

Universidad de las Ciencias Informáticas



Facultad 8

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO
EN CIENCIAS INFORMÁTICAS**

Sistema para la Gestión de Postgrados en la UCI.

Autores:

Grisel Reyes Lominchar
Máximo Alberto Márquez

Tutor:

Ing. Joe Fernández Vanega

Ciudad de la Habana, junio de 2009
“Año del 50 aniversario del Triunfo de la Revolución”



"SI EL PRESENTE ES DE LUCHA, EL FUTURO ES NUESTRO."

che

DEDICATORIA

... A MIS PADRES POR TODO EL APOYO Y AMOR QUE SIEMPRE ME
HAN DADO.

... A MI ESPOSO Y A MI NIÑA POR ESPERARME TANTO TIEMPO.

GRISSEL

... A MIS PADRES POR HABERME AYUDADO EN TODOS ESTOS AÑOS
DE ESTUDIO, POR LA CONFIANZA Y EL AMOR QUE SIEMPRE ME HAN
DADO, SIN SU APOYO NO HUBIERA PODIDO LOGRAR LO QUE HE SIDO
HASTA HOY.

MÁXIMO

AGRADECIMIENTOS

... A MIS PADRES POR SER LOS MEJORES.
... A MI NIÑA POR QUERERME TANTO, POR SER LA RAZÓN DE MI
VIDA.
... A MI ESPOSO POR TENER PACIENCIA Y APOYARME EN TODO.
... A MI FAMILIA Y EN ESPECIAL A MI ABUELA HILDA Y MI ABUELO
RIGOBERTO DONDE QUIERA QUE ESTÉ.
... A EMILIA POR AYUDARME A CUIDAR A MI HIJA.
... A MIS COMPAÑEROS DE GRUPO Y AMIGOS, EN ESPECIAL A YENI,
SÁNCHEZ, CORRE Y HÉCTOR POR AYUDARME SIEMPRE.
... A MI TUTOR Y MI COMPAÑERO DE TESIS.
GRISEL

...A MIS PADRES POR SER LO MÁS VALIOSO DE MI VIDA.
...A YIYO POR QUERERME Y APOYARME CADA VEZ QUE LO
NECESITO.
...A TODA MI FAMILIA Y MUÝ ESPECIAL A MI ABUELA.
...A MI NOVIA QUE TANTO ME AYUDÓ A LO LARGO DE TODA LA
CARRERA.
...A MI TUTOR POR TODA LA AYUDA Y APOYO QUE NOS BRINDÓ EN
ESTOS ÚLTIMOS MESES.
...A DAINYS POR TODA LA AYUDA QUE ME BRINDÓ EN ESTE
TRABAJO Y A LO LARGO DE TODA MI CARRERA
...A ALEXEY POR HABER SIDO UN AMIGO INCONDICIONAL, ESTE
TRABAJO TAMBIÉN FUE POR TI HERMANO.
...A TODOS MIS COMPAÑEROS DE GRUPO Y AMIGOS, EN ESPECIAL A
HÉCTOR Y ORESTES POR TODA SU AYUDA Y PACIENCIA
INCONDICIONAL.
...A TODAS LAS PERSONAS QUE ME QUIEREN Y QUE SIN ESPERAR
NADA DE MÍ DE UNA FORMA U OTRA SIEMPRE ME HAN AYUDADO.
MÁXIMO

Resumen

La Dirección de Formación Postgraduada requiere un control de todas las actividades que se desarrollan en la misma y los reportes que se necesitan. Por lo expuesto anteriormente es necesario desarrollar un sistema que permita automatizar el control y la gestión de la información generada por las actividades de postgrado. El presente trabajo se centró en la actualización del análisis y diseño realizado en la tesis Análisis y Diseño del Sistema para la Gestión de Postgrados en la UCI por las ingenieras Giselle Barreto y Wendy Acedo y la implementación del sistema con el objetivo de mejorar la gestión de la información y minimizar el tiempo de búsqueda de información accediendo a ella en cualquier momento y desde cualquier lugar dentro de la UCI. Para la realización de este proyecto se siguieron los flujos de trabajo que propone el Proceso Unificado de Desarrollo, como lenguaje de programación por parte del servidor se utilizó PHP y por parte del cliente, Java Script y HTML y se hizo uso del Lenguaje Unificado de Modelado (UML).

Palabras claves: Sistema de Gestión de la Información, Dirección de Formación Postgraduada.

Índice

INTRODUCCIÓN.....	1
CAPÍTULO 1.....	3
FUNDAMENTACIÓN TEÓRICA.....	3
<i>Introducción</i>	3
1.1 <i>Generalidades de postgrado.</i>	3
1.2 <i>Sistemas de Gestión.</i>	4
1.3 <i>Ejemplos de Sistemas de Gestión de postgrado existentes a nivel nacional e internacional.</i>	5
1.4 <i>Tendencias y tecnologías actuales.</i>	6
1.4.1 <i>Lenguaje de Programación y Tecnologías del lado del Cliente.</i>	6
1.4.1.1 <i>Java Script.</i>	6
1.4.1.2 <i>Lenguaje de Marcación de Texto (HTML). (9)</i>	7
1.4.2 <i>Lenguaje de Programación y Tecnologías del lado del Servidor.</i>	8
1.4.2.1 <i>PHP (HyperText Preprocesor). (10)</i>	8
1.4.3 <i>Sistemas Gestores de Base de Datos (SGBD). (11)</i>	8
1.4.3.1 <i>MySQL. (12)</i>	9
1.4.4 <i>Metodologías de desarrollo de software.</i>	10
1.4.4.1 <i>Proceso Unificado de Desarrollo (RUP). (14)</i>	10
1.4.5 <i>Lenguaje de Modelado. (15)</i>	11
1.4.6 <i>Servidores Web</i>	12
1.4.7 <i>Arquitectura</i>	12
1.4.7.1 <i>Arquitectura en capas.</i>	12
1.4.7.2 <i>Modelo Vista Controlador (MVC).</i>	13
1.4.7.3 <i>Arquitectura Orienta a Servicios (SOA). (22)</i>	14
1.4.7.4 <i>¿Por qué utilizar Arquitectura en capas?</i>	15
1.5 <i>Herramientas Utilizadas.</i>	15
1.5.1 <i>Embarcadero ERStudio 7.1. (23)</i>	15
1.5.2 <i>Dreamweaver.</i>	16
1.5.3 <i>Visual Paradigm.</i>	17
1.5.4 <i>Gestor de Base de Datos MySQL. (28)</i>	17
1.5.5 <i>Zend Studio. (29)</i>	18
1.5.6 <i>WAMP. (30)</i>	19
<i>Conclusiones</i>	19
CAPÍTULO 2.....	20
CARACTERÍSTICAS DEL SISTEMA.....	20
<i>Introducción</i>	20
2.1 <i>Modelo de Negocio.</i>	20
2.2 <i>Reglas del Negocio.</i>	21
2.3 <i>Actores del Negocio</i>	22
2.4 <i>Trabajadores del Negocio</i>	23
2.5 <i>Entidades del Negocio.</i>	24
2.6 <i>Casos de Usos del Negocio.</i>	24
2.6.1 <i>Casos de Usos del Negocio a tratar</i>	24
2.6.2 <i>Diagrama de casos de uso del negocio.</i>	24
2.7 <i>Diagramas de Actividades.</i>	25
2.7.1 <i>Diagrama de Inscribir_Actividad.</i>	25
2.7.2 <i>Diagrama de Reportar</i>	25
2.7.3 <i>Diagrama de Matricular.</i>	26

2.7.4 Diagrama de Recoger_Título.....	26
2.8 Requerimientos de Software.....	27
2.8.1 Requerimientos Funcionales.....	27
2.8.2 Requerimientos No Funcionales.....	28
2.9 Actores del Sistema.....	29
2.10 Diagrama de casos de uso del sistema.....	30
2.11 Definición de los casos de uso.....	30
2.12 Descripción de los casos de usos del sistema.....	33
2.12.1 Descripción del Caso de Uso Gestionar Persona.....	33
2.12.2 Descripción del Caso de Uso Autenticar Usuario.....	35
2.12.3 Descripción del Caso de Uso Evaluar Estudiante.....	35
2.12.4 Descripción del Caso de Uso Gestionar Actividad.....	36
2.12.5 Descripción del Caso de Uso Solicitar Matrícula.....	38
2.12.6 Descripción del Caso de Uso Aprobar solicitar de matrícula.....	38
2.12.7 Descripción del Caso de Uso Gestionar Noticia.....	39
2.12.8 Descripción del Caso de Uso Generar Reportes.....	40
2.12.9 Descripción del Caso de Uso Solicitar Información.....	44
2.12.10 Descripción del Caso de Uso Buscar Persona.....	45
2.12.11 Descripción del Caso de Uso Buscar Actividad.....	46
Conclusiones.....	46
CAPÍTULO 3.....	47
ANÁLISIS Y DISEÑO.....	47
Introducción.....	47
3.1 Análisis.....	47
3.1.1 Modelo de análisis.....	47
3.1.2 Diagramas de clases del análisis.....	47
3.1.2.1 Diagrama de clases de análisis de Autenticar.....	47
3.1.2.2 Diagrama de clases de análisis de Gestionar Persona.....	48
3.1.2.3 Diagrama de clases de análisis de Gestionar Actividad.....	48
3.1.2.4 Diagrama de clases de análisis de Buscar Actividad.....	48
3.1.2.5 Diagrama de clases de análisis de Aprobar Matrícula.....	49
3.1.2.6 Diagrama de clases de análisis de Evaluar.....	49
3.1.2.7 Diagrama de clases de análisis de Gestionar Noticia.....	50
3.1.2.8 Diagrama de clases de análisis de Buscar Persona.....	50
3.1.2.9 Diagrama de clases de análisis de Generar Reportes.....	51
3.1.2.10 Diagrama de clases de análisis de Solicitar Información.....	51
3.1.2.11 Diagrama de clases de análisis de Solicitar Matrícula.....	52
3.2 Diseño.....	52
3.2.1 Modelo de diseño.....	52
3.2.2 Diagrama de clases del diseño.....	52
3.2.2.1 Diagrama de clases de diseño de Gestionar Persona.....	53
3.2.2.2 Diagrama de clases de diseño de Gestionar Actividad.....	53
3.2.2.3 Diagrama de clases de diseño de Autenticar.....	54
3.2.2.4 Diagrama de clases de diseño de Buscar Actividad.....	54
3.2.2.5 Diagrama de clases de diseño de Buscar Persona.....	55
3.2.2.6 Diagrama de clases de diseño de Evaluar.....	55
3.2.2.7 Diagrama de clases de diseño de Generar Reportes.....	56
3.2.2.8 Diagrama de clases de diseño de Solicitar Información.....	57
3.2.2.9 Diagrama de clases de diseño de Gestionar Noticias.....	57
3.2.2.10 Diagrama de clases de diseño de Solicitar Matrícula.....	58
3.2.2.11 Diagrama de clases de diseño de Aprobar Matrícula.....	58
3.2.3 Descripción de las clases del diseño.....	59

3.2.3.1 Descripción de la clase Factory.....	59
3.2.3.2 Descripción de las clases Controladoras.....	59
3.2.3.3 Descripción de las clases de Acceso a Datos.....	62
3.2.3.4 Descripción de las clases Entidad.....	64
3.3 <i>Diagramas de interacción</i>	70
3.3.1 Diagrama de Secuencia de Aprobar Matrícula.....	70
3.3.2 Diagrama de Secuencia de Gestionar Persona.....	71
3.3.3 Diagrama de Secuencia de Gestionar Actividad.....	72
3.3.4 Diagrama de Secuencia de Evaluar Estudiante.....	73
3.3.5 Diagrama de Secuencia de Solicitar Matrícula.....	73
3.4 Modelo de Clases Persistentes.....	74
<i>Conclusiones</i>	74
CAPÍTULO 4.....	75
IMPLEMENTACIÓN.....	75
<i>Introducción</i>	75
4.1 Modelo de Implementación.....	75
4.2 Diagrama de Despliegue.....	75
4.3 Diagrama de Componentes.....	76
4.3.1 Diagrama de Componentes.....	76
4.3.2 Diagrama de Componentes Estado de la Matrícula.....	77
4.3.3 Diagrama de Componentes Actividades por Áreas.....	77
4.3.4 Diagrama de Componentes Actividades por estado.....	77
4.3.5 Diagrama de Componentes Actividades por profesor.....	78
4.3.6 Diagrama de Componentes Estudiantes adiestrados.....	78
4.3.7 Diagrama de Componentes Aprobar Solicitud Matrícula.....	78
4.3.8 Diagrama de Componentes Actividades vencidas.....	79
4.3.9 Diagrama de Componentes Retención de Matrícula.....	79
4.3.10 Diagrama de Componentes Matrícula por Actividad.....	79
4.3.11 Diagrama de Componentes Recogida de certificado.....	80
4.3.12 Diagrama de Componentes Evaluar estudiantes.....	80
4.3.13 Diagrama de Componentes Información Actividades.....	80
4.3.14 Diagrama de Componentes Gestionar Actividad.....	81
4.3.15 Diagrama de Componentes Gestionar Persona.....	81
4.3.16 Diagrama de Componentes Solicitar Matrícula.....	81
4.3.17 Diagrama de Componentes Gestionar Noticia.....	82
<i>Conclusiones</i>	82
CONCLUSIONES.....	83
RECOMENDACIONES.....	84
REFERENCIAS BIBLIOGRÁFICAS.....	85
BIBLIOGRAFÍA.....	87
GLOSARIO DE TÉRMINOS.....	88

Introducción

En la era de la informatización, los avances tecnológicos han marcado el desarrollo de la sociedad. Con el propósito de ayudar al hombre en la realización de sus actividades en los procesos productivos y en el desarrollo social, la informática como ciencia ha posibilitado la realización de aplicaciones capaces de mejorar las condiciones de trabajo y de vida al permitirle tener acceso a la información de manera fácil y rápida.

Como parte de esta revolución tecnológica en el que se encuentra inmerso el mundo actualmente se han tomado varias acciones destinadas a la informatización de la sociedad. Cuba, no ajeno al mismo, ha implementado medidas para la inserción del software cubano en el mercado mundial, entre las que se encuentra la creación de la Universidad de las Ciencias Informáticas (UCI).

La UCI tiene como premisa convertirse en un centro de enseñanza superior con fortaleza científica en las ciencias informáticas, para lo cual ha implementado el vínculo del estudio con la producción, lo que ha contribuido al desarrollo de las capacidades cognoscitivas y al aumento de la competencia de sus futuros profesionales. Actualmente la universidad tiene entre sus principales tareas informatizar todas las áreas de la misma, pues en la mayoría de ellas las actividades que se realizan no se pueden controlar con facilidad. Ejemplo de ello lo tenemos en la Dirección de Formación Postgraduada (DFP).

En la educación superior, el postgrado representa el nivel más elevado y constituye la estrategia principal para la formación de profesionales del más alto nivel. Este tipo de enseñanza se desarrolla en dos vertientes: la superación profesional que propicia cursos, entrenamientos, talleres y diplomados; y la superación académica con maestrías, especialidades y doctorados. Estas formas organizativas pueden desarrollarse en diferentes modalidades como: a tiempo completo o parcial y con diferentes grados de comparecencia: semipresencial o a distancia.

En la UCI, la gestión de las actividades de postgrado se realiza con la ayuda de una aplicación en Access en la que se puede, entre otras funcionalidades: inscribir actividades, matricular estudiantes, llevar un control de evaluaciones y emitir algunos reportes; sin embargo esta aplicación no cumple con todas las necesidades de la DFP. No se cuenta con un medio de información para que el usuario pueda estar actualizado con respecto al estado de su superación postgraduada y de sus resultados, pues esta forma de gestión provoca que los mismos empleen mucho tiempo en averiguar sus calificaciones, esto se hace de manera directa en la dirección, por lo que trae consigo un aumento

del margen de error. Además los compañeros que trabajan en dicha dirección no tienen manera de emitir los reportes necesarios en el menor tiempo posible, pues esta actividad hay que realizarla de manera manual y en el peor de los casos la información está dividida y se hace necesario centralizar todo lo referente a la actividad que se realiza en la dirección.

