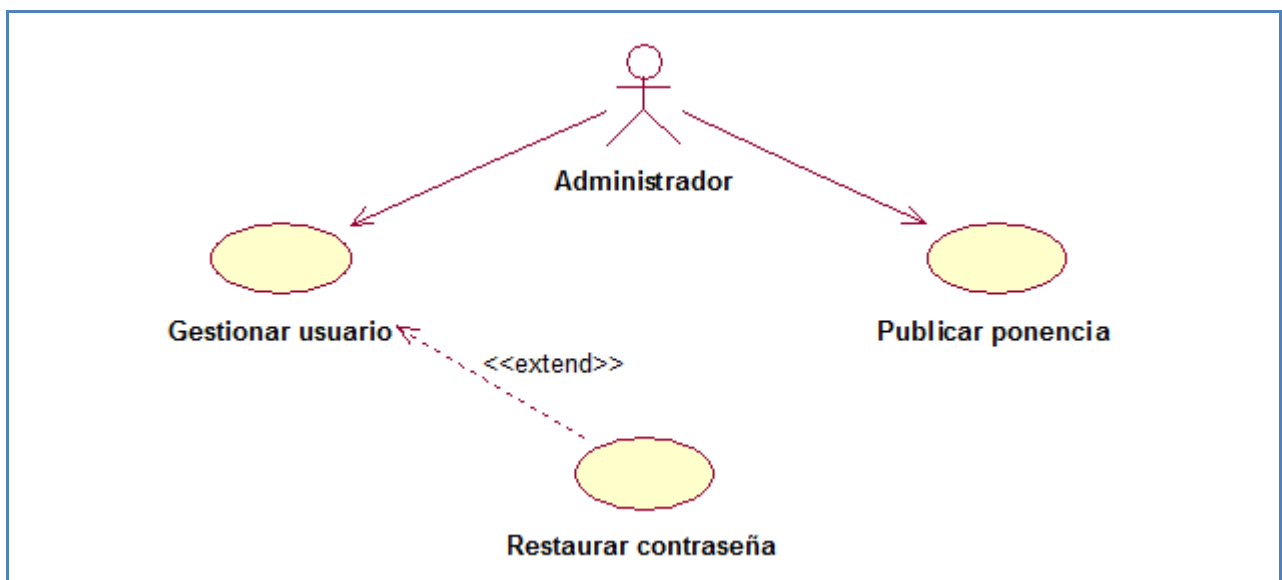


Diagrama 7. Paquete de gestión

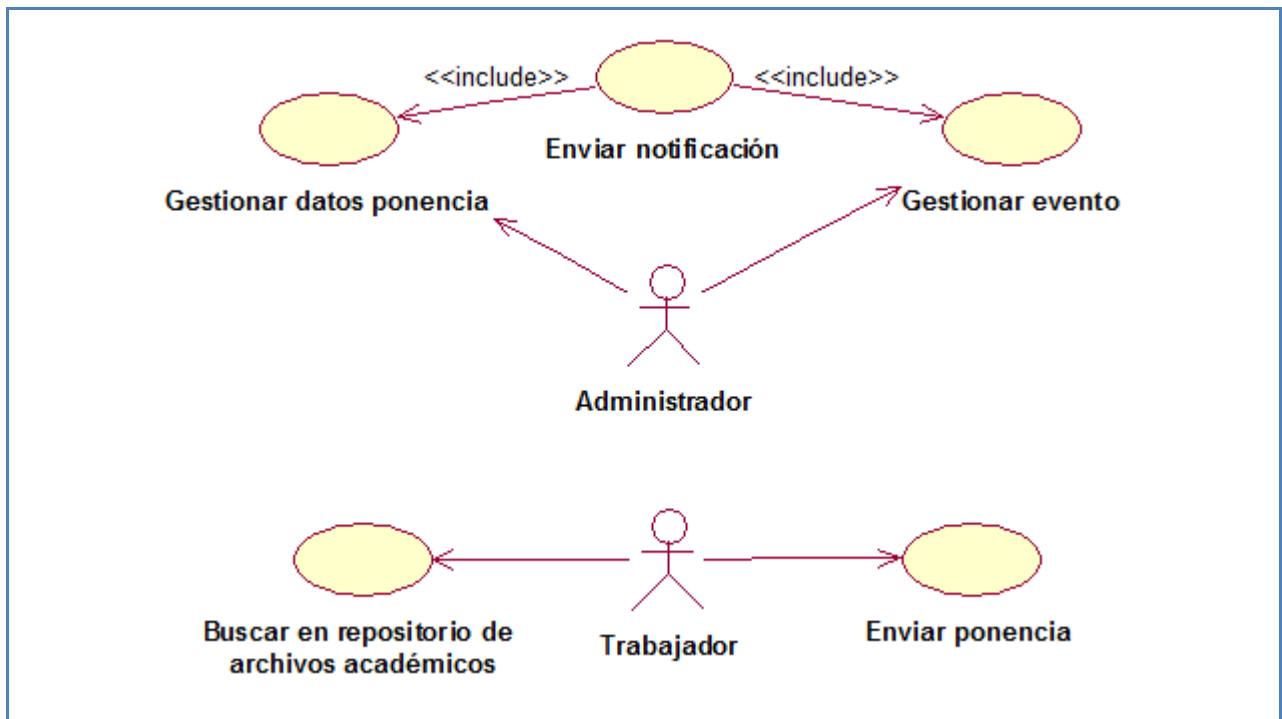
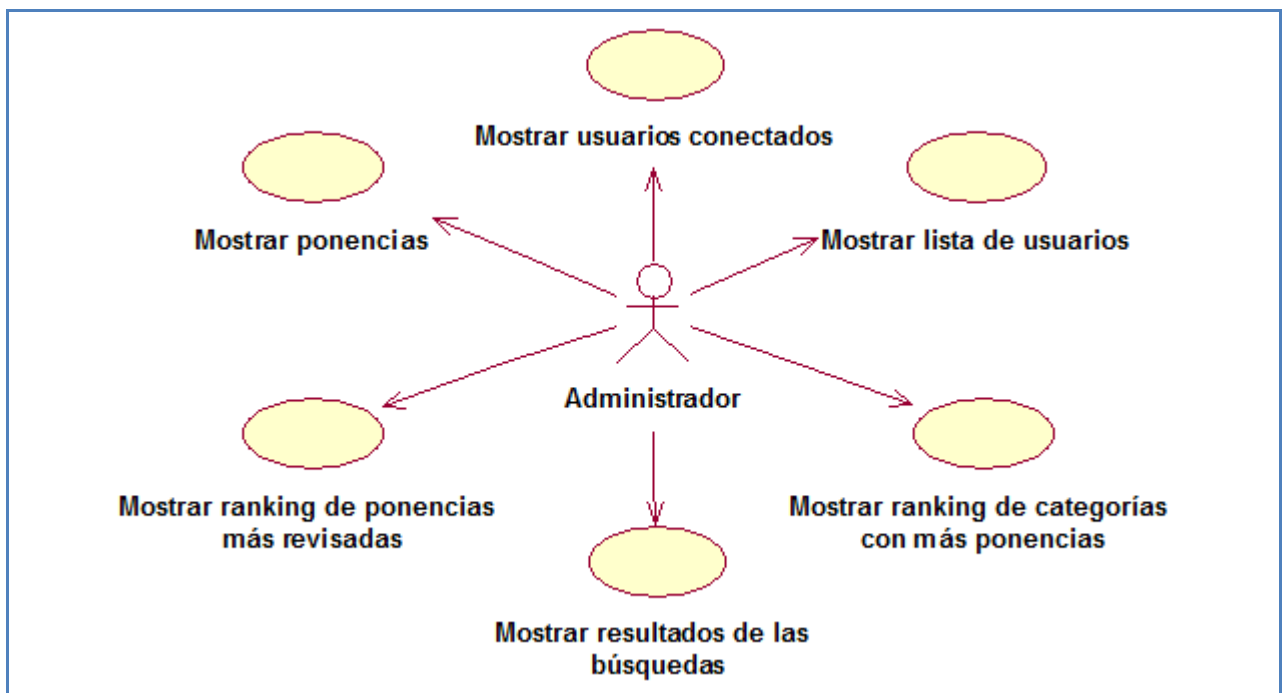


Diagrama 8. Paquete de reportes



### **2.9 Conclusiones**

En el presente capítulo se hizo un análisis del negocio, estudiándose los procesos que lo conforman, permitiendo así modelarlo mediante reglas, casos de uso, actores y trabajadores que en él participan. Quedando definidos los principales requerimientos del sistema y los casos de uso, que luego se convertirán en módulos de programación. Se definieron además los actores del sistema y con cuáles requerimientos funcionales se relacionan.

# 3

## Capítulo

### Construcción de la solución propuesta

#### 3.1 Introducción

En este capítulo se plasman los resultados de las etapas de diseño e implementación del sistema, utilizando UML para su modelado. Se describen los casos de uso del sistema, se elaboran los diagramas de clases de diseño y se representa el diseño de la base de datos a través del diagrama de clases persistentes y el modelo de datos. Además se describe la implementación a través del diagrama de despliegue y del diagrama de componentes.

#### 3.2 Descripción de los casos de uso del sistema

En las siguientes tablas se muestran las descripciones de los casos de uso del sistema agrupados por paquetes.

##### Paquete Administración

Tabla 5. Descripción del caso de uso del sistema “Gestionar usuario”

Caso de uso	Gestionar usuario
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite al administrador el trabajo con los diferentes usuarios creación de nuevos usuarios, modificación y eliminación de los mismos.
<b>Resumen</b>	

El caso de uso se inicia cuando el administrador necesita insertar, actualizar o eliminar usuarios y restaurar contraseñas, la aplicación muestra los usuarios existentes y opciones, el administrador selecciona alguna de estas opciones y completa el formulario, en caso de restaurar contraseña se extiende el caso de uso Restaurar contraseña, la aplicación registra la información. Culmina el caso de uso.	
<b>Referencias</b>	RF– (R6)

**Tabla 6. Descripción del caso de uso del sistema “Restaurar contraseña”**

<b>Caso de uso</b>	<b>Restaurar contraseña</b>
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite al administrador restaurar la contraseña de usuario de cualquier usuario de la aplicación que haya olvidado la misma.
<b>Resumen</b> El caso de uso se inicia cuando el administrador selecciona el usuario al que le va a restaurar la contraseña debido a que ese trabajador la olvidó, el sistema actualiza la contraseña con el valor por defecto. Culmina el caso de uso.	
<b>Referencias</b>	RF– (R3)

**Tabla 7. Descripción del caso de uso del sistema “Publicar ponencia”**

<b>Caso de uso</b>	<b>Publicar ponencia</b>
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite al administrador el trabajo con las ponencias, que en caso de estar correctas, las publica en la aplicación para que todos los usuarios tengan acceso a ellas.
<b>Resumen</b> El caso de uso se inicia cuando el administrador revisa los datos de las ponencias enviadas por los usuarios, que en caso de estar correctos, la aplicación muestra la opción de publicar, la cual el administrador selecciona. Culmina el caso de uso.	
<b>Referencias</b>	RF– (R8)

**Paquete Gestión**

**Tabla 8. Descripción del caso de uso del sistema “Gestionar datos ponencia”**

<b>Caso de uso</b>	<b>Gestionar datos ponencia</b>
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite al administrador el trabajo con las diferentes ponencias, creación de nuevas ponencias, modificación y eliminación de las mismas.
<b>Resumen</b> El caso de uso se inicia cuando el administrador necesita insertar, actualizar o eliminar ponencias, la aplicación muestra las ponencias existentes y opciones, el administrador selecciona alguna de estas opciones y completa el formulario. Culmina el caso de uso.	
<b>Referencias</b>	RF– (R4)

**Tabla 9. Descripción del caso de uso del sistema “Gestionar evento”**

<b>Caso de uso</b>	<b>Gestionar evento</b>
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite al administrador el trabajo con los diferentes eventos, creación de nuevos eventos, modificación y eliminación de los mismos.
<b>Resumen</b> El caso de uso se inicia cuando el administrador necesita insertar, actualizar o eliminar eventos, la aplicación muestra los eventos existentes y opciones, el administrador selecciona alguna de estas opciones y completa el formulario. Culmina el caso de uso.	
<b>Referencias</b>	RF– (R5)

**Tabla 10. Descripción del caso de uso del sistema “Enviar notificación”**

<b>Caso de uso</b>	<b>Enviar notificación</b>
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite al administrador notificar a todos los usuarios registrados cuando la ponencia es publicada.



<b>Resumen</b>	
El caso de uso se inicia cuando el administrador envía una notificación a través del sistema, a todos los usuarios registrados, comunicándoles que ha sido publicada una nueva ponencia. Culmina el caso de uso.	
<b>Referencias</b>	RF– (R10)

Tabla 11. Descripción del caso de uso del sistema “Buscar en repositorio de archivos académicos”

<b>Caso de uso</b>	<b>Buscar en repositorio de archivos académicos</b>
<b>Actores</b>	Trabajador
<b>Propósito</b>	Permite al trabajador buscar ponencias en la base de datos del sistema.
<b>Resumen</b>	
El caso de uso se inicia cuando el trabajador se dirige a la barra de búsqueda, escribiendo lo que desea buscar. En caso de coincidencia el sistema muestra la lista con todos los trabajos que contiene dicha información. Culmina el caso de uso.	
<b>Referencias</b>	RF– (R9)

Tabla 12. Descripción del caso de uso del sistema “Enviar ponencia”

<b>Caso de uso</b>	<b>Enviar ponencia</b>
<b>Actores</b>	Trabajador
<b>Propósito</b>	Enviar ponencia para publicación.
<b>Resumen</b>	
El caso de uso se inicia cuando el trabajador llena un formulario con los datos de la ponencia y la envía a través del sistema, para su posterior publicación. El administrador la revisa, y si cumple con todos los requisitos, entonces se publica en el sistema para que todos tengan acceso a ella. Culmina el caso de uso.	
<b>Referencias</b>	RF– (R7)

**Paquete Seguridad**

Tabla 13. Descripción del caso de uso del sistema “Autenticarse”

Caso de uso	Autenticarse
Actores	Usuario
Propósito	Iniciar sesión
<b>Resumen</b> El caso de uso inicia cuando el usuario procede a entrar en el sistema. Para ello el sistema muestra un formulario de autenticación. El usuario introduce su nombre de usuario y su clave de acceso, a continuación la aplicación verifica la información. Si es válida, el usuario podrá acceder a la aplicación de acuerdo a sus permisos, en caso contrario el sistema muestra un mensaje de error denegando el acceso. Culmina el caso de uso.	
Referencias	RF– (R1)

Tabla 14. Descripción del caso de uso del sistema “Cambiar contraseña”

Caso de uso	Cambiar contraseña
Actores	Usuario
Propósito	Cambiar contraseña de acceso al sistema
<b>Resumen</b> El caso de uso se inicia cuando el usuario luego de haber iniciado sesión, necesita cambiar la contraseña de acceso al sistema; el sistema muestra formulario donde el usuario escribe la nueva contraseña y la confirma, el sistema actualiza la base datos. Culmina el caso de uso.	
Referencias	RF– (R2)

**Paquete Reportes**

Tabla 15. Descripción del caso de uso del sistema “Mostrar ponencias”

Caso de uso	Mostrar ponencias
Actores	Administrador

<b>Propósito</b>	Mostrar las ponencias existentes.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador desea conocer la cantidad de ponencias existentes en el sistema, con un parámetro de búsqueda determinado; ya sea por año, por categoría, por autor, etc. Para esto se dirige a la barra de búsqueda y escribe en la misma lo que desea buscar. En caso de coincidencia el sistema muestra un listado con el resultado de la búsqueda. Culmina el caso de uso.
<b>Referencias</b>	RF- (R11)

Tabla 16. Descripción del caso de uso del sistema “Mostrar lista de usuarios”

<b>Caso de uso</b>	<b>Mostrar lista de usuarios</b>
<b>Actores</b>	Administrador
<b>Propósito</b>	Mostrar los usuarios existentes en el sistema.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador desea conocer la cantidad de usuarios existentes en el sistema, ya sea por nombre o por categoría científica, para ello se dirige al panel de administración y selecciona la opción “Usuarios”. El sistema muestra la lista de usuarios con el nombre y apellido, además del e-mail de cada uno. Culmina el caso de uso.
<b>Referencias</b>	RF- (R12)

Tabla 17. Descripción del caso de uso del sistema “Mostrar ranking de categorías con más ponencias”

<b>Caso de uso</b>	<b>Mostrar ranking de categorías con más ponencias</b>
<b>Actores</b>	Administrador
<b>Propósito</b>	Muestra las categorías más publicadas.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador desea conocer la categoría con más trabajos publicados, para ello el sistema muestra una lista con las categorías con más trabajos publicados. Culmina el caso de uso.

<b>Referencias</b>	RF– (R13)
--------------------	-----------

Tabla 18. Descripción del caso de uso del sistema “Mostrar ranking de ponencias más revisadas”

<b>Caso de uso</b>	<b>Mostrar ranking de ponencias más revisadas</b>
<b>Actores</b>	Administrador
<b>Propósito</b>	Muestra las ponencias más revisadas
<b>Resumen</b>	
El caso de uso se inicia cuando el administrador desea conocer los trabajos más revisadas por los usuarios, para ello el sistema muestra una lista con los 10 trabajos que más visitas han recibido. Culmina el caso de uso.	
<b>Referencias</b>	RF– (R14)

Tabla 19. Descripción del caso de uso del sistema “Mostrar usuarios conectados”

<b>Caso de uso</b>	<b>Mostrar usuarios conectados</b>
<b>Actores</b>	Administrador
<b>Propósito</b>	Muestra los usuarios conectados
<b>Resumen</b>	
El caso de uso se inicia cuando el administrador desea conocer los usuarios que están conectados al sistema en ese momento, para ello el sistema muestra una lista con todos los usuarios que se encuentran trabajando en el sistema. Culmina el caso de uso.	
<b>Referencias</b>	RF– (R15)

Tabla 20. Descripción del caso de uso del sistema “Mostrar resultados de las búsquedas”

<b>Caso de uso</b>	<b>Mostrar resultados de las búsquedas</b>
<b>Actores</b>	Administrador
<b>Propósito</b>	Muestra un listado por cada búsqueda
<b>Resumen</b>	
El caso de uso se inicia cuando el administrador desea buscar un trabajo o un usuario,	

ingresa en la barra de búsqueda los datos, y de existir alguna coincidencia, el sistema muestra una lista con el resultado. Culmina el caso de uso.

<b>Referencias</b>	RF– (R16)
--------------------	-----------

### 3.3 Diagrama de clases web

Un diagrama de clases presenta las clases del sistema con sus relaciones (estructurales y de herencia). En el caso de las aplicaciones web, el diagrama de clases representa las colaboraciones entre las páginas, donde cada página lógica puede ser representada como una clase. Los diagramas de clases web, fueron realizados, a partir de los casos de uso definidos anteriormente. La siguiente tabla muestra una relación de donde se encuentran ubicados.

Los diagramas de clases web, fueron realizados, a partir de los casos de uso definidos anteriormente. La siguiente tabla muestra una relación de donde se encuentran ubicados.

Tabla 21. Diagrama de clases web

Diagrama de clases Web	Ubicación
Gestionar usuario	Anexos. Diagrama 12
Restaurar contraseña	Anexos. Diagrama 13
Publicar ponencia	Anexos. Diagrama 14
Enviar notificación	Anexos. Diagrama 15
Gestionar datos ponencia	Anexos. Diagrama 16
Gestionar evento	Anexos. Diagrama 17
Buscar en repositorio de archivos académicos	Anexos. Diagrama 18
Enviar ponencia	Anexos. Diagrama 19
Autenticarse	Anexos. Diagrama 20
Cambiar contraseña	Anexos. Diagrama 21
Mostrar ponencias	Anexos. Diagrama 22
Mostrar lista de usuarios	Anexos. Diagrama 23
Mostrar ranking de usuarios con más ponencias	Anexos. Diagrama 24

Mostrar ranking de ponencias más revisadas	Anexos. Diagrama 25
Mostrar usuarios conectados	Anexos. Diagrama 26
Mostrar resultados de las búsquedas	Anexos. Diagrama 27

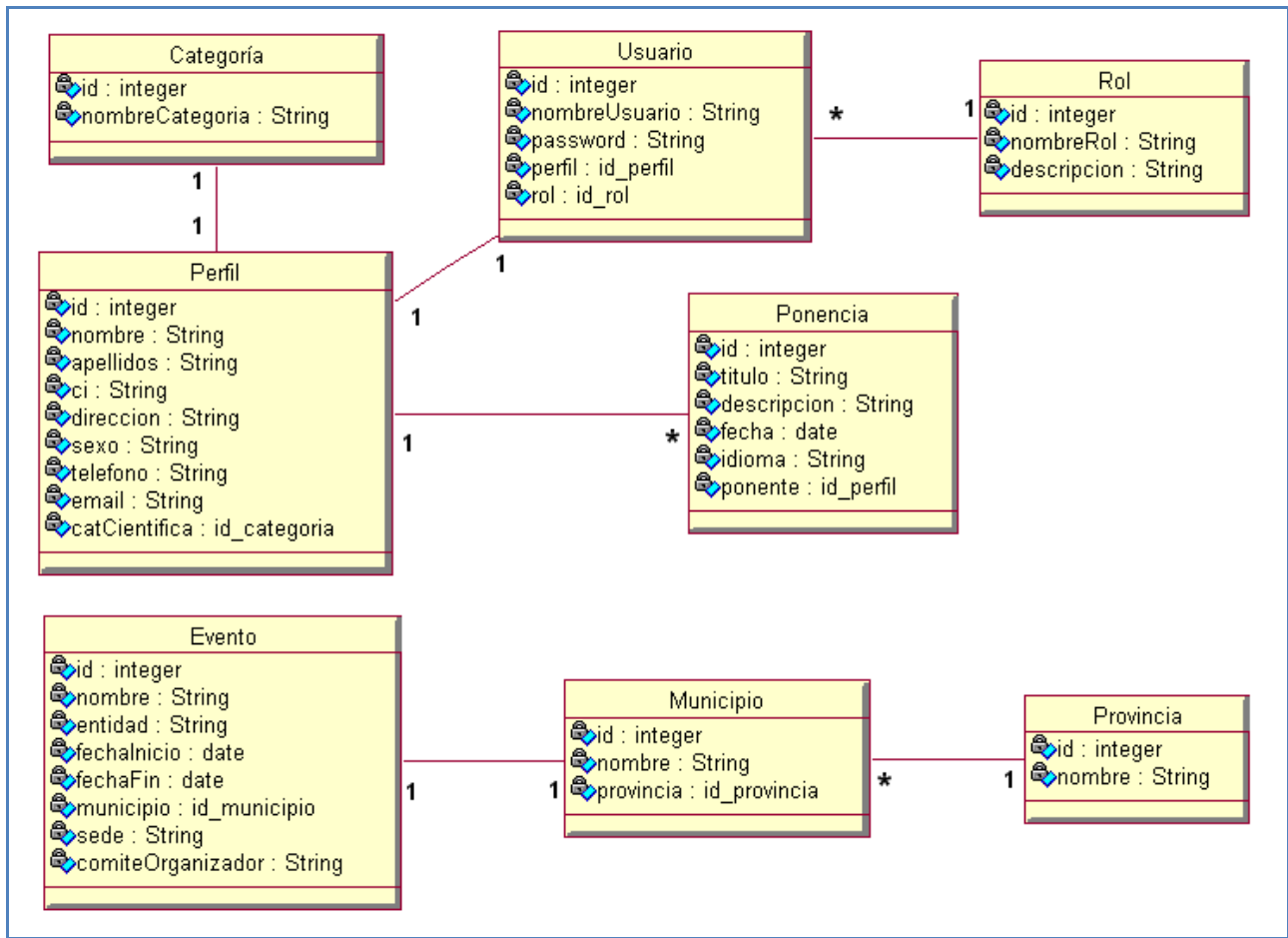
### **3.4 Diseño de la base de datos**

Cuando se va a realizar el diseño de la base de datos para un sistema dado, es necesario determinar los datos que se deben tomar en cuenta y las dependencias funcionales existentes entre ellos. Esto se obtiene luego de realizada la etapa de análisis del sistema y partiendo de lo obtenido en ésta. (Mato García, 2006)

#### **3.4.1 Diagrama de clases persistentes**

El diagrama del modelo lógico de datos o diagrama de clases persistentes, muestra las clases capaces de mantener su valor en el espacio y en el tiempo. (Jacobson, Booch, & Rumbaugh, 2006)

Diagrama 9. Diagrama de clase persistentes



### 3.4.2 Modelo de datos

Considerado el último paso en relación con los datos que utilizará un sistema de información, es la elección de la organización física que soporte los métodos de acceso a los datos establecidos anteriormente, está orientado a la forma en que se almacenarán en memoria. Se puede apreciar en el mismo los nombres de los campos, el tipo de datos y los índices de las relaciones. Es obtenido a partir del modelo de clases persistentes. (Jacobson, Booch, & Rumbaugh, 2006) Ver Anexos Diagrama 28.

## **3.5 Principios de diseño**

El diseño de la interfaz de una aplicación, el formato de los reportes, el tratamiento de errores y la seguridad, tienen gran influencia en el éxito o fracaso de una aplicación. A continuación se describen los principios de diseño seguidos para el desarrollo del sistema en cuestión.

### **3.5.1 Estándares en la interfaz de la aplicación**

La primera impresión del usuario cuando visita una aplicación web la brinda el diseño de la interfaz. Es por ello que, para lograr la apariencia adecuada y que el usuario se sienta cómodo, se tienen en cuenta varios aspectos, sobre todo relacionados con gráficos, navegación, composición del sitio, etc., que a continuación se detallan.

Para el diseño de la interfaz del sistema se tuvieron en cuenta aspectos necesarios, que garanticen la comodidad por parte del usuario, teniendo presente la organización de la información que se muestra y su distribución en la página. Los elementos que se repiten en varias páginas son ubicados en el mismo lugar permitiéndole al usuario acostumbrarse al ambiente y que éste no se vea desorientado. Éstas sólo contienen la información necesaria para el usuario, evitando que estén sobrecargadas. Las páginas son uniformes, no presentan muchos objetos que distraigan ni le roben la atención al usuario dado que es un sitio de trabajo solamente.

El vocabulario manejado es lo menos técnico posible, acercándose al utilizado por los usuarios. La navegación del sistema es sencilla. Cada ventana muestra un encabezado, que orienta al usuario sobre que se hace en ella.

### **3.5.2 Formatos de reportes**

Los reportes se obtendrán en tablas pues esto hará más comprensible la información que muestran, las tablas contendrán como máximo 30 resultados y cuando tengan más cantidad de filas, la información será paginada. Se ha concebido que cada reporte tenga un encabezado que lo identifica, este estará centrado y con una letra de tamaño mayor que el resto de la información y en negrita. La información será organizada en tablas.