Por lo cual se identifica el siguiente **Problema Científico**: ¿Cómo mejorar la actividad de gestión de la formación postgraduada en la UCI? Identificándose como **Objeto de Estudio** los procesos de formación de postgrado en Cuba cuyo **Campo de Acción** se centra en los procesos de formación de postgrado en la UCI.

Como **Idea a Defender** se plantea que si se desarrolla un sistema informático para la gestión de las actividades de postgrado en la UCI, que cumpla con las exigencias de los procesos de la Dirección de Formación Postgraduada, entonces se mejorará la gestión de los procesos de formación de postgrado en la UCI.

El **Objetivo General** de este trabajo es Desarrollar un sistema para la Gestión del Postgrado en la UCI, que automatice los procesos de formación, para lo cual se trazaron los siguientes **Objetivos Específicos**:

1. Realizar una aplicación capaz de satisfacer todas las necesidades posibles, de la Dirección de Formación Postgraduada de la UCI.
2. Crear una Interfaz atractiva y con facilidad en el trabajo de los usuarios.

Tareas Investigativas

1. Analizar los procesos de la Dirección de Formación Postgraduada de la UCI.
2. Realizar entrevistas al personal de la Dirección de Formación Postgraduada de la UCI, identificando quiénes participan en los procesos llevados a cabo por la misma y las actividades que requieren de automatización.
3. Estudiar y actualizar el análisis y diseño realizado.
4. Estudio del estado del Arte de Sistemas de Gestión Postgraduada.
5. Estudiar las tendencias actuales para el desarrollo de Sistemas de Gestión de la Información.
6. Realizar la Implementación de mecanismos que permitan gestionar la información.

Capítulo 1

Fundamentación Teórica

Introducción

En el presente capítulo se realiza un estudio del arte de acuerdo al objeto de estudio y campo de acción, haciéndose referencia a las tendencias y tecnologías que se emplean actualmente y han sido propuestas por las ingenieras Giselle Barreto y Wendy Acedo en su trabajo de diploma Análisis y Diseño del Sistema para la Gestión de Postgrados en la UCI, que puedan formar parte de la propuesta de solución del problema científico.

Además se analizan los principales servidores Web y las herramientas que se necesitan para desarrollar el sistema informático requerido, de esta manera se seleccionan las técnicas, metodología y herramientas que serán empleados para cumplir los objetivos propuestos.

1.1 Generalidades de postgrado.

En Cuba la enseñanza de postgrado ha atravesado por diferentes etapas. En 1976 se crea el Ministerio de Educación Superior (MES) convirtiéndose así la educación de postgrado en el nivel más elevado del Sistema Nacional de Educación para la superación continua de los egresados universitarios durante su carrera profesional.

La educación de postgrado integra un conjunto de actividades que se organizan en forma de dos subsistemas diferentes e interrelacionados:

➤ Superación Profesional (1)

- Curso: posibilita la formación básica y especializada de los graduados universitarios; comprende la organización de un conjunto de contenidos que abordan resultados de investigación relevantes o asuntos trascendentes con el propósito de complementar o actualizar los conocimientos de los profesionales que los reciben. Tiene una extensión mínima de dos créditos.
- Entrenamiento: posibilita la formación básica y especializada de los graduados universitarios, particularmente en la adquisición de habilidades y destrezas y en la asimilación e introducción de nuevos procedimientos y tecnologías con el propósito de complementar, actualizar, perfeccionar y consolidar conocimientos y habilidades prácticas. Tiene una extensión mínima de dos créditos.

- **Diplomado:** tiene como objetivo la especialización de un área particular del desempeño, y propicia la adquisición de conocimientos y habilidades académicas, científicas y/o profesionales en cualquier etapa del desarrollo de un graduado universitario, de acuerdo con las necesidades de su formación profesional o cultural. El diplomado está compuesto por un sistema de cursos y/o entrenamientos y otras formas articulados entre sí, que culmina con la realización y defensa de un trabajo ante tribunal. La extensión mínima de cada diplomado es de 15 créditos.

➤ **Superación Académica**

- **Maestría:** proporciona a los graduados universitarios una amplia cultura científica y conocimientos avanzados en las áreas correspondientes del saber, una mayor capacidad para la actividad docente, científica, la innovación o la creación artística, en correspondencia con las necesidades del desarrollo económico, social y cultural del país. Tiene extensión mínima de 70 créditos.
- **Especialidad:** proporciona la actualización, profundización, perfeccionamiento o ampliación de las competencias laborales para el desempeño profesional que requiere un puesto de trabajo -o familia de puestos de trabajo-, en correspondencia con las necesidades del desarrollo económico, social y cultural del país. Tiene extensión mínima de 100 créditos.
- **Doctorado:** se otorga a los doctores en ciencias de determinada especialidad que hayan realizado un trabajo de alto nivel de especialización en el campo del conocimiento al que se dediquen, con la defensa ante un tribunal competente, de una tesis que contenga la solución y generalización de un problema de carácter científico que constituya un aporte a la rama del conocimiento de que se trate y científico – técnico del país.

1.2 Sistemas de Gestión.

Un Sistema de Gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización; propicia el manejo de grandes volúmenes de datos, almacenarlos y transmitirlos en muy poco tiempo.

¿Por qué son necesarios?

Las instituciones del siglo XXI se enfrentan a muchos retos significativos, entre ellos: rentabilidad, competitividad, velocidad de los cambios, crecimiento y tecnología.

La implementación de un sistema de gestión eficaz puede ayudar a: reducir costos, lograr mejoras continuas y aumentar la satisfacción de clientes y partes interesadas.

Sistema de Gestión Académica.

El propósito de los Sistemas de Gestión Académica es brindar información útil, oportuna, segura e inmediata a la Dirección Docente del centro en que se encuentre y a sus estudiantes, con el objetivo de satisfacer sus necesidades de forma autónoma y flexible.

1.3 Ejemplos de Sistemas de Gestión de postgrado existentes a nivel nacional e internacional.

Debido al desarrollo que existe actualmente en el mundo, toda institución que aspira superarse debe automatizar sus servicios, para lograr una mayor confiabilidad de la información y rapidez en su búsqueda.

El Sistema Integrado de Gestión Académica (SIGA) es un programa fácil y rápido de utilizar que permite la gestión completa de cualquier centro de formación pues está compuesto de diferentes módulos entre los que se encuentran: Alumnos, Opciones de alumnos, Diplomas, Profesores, Horarios, Ingresos-Gastos, Generador de listados, Automatización de exámenes, Automatización de encuestas y Automatización de matriculación de alumnos. (2)

La Universidad de Murcia tiene un sistema de gestión en el cual se publican documentos legales, información general de actividades que se ofertan (proceso de preinscripción y proceso de matrícula), además de información de interés tanto para alumnos como para profesores. (3)

La Universidad de Cantabria tiene un sistema que además de informaciones necesarias, permite la preinscripción y matrícula de los estudiantes a las actividades ofertadas, de manera online. (4)

Actualmente, en el país no se cuenta en todas las universidades con un sistema de gestión de postgrado, solamente la Universidad Central de las Villas cuenta con un sistema, el que permite llevar el control de las actividades tanto a nivel central como por sus facultades. En su versión actual esta ajustado al Reglamento de Postgrado de la Educación Superior, en vigor desde finales de 2004.

El sistema de gestión de postgrado de la **Universidad Central de las Villas**: permite controlar la planificación, ejecución y terminación tanto de las actividades de Superación Profesional (cursos, entrenamientos y diplomados) como Superación Académica (maestrías, especialidades y doctorados). Posibilita la planificación de los postgrados, así como el control de las matrículas y evaluaciones. Brinda algunos informes predeterminados como planificaciones de cursos, listado de matrícula por cursos, actas de exámenes, informes finales de postgrados concluidos, sábanas de notas de postgrados múltiples, certificados de evaluación de cursos, entrenamientos y diplomados, certificaciones de estudios terminados de maestrías y especialidades.

Los sistemas de gestión de postgrado antes mencionados pudieran ser una alternativa, principalmente SIGA que es el más completo y se puede adaptar a cualquier centro de enseñanza, por estar compuesto por módulos que se pueden adquirir por separado, pero los sistemas que se utilicen en los centros cubanos tienen que regirse por la Resolución No 132/ 2004 y el sistema que se utiliza actualmente en la universidad es el que se emplea en la Universidad Central de las Villas, el cual no satisface las necesidades de la Dirección de Formación Postgraduada, por lo que se requiere la implementación de un nuevo sistema de acuerdo a las exigencias de la misma.

1.4 Tendencias y tecnologías actuales.

1.4.1 Lenguaje de Programación y Tecnologías del lado del Cliente.

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. Permite crear programas y software. (5)

1.4.1.1 Java Script.

A principios de los años 90, la mayoría de usuarios que se conectaban a Internet lo hacían con módems a una velocidad máxima de 28.8 Kbps. En esa época, empezaban a desarrollarse las primeras aplicaciones Web y por tanto, comenzaban a incluir formularios complejos.

Con unas aplicaciones Web cada vez más complejas y una velocidad de navegación tan lenta, surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario. De esta forma, si el usuario no rellenaba correctamente un formulario, no se le hacía esperar mucho tiempo hasta que el servidor volviera a mostrar el formulario indicando los errores existentes.

Brendan Eich, un programador que trabajaba en Netscape, pensó que podría solucionar este problema adaptando otras tecnologías existentes al navegador Netscape Navigator 2.0, que iba a lanzarse en 1995. Inicialmente, Eich denominó a su lenguaje LiveScript (6) que permitía crear pequeños programas en las páginas y que era mucho más sencillo de utilizar que Java, pero no duró mucho ese nombre, pues antes de lanzar la primera versión del producto se forjó una alianza con Sun Microsystems, creador de Java, para desarrollar en conjunto ese nuevo lenguaje.

La alianza hizo que Java Script se diseñara como un hermano pequeño de Java, solamente útil dentro de las páginas Web y mucho más fácil de utilizar, de modo que cualquier persona, sin conocimientos de programación pudiese adentrarse en el lenguaje y utilizarlo a sus anchas.

Netscape 2.0 fue el primer navegador que entendía Java Script y su estela fue seguida por los navegadores de la compañía Microsoft a partir de la versión 3.0. (7)

Java Script es un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento, permite crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web.

Por su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Java Script permite crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Entre las acciones típicas que se pueden realizar en Java Script se identifican dos vertientes. Por un lado los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se puede crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Java Script es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas. Además, pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente. (8)

Es bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad.

1.4.1.2 Lenguaje de Marcación de Texto (HTML). (9)

El físico nuclear Tim Berners Lee definió la primera versión de HTML en el año 1989 mientras trabajaba en el Centro Europeo para la investigación Nuclear (CERN), pretendía facilitar el acceso a todos los datos del Centro independientemente del ordenador en el que se encontrara esa información, tanto si estaba en el propio centro o en los ordenadores de las diferentes instituciones que colaboraban con el CERN.

Esta primera versión solamente marcaba encabezados, listas y anclas pero fue suficiente para que se creara la World Wide Web. Surgieron varias versiones de este primer HTML pero ninguna de ellas

llegaría a convertirse en el estándar oficial. Lo más cercano fue el HTML 2.0, al cual se le puso el nombre para distinguirlo de la colección de estándares no oficiales que iban surgiendo.

Después evolucionaría hasta llegar a cuatro versiones más. HTML 4.0 fue la última en 1998 y posteriormente llegaría el lenguaje que aún utilizamos, el XHTML hijo del XML y el HTML.

El lenguaje HTML permite representar cualquier clase de información que se encuentre almacenada en una página Web, codificar o preparar documentos de hipertexto.

HTML es el lenguaje con el que se definen las páginas Web. Básicamente se trata de un conjunto de etiquetas que sirven para definir la forma en la que se presentará el texto y otros elementos de la página Web.

1.4.2 Lenguaje de Programación y Tecnologías del lado del Servidor.

1.4.2.1 PHP (HyperText Preprocesor). (10)

PHP es uno de los lenguajes de lado del servidor más extendidos en la Web usado frecuentemente para la creación de contenido para sitios Web con los cuales se puede programar las páginas HTML y los códigos fuente. Se trata de un "lenguaje interpretado" usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web.

Es un lenguaje de programación gratuito e independiente de plataforma, rápido, seguro, con una gran librería de funciones y mucha documentación.

PHP está desarrollado en política de código abierto por lo que a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores.

Presenta compatibilidad con las bases de datos más comunes, como MySQL, MS SQL Server, Oracle, Informix, entre otras, aunque destaca su conectividad con MySQL.

PHP es un desarrollo "Open Source" pues permite ver y modificar el código fuente de la aplicación.

1.4.3 Sistemas Gestores de Base de Datos (SGBD). (11)

Un Sistema Gestor de base de datos (SGBD) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Algunas de las características deseables en un Sistema Gestor de base de datos son:

- Control de la redundancia: La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.
- Restricción de los accesos no autorizados: cada usuario ha de tener unos permisos de acceso y autorización.
- Cumplimiento de las restricciones de integridad: el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad.

1.4.3.1 MySQL. (12)

MySQL es un gestor de base de datos sencillo de usar e increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

Es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las características principales de MySQL son:

- El servidor de base de datos MySQL es muy rápido, fiable y fácil de usar.
- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo muy buen nivel de seguridad en los datos.
- Es Open Source. El código fuente de MySQL se puede descargar y está accesible a cualquiera que desee usar y modificar el software, por otra parte, usa la licencia GPL para aplicaciones no comerciales.

1.4.4 Metodologías de desarrollo de software.

Las Metodologías de Desarrollo de Software son el conjunto de técnicas y procedimientos que nos permiten conocer los elementos necesarios para definir un proyecto de software; en otras palabras, es la base para la edificación de un proyecto de software.

Sirven para aumentar la "calidad" del software, al permitir controlar de manera transparente todo el proceso de desarrollo, fundamentalmente posibilita producir lo deseado en el tiempo y con el costo esperado. (13)

1.4.4.1 Proceso Unificado de Desarrollo (RUP). (14)

RUP es uno de los procesos más generales que existe, está enfocado a cualquier tipo de proyecto así no sea de software, se basa en la documentación generada en cada uno de sus cuatro fases:

1. **Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
2. **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable.
3. **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario, poniéndose a consideración de un subconjunto de usuarios.
4. **Transición:** El producto está listo para su instalación en las condiciones reales, lo que puede implicar reparación de errores.

RUP se basa en casos de uso para describir lo que se tiene y lo que se espera del software, está muy orientado a la arquitectura del sistema a implementarse, documentándose de la mejor manera, basándose en UML (Lenguaje de Modelado Unificado).

Las características principales del proceso son:

- **Dirigido por Casos de Uso:** Los Casos de Uso reflejan las necesidades de los usuarios que se capta cuando se modela el negocio y se representa en los requerimientos.
- **Centrado en la Arquitectura:** La arquitectura muestra la visión del sistema completo describiendo los elementos del modelo que son más importantes para su construcción.
- **Iterativo e incremental:** Cada fase se desarrolla en iteraciones, las que involucran actividades de todos los flujos de trabajo.

RUP propone los siguientes flujos de ingeniería:

Modelo de Negocios: Entendiendo las necesidades del negocio.

Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.

Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.

Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.

Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Flujos de apoyo:

Configuración y administración del cambio: Guardando todas las versiones del proyecto.

Administrando el proyecto: Administrando horarios y recursos.

Ambiente: Administrando el ambiente de desarrollo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

1.4.5 Lenguaje de Modelado. (15)

UML es un lenguaje que proporciona un vocabulario y unas reglas para permitir una comunicación, posibilitando expresar de una forma gráfica un sistema de manera que otro lo puede entender y permite especificar cuáles son las características de un sistema antes de su construcción.

Es independiente del lenguaje de implementación, de tal forma que los diseños realizados se pueden implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).

Prácticamente todas las herramientas CASE y de desarrollo la han adaptado como lenguaje de modelado por lo que es considerado como un método formal de modelado, lo que aporta ventajas como: mayor rigor en la especificación, permite realizar una verificación y validación del modelo realizado y se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.