### **3.5.3 Tratamiento de errores**

El diseño e implementación del sistema facilita que las posibilidades de introducir información errónea por parte del usuario sean mínimas, pues en muchas ocasiones él selecciona elementos de la pantalla, lo que provoca que los errores se mantengan a un nivel de validación de la información a través de funciones o código JavaScript. En el caso de que el usuario teclee la información, los campos están validados y los errores se le comunican a través de mensajes de error en pantalla o cuadros de alerta, los que se muestran en un lenguaje de fácil comprensión para los usuarios.

### **3.5.4 Seguridad**

El sistema mantiene un mecanismo de seguridad y protección, basado en usuario y contraseña para el acceso al mismo. Estos son creados de acuerdo a una política de usuarios con privilegios de acuerdo a distintos perfiles, definiendo los niveles de acceso que tiene cada uno. Se debe mantener una traza que permita saber cuando entraron los usuarios al sistema y a que funcionalidades accedieron. La información almacenada es consistente y se utilizan validaciones que limitan la entrada de datos erróneos.

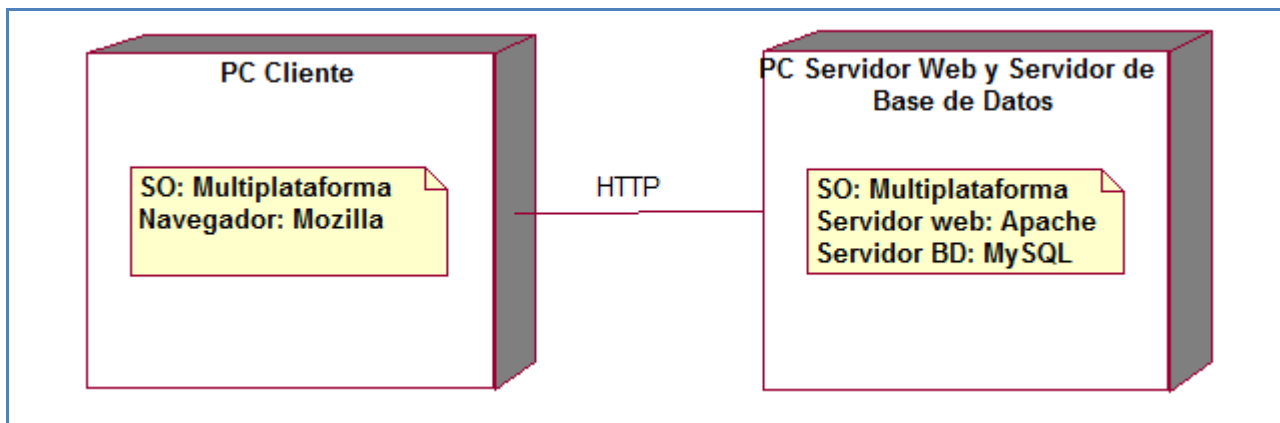
### **3.6 Diagrama de despliegue**

Un diagrama de despliegue es un diagrama que muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. (Jacobson, Booch, & Rumbaugh, 2006)

Para el desarrollo de la aplicación se utilizó la arquitectura de tres capas a través del patrón de diseño Modelo-Vista-Controlador.

La siguiente figura muestra el diagrama de despliegue de la aplicación:

Diagrama 10. Diagrama de despliegue

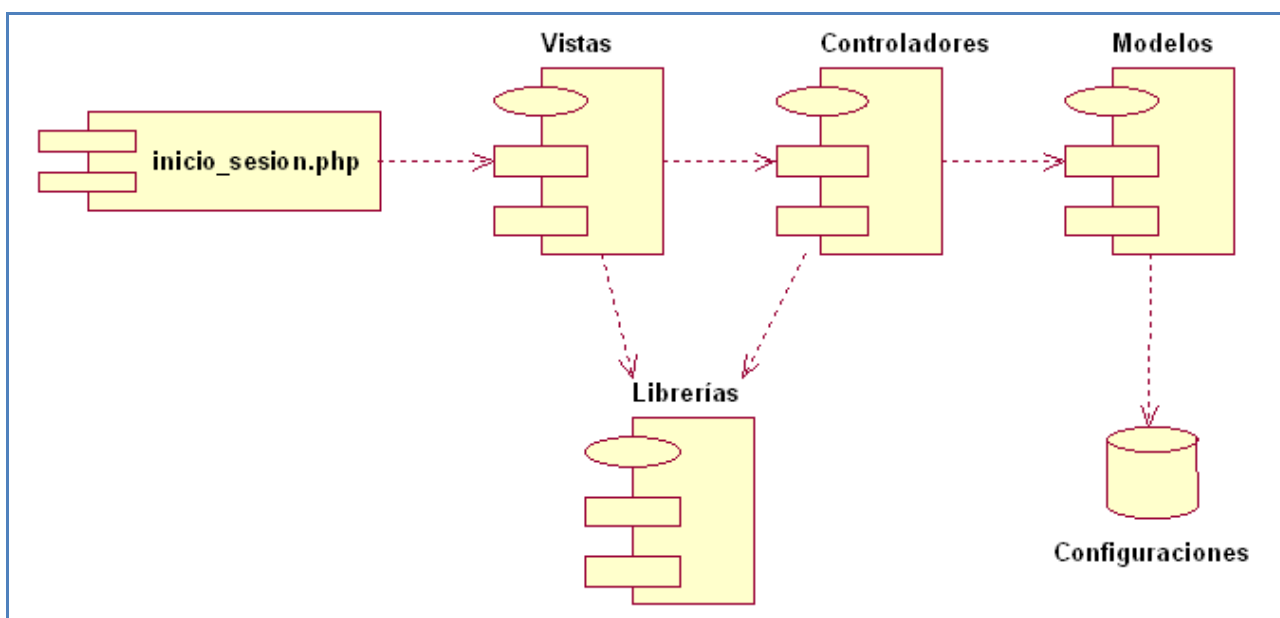


### 3.7 Diagrama de componentes

Un diagrama de componentes muestra la organización y las dependencias entre un conjunto de componentes. Los diagramas de componentes cubren la vista de implementación estática de un sistema. Se relacionan con los diagramas de clases en que un componente se corresponde, por lo común, con una o más clases, interfaces o colaboraciones. (Jacobson, Booch, & Rumbaugh, 2006)

La siguiente figura muestra el diagrama de componentes:

Diagrama 11. Diagrama de componentes



En la siguiente tabla se muestra una pequeña descripción de cada uno de los paquetes de componentes.

**Tabla 22. Descripción de los paquetes de componentes**

Componentes	Descripción
Paquete Vistas	Paquete que agrupa los ficheros de las clases de interfaz del usuario para los diferentes perfiles, al igual que los reportes.
Paquete Controladores	Paquete que agrupa los ficheros de las clases controladoras, encargadas de manipular la lógica del negocio y que actúan de intermediaria entre las clases del paquete vista y las del modelo.
Paquete Modelos	Este paquete contiene los ficheros de las clases de acceso a datos que contienen las consultas a la base de datos.
Librerías	Este paquete contiene las librerías para la validación, generación de tablas HTML, disponibles en el framework Symfony, además de las librerías JavaScript utilizadas para las interfaces que se muestran.
Configuración	Representa el Sistema Gestor de Base de Datos donde se almacena toda la información.

### 3.8 Conclusiones

En el presente capítulo se describieron los casos de uso del sistema, se mostraron los resultados de la etapa de implementación del sistema. Se desarrollaron los diagramas de clases web, el diseño de la base de datos, el diagrama de componentes y el de despliegue, además se definieron una serie de políticas que se tienen en cuenta para el diseño de la interfaz, el tratamiento de los posibles errores, así como la seguridad y protección de la aplicación. Todo ello con el objetivo de lograr una herramienta automatizada de trabajo que cumpla con las expectativas del cliente.

## *Conclusiones generales*

Con la realización de la presente investigación se arribaron a las siguientes conclusiones:

- 1 El estudio de los fundamentos teóricos y metodológicos para la elaboración de un software que contribuya al almacenamiento y recuperación de información en formato digital en el Centro de Estudios Energéticos y Procesos Industriales (CEEPI), de la Universidad de Sancti Spíritus “José Martí Pérez” permitió determinar que la metodología RUP es la adecuada por ser un proceso bien definido, que junto a UML, constituye la metodología más utilizada para el análisis, diseño, implementación y documentación de la solución propuesta.
- 2 Se seleccionaron como lenguaje de programación para la lógica del negocio PHP y como SGBD MySQL, por ser tecnologías de código libre y tener la característica de integrarse para funcionar en cualquier plataforma.
- 3 Se diseñó sustentado en las bases del software libre, la programación orientada a objeto y la arquitectura en tres capas lo cual le brinda al sistema una mayor flexibilidad y capacidad de mantenimiento.
- 4 Se implementó un software utilizando el lenguaje de programación PHP con funcionalidades que se ajustan a las necesidades del cliente y que brinda al usuario la posibilidad de interactuar de manera dinámica con los datos empleando la arquitectura de red cliente-servidor de manera que se pueda acceder a la información desde cualquier máquina.

## *Recomendaciones*

Al concluir esta investigación se hacen las siguientes recomendaciones:

- 1 Evaluar el software luego de su introducción en el CEEPI.
- 2 Incorporar imagen al perfil del investigador.
- 3 Realizar el análisis correspondiente con los usuarios para incorporar en versiones futuras, salidas que puedan ser de interés y que actualmente no fueron concebidas.

- Bologna, J., & Walsh, A. M. (1997). *The Accountant's Handbook of Information Technology [El Manual de la Historia Informática]* (Vol. I). John Wiley and Sons.
- Catalani, E. (2007). *ARQUITECTURA Modelo/Vista/Controlador*. Extraído de <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>
- Chappell, D., & Kirk, S. (2006). *De N-capas a .NET. Desarrollo de aplicaciones*. Extraído de <http://www.microsoft.com/spanish/msdn/articulos/archivo/081102/voices/dncapas.asp>
- Concepto de gestión. (1924). *Enciclopedia Universal Ilustrada Europeo Americana*.
- Cuenca, C. L. (2006). *Arquitectura del servidor Apache*. Extraído de <http://www.desarrolloweb.com/articulos/1112.php>
- Free Software Foundation. (2008). *Free software definition [Definición del software libre]*. Extraído de <http://www.gnu.org/philosophy/free-sw.html>.
- Gutiérrez, J. J. (2006). *¿Qué es un framework web?* Extraído de <http://jordisan.net/blog/2006/que-es-un-framework/> ¿Qué es un framework?
- Hurtado Jara, O. (2008). *Sistemas distribuidos*. Extraído de Monografías: <http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtmls>
- Jacobson, I., Booch, G., & Rumbaugh, J. (2006). *El lenguaje Unificado de Modelado* (Vol. I). La Habana: Félix Varela.
- Jacobson, I., Booch, G., & Rumbaugh, J. (2006). *El Proceso Unificado de Desarrollo de Software* (Vol. I). La Habana: Félix Varela.
- Johnson, A. (2007). *MySQL*. Extraído de Software Libre: [http://softwarelibre.net63.net/?page\\_id=69](http://softwarelibre.net63.net/?page_id=69)
- Johnson, A. (2007). *PostgreSQL*. Extraído de Software Libre: [http://softwarelibre.net63.net/?page\\_id=69](http://softwarelibre.net63.net/?page_id=69)
- Martínez, P. (2007). *Patrones*. Extraído de Patrón "Modelo-Vista-Controlador": <http://www.proactiva-calidad.com/java/patrones/mvc.html>
- Mato García, R. M. (2006). *Sistema de Base de Datos*. La Habana: Editorial Félix Varela.

- Molina, M., & Castro, A. (2009). *Cliente - servidor*. Extraído de <http://deymar-clienteservidor.blogspot.com/2009/04/si-bien-la-clasica-arquitectura-cs.html>
- Peláez, J. (2009). *Arquitectura basada en capas*. Extraído de <http://geeks.ms/blogs/jkpelaez/archive/2009/05/29/arquitectura-basada-en-capas.aspx>
- Pérez Sandoval, H. (2009). *Programación por capas*. Extraído de MTY.Coders. Comunidad de Desarrolladores de Monterrey: <http://mtycoders.com/programacion-por-capas/>
- Pérez Valdés, D. (2007). *Los Frameworks de PHP agilizan tu trabajo*. Extraído de <http://www.maestrosdelweb.com/principiantes/los-framework-de-php-agilizan-tu-trabajo/>
- Pressman, R. S. (2007). *Ingeniería del Software Un enfoque práctico*. La Habana: Editorial Félix Varela.
- Riveros, F. (2008). *Gestor de Base de Datos: MySQL, PostgreSQL, SQLite*. Extraído de [http://www.eaprende.com/base\\_de\\_datos\\_SQL\\_Server\\_con\\_PHP\\_y\\_ADODB.html](http://www.eaprende.com/base_de_datos_SQL_Server_con_PHP_y_ADODB.html)
- Rodríguez Febles, J. (2005). *Sistema Informático para la Gestión Integral de Comedores (SISCOMED)*. CUJAE. (UH): Trabajo de Diploma.
- Sandoval Méndez, S. (2009). *Gestores de Bases de Datos*. Extraído de <http://www.chacharaselnido.com/ITVG/GBD%20EXPO.pptx>
- Una introducción a Apache. (2006). *Una introducción a Apache*. Extraído de [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/)
- Webmaster. (2009). *HTML, lenguaje de Hipertexto*. Extraído de Mundo Webmaster: <http://mundowebmaster.wordpress.com/2009/02/21/html-lenguaje-de-hipertexto/>
- Woodman, L. (1985). Information management in large organizations. En W. L, *Information management from strategies to action*. London: ASLIB.

Diagrama 12. Diagrama de clases web “Gestionar usuario”

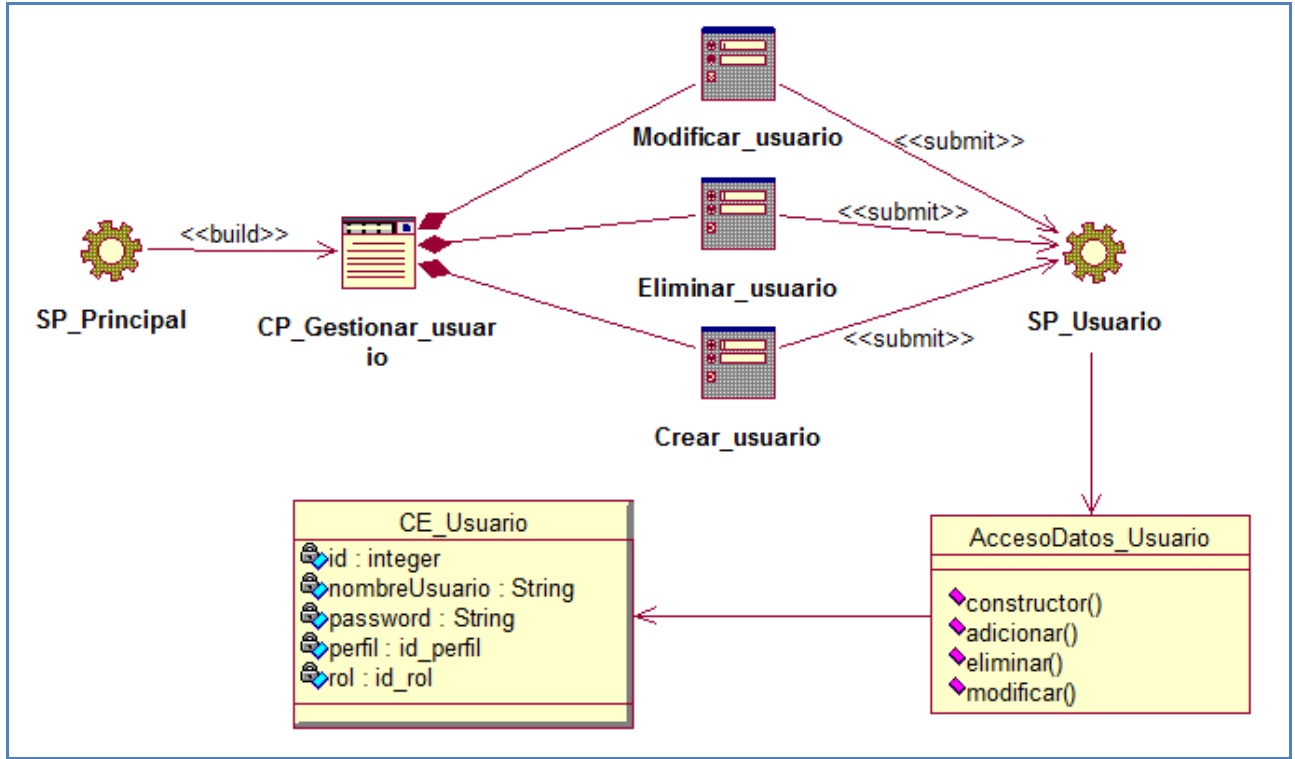


Diagrama 13. Diagrama de clases web “Restaurar contraseña”

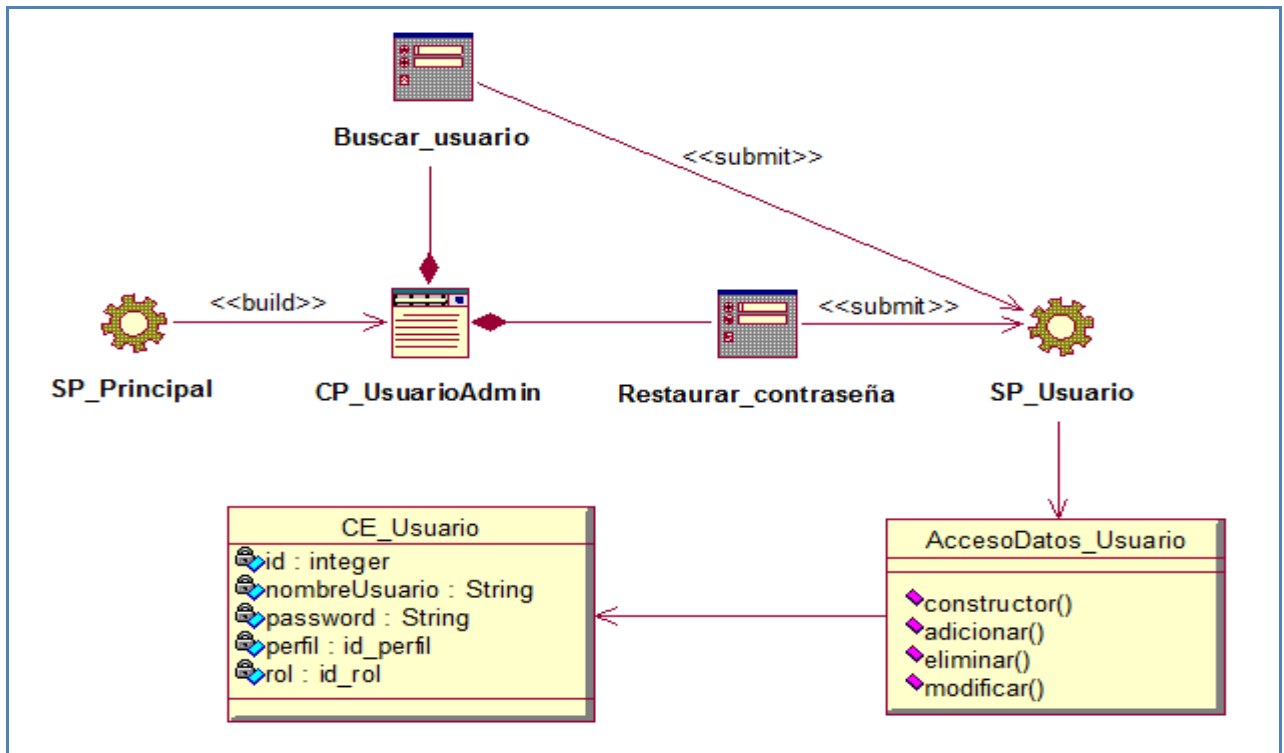




Diagrama 14. Diagrama de clases web “Publicar ponencia”

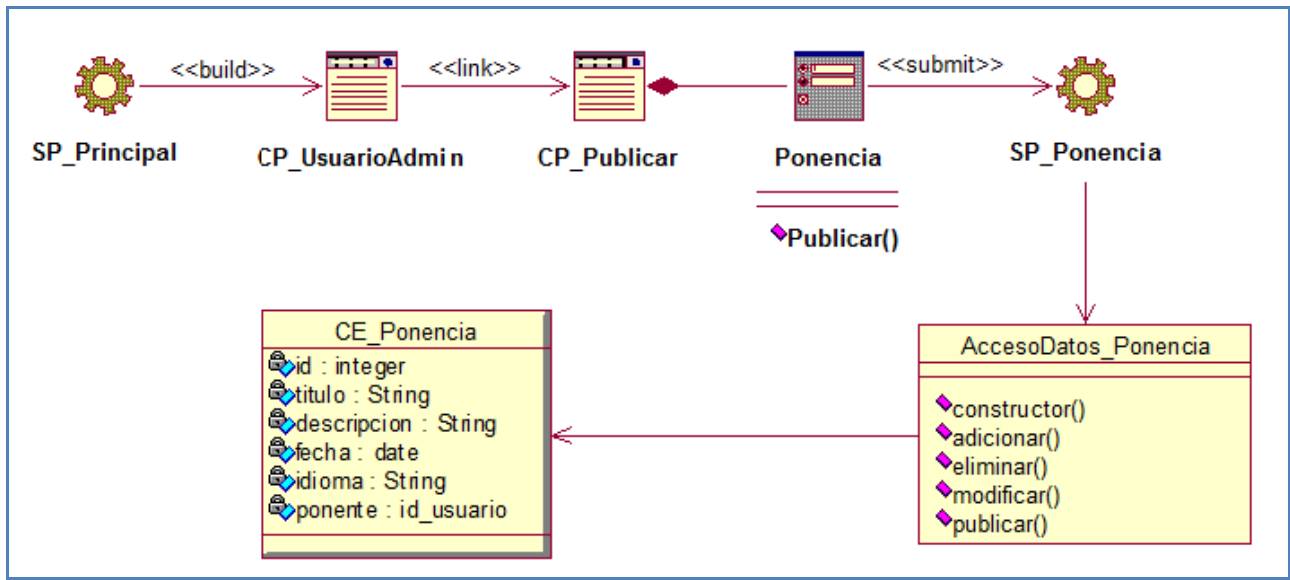


Diagrama 15. Diagrama de clases web “Enviar notificación”

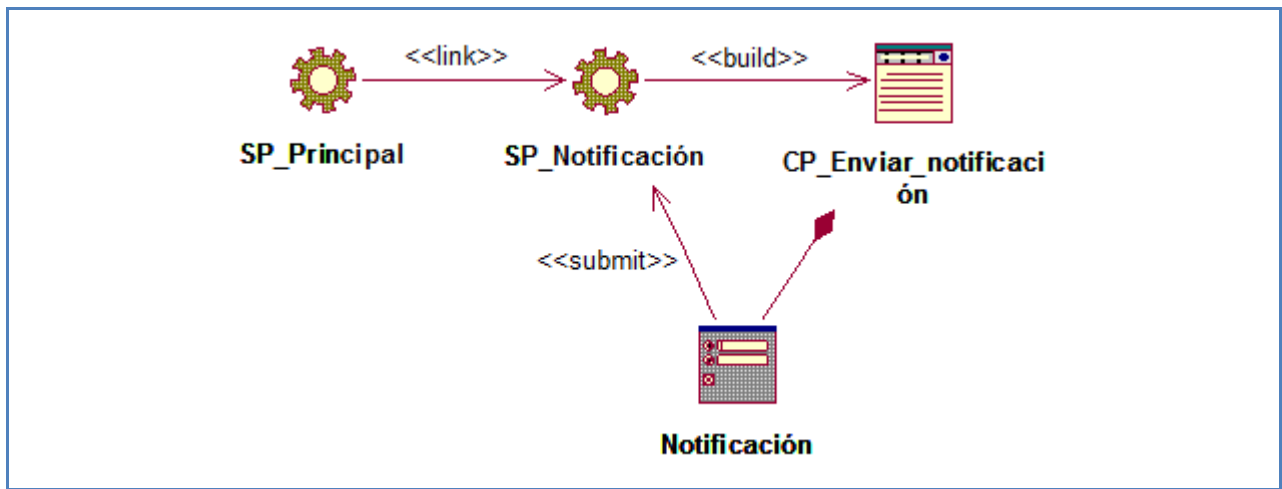


Diagrama 16. Diagrama de clases web “Gestionar datos ponencia”

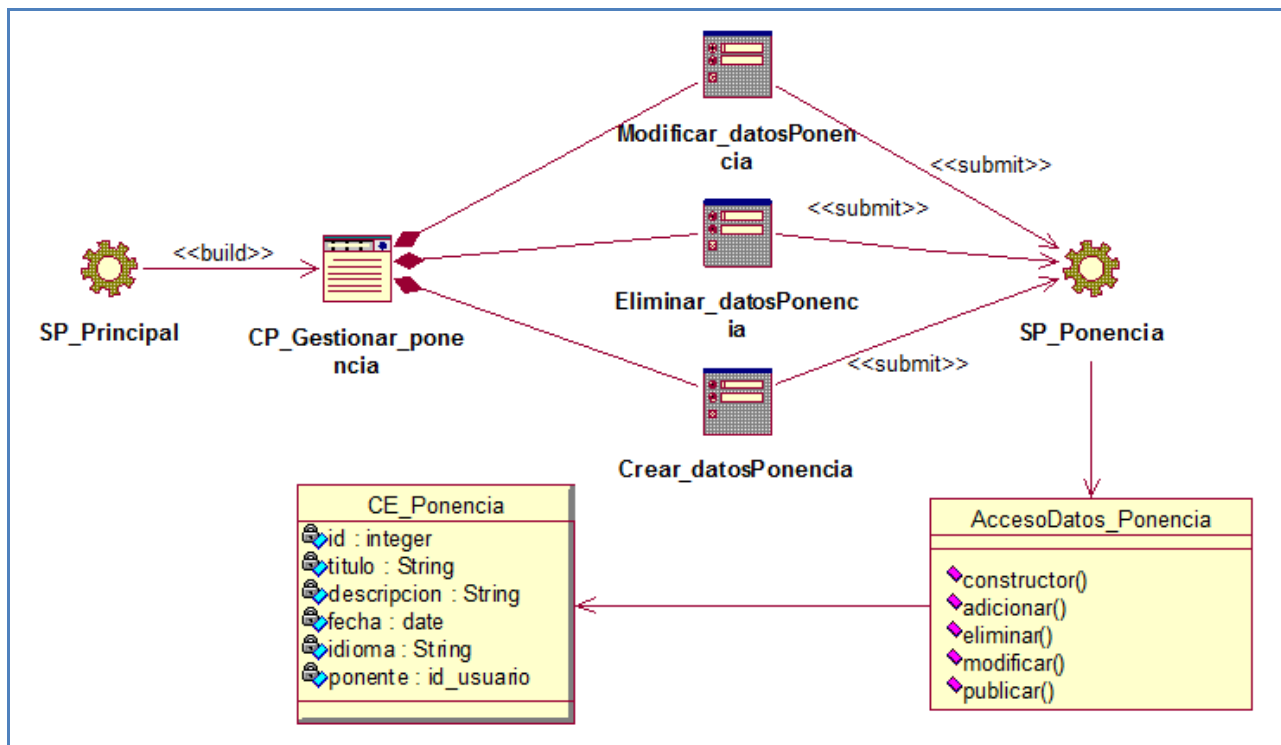


Diagrama 17. Diagrama de clases web “Gestionar evento”