1.4.6 Servidores Web.

Un servidor Web es un software que se encuentra en un ordenador y que se encarga de atender las solicitudes que recibe de los distintos usuarios que acceden a la red. Los más utilizados actualmente son el Internet Information Server (IIS) y el APACHE

IIS es un software privativo que pertenece a Microsoft y funciona bajo Windows, APACHE es gratuito, multiplataforma y dispone de versiones para la mayoría de sistemas operativos. (16)

1.4.7 Arquitectura.

La arquitectura de una aplicación es la vista conceptual de la estructura de ésta. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como está distribuido este código. (17)

1.4.7.1 Arquitectura en capas.

El modelo n- capas ha emergido como la arquitectura predominante para la construcción de aplicaciones multiplataforma, se está posicionando rápidamente como la piedra angular de los desarrolladores de aplicaciones empresariales. Proporciona gran cantidad de beneficios para las empresas que necesiten soluciones flexibles y fiables para resolver complejos problemas inmersos en cambios constantes.

Este modelo ofrece ventajas como:

- Desarrollos paralelos (en cada capa).
- Aplicaciones más robustas debido al encapsulamiento.
- Mantenimiento y soporte más sencillo.
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad). (18)

La arquitectura tres capas es una de las tres propuestas de arquitecturas de capas para sistemas de información la cual dispone de tres tipos de nodos:

Clientes: interactúan con los usuarios.

Servidores de aplicación: procesan los datos.

Servidores de base de datos: almacenan todos los datos.

Esta arquitectura es considerada, desde el punto de vista tecnológico, lo más avanzado que puede implementarse como estructura de las aplicaciones informáticas. Provee gran flexibilidad y escalabilidad en la configuración.

Las ventajas que proporciona la arquitectura de tres capas son:

- Reduce el tráfico de información en la red por lo que mejora el rendimiento de los sistemas.
- Brinda una mayor flexibilidad de desarrollo y de elección de plataformas sobre la cual montar las aplicaciones.
- Provee escalabilidad horizontal y vertical.
- Se mantiene la independencia entre el código de la aplicación (reglas y conocimiento del negocio) y los datos, mejorando la portabilidad de las aplicaciones.
- Los lenguajes sobre los cuales se desarrollan las aplicaciones son estándares lo que hace más exportables las aplicaciones entre plataformas.
- Dado que mejora el rendimiento al optimizar el flujo de información entre componentes, permite construir sistemas críticos de alta fiabilidad.
- El mismo hecho de localizar las reglas del negocio en su propio ambiente, en vez de distribuirlos en la de interfaz de usuario, permite reducir el impacto de hacer mantenimiento, cambios urgentes de última hora o mejoras al sistema.
- Disminuye el número de usuarios conectados a la base de datos.

Estas tres capas pueden estar todas en el mismo equipo o distribuidas en una red local o en una extranet (comunicación remota a través de internet). Esto le permite balancear los recursos según las exigencias de procesamiento de su entorno. (19)

1.4.7.2 Modelo Vista Controlador (MVC).

El patrón de diseño de arquitectura MVC se utiliza principalmente en aplicaciones que manejan gran volumen de datos y transacciones complejas donde se requiere una mejor separación de conceptos permitiendo la programación en diferentes capas de manera paralela e independiente: modelo, vista y controlador.

Modelo: la representación de la información que contiene la aplicación, es decir, los datos que proveen de información al usuario o a la aplicación misma.

Vista: la representación del modelo en forma gráfica disponible para interactuar con el usuario, en una aplicación Web es una página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones.

Controlador: la capa que se encarga de manejar y responder las solicitudes del usuario, procesando la información correspondiente y en caso de ser necesario, modificar el modelo.

Algunas de las ventajas que ofrecen son:

- Separar los datos de la representación visual de los mismos.
- Facilita agregar nuevos tipos de datos ya que son independientes del funcionamiento de las otras capas.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Permite el escalamiento de la aplicación.

Además presenta desventajas como:

- La separación de conceptos en capas agrega complejidad al sistema.
- Se incrementa la cantidad de archivos a mantener y desarrollar.
- La curva de aprendizaje del patrón es más alta. (20)

Su característica principal es 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.

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. No tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

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. Interactúa con el Modelo a través de una referencia al propio Modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo. (21)

1.4.7.3 Arquitectura Orienta a Servicios (SOA). (22)

La arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios.

La estrategia de orientación a servicios permite la creación de servicios y aplicaciones compuestas que pueden existir con independencia de las tecnologías subyacentes. En lugar de exigir que todos

los datos y lógica de negocio residan en un mismo ordenador, el modelo de servicios facilita el acceso y que los servicios están diseñados para ser independientes, autónomos y para interconectarse adecuadamente, pueden combinarse y recombinarse con suma facilidad en aplicaciones complejas que respondan a las necesidades de cada momento de una organización.

El resultado final es que las organizaciones que adoptan la orientación a servicios pueden crear y reutilizar servicios y aplicaciones y adaptarlos ante los cambios evolutivos que se producen dentro y fuera de ellas, y con ello adquirir la agilidad necesaria para ganar ventaja competitiva.

Las empresas que mejor pueden aprovechar SOA son las de gran tamaño que tienen un gran número de sistemas de información adquiridos para ir dando solución a diferentes problemas. Pero que buscan una creciente integración de todos los sistemas para obtener mejor calidad de datos, evitar procesos redundantes y lograr una visión más clara y completa del negocio.

1.4.7.4 ¿Por qué utilizar Arquitectura en capas?

MVC posee muchas ventajas para ser considerado como arquitectura candidata, sin embargo, se utiliza principalmente en aplicaciones que manejan gran volumen de datos y transacciones complejas; mientras que SOA es un patrón de arquitectura que es ideal para grandes empresas con gran número de sistemas de información que se necesiten integrar.

La arquitectura en 3 capas permite una centralización de los aspectos de seguridad y transaccionalidad, no replicación de lógica de negocio en los clientes permitiendo que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento.

1.5 Herramientas Utilizadas.

1.5.1 Embarcadero ERStudio 7.1. (23)

ERStudio 7.1 es la última versión liberada de la solución para la arquitectura y el modelado de base de datos que provee el descubrimiento, documentación y reutilización de los activos de información a través de modelos visuales. Facilita las mejores prácticas para la seguridad y calidad en el diseño y modelado de base de datos, provee nuevas propiedades para la seguridad y la clasificación de información sensible, introduce mejoras en la validación y simplifica los modelos de revisión.

Con ERStudio 7.1 se facilita el cumplimiento de requerimientos y reducir el riesgo de que la información sea utilizada en forma inapropiada utilizando capacidades mejoradas para la identificación, clasificación y comunicación de políticas alrededor de la información sensible, además,

mejoró los modelos de consistencia y calidad al proveer reglas de validación para reforzar estándares específicos.

1.5.2 Dreamweaver.

Dreamweaver es la herramienta de diseño de páginas Web más avanzada, cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- Hojas de estilo y capas
- Java Script para crear efectos e interactividades
- Inserción de archivos multimedia (24)

Este producto es el editor Web más ampliamente utilizado por los profesionales. Puede realizar diseño, desarrollo y mantenimiento de sitios Web y aplicaciones para Internet. Dreamweaver es el producto más avanzado del mercado. Una de las características que diferencia a Dreamweaver de la mayoría de las herramientas similares es la funcionalidad de sus plantillas. Esto permite que el diseño y mantenimiento de páginas Web sea una tarea mucho más simple. Además, las plantillas permiten dar seguridad al diseño de manera que los editores Web no tengan que preocuparse en realizar esta tarea, ni teman estropear el sitio. (25)

Además es un programa que se puede actualizar con componentes, que fabrica tanto Macromedia como otras compañías, para realizar otras acciones más avanzadas.

Los lenguajes de programación que domina Dreamweaver MX 2004 son ASP, CSS, PHP, SQL, JSP, y XML. El potencial del software en cuanto a la capacidad de programar bajo los lenguajes que acabados de citar es de lo más amplio, permitiendo la creación de aplicaciones y diseños Web complejos.

Uno de los puntos de mayor énfasis en esta edición de Dreamweaver son el soporte y las características de desarrollo en Cascading Style Sheet (CSS), haciendo posible creaciones con más facilidad y precisión, aplicando herramientas capaces de inspeccionar el código escrito. Otro aspecto capaz de ser analizado es la compatibilidad con los diversos navegadores, para que todos puedan visualizar la página correctamente.

La posibilidad de crear botones flash, formularios, Java Scripts, y más, es de gran ayuda. Además de poder insertar elementos Web, se encuentra gran precisión en la importación de información de Word y Excel, con las funciones de copiar y pegar. (26)

1.5.3 Visual Paradigm.

Esta herramienta está desarrollada por Visual Paradigm Internacional una de las principales compañías de herramientas CASE. Su mayor éxito consiste en la capacidad de ejecutarse sobre diferentes sistemas operativos lo que le confiere la característica de ser multiplataforma. Visual Paradigm utiliza UML como lenguaje de modelado ofreciendo soluciones de software que permiten a las organizaciones desarrollar las aplicaciones de calidad más rápido, bien y más barato. Es muy fácil de usar y presenta un ambiente gráfico agradable para el usuario. Su notación es muy parecida a la estándar, permite configurar las líneas de redacción, el modelado de base de datos, el modelado de requerimientos, el modelado del proceso de negocio, la interoperabilidad, la generación de documentación y la generación de código base para diferentes lenguajes de programación como Java, C# y PHP además de permitir la integración con herramientas de desarrollo. (27)

1.5.4 Gestor de Base de Datos MySQL. (28)

MySQL un gestor de bases de datos que ha sido diseñado bajo el objetivo primordial de la velocidad, destaca también sobre todo por su bajísimo consumo en recursos, tanto a nivel de CPU como de memoria.

Dentro de sus características destacan:

- Velocidad
 - Arquitectura multihilo: Múltiples clientes tienen acceso concurrente.
 - Caché de consultas: Cachea los resultados de las consultas comunes.
- Fiabilidad
- Facilidad de uso
 - Interfaz de línea de comandos.
 - Herramientas gráficas-Escritorio/ basadas en Web.
- Soporte Multi-Usuario
 - Múltiples clientes tienen acceso concurrente a una (o más) bases de datos simultáneamente.
 - Sistema de privilegios de usuarios potente y flexible.
 - Esquemas de autenticación basados en usuario-máquina.
- Escalabilidad
 - Bases de datos conteniendo cerca de 50 millones de registros.
- Portabilidad
 - UNIX/ No-Unix: Linux, Solaris, Windows.

- Internacionalización
 - Soporte para varios lenguajes y juegos de caracteres.
- Amplio soporte de aplicaciones
 - Base de datos MySQL de aplicaciones para el escritorio y la Web.
 - APIs para: C/C++, Java, PHP, Perl, y otros.

1.5.5 Zend Studio. (29)

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones Web en lenguaje PHP.

El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS.

El editor, la parte del programa que nos permite escribir los scripts es bastante útil para la programación en PHP. Lo más destacable es que contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir, como permitir editar varios archivos, y moverse fácilmente entre ellos, marcar a qué elementos corresponden los inicios y cierres de las etiquetas, paréntesis o llaves, moverse al principio o al final de una función, identificación automática del código.

Dispone de herramientas para gestionar los proyectos, muy útiles para mejorar la productividad en la programación. Los proyectos permiten guardar mucha más información al programa sobre los archivos, discos, servidores, que se gestionen en las aplicaciones PHP.

Zend Studio dispone de una herramienta muy interesante de debug o depuración. Permitiendo ejecutar páginas y conocer en todo momento el contenido de las variables de la aplicación y las variables del entorno como las cookies, las recibidas por formulario o en la sesión.

1.5.6 WAMP. (30)

(Windows-Apache-MySQL- PHP/Python/PERL). El término hace referencia al sistema creado por la conjunción de esas aplicaciones libres (de código abierto) y el sistema operativo Windows. Este grupo de aplicaciones generalmente son usados para crear servidores web.

WAMP provee a los desarrolladores con los cuatro elementos necesarios para un servidor web: un sistema operativo (Windows), un manejador de base de datos (MySQL), un software para servidor web (Apache) y un software de programación script web (PHP, Python o PERL).

Conclusiones

En este capítulo se realizó un estudio de las tendencias actuales para el desarrollo de Sistemas de Gestión de la Información, fundamentándose las tecnologías que son más apropiadas para desarrollar el sistema, donde se propone que la programación por parte del cliente se realice en Java Script y HTML; y por parte del servidor con PHP, se utilizará la metodología RUP y como herramienta case Visual Paradigm, además MySQL como gestor de base de datos y APACHE como servidor Web.

Capítulo 2

Características del Sistema

Introducción

Para desarrollar una aplicación es necesario comprender la estructura y dinámica de la organización en la cual se va a implantar para delimitar el problema e identificar las mejoras potenciales, asegurando que los clientes y desarrolladores tengan un entendimiento común de la misma, para lo que se realiza el modelamiento del negocio.

Del modelo de negocio se derivan los requerimientos funcionales y no funcionales del sistema que darán solución al problema planteado.

En el presente capítulo se exponen las reglas del negocio asociadas al campo de acción, se describen los actores y trabajadores del negocio, los actores del sistema y las funcionalidades que les ofrecerán.

2.1 Modelo de Negocio.

A la Dirección de Formación Postgraduada de la UCI se le realiza la solicitud para la inscripción de una actividad por parte del solicitante que para la inscripción de cursos, entrenamientos, talleres y diplomados es un profesor, mientras que para las maestrías es la Comisión Asesora para la Educación de Postgrado (COPED) y para los doctorados, la Comisión Nacional de Grados Científicos (CNGC), proceso que se denomina “Inscribir_Actividad”, esta actividad inscrita permanece un período de tiempo publicada para informar a los interesados, posibilitando que se realice la matrícula de la actividad, “Matricular”.

Cuando se cierra la matrícula y se comienza a desarrollar la actividad, los estudiantes son evaluados y surge la necesidad de controlar sus evaluaciones, proceso denominado “Evaluar_Estudiante”. Concluida la misma, el estudiante aprobado realiza la recogida del título, “Recoger_Título”.

Por último se tiene el proceso de control de reportes, “Reportar”, que es la respuesta a toda la información que necesite el Director de postgrado.

2.2 Reglas del Negocio.

Las reglas del negocio definen políticas que deben cumplirse o condiciones que deben de satisfacerse, por lo que regulan algún aspecto del negocio.

1. Sólo los graduados universitarios pueden matricular en las actividades de postgrado, además deben estar autorizados y avalados por la dirección de su centro de trabajo; cumpliendo con las exigencias que para un programa de postgrado, establezcan los Comité Académicos o Coordinadores para el ingreso en la actividad.
2. Para ejecutarse las actividades de postgrado los coordinadores deberán establecer la cantidad de estudiantes a matricular.
3. Para la inscripción de las actividades de postgrado, es necesario que estas cumplan con todas las exigencias de inscripción.
4. Dentro de la Superación Profesional se encuentran los cursos y entrenamientos, ambos con extensión mínima de dos créditos, los diplomados con extensión mínima de quince créditos y otras formas organizativas que se establezcan en el reglamento y que otorgarán como mínimo un crédito.
5. Los programas de maestría y de especialidad de postgrado son aprobados por el MES, a propuesta de la Comisión Asesora para la Educación de Postgrado (COPED).
6. La extensión mínima de los programas de maestría y de especialidad de postgrado son de 70 y 100 créditos, respectivamente.
7. Para obtener un título de cualquier actividad de postgrado, deben cumplirse con los requisitos específicos que se aprobaron en su programa.
8. El Profesor es el encargado de registrar las evaluaciones de los estudiantes matriculados en las actividades que imparte.
9. Las actividades de postgrado transitan por los siguientes estados: inscritos, en curso y culminado.
10. La Técnico En Control y Análisis del trabajo Docente Investigativo en la Educación Superior (TECADIES) tendrá total acceso al registro de evaluaciones de todas las actividades de postgrado inscritas y podrá modificar datos de estudiantes, profesores y actividades de postgrado.
11. Las evaluaciones se expresan de la siguiente manera: excelente (5), bien (4), aprobado (3) o desaprobado (2).