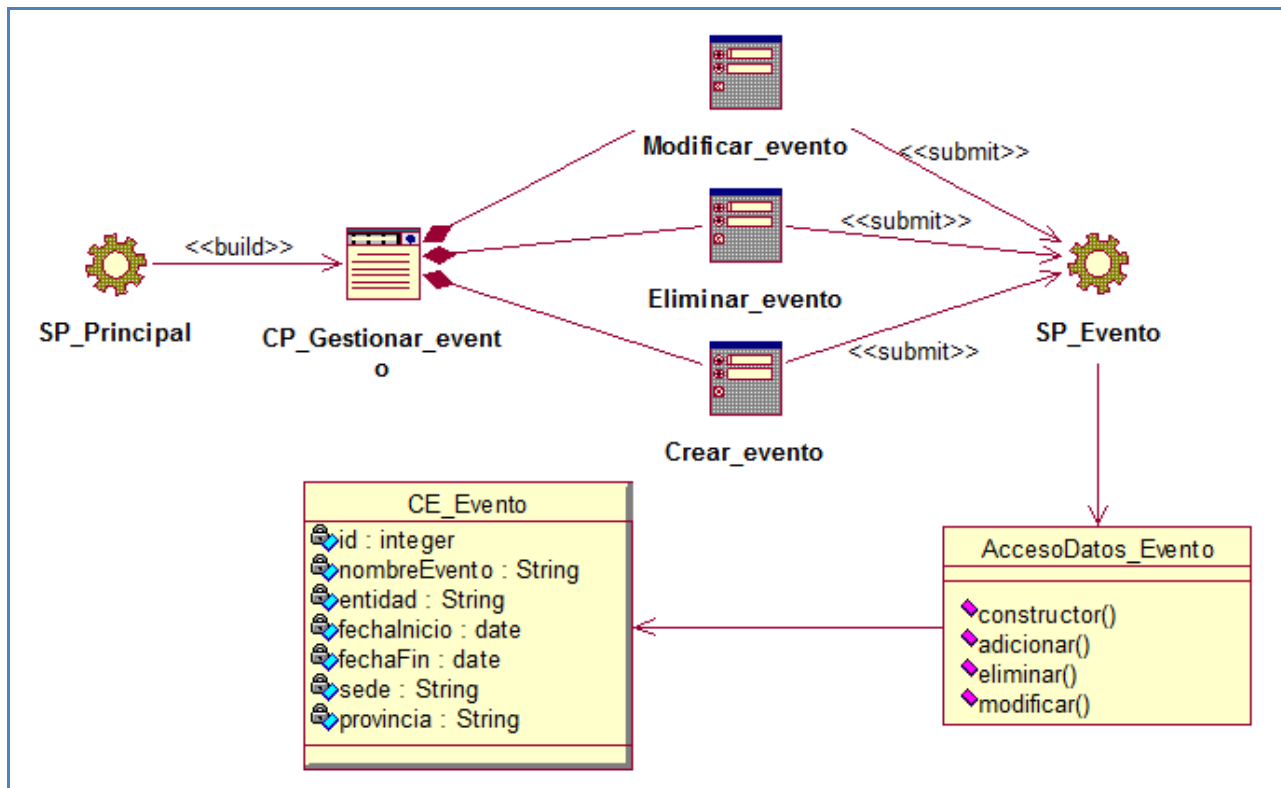


Diagrama 18. Diagrama de clases web “Buscar en repositorio de archivos académicos”

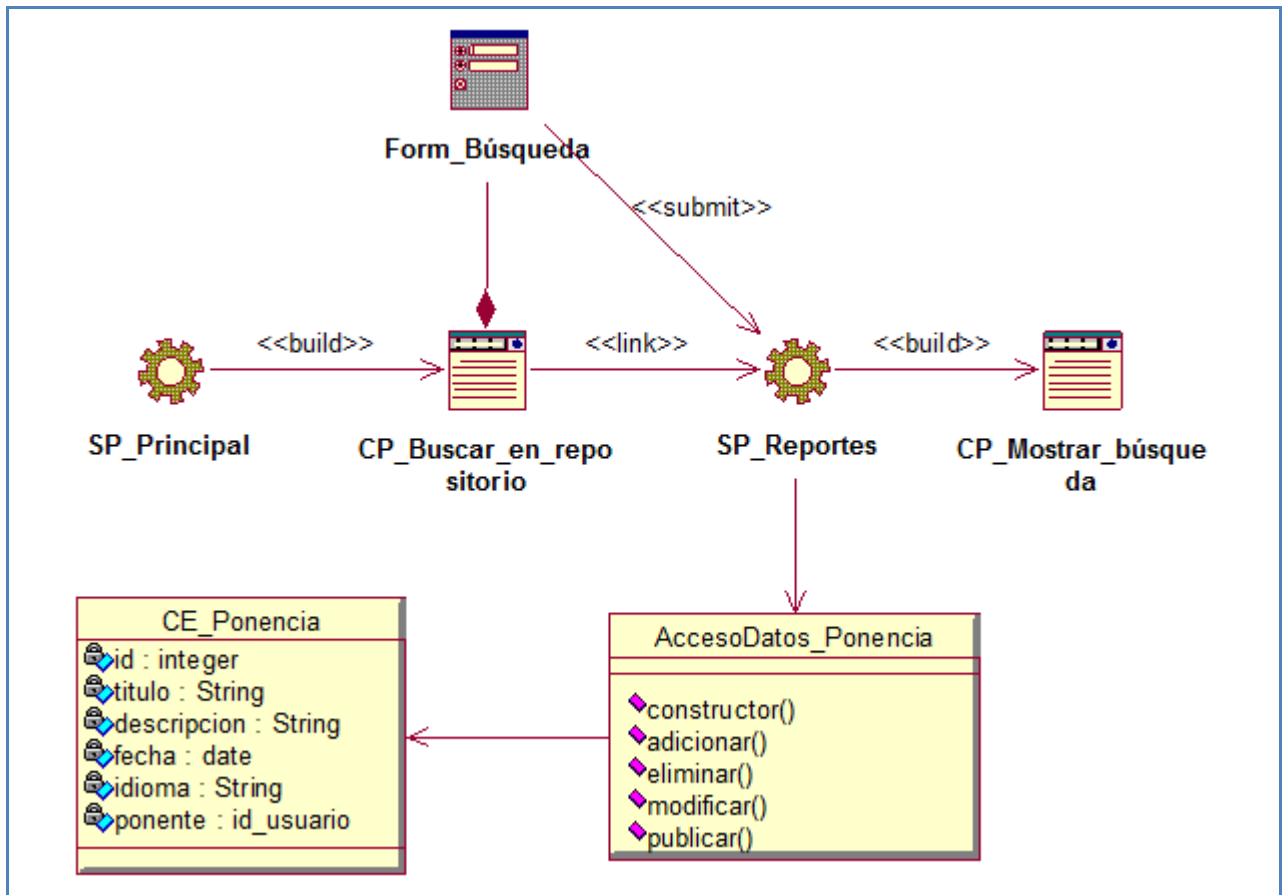


Diagrama 19. Diagrama de clases web “Enviar ponencia”

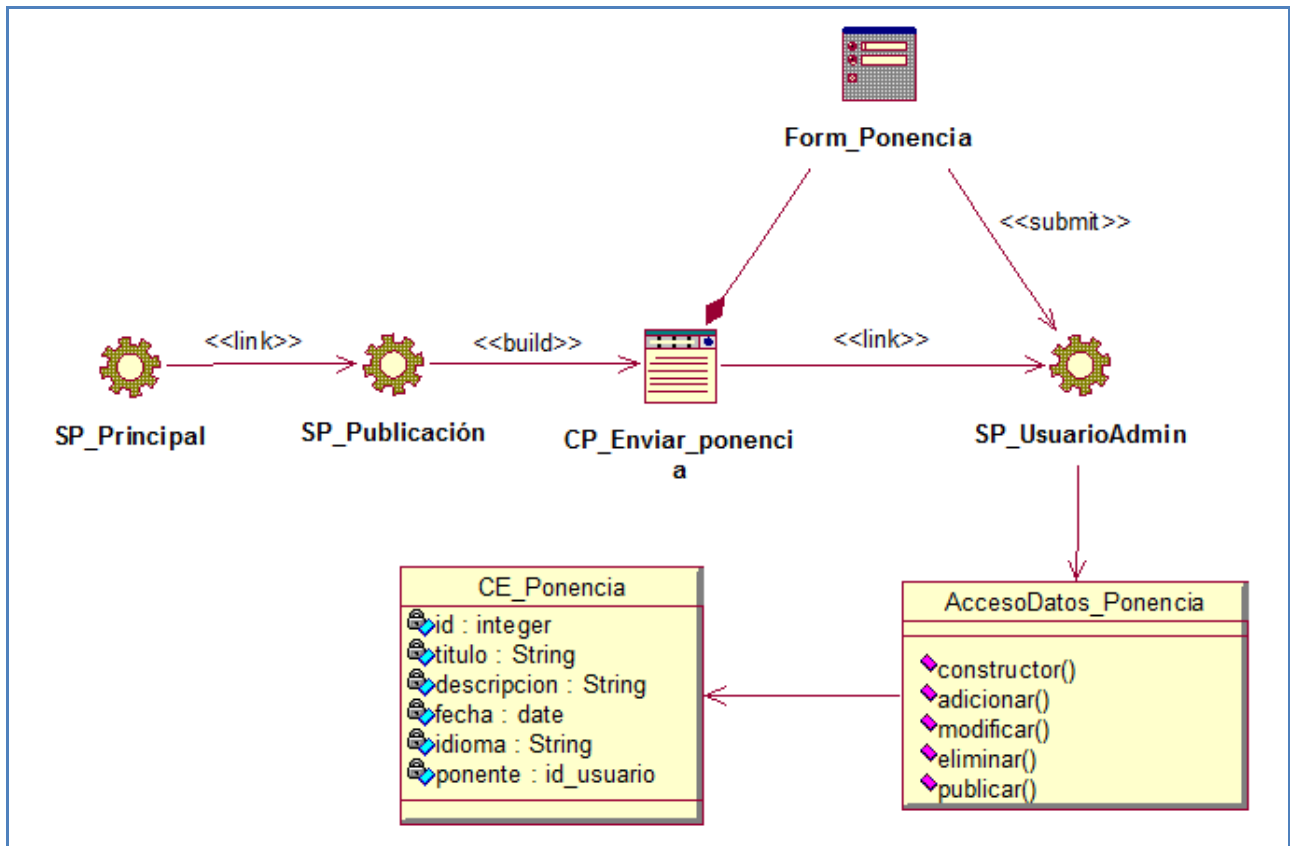


Diagrama 20. Diagrama de clases web “Autenticarse”

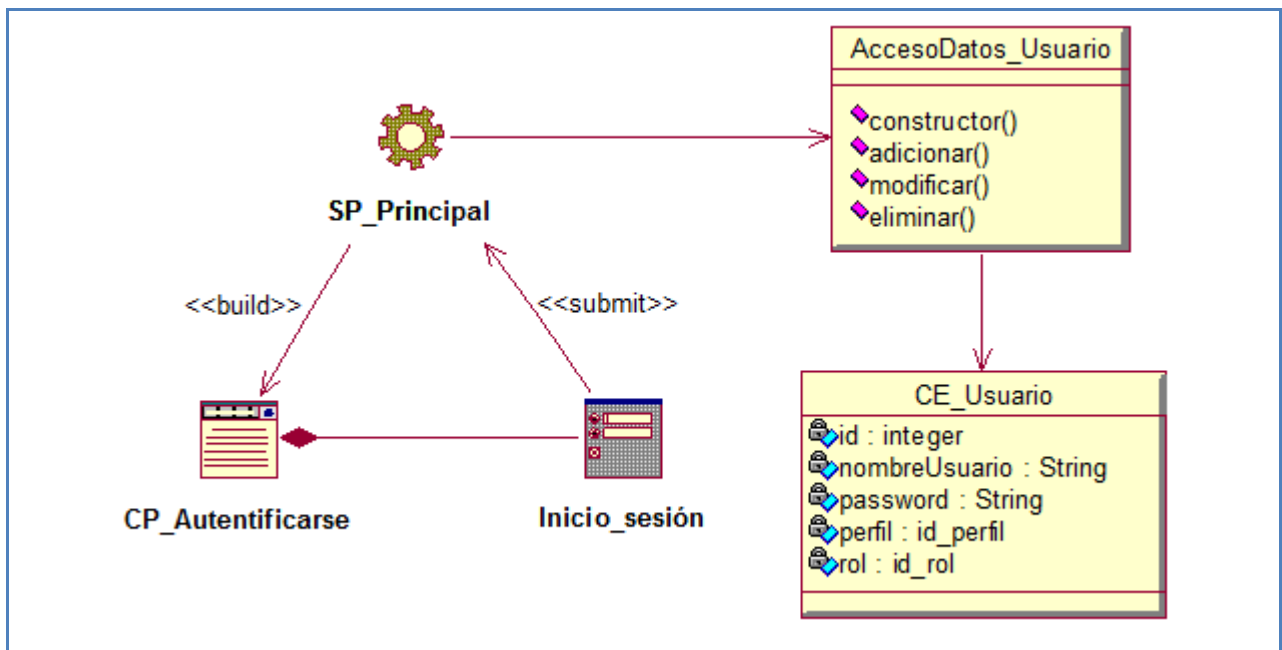


Diagrama 21. Diagrama de clases web “Cambiar contraseña”

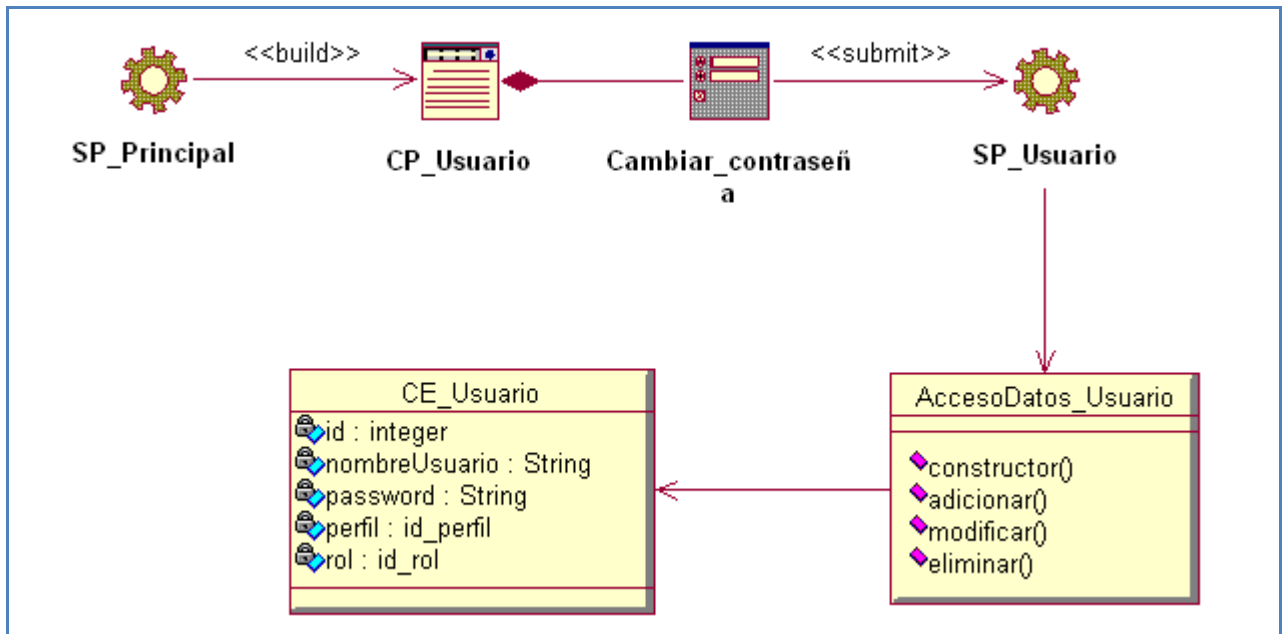


Diagrama 22. Diagrama de clases web “Mostrar ponencias”

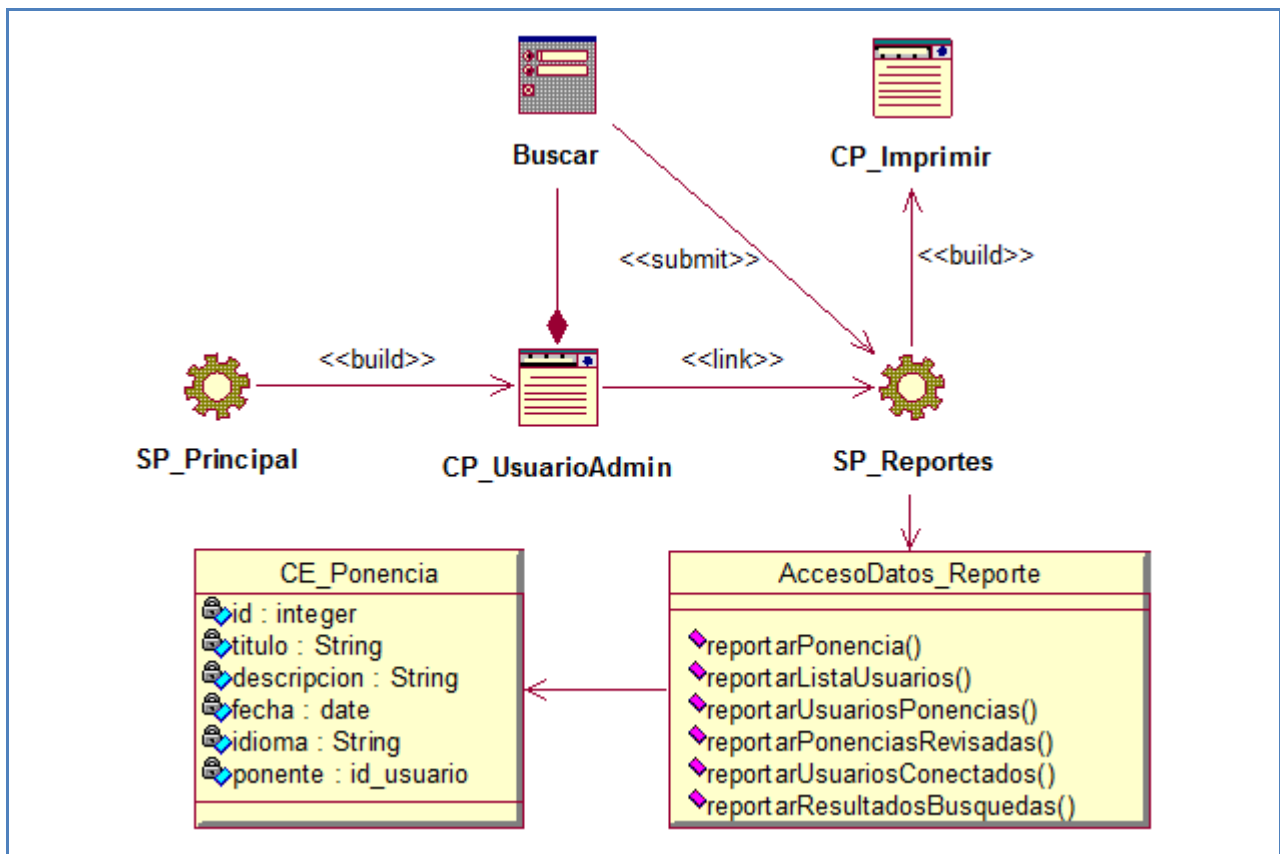


Diagrama 23. Diagrama de clases web “Mostrar lista de usuarios”

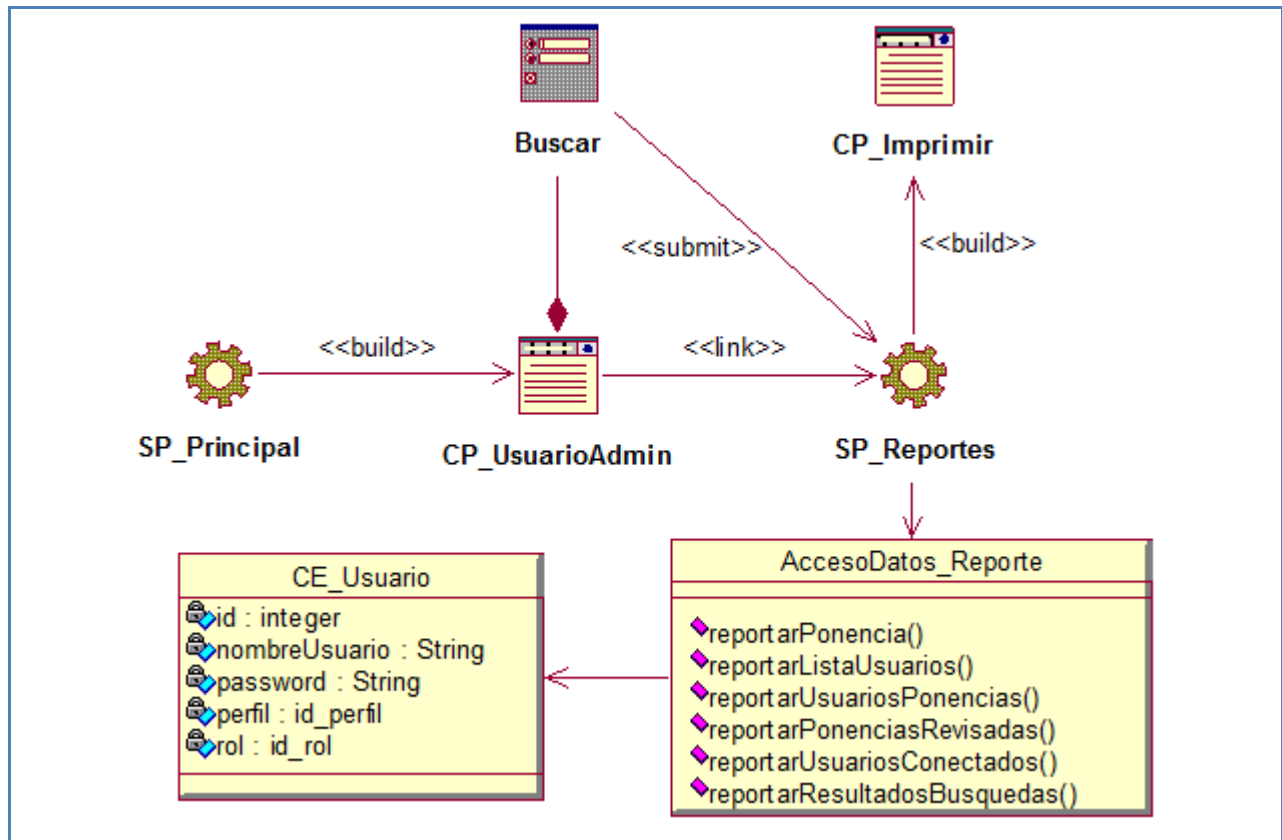


Diagrama 24. Diagrama de clases web “Mostrar ranking de categorías con más ponencias”

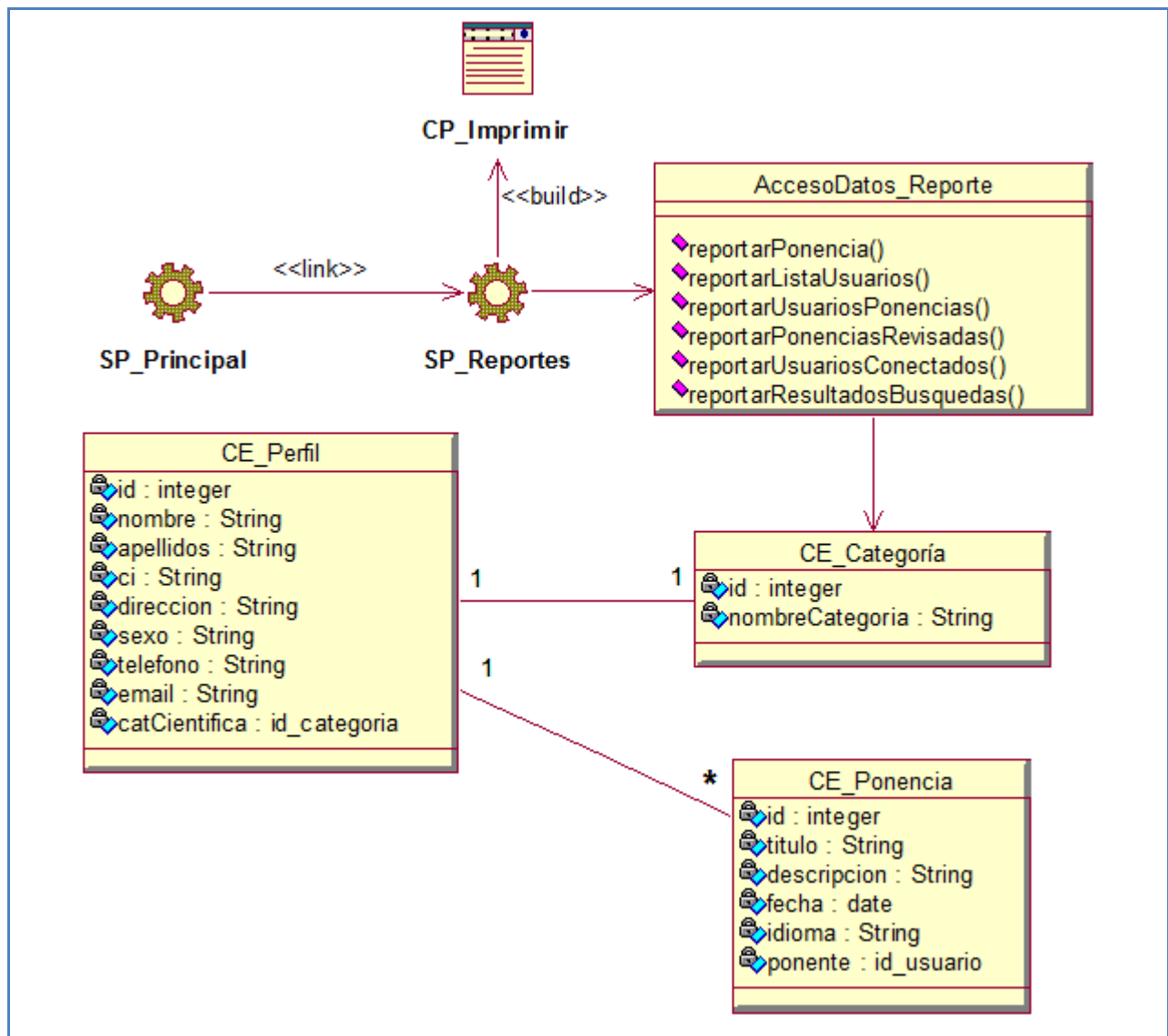


Diagrama 25. Diagrama de clases web “Mostrar ranking de ponencias más revisadas”