2.3 Actores del Negocio.

Un actor del negocio es cualquier individuo, grupo organización o máquina que interactúa con el negocio; éste siempre permanece fuera de las fronteras del negocio. El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio, por lo que no representa un usuario físico, pues varios usuarios físicos pueden realizar el mismo rol. (Ver tabla 2.1)

Actores del Negocio	Descripción
Director Postgrado	Inicia el proceso "Generar_Reportes". Envía Datos para la solicitud de reportes y recibe el informe pedido.
Solicitante	Inicia el proceso "Inscribir_Actividad". Solicita convocatoria y únicamente recibe notificación de rechazo.
Estudiante	Inicia el proceso "Matricular". Solicita matrícula a la actividad de postgrado deseada y recibe notificación de aceptación o rechazo. Inicia el proceso "Recoger_Título". Solicita el título después de haber culminado una actividad de postgrado.
Profesor	Se comporta como solicitante.
COPED	Se comporta como solicitante.
CNGC	Se comporta como solicitante.

Tabla 2.1 Actores del Negocio

2.4 Trabajadores del Negocio.

Un trabajador del negocio representa a personas o sistemas (software) dentro del negocio que son las que realizan las actividades que están comprendidas dentro de un caso de uso; está dentro de la frontera del negocio y se convertirá en usuario del sistema a construir.

(Ver tabla 2.2).

Trabajadores del Negocio	Descripción
Jefe Superación Profesional	Participa en el proceso "Inscribir_Actividad", revisando todas las solicitudes asociadas a los cursos, talleres y entrenamientos para aceptarlas o denegarlas.
Jefe Superación Académica	Participa en el proceso "Inscribir_Actividad", revisando las solicitudes referentes a las maestrías y doctorados.
Administrador	Interviene en todos los procesos del negocio: "Matricular", gestiona la solicitud de matrícula del estudiante e informa cuando culmine el proceso. "Generar_Reportes", es el encargado de generar la información solicitada y comunicarla. "Inscribir_Actividad", responsable realizar la inscripción de actividades y publicarla. "Recoger_Título", responsable de entregar el título a los estudiantes que culminaron las actividades de postgrado.
TECADIES	Se comporta como Administrador.
Secretaria	Se comporta como Administrador.
Comité Académico	Participa en "Matricular", verificando el expediente del estudiante a matricular en maestrías o doctorados.
Consejo Postgrado	Participa en el proceso "Inscribir_Actividad", revisando las solicitudes asociadas a los diplomados.
Profesor	Participa en "Evaluar _ estudiante", otorgando evaluación a cada estudiante matriculado en la actividad que imparte.

Tabla 2.2 Trabajadores del Negocio

2.5 Entidades del Negocio.

1. Reporte.
2. Expediente.
3. Matrícula
4. Acta de evaluación.
5. Registro de actividad.

2.6 Casos de Usos del Negocio.

Los casos de uso del negocio representan a un proceso de negocio, que se corresponde con una secuencia de acciones que producen un resultado observable para determinados actores del negocio. Desde la perspectiva de un actor individual, define un flujo de trabajo completo que produce resultados deseables.

2.6.1 Casos de Usos del Negocio a tratar.

1. Inscribir_Actividad.
2. Generar_Reportes.
3. Matricular_Estudiantes.
4. Recoger_Título.

2.6.2 Diagrama de casos de uso del negocio.

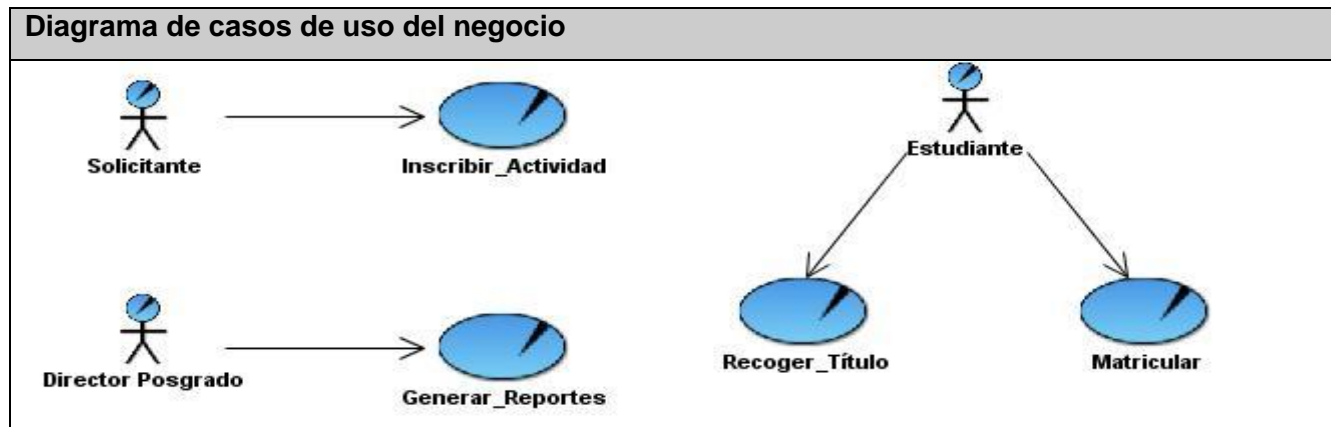


Tabla 2.3 Diagrama de casos de uso del negocio.

2.7 Diagramas de Actividades.

2.7.1 Diagrama de Inscribir_Actividad.

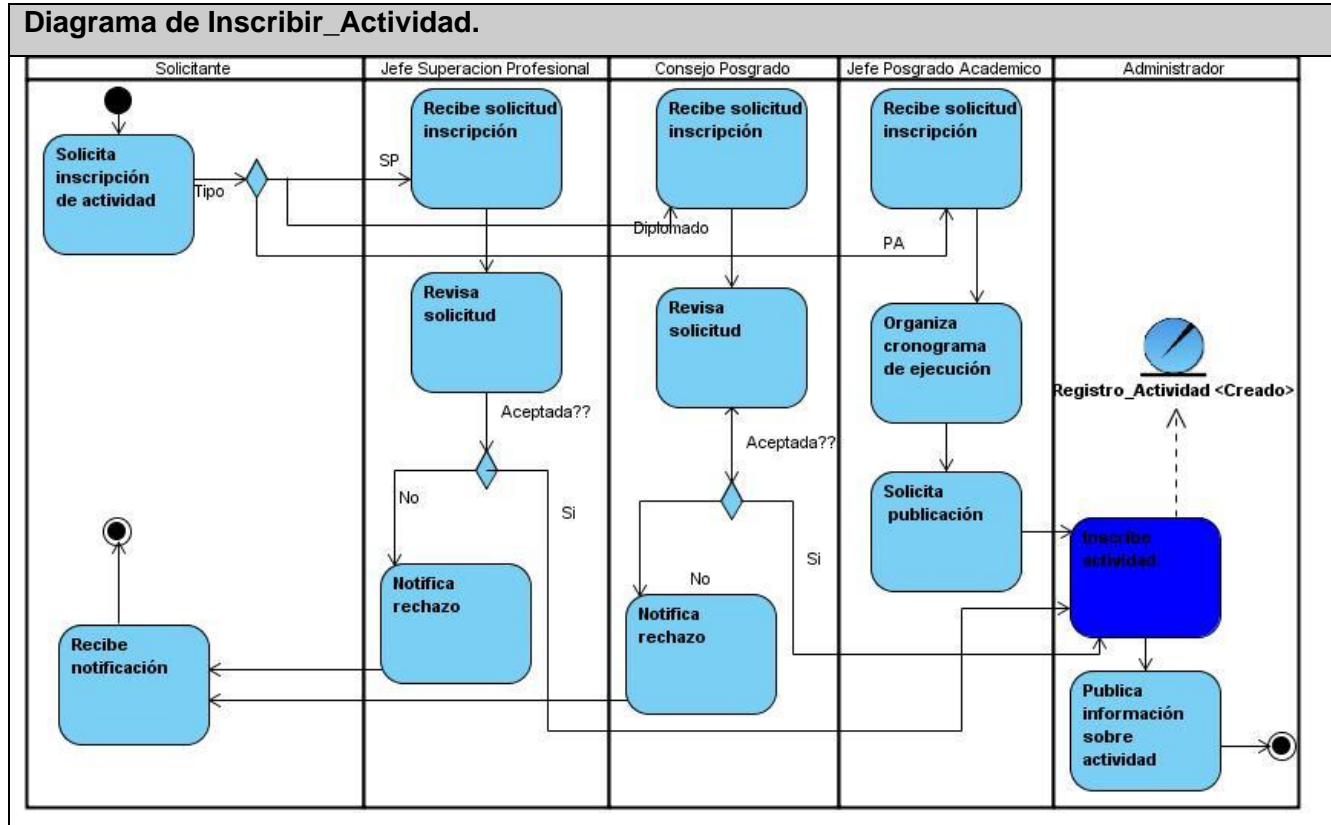


Tabla 2.4 Inscribir_Actividad.

2.7.2 Diagrama de Reportar.

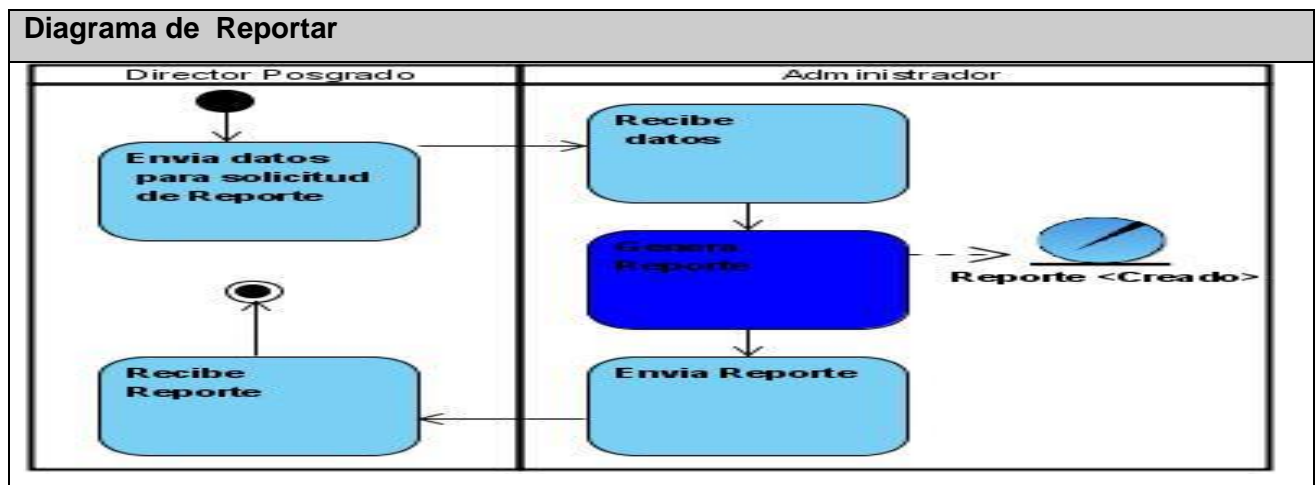


Tabla 2.5 Diagrama de Reportar.

2.7.3 Diagrama de Matricular.

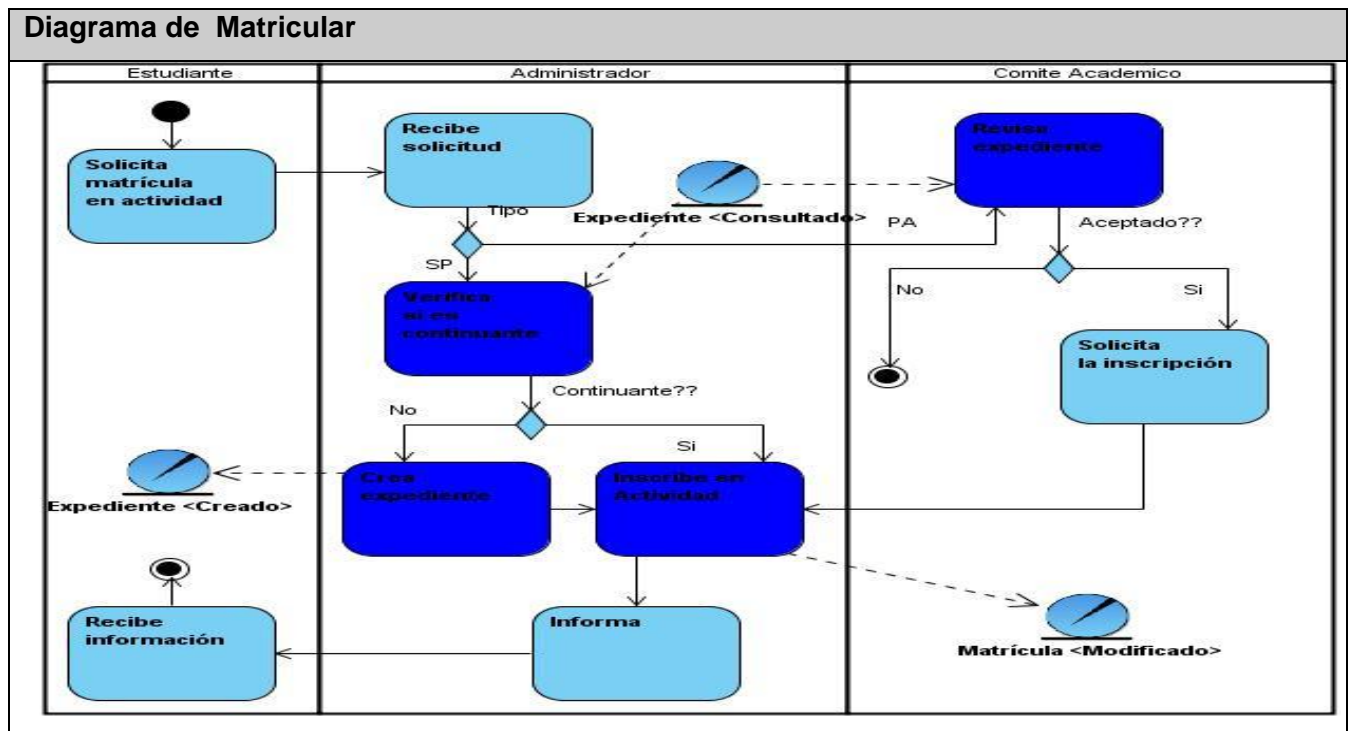


Tabla 2.7 Diagrama de Matricular

2.7.4 Diagrama de Recoger_Título.

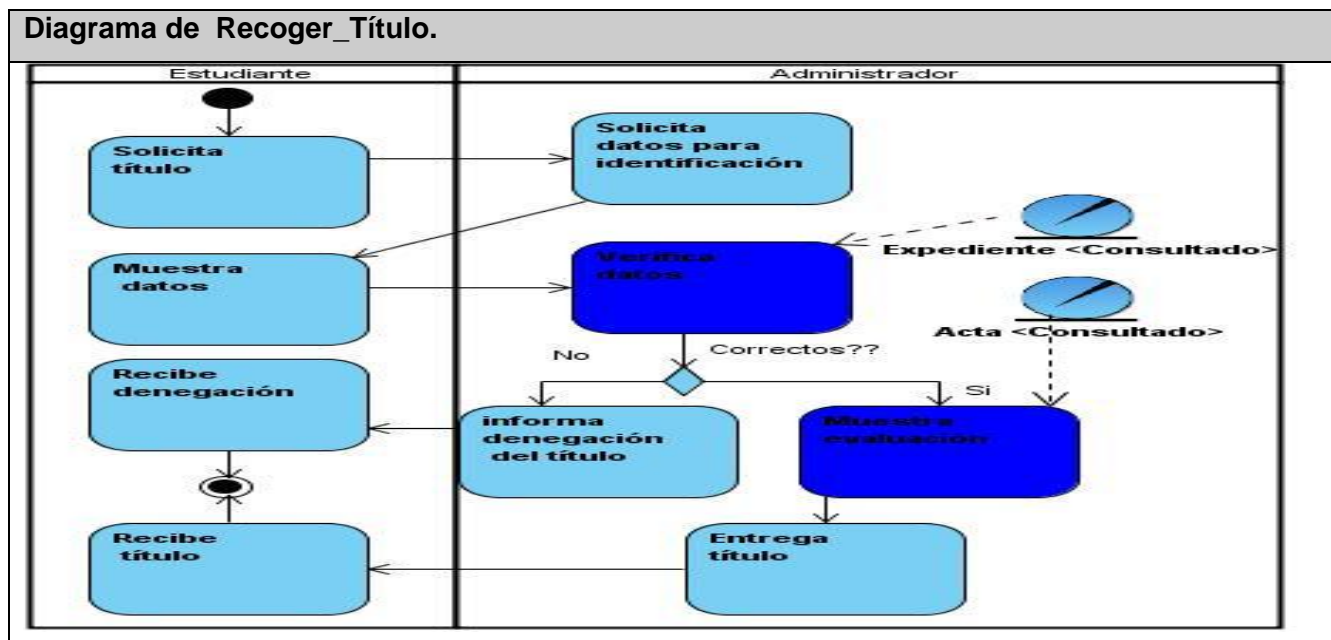


Tabla 2.8 Diagrama de Recoger_Título.

2.8 Requerimientos de Software.

2.8.1 Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir de acuerdo a las necesidades del usuario, por lo que el sistema debe ser capaz de:

RF1 Insertar persona.

RF2 Modificar persona.

RF3 Eliminar persona.

RF4 Insertar actividad.

RF5 Eliminar actividad.

RF6 Modificar actividad.

RF7 Insertar noticia.

RF8 Modificar noticia.

RF9 Eliminar noticia.

RF10 Autenticar usuario.

10.1. Verificar privilegios de usuarios.

RF11 Asignar nota en actividad.

RF12 Otorgar crédito.

RF13 Buscar persona.

RF14 Mostrar listado de estudiantes que no han recogido su certificado.

RF15 Mostrar la cantidad de estudiantes que han abandonado la actividad de postgrado y cuántos lo han finalizado.

RF16 Mostrar listado de estudiantes adiestrados con actividades acreditadas.

RF17 Mostrar actividades por estado de ejecución.

RF18 Mostrar actividades impartidas por un profesor determinado.

RF19 Mostrar listado de estudiantes por actividad.

RF20 Mostrar actividades por área.

RF21 Mostrar las actividades que se ofertan de Superación Profesional.

RF22 Mostrar las actividades que se ofertan de Superación Académica.

RF23 Mostrar las actividades vencidas y evaluaciones.

RF24 Mostrar las actividades matriculadas.

RF25 Mostrar crédito activo.

RF26 Buscar actividad.

RF27 Solicitar matrícula en actividad

RF28 Aprobar solicitud de matrícula

2.8.2 Requerimientos No Funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el sistema debe tener, es decir, las características que lo hacen atractivo, usable, rápido o confiable.

RNF1. Seguridad

- 1.1 Confidencialidad, se establecerán diferentes niveles de acceso (roles) para los diferentes usuarios que interactúen con el sistema para garantizar que la información manejada esté protegida de acceso no autorizado y divulgación.
- 1.2 Integridad, debido a la importancia de la información manejada será necesario un extremo cuidado en su protección, por lo que la transmisión de datos por la red se realizará a través de un protocolo seguro.
- 1.3. Disponibilidad, la información estará disponible las 24 horas del día, tanto para el trabajo de los usuarios como para las acciones de mantenimiento.

RNF2. Legales

- 2.1. El sistema se ajustará a lo planteado en la versión vigente de la Resolución No 132.
- 2.2. Estará regido por el manual de normas y procedimientos de la Resolución No 132/2004.

RNF3. Software

- 3.1. Lenguaje de programación: PHP.
- 3.2. Herramienta para el almacenamiento de información: Sistema gestor de bases de datos MySQL.

RNF4. Apariencia o Interfaz Externa

- 4.1. Tendrá apariencia profesional, pues será desarrollado, para llevar acabo la gestión del postgrado en una institución de enseñanza superior (UCI).
- 4.2. Sin gran cantidad de imágenes para agilizar el flujo de información, además, el objetivo del sistema es priorizar la gestión de la información.
- 4.3. Debe poseer una interfaz que este en concordancia con el estilo de diseño de las aplicaciones UCI.

RNF5. Rendimiento.

- 5.1. Ante cualquier acción el sistema debe dar respuesta en un tiempo 15 segundos.
- 5.2. El sistema debe soportar un promedio de transacciones por segundo.

2.9 Actores del Sistema.

Actores	Descripción
TECADIES	Se comporta como Administrador
Secretaria	Se comporta como Administrador
Profesor de postgrado	Es el encargado de la gestión de las evaluaciones. Se comporta como usuario
Coordinador	Es el encargado de aprobar las solicitudes de matrícula. Se comporta como usuario.
Administrador	Es el encargado de realizar la gestión de personas, inscripción de actividades y la matrícula de estudiantes en las actividades de postgrado. Se comporta como Profesor y usuario.
Supervisor	Se comporta como: Administrador y usuario.
Usuario	Se autentica.
Estudiante	Es el encargado de solicitar inscripción de matrícula en las actividades, además de poder acceder a información de interés. Se comporta como usuario.

Tabla 2.9 Actores del Sistema.

2.10 Diagrama de casos de uso del sistema.

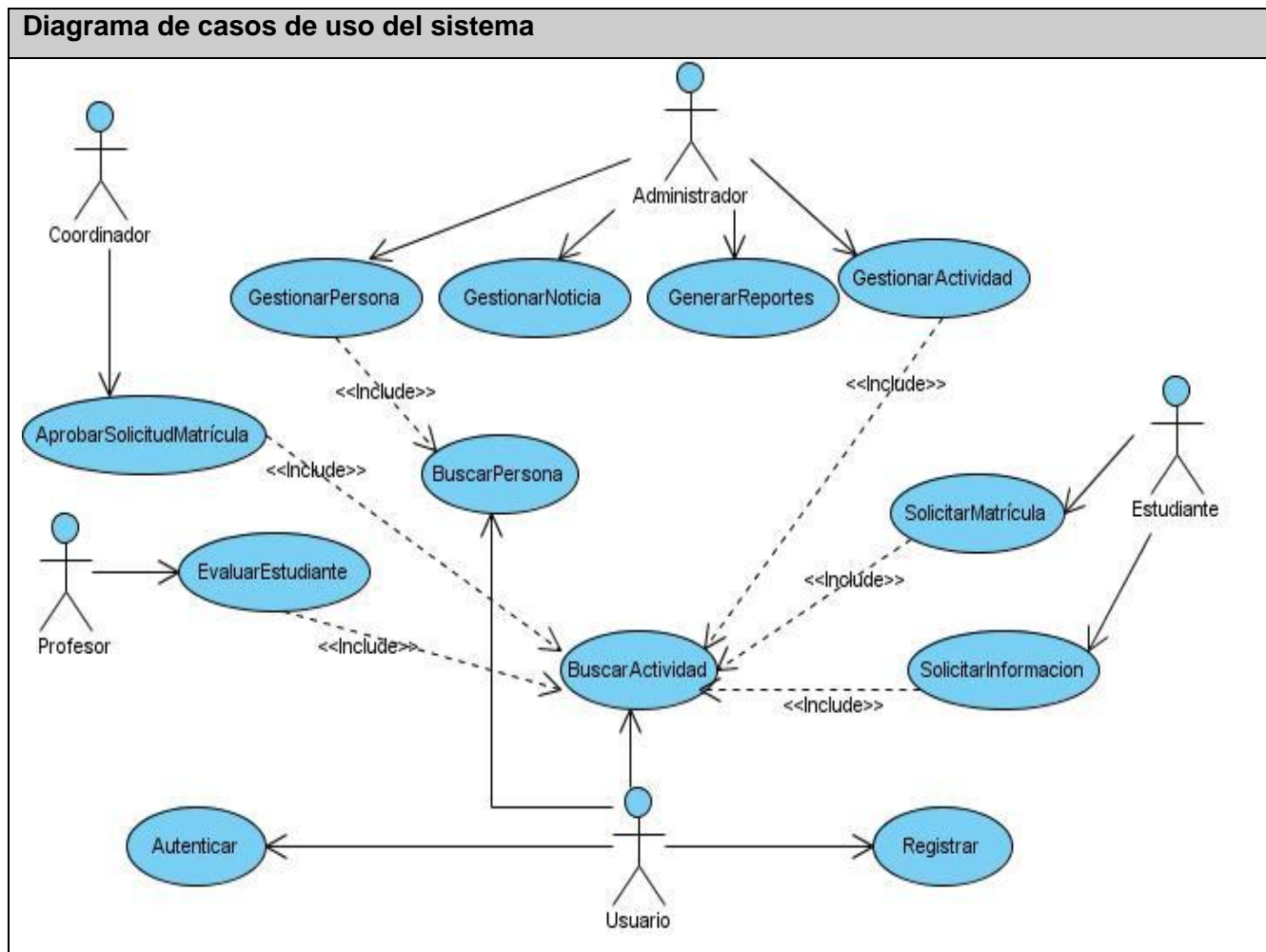


Tabla 2.10 Diagrama de casos de uso del sistema

2.11 Definición de los casos de uso.

CU-1	Gestionar persona
Actor	Administrador
Descripción	Consiste en la inserción, modificación y eliminación de personas vinculadas con las actividades de postgrado.
Referencia	RF 1, RF 2, RF 3.

Tabla 2.11 Gestionar persona

CU-2	Autenticar usuario
Actor	Usuario
Descripción	Consiste en la autenticación en el sistema y verificación de privilegios.
Referencia	RF 10.

Tabla 2.12 Autenticar usuario

CU-3	Evaluar estudiantes
Actor	Profesor
Descripción	Consiste en la inserción de evaluaciones y asignación de créditos a los estudiantes que han vencido una actividad.
Referencia	RF 11, RF 12.

Tabla 2.13 Evaluar estudiantes

CU-4	Solicitar matrícula
Actor	Estudiante
Descripción	Consiste en la solicitud de matrícula en una actividad de postgrado, por parte del estudiante.
Referencia	RF 27

Tabla 2.14 Solicitar matrícula

U-5	Aprobar solicitud de matrícula
Actor	Coordinador
Descripción	Consiste en la revisión de la solicitud de matrícula en actividad, por parte del coordinador, determinando si será aprobada o no.
Referencia	RF 28

Tabla 2.15 Aprobar solicitud de matrícula

CU-6	Generar Reportes
Actor	Administrador
Descripción	Consiste en mostrar información de los estudiantes y las actividades inscritas.
Referencia	RF 14, RF 15, RF 16, RF 17, RF 18, RF 19, RF 20.

Tabla 2.16 Generar Reportes

CU-7	Gestionar Actividad
Actor	Administrador
Descripción	Consiste en insertar, modificar o eliminar una actividad inscrita.
Referencia	RF 4, RF 5, RF 6.

Tabla 2.17 Gestionar Actividad

CU-8	Gestionar noticia
Actor	Administrador
Descripción	Consiste en insertar, modificar o eliminar una noticia.
Referencia	RF 7, RF 8, RF 9.

Tabla 2.18 Gestionar noticias

CU-9	Solicitar Información
Actor	Estudiante
Descripción	Consiste en mostrar información de interés sobre las actividades, la certificación de notas y otros datos sobre las evaluaciones.
Referencia	RF 21, RF 22, RF 23, RF 24, RF 25.

Tabla 2.19 Solicitar Información

CU-10	Buscar persona
Actor	Usuario
Descripción	Consiste en buscar información sobre una persona que esté en la base de datos.
Referencia	RF 13.

Tabla 2.20 Buscar persona

CU-11	Buscar actividad
Actor	Usuario
Descripción	Consiste en buscar información sobre una actividad.
Referencia	RF 26.

Tabla 2.21 Buscar actividad

2.12 Descripción de los casos de usos del sistema.

2.12.1 Descripción del Caso de Uso Gestionar Persona.

Nombre del CU	Gestionar Persona	
Actores	Administrador (TECADIES y Secretaria)	
Propósito	Insertar, modificar o eliminar una persona.	
Resumen	El caso de uso se inicia cuando el Administrador solicita la inserción, modificación o eliminación de personas.	
Referencias	RF 1, RF 2, RF 3.	
Precondiciones	Que el Administrador esté autenticado en el sistema. Se realizó la inscripción de una actividad o la matrícula en alguna de las ofertas.	
Prioridad	Crítico	
Sección "Insertar Persona": Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Administrador selecciona la opción: "Crear Persona UCI", si la persona está en el dominio UCI. "Crear Persona", si la persona no está en el dominio UCI.	2. El sistema muestra la interfaz para insertar los datos de la persona.	
3. El Administrador inserta los datos de la persona.	4. El sistema verifica que no hayan campos vacíos. 5. El sistema registra los datos.	
Sección "Insertar Persona": Flujo Alternativo		

1. Si el Administrador no desea realizar otra acción, termina el CU.	2. Si quedó un campo vacío, el sistema envía un mensaje.
Sección “Modificar Persona”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción “Buscar Personas Según Rol”.	2. El sistema muestra una interfaz para seleccionar el rol.
3. El Administrador selecciona el rol.	4. El sistema muestra las personas con el rol seleccionado.
5. El Administrador selecciona la opción “Modificar”.	6. El sistema muestra los datos de la persona.
7. El Administrador modifica los datos.	8. El sistema verifica que no haya ningún campo vacío. 9. El sistema actualiza los datos de la persona.
Sección “Modificar Persona”: Flujo Alternativo	
1. Si el Administrador no desea realizar otra acción, termina el CU.	2. Si quedó un campo vacío, el sistema envía un mensaje.
Sección “Eliminar Persona”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción “Buscar Personas Según Rol”.	2. El sistema muestra una interfaz para seleccionar el rol.
3. El Administrador selecciona el rol.	4. El sistema muestra las personas con el rol seleccionado.
5. El Administrador selecciona la opción “Eliminar”.	6. El sistema elimina la persona.
Sección “Eliminar Persona”: Flujo Alternativo	
1. Si el Administrador no desea realizar otra acción, termina el CU.	
Poscondiciones	Los datos insertados o modificados deben actualizarse en la base de datos.

2.12.2 Descripción del Caso de Uso Autenticar Usuario.

Nombre del CU	Autenticar Usuario	
Actores	Usuario	
Propósito	Conceder los permisos por rol.	
Resumen	El caso de uso se inicia cuando el usuario envía al sistema usuario y contraseña.	
Referencias	RF 10	
Precondiciones	Debe ser usuario de la base de datos de postgrado.	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario accede a la página	2. El sistema solicita el usuario y la contraseña.	
3. El usuario introduce el usuario y la contraseña.	4. El sistema verifica los datos. 5. De ser correctos, brinda privilegios de acceso.	
Flujos Alternativos		
Acción del Actor	Respuesta del Sistema	
	5. En caso contrario, muestra un mensaje de error.	
Poscondiciones	Se ha autenticado un usuario en el sistema.	

2.12.3 Descripción del Caso de Uso Evaluar Estudiante.

Nombre del CU	Evaluar Estudiante	
Actores	Profesor	
Propósito	Dar evaluación a los estudiantes matriculados.	
Resumen	El caso de uso se inicia cuando el profesor accede al registro de los estudiantes matriculados en la actividad que imparte y se dispone a evaluarlos, inserta las evaluaciones por estudiante y al culminar se realiza la asignación de crédito.	
Referencias	RF 11, RF 12.	

Precondiciones	El usuario autenticado debe tener los privilegios de acceso y además, existir estudiantes matriculados en la actividad que se desea evaluar.
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El profesor accede a la interfaz Evaluación y Estado.	2. El sistema solicita el identificador de la actividad.
3. El profesor introduce el identificador de la actividad.	4. El sistema muestra un listado con los estudiantes matriculados en dicha actividad.
5. El profesor introduce las evaluaciones.	6. El sistema asigna la nota dada por el profesor. 8. Asigna crédito automáticamente.
Flujos Alternativos	
Poscondiciones	Se han registrado las evaluaciones de los estudiantes, asignado crédito activo, la fecha de vencimiento del crédito y el estado de la actividad de cada estudiante.

2.12.4 Descripción del Caso de Uso Gestionar Actividad.

Nombre del CU	Gestionar Actividad
Actores	Administrador (TECADIES y Secretaria)
Propósito	Crear, modificar o eliminar una actividad.
Resumen	El caso de uso se inicia cuando el Administrador solicita la inserción, modificación o eliminación de una actividad de postgrado.
Referencias	RF 4, RF 5, RF 6.
Precondiciones	La actividad debe de haber sido aceptada.
Prioridad	Crítico
Sección "Insertar Actividad": Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción "Crear Actividad".	2. El sistema muestra la interfaz para insertar los datos de la actividad.