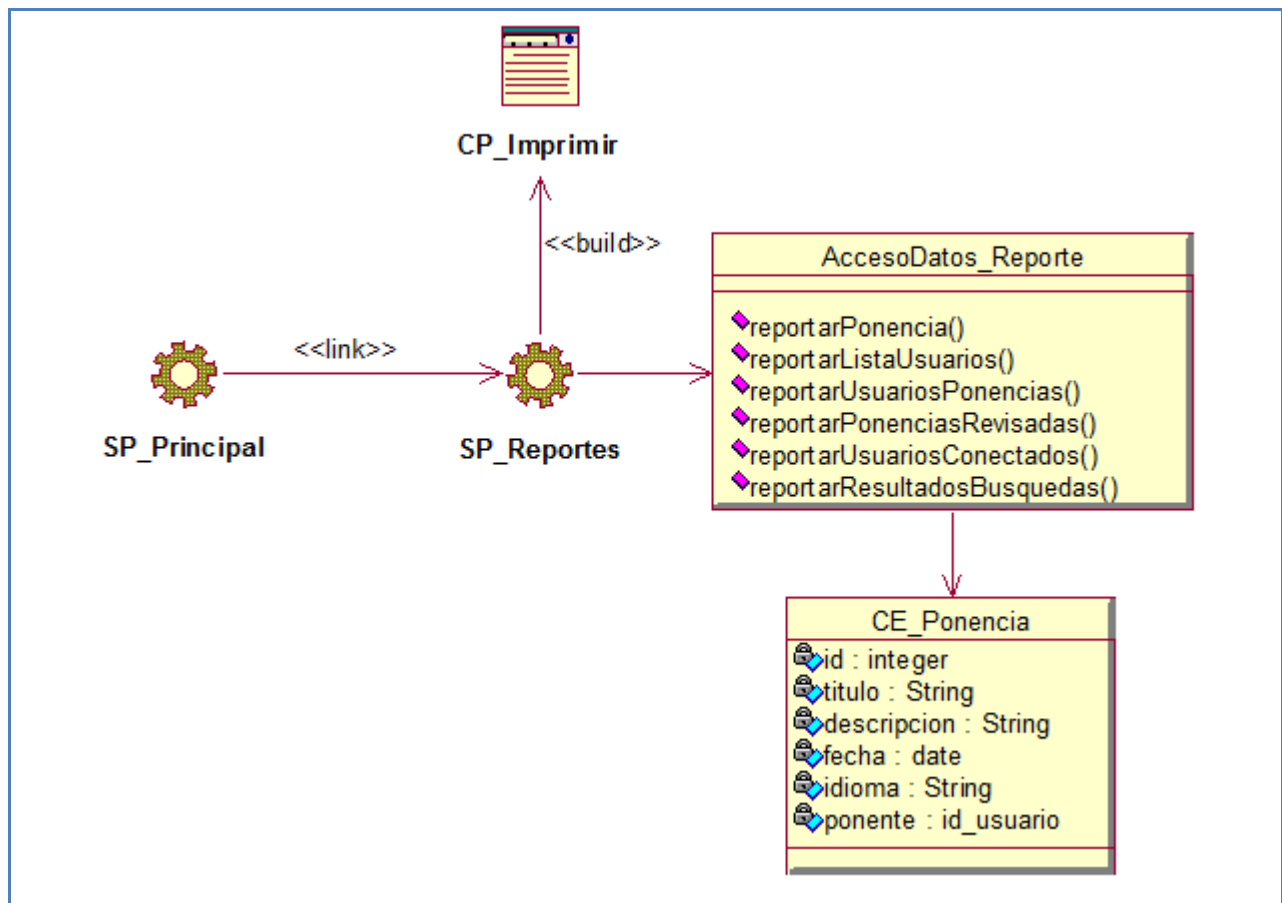




Diagrama 26. Diagrama de clases web “Mostrar usuarios conectados”

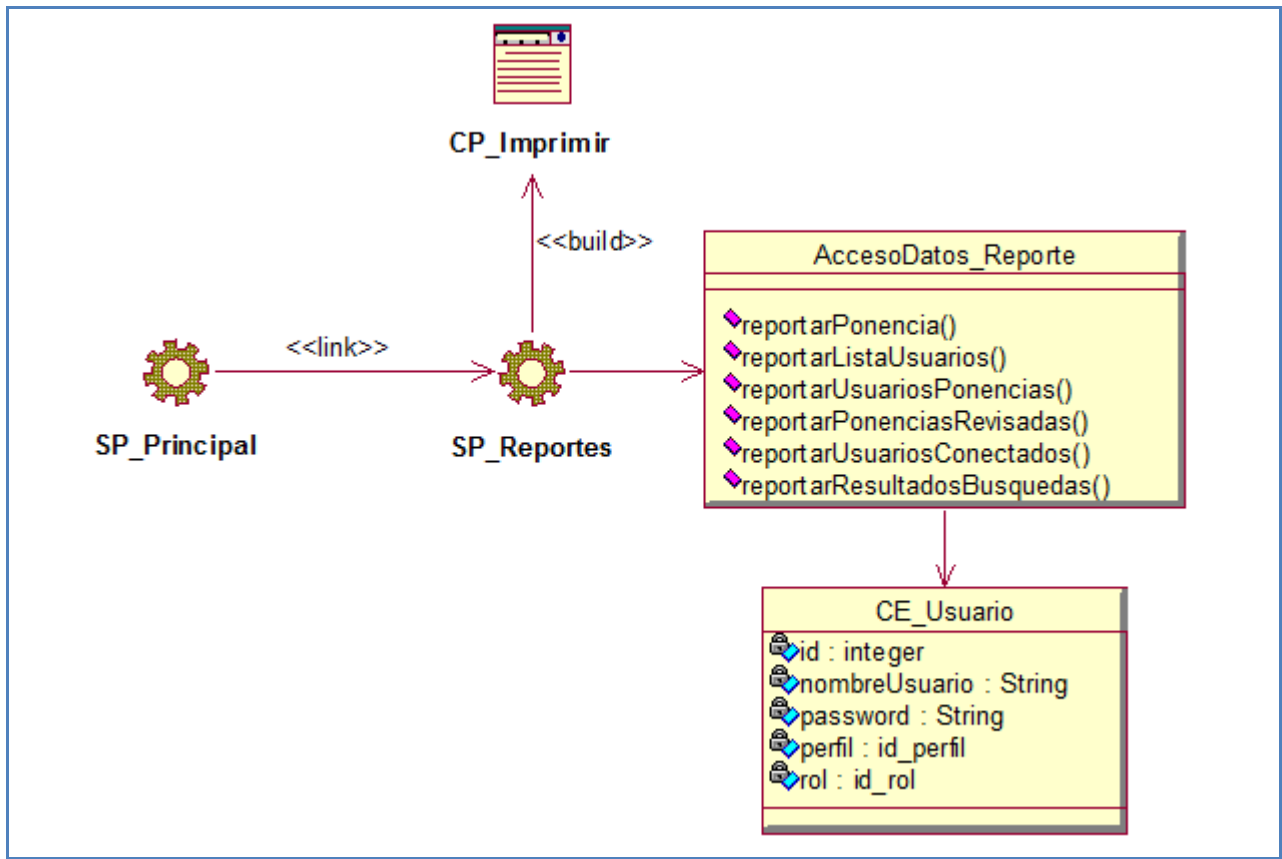


Diagrama 27. Diagrama de clases web “Mostrar resultados de las búsquedas”

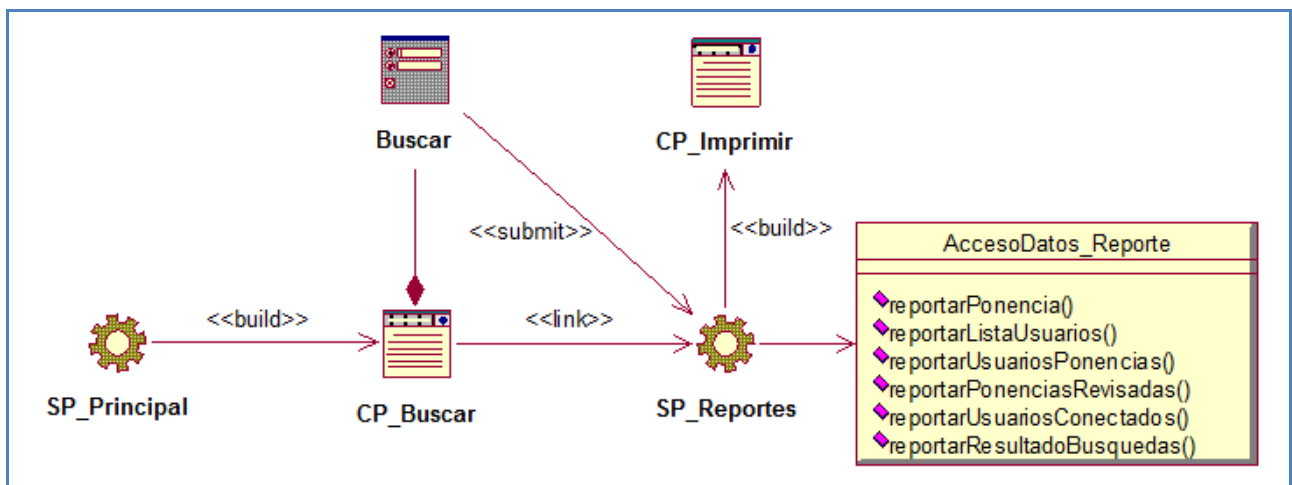
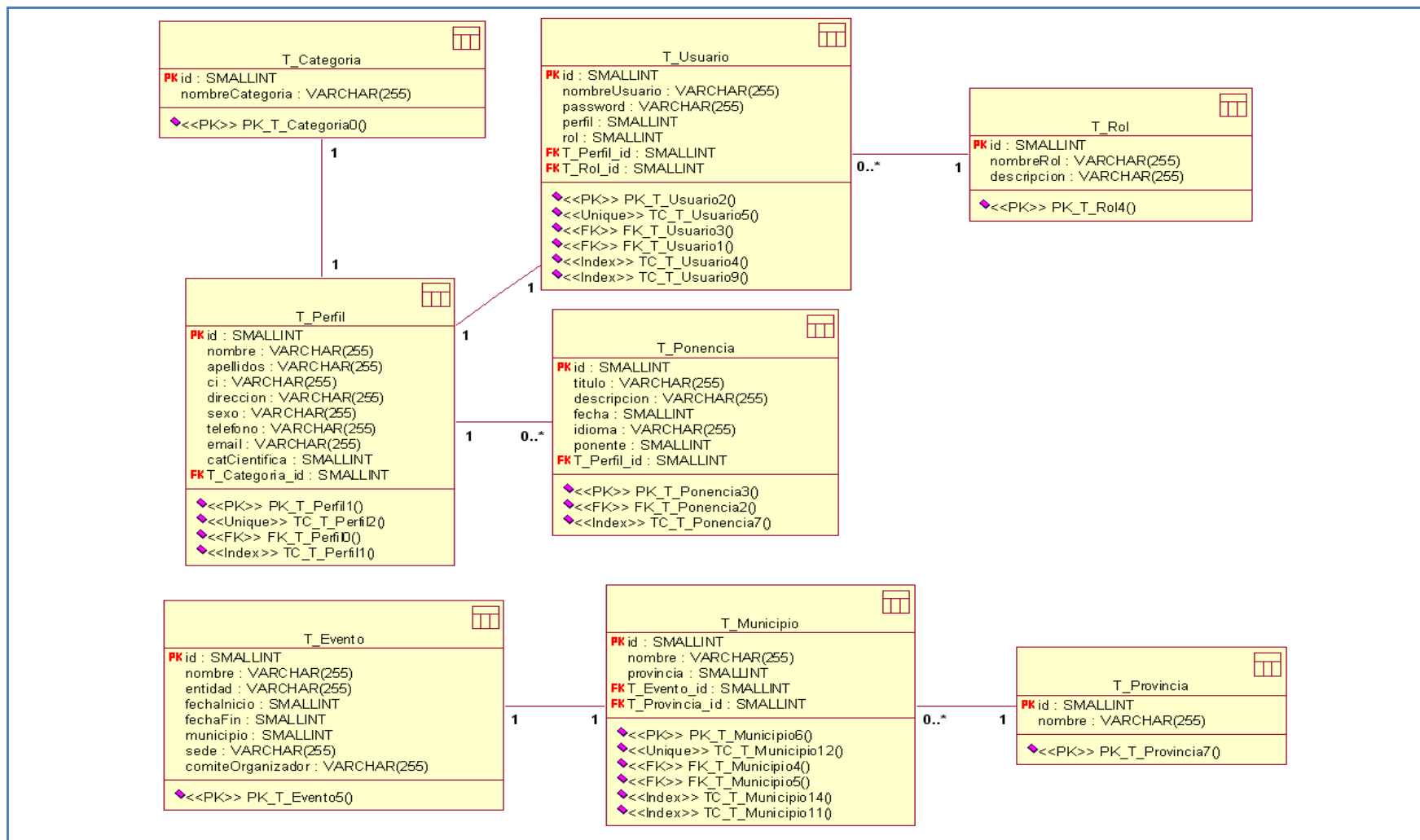
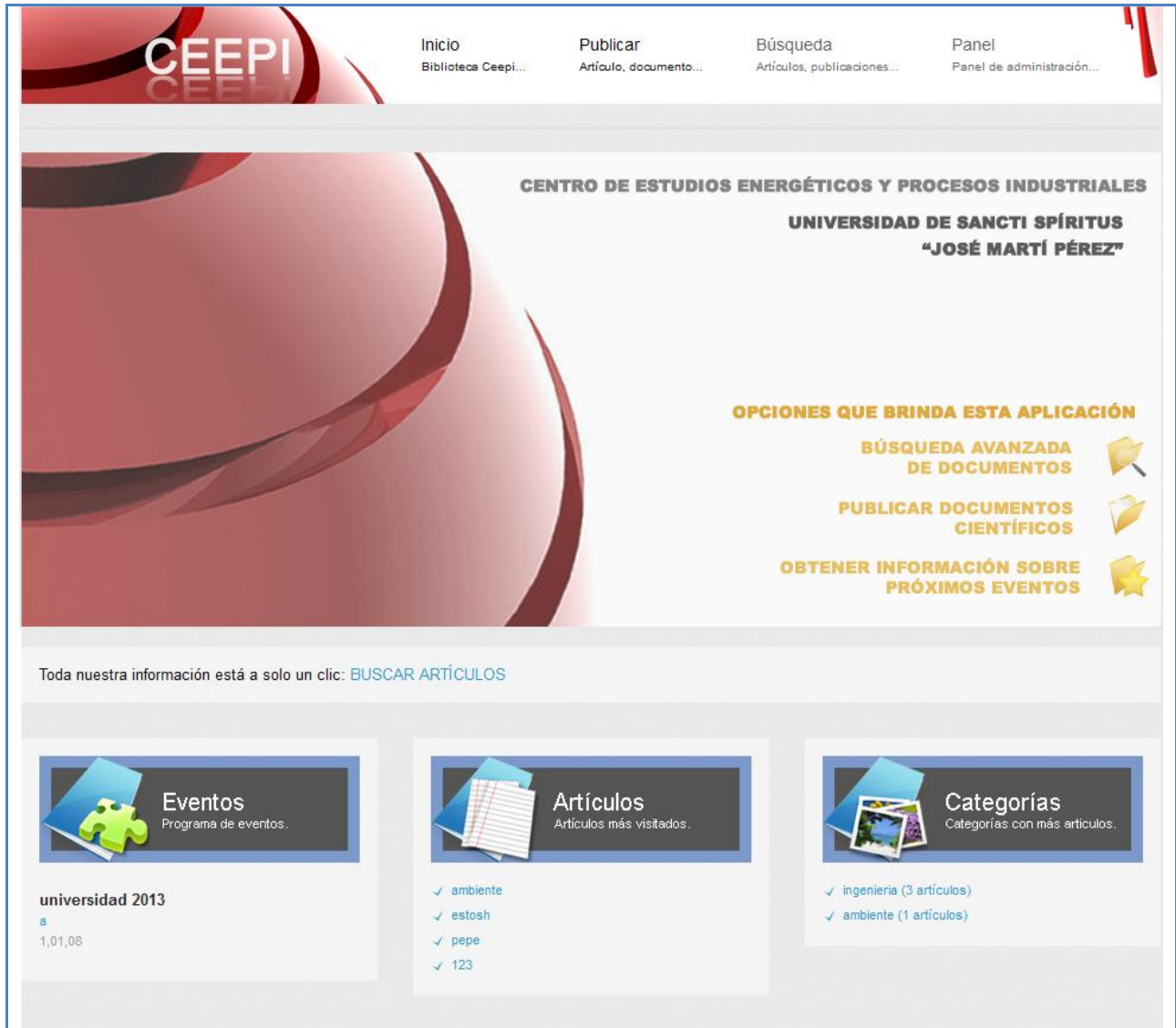