3. El Administrador inserta los datos de la actividad.	4. El sistema verifica que no hayan campos vacíos. 5. El sistema registra los datos.
Sección “Insertar Actividad”: Flujo Alternativo	
1. Si el Administrador no desea realizar otra acción, termina el CU.	2. Si quedó un campo vacío, el sistema envía un mensaje.
Sección “Modificar Actividad”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción “Listar Actividad”, que muestra todas las actividades.	2. El sistema muestra una interfaz con las actividades.
3. El Administrador selecciona la opción “Modificar”.	4. El sistema muestra la interfaz para modificar datos de la actividad.
5. El Administrador modifica los datos.	6. El sistema verifica que no haya campos vacíos. 7. El sistema actualiza los datos de la actividad.
Sección “Modificar Actividad”: Flujo Alternativo	
1. Si el Administrador no desea realizar otra acción, termina el CU.	
Sección “Eliminar Actividad”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción “Listar Actividad”, que muestra todas las actividades.	2. El sistema muestra una interfaz con las actividades.
3. El Administrador selecciona la opción “Eliminar”.	4. El sistema elimina la actividad.
Sección “Eliminar Actividad”: Flujo Alternativo	
1. Si el Administrador no desea realizar otra acción, termina el CU.	
Poscondiciones	Los datos insertados o modificados deben actualizarse en la base de datos.

2.12.5 Descripción del Caso de Uso Solicitar Matrícula.

Nombre del CU	Solicitar Matrícula	
Actores	Estudiante	
Propósito	Matricular en una actividad.	
Resumen	El caso de uso se inicia cuando se realiza la solicitud de matrícula en alguna actividad.	
Referencias	RF 27.	
Precondiciones	La actividad a la que se solicite matricular debe estar inscrita y en proceso de matrícula.	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El estudiante accede a "Solicitar Matrícula"	2. El sistema muestra un listado con las actividades ofertadas.	
3. El estudiante selecciona la(s) actividad(es) y da enviar.	4. El sistema guarda la(s) solicitud(es).	
Flujos Alternativos		
Poscondiciones	Se adiciona una persona a las solicitudes de matrícula.	

2.12.6 Descripción del Caso de Uso Aprobar solicitar de matrícula.

Nombre del CU	Aprobar solicitar de matrícula
Actores	Coordinador
Propósito	Aprobar o no la matrícula solicitada por los estudiantes.
Resumen	El caso de uso se inicia cuando se solicita realizar la revisión de las solicitudes de matrícula de cualquier actividad de postgrado, determinando los estudiantes que serán matriculados.
Referencias	RF 28.
Precondiciones	El usuario autenticado debe tener los privilegios de acceso y además, existir solicitudes de matrícula en la actividad.
Prioridad	Crítico

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Coordinador solicita la opción aprobar matrícula.	2. El sistema le muestra la opción de buscar la actividad.
3. El coordinador introduce el identificador de la actividad.	4. El sistema muestra el usuario y nombre de los estudiantes que han solicitado la matrícula.
5. El coordinador acepta o deniega las solicitudes.	6. El sistema actualiza la solicitud como aceptada o denegada.
Flujos Alternativos	
Poscondiciones	Se determinan los estudiantes que serán matriculados.

2.12.7 Descripción del Caso de Uso Gestionar Noticia.

Nombre del CU	Gestionar Noticia
Actores	Administrador (TECADIES y Secretaria)
Propósito	Insertar, modificar o eliminar una noticia.
Resumen	El caso de uso se inicia cuando el Administrador solicita la inserción, modificación o eliminación de noticia.
Referencias	RF 7, RF 8, RF 9.
Precondiciones	Que el Administrador esté autenticado en el sistema.
Prioridad	Crítico
Sección “Insertar Noticia”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción: “Crear Noticia.	2. El sistema muestra la interfaz para insertar la noticia.
3. El Administrador inserta la noticia.	4. El sistema guarda la noticia y la publica.
Sección “Insertar Noticia”: Flujo Alternativo	
1. Si el Administrador no desea realizar otra acción, termina el CU.	
Sección “Modificar Noticia”: Flujo Normal de Eventos	

Acciones del Actor		Respuesta del Sistema
1. El Administrador selecciona la opción "Listar Noticias".		2. El sistema muestra una interfaz con todas las noticias.
3. El Administrador selecciona modificar en la noticia que desea.		4. El sistema muestra los datos de la noticia.
5. El Administrador modifica los datos.		6. El sistema verifica que no haya ningún campo vacío. 7. El sistema actualiza la noticia.
Sección "Modificar Noticia": Flujo Alternativo		
1. Si el Administrador no desea realizar otra acción, termina el CU.		
Sección "Eliminar Noticia": Flujo Normal de Eventos		
Acciones del Actor		Respuesta del Sistema
1. El Administrador selecciona la opción "Listar Noticias".		2. El sistema muestra una interfaz con todas las noticias.
3. El Administrador selecciona eliminar en la noticia que desea		4. El sistema elimina la noticia.
Sección "Eliminar Noticia": Flujo Alternativo		
1. Si el Administrador no desea realizar otra acción, termina el CU.		
Poscondiciones	Las noticias insertadas o modificadas deben actualizarse en la base de datos.	

2.12.8 Descripción del Caso de Uso Generar Reportes.

Nombre del CU	Generar Reportes
Actores	Administrador
Propósito	Permite generar reportes de los estudiantes y las actividades inscritas.
Resumen	El caso de uso comienza cuando el Administrador selecciona la opción Solicitar Reporte, el sistema muestra los diferentes tipos de reportes: Reporte de estudiantes que no han recogido su certificado, reporte con la

	cantidad de estudiantes que han abandonado una actividad de postgrado y cuántos lo han finalizado, reporte de estudiantes de alto rendimiento, reporte de actividades por estado de ejecución, reporte de actividades impartidas por un profesor determinado y reporte de estudiantes por actividad.
Referencias	RF 14, RF 15, RF 16, RF 17, RF 18, RF 19, RF 20.
Precondiciones	La información solicitada debe estar en la base de datos.
Prioridad	Critico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción "Reportes".	2. El sistema muestra las opciones correspondientes: a. Recogida de Certificado. b. Retención de Matrícula. c. Listado de Adiestrados. d. Actividades por su Estado. e. Actividades de un profesor. f. Matrícula de una Actividad. g. Actividades por Área.
Sección " Recogida de Certificado.": Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción Recogida de Certificado.	2. El sistema muestra todos los estudiantes de las actividades culminadas que no han recogido el certificado.
Sección " Recogida de Certificado.": Flujos Alternativos	
Sección " Retención de Matrícula ": Flujo Normal de Eventos	
Acción del Actor	Respuesta de Sistema
1. El Administrador selecciona la opción "Retención de Matrícula".	2. El sistema solicita el identificador de la actividad.
3. El Administrador introduce el	4. El sistema muestra el reporte y la opción de exportar a

identificador.	un Word.
5. El Administrador selecciona la opción exportar a un Word.	6. El sistema exporta la solicitud correspondiente.
Sección “ Retención de Matrícula ”: Flujos Alternativos	
Sección “ Listado de Adiestrados ”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción “Listado de Adiestrados”.	2. El sistema muestra una interfaz y solicita el área.
3. El Administrador selecciona el área correspondiente.	4. El sistema muestra los estudiantes adiestrados por áreas y los cursos acreditados y la opción de exportar a un Word.
5. El Administrador selecciona la opción exportar a un Word.	6. El sistema exporta la solicitud correspondiente.
Sección “ Listado de Adiestrados ”: Flujos Alternativos	
Sección “ Actividades por su Estado ”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción “Actividades por su Estado”.	2. El sistema solicita el estado.
3. El Administrador selecciona el estado correspondiente.	4. El sistema muestra las actividades que están en ese estado y la opción de exportar a un Word.
5. El Administrador selecciona la opción exportar a un Word.	6. El sistema exporta la solicitud correspondiente.
Sección “ Actividades por su Estado ”: Flujos Alternativos	
Sección “ Actividades de un profesor ”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

1. El Administrador selecciona la opción “Actividades de un profesor”.	2. El sistema solicita el usuario del profesor.
3. El Administrador introduce el usuario del profesor.	4. El sistema muestra las actividades que ha impartido el profesor y la opción de exportar a un Word.
5. El Administrador selecciona la opción exportar a un Word.	6. El sistema exporta la solicitud correspondiente.
Sección “ Actividades de un profesor ”: Flujos Alternativos	
Sección “ Matrícula de una Actividad ”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción “Matrícula de una Actividad “.	2. El sistema solicita el identificador de la actividad.
3. El Administrador introduce el identificador correspondiente.	4. El sistema muestra un listado con los estudiantes matriculados en la actividad y la opción de exportar a un Word.
5. El Administrador selecciona la opción exportar a un Word.	6. El sistema exporta la solicitud correspondiente.
Sección “ Matrícula de una Actividad ”: Flujos Alternativos	
Sección “ Actividades por Área ”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción “Actividades por Área “.	2. El sistema solicita el área.
3. El Administrador introduce el área correspondiente.	4. El sistema muestra un listado con las actividades del área y la opción de exportar a un Word.
5. El Administrador selecciona la opción exportar a un Word.	6. El sistema exporta la solicitud correspondiente.
Sección “ Actividades por Área ”: Flujos Alternativos	

2.12.9 Descripción del Caso de Uso Solicitar Información.

Nombre del CU	Solicitar Información	
Actores	Estudiante	
Propósito	Consiste en mostrar información de interés sobre las actividades, la certificación de notas y otros datos sobre las evaluaciones.	
Resumen	El caso de uso comienza cuando el Estudiante selecciona la opción Solicitar Información, el sistema muestra los diferentes tipos de información: Información de actividades, información sobre actividades vencidas, y actividades matriculadas.	
Referencias	RF 21, RF 22, RF 23, RF 24, RF 25.	
Precondiciones	La información solicitada debe estar en la base de datos.	
Prioridad	Critico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	1. El sistema muestra las opciones correspondientes: a. Información de actividades. b. Actividades vencidas c. Estado de la Matrícula.	
Sección “Información de actividades”: Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	1. El sistema muestra los Cursos, Entrenamientos, Diplomados, Seminarios y Talleres, Maestrías y Doctorados.	
3. El Estudiante selecciona la actividad.	4. El sistema muestra información sobre la actividad.	
Sección “Información de actividades”: Flujos Alternativos		
Sección “Actividades vencidas”: Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Estudiante selecciona la opción	2. El sistema muestra las actividades vencidas, con la	

Actividades vencidas.	nota y el crédito por actividad.
Sección “Actividades vencidas”: Flujos Alternativos	
Sección “Estado de la Matrícula”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Estudiante selecciona la opción Estado de la Matrícula.	2. El sistema muestra las actividades en las que está matriculado
Sección “Estado de la Matrícula”: Flujos Alternativos	
Poscondiciones	Curso normal de los eventos.

2.12.10 Descripción del Caso de Uso Buscar Persona.

Nombre del CU	Buscar Persona
Actores	Usuario
Propósito	Buscar información de usuarios.
Resumen	El caso de uso inicia cuando el usuario solicita la búsqueda de una persona en la base de datos
Referencias	RF 13.
Precondiciones	Debe estar la persona en la base de datos
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Usuario solicita la opción “Buscar Personas Según Rol” o “Buscar Persona”	2 Si seleccionó: “Buscar Personas Según Rol”, el sistema solicita el rol. Si seleccionó: “Buscar Persona”, el sistema solicita el usuario.
3. El usuario introduce los datos correspondientes.	4. Si los datos están correctos, el sistema muestra información sobre la persona solicitada.
Flujo Alternativo	
	5. Si los datos están incorrectos, el sistema envía

	mensaje de error.
Poscondiciones	Curso normal de los eventos.

2.12.11 Descripción del Caso de Uso Buscar Actividad.

Nombre del CU	Buscar Actividad	
Actores	Usuario	
Propósito	Buscar información de actividades.	
Resumen	El caso de uso inicia cuando el usuario solicita la búsqueda de una actividad en la base de datos	
Referencias	RF 26.	
Precondiciones	Debe estar la actividad en la base de datos	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Usuario solicita la opción Buscar Actividad	2. El sistema solicita el identificador de la actividad.	
3. El usuario introduce el identificador.	4. Si el identificador está correcto, el sistema muestra información sobre la actividad solicitada.	
Flujo Alternativo		
	5. Si el identificador está incorrecto, el sistema envía mensaje de error.	
Poscondiciones	Curso normal de los eventos.	

Conclusiones

En este capítulo se identificaron los actores del negocio, trabajadores y actores del sistema y su relación en los procesos a través de los diagramas de actividades. Se establecieron las principales funcionalidades que debe cumplir el sistema. Se expusieron los casos de uso del negocio y los casos de uso del sistema, y se describieron las acciones de los actores y las respuestas del sistema.

Capítulo 3

Análisis y Diseño

Introducción

En el presente capítulo se realiza el Análisis y Diseño, flujo de trabajo que propone la metodología RUP, donde se modelan los principales casos de uso a través de los artefactos más importantes: el Diagrama de Análisis, Diagrama de Clases del Diseño, Diagrama de interacción y Modelo de Clases Persistentes. Estos procesos son importantes en el desarrollo de software pues describen como el sistema será realizado a partir de las funcionalidades previstas y las restricciones impuestas.

3.1 Análisis.

Durante el análisis se refinan y estructuran los requerimientos que se describieron en la captura de requisitos, posibilitando una comprensión más precisa y una descripción que sea fácil de mantener y que permita estructurar el sistema completo, incluyendo la arquitectura.

3.1.1 Modelo de análisis.

El modelo de análisis ofrece una especificación más precisa de los requisitos, puede ser considerado como una primera aproximación al modelo de diseño que se describe utilizando el lenguaje de los desarrolladores y estructura los requisitos de modo que facilita su comprensión, preparación, modificación y mantenimiento.

3.1.2 Diagramas de clases del análisis.

El diagrama de clase del análisis es un artefacto donde se adjuntan los diagramas de clases a las realizaciones de casos de uso, mostrando sus clases participantes y sus relaciones.

3.1.2.1 Diagrama de clases de análisis de Autenticar.



Tabla 3.1 Diagrama de clases de análisis de Autenticar.

3.1.2.2 Diagrama de clases de análisis de Gestionar Persona.

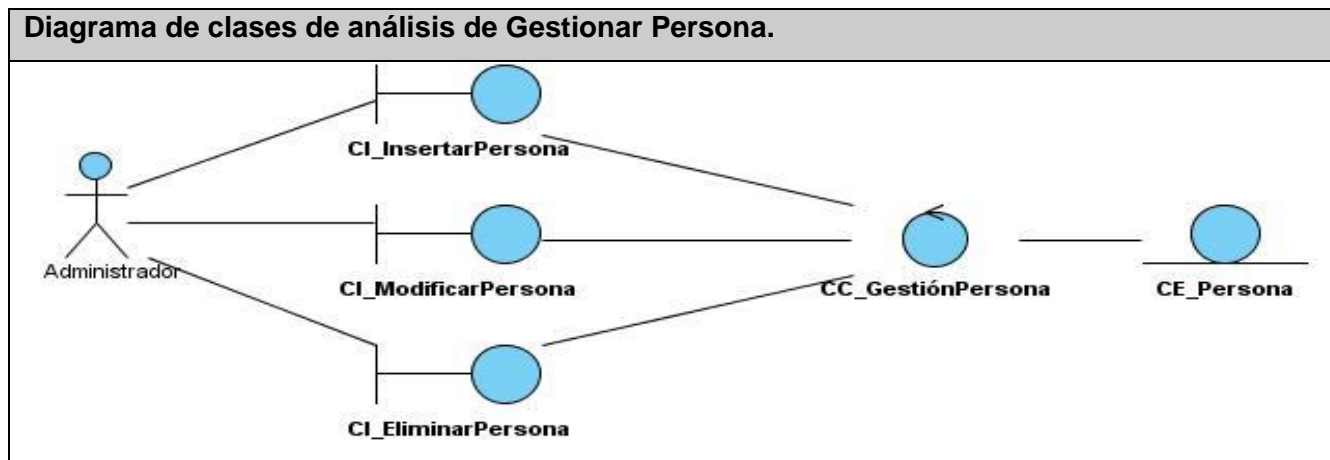


Tabla 3.2 Diagrama de clases de análisis de Gestionar Persona.

3.1.2.3 Diagrama de clases de análisis de Gestionar Actividad.

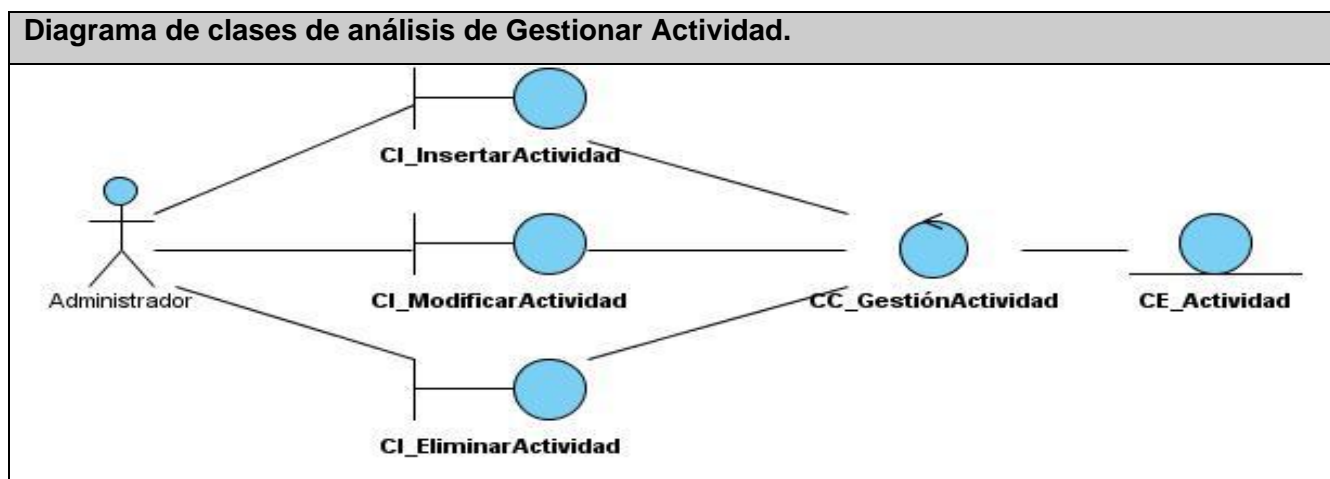


Tabla 3.3 Diagrama de clases de análisis de Gestionar Actividad.

3.1.2.4 Diagrama de clases de análisis de Buscar Actividad.

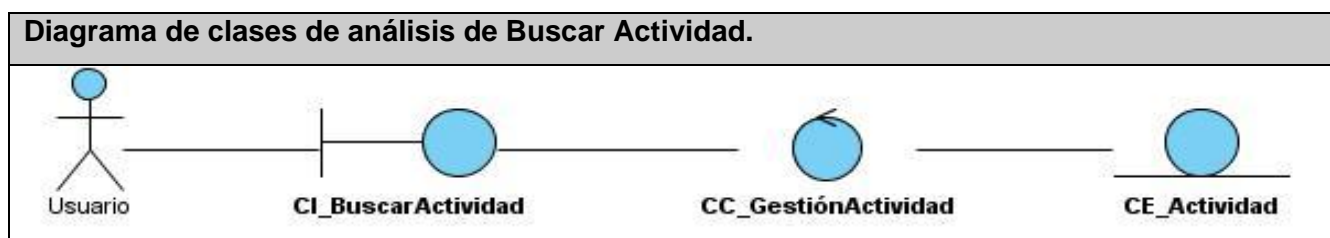


Tabla 3.4 Diagrama de clases de análisis de Buscar Actividad.

3.1.2.5 Diagrama de clases de análisis de Aprobar Matrícula.

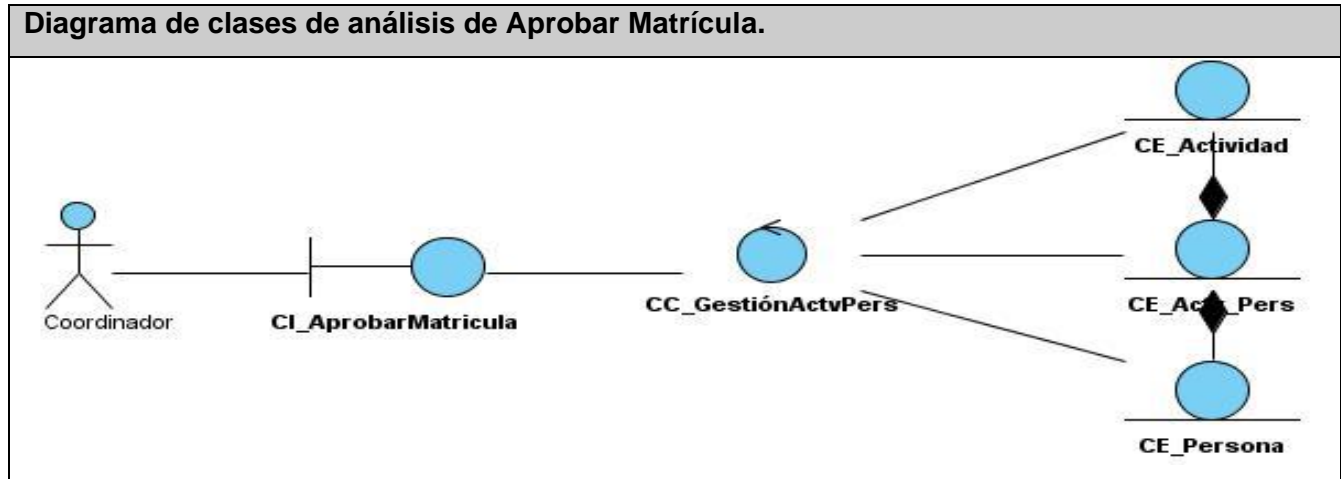


Tabla 3.5 Diagrama de clases de análisis de Aprobar Matrícula.

3.1.2.6 Diagrama de clases de análisis de Evaluar.

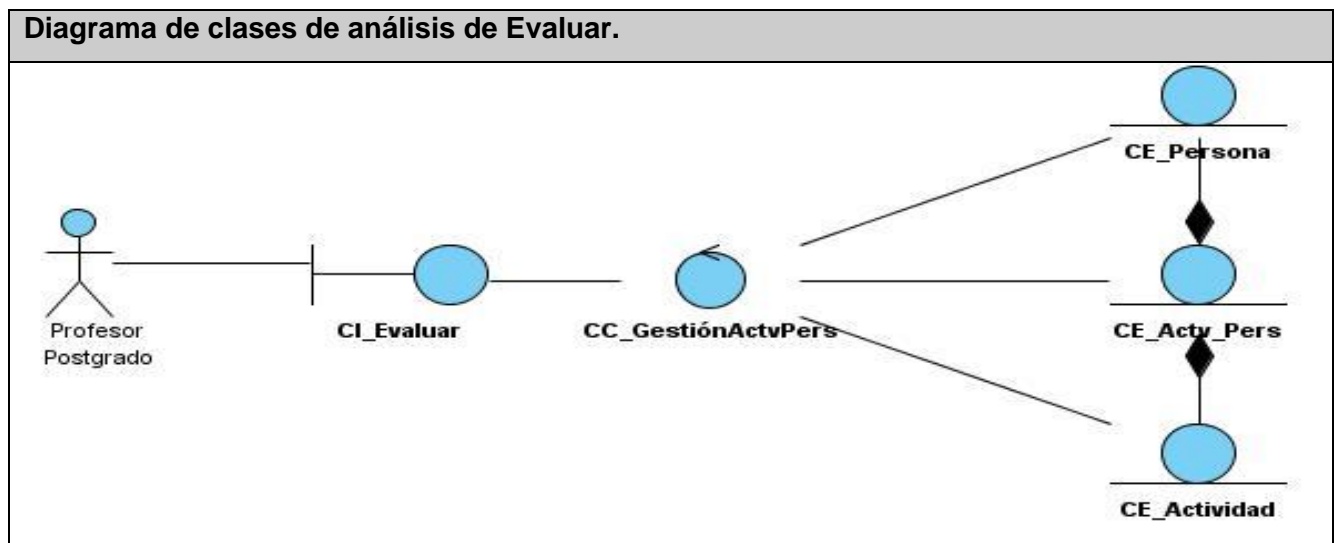


Tabla 3.6 Diagrama de clases de análisis de Evaluar.

3.1.2.7 Diagrama de clases de análisis de Gestionar Noticia.

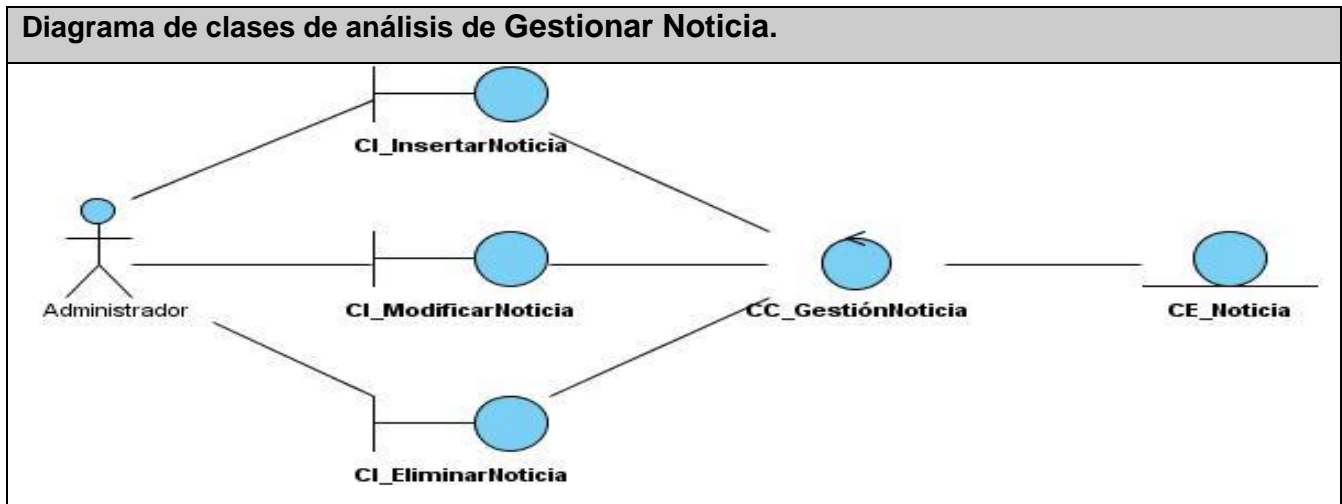


Tabla 3.7 Diagrama de clases de análisis de Gestionar Noticia.

3.1.2.8 Diagrama de clases de análisis de Buscar Persona.

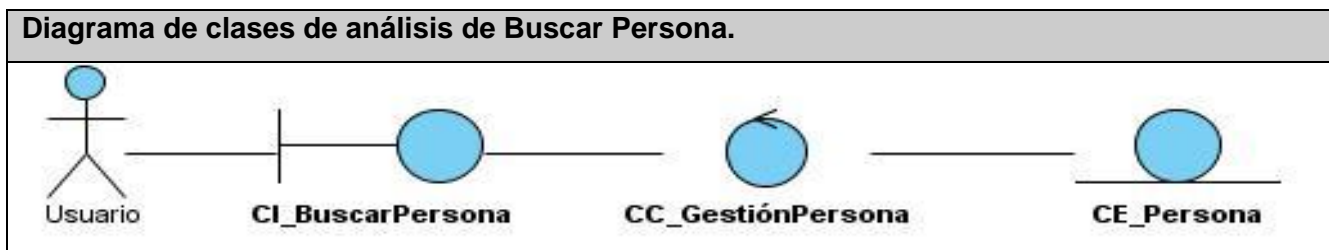


Tabla 3.8 Diagrama de clases de análisis de Buscar Persona.

3.1.2.9 Diagrama de clases de análisis de Generar Reportes.

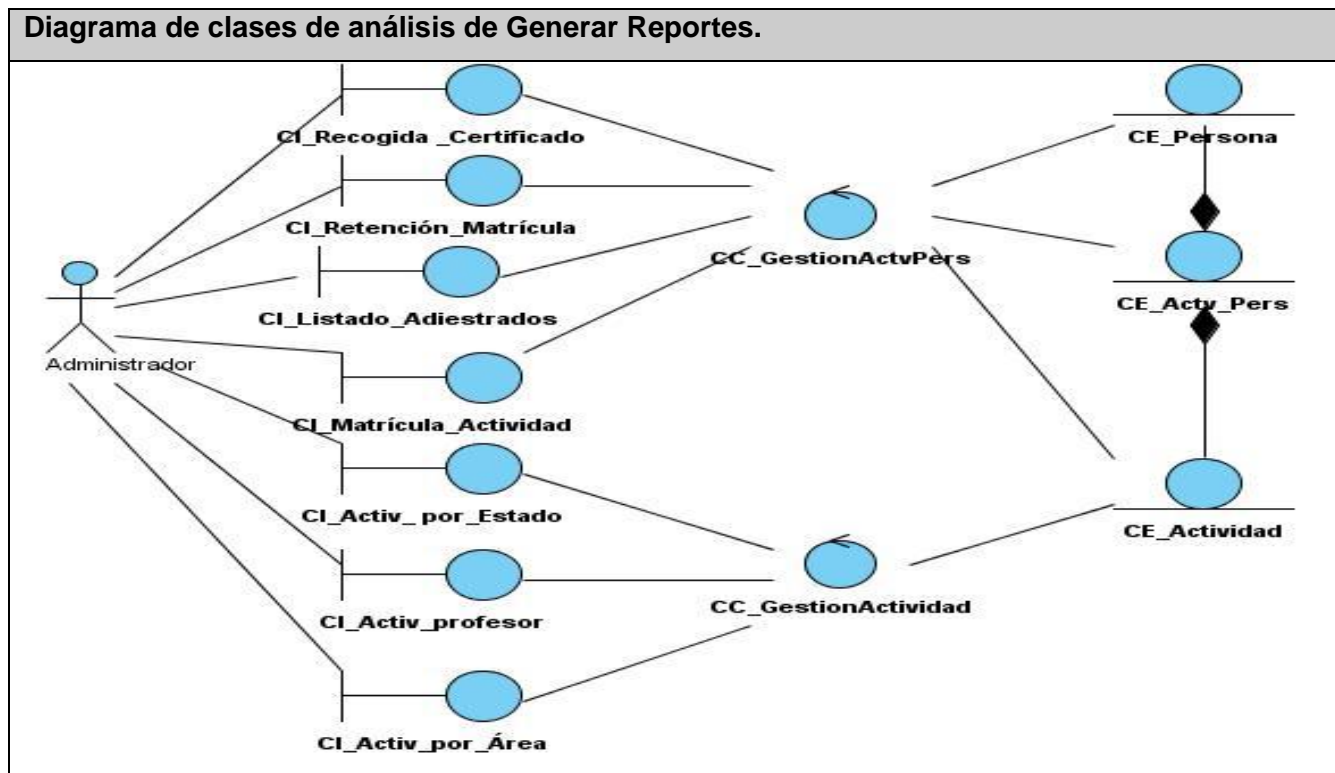


Tabla 3.9 Diagrama de clases de análisis de Generar Reportes.