Diagrama 28. Diagrama del modelo de datos



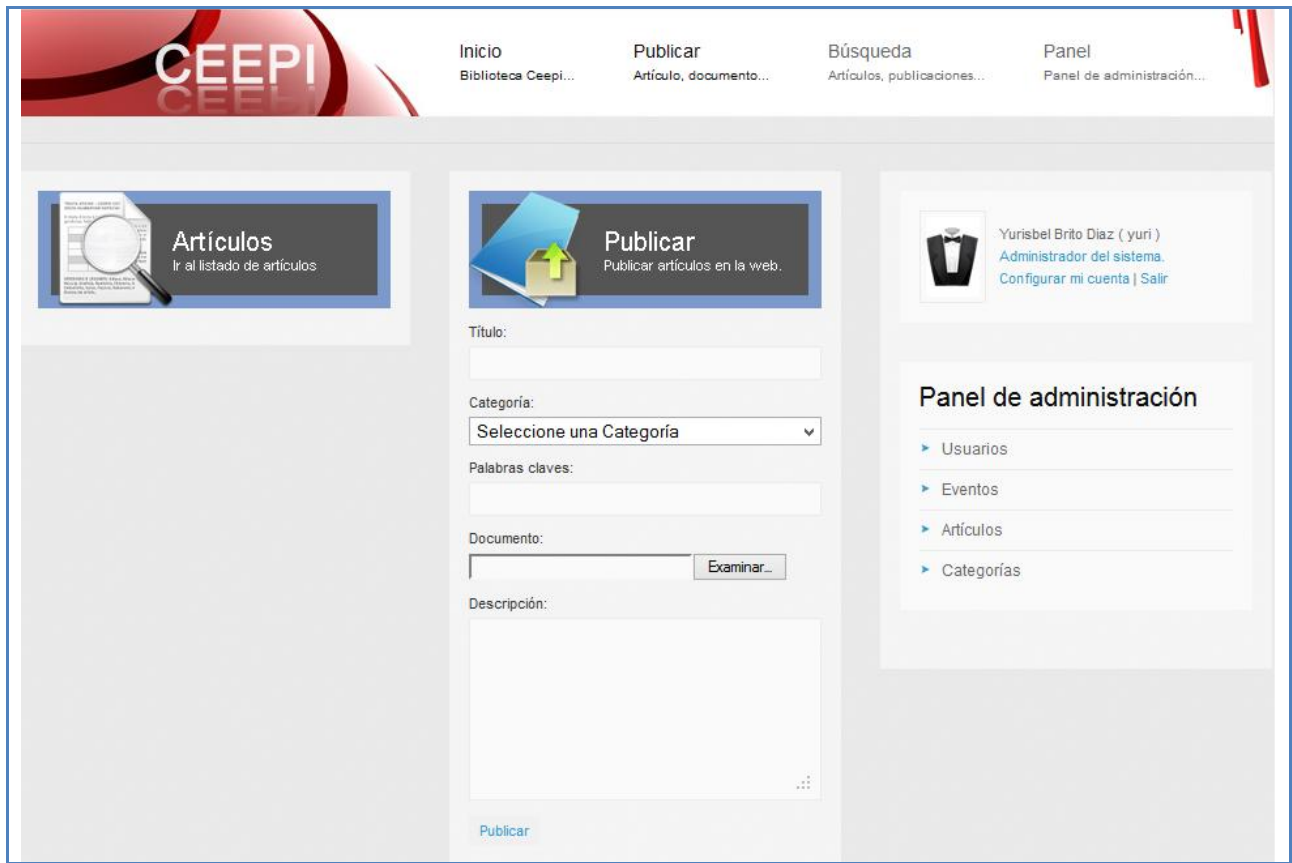
Prototipo de la ventana de la aplicación "Inicio"



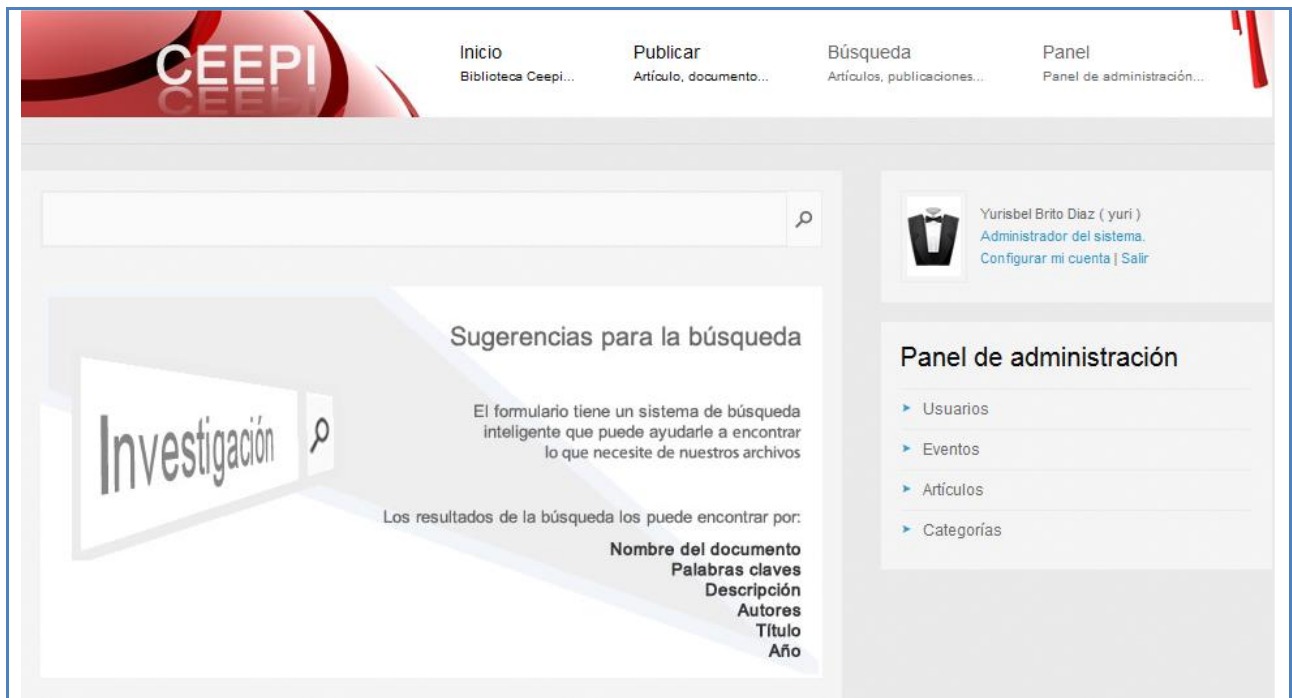
Prototipo de la ventana de la aplicación "Autenticarse"

El prototipo de la ventana de la aplicación "Autenticarse" se muestra dentro de un recuadro con un borde azul. En la parte superior izquierda, hay un icono que representa un documento azul con un símbolo de advertencia rojo (un octágono con un signo de exclamación). A la derecha de este icono, el texto "Entrar" está escrito en un tamaño grande y en negrita, con "Autenticarse en el sistema" debajo en un tamaño más pequeño. Debajo de esta sección de encabezado, hay un campo de entrada etiquetado "Usuario:" y un campo de entrada etiquetado "Contraseña:". Debajo de los campos de entrada, hay un botón con el texto "Entrar". En la parte inferior de la ventana, hay un recuadro blanco con un signo de interrogación gris en el centro, lo que sugiere un estado de carga o un mensaje de error.

Prototipo de la ventana de la aplicación “Publicar”



Prototipo de la ventana de la aplicación “Buscar”



Prototipo de la ventana de la aplicación “Panel de administración”

**CEEPI**

Inicio Biblioteca Ceepi... Publicar Artículo, documento... Búsqueda Artículos, publicaciones... Panel Panel de administración...

### Listado de Usuarios

Nombre	Usuario	Correo		
Yurisbel Brito Diaz	yuri	yuri@correo.cu	<a href="#">Editar</a>	<a href="#">Borrar</a>
Otro Perez	otro	otro@suss.co.cu	<a href="#">Editar</a>	<a href="#">Borrar</a>

[Nuevo usuario](#)

**Yurisbel Brito Diaz ( yuri )**  
 Administrador del sistema.  
[Configurar mi cuenta](#) | [Salir](#)

### Panel de administración

- ▶ Usuarios
- ▶ Eventos
- ▶ Artículos
- ▶ Categorías

Prototipo de la ventana de la aplicación “Editar”

**CEEPI**

Inicio Biblioteca Ceepi... Publicar Artículo, documento... Búsqueda Artículos, publicaciones... Panel Panel de administración...

### Editar usuario: Yurisbel Brito Diaz

Nombre y Apellidos:

Usuario:

Correo:

Contraseña:

Repita la contraseña

[Volver al listado](#)

**Yurisbel Brito Diaz ( yuri )**  
 Administrador del sistema.  
[Configurar mi cuenta](#) | [Salir](#)

### Panel de administración

- ▶ Usuarios
- ▶ Eventos
- ▶ Artículos
- ▶ Categorías