3.1.2.10 Diagrama de clases de análisis de Solicitar Información.

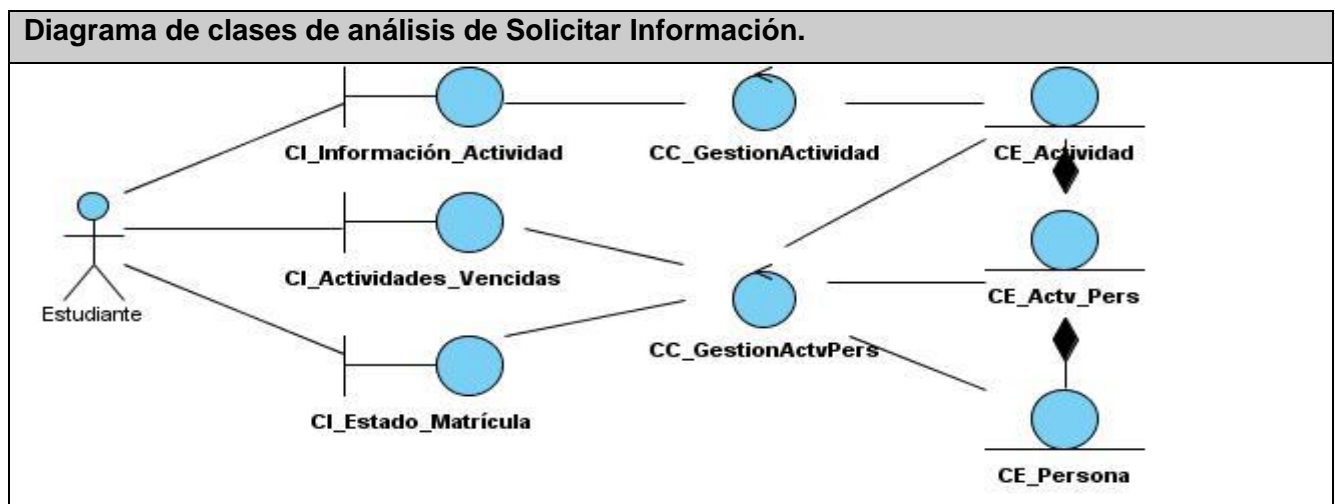


Tabla 3.10 Diagrama de clases de análisis de Solicitar Información.

3.1.2.11 Diagrama de clases de análisis de Solicitar Matrícula.

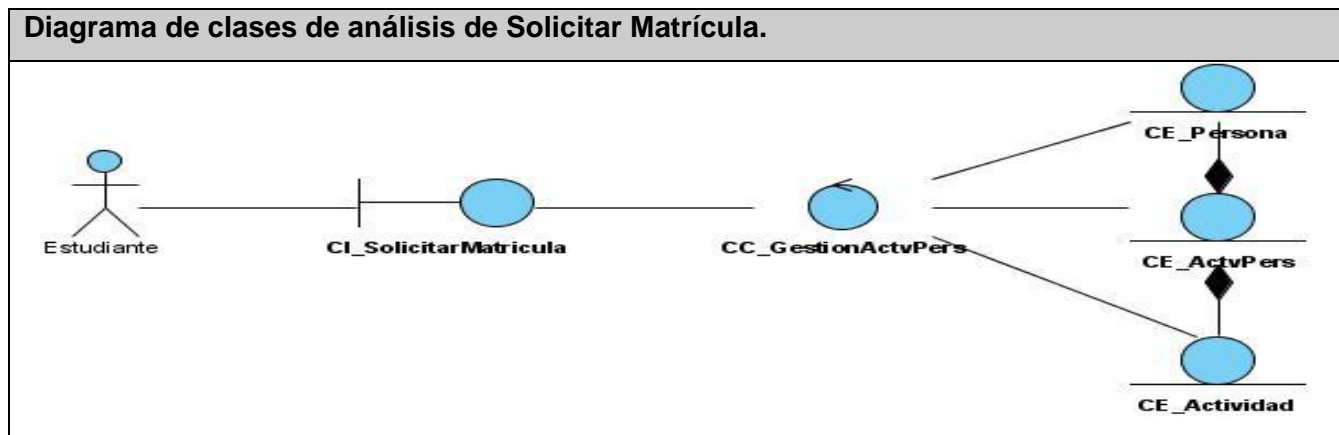


Tabla 3.11 Diagrama de clases de análisis de Solicitar Matrícula.

3.2 Diseño.

En el diseño se modela el sistema y se adquiere una comprensión en profundidad de los aspectos relacionados con todos los requisitos, tanto los funcionales como los no funcionales y otras restricciones, y además crea una entrada apropiada y un punto de partida para actividades de implementación.

3.2.1 Modelo de diseño.

El modelo de diseño describe la realización física de los casos de uso centrándose en los requisitos funcionales, no funcionales y otras restricciones que tienen impacto en el sistema. Además sirve de abstracción de la implementación y como entrada fundamental de sus actividades.

3.2.2 Diagrama de clases del diseño.

En el diagrama de clases del diseño se describen gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación.

Contiene la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos
- Información sobre los tipos de los atributos
- Navegabilidad
- Dependencias

3.2.2.1 Diagrama de clases de diseño de Gestionar Persona.

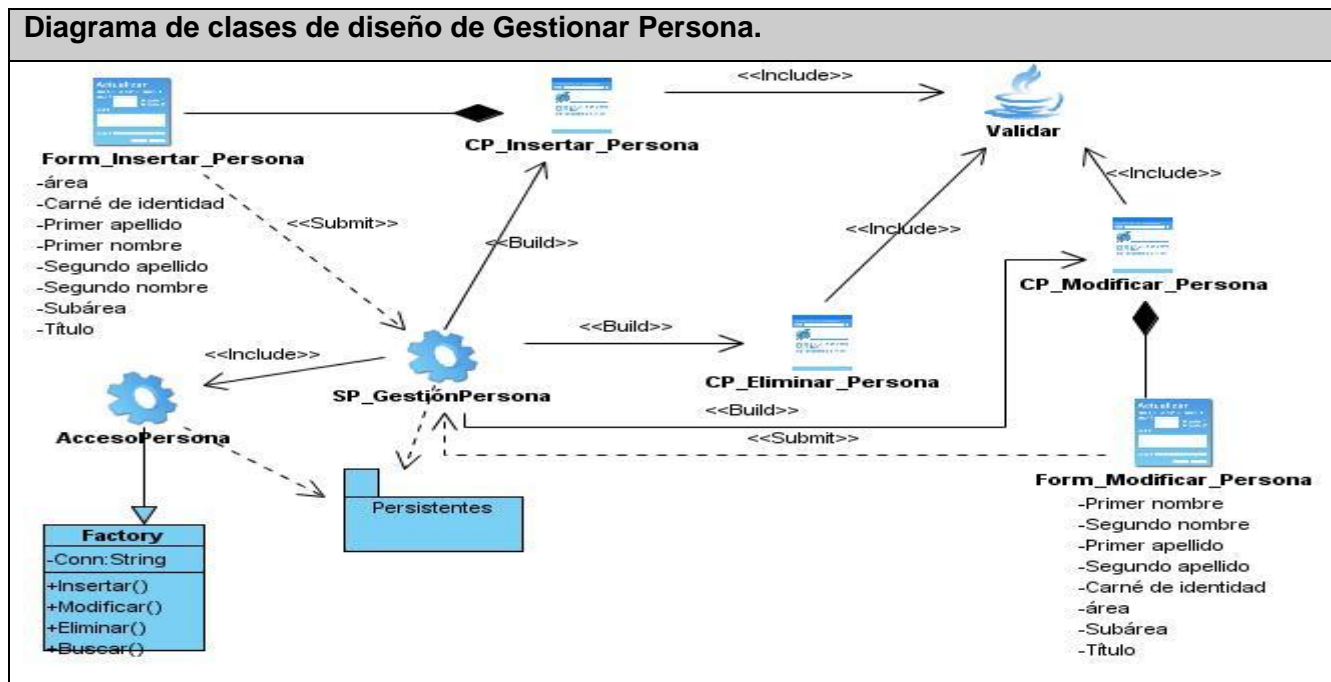


Tabla 3.12 Diagrama de clases de diseño de Gestionar Persona.

3.2.2.2 Diagrama de clases de diseño de Gestionar Actividad.

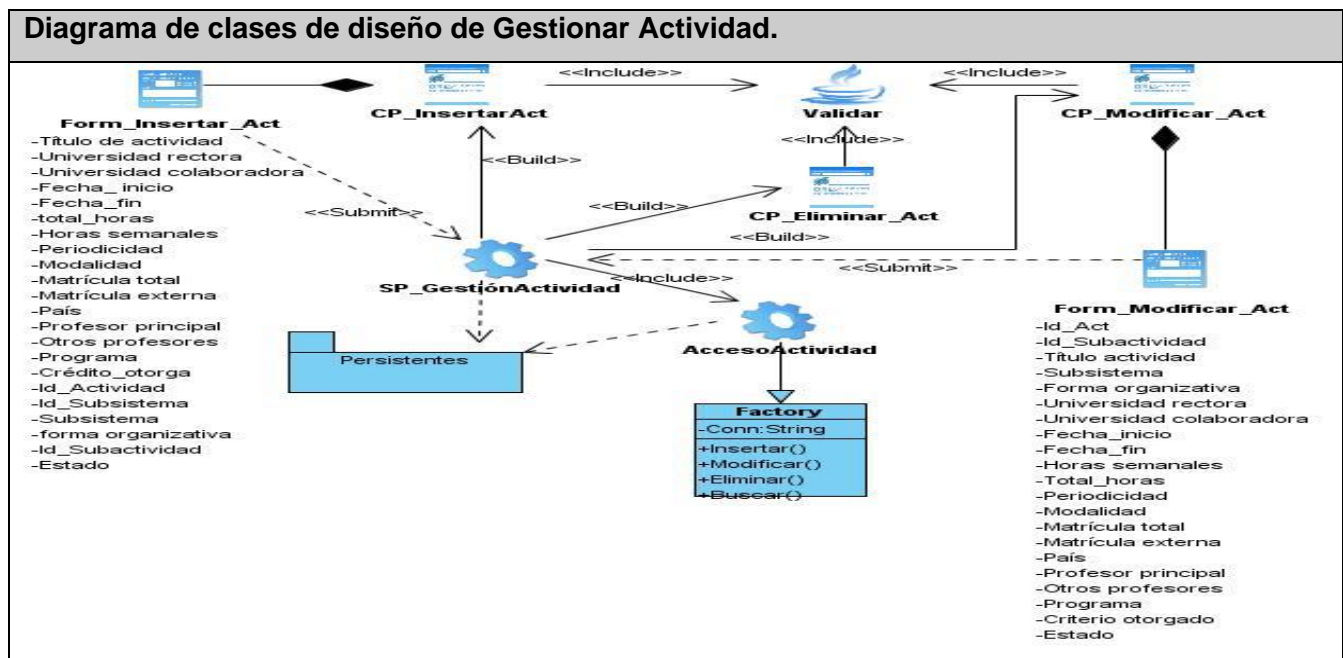


Tabla 3.13 Diagrama de clases de diseño de Gestionar Actividad.

3.2.2.3 Diagrama de clases de diseño de Autenticar.

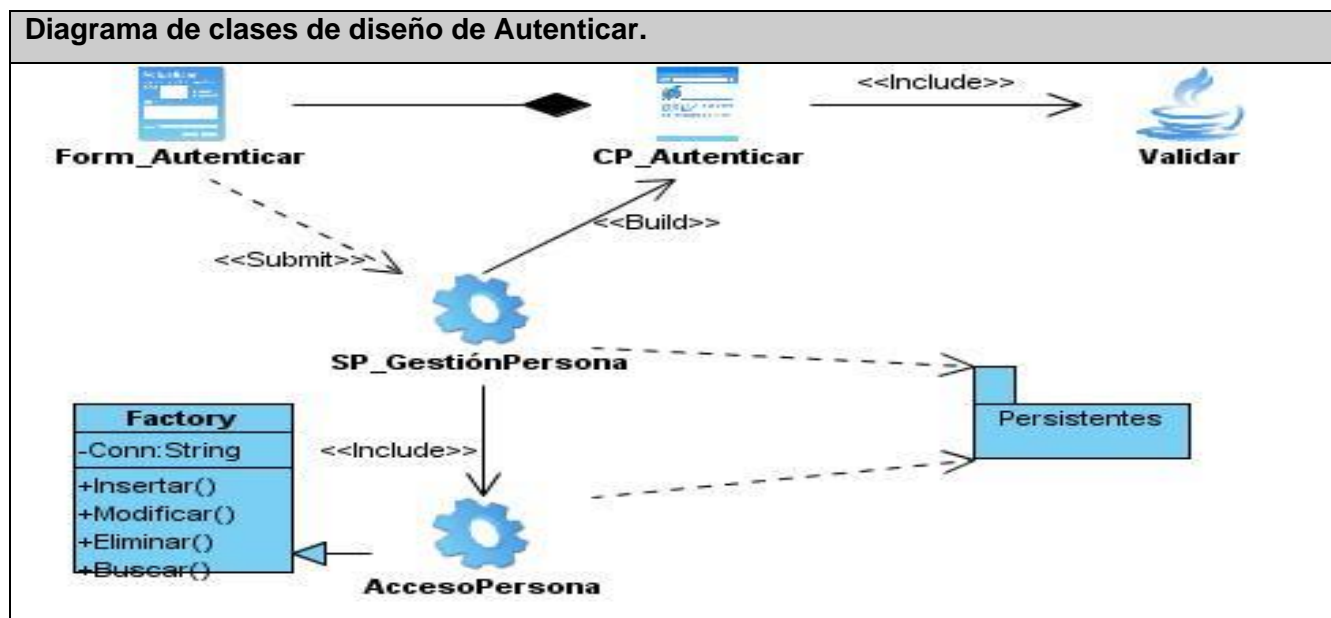


Tabla 3.15 Diagrama de clases de diseño de Autenticar.

3.2.2.4 Diagrama de clases de diseño de Buscar Actividad.

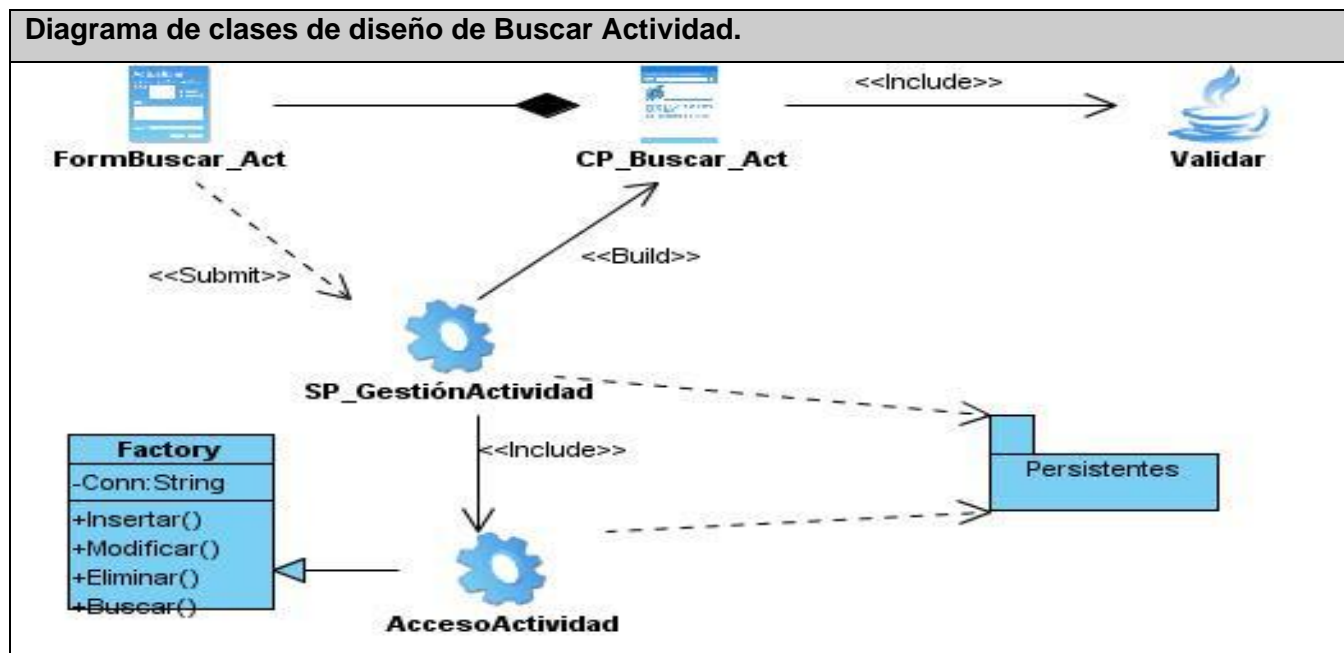


Tabla 3.16 Diagrama de clases de diseño de Buscar Actividad.

3.2.2.5 Diagrama de clases de diseño de Buscar Persona.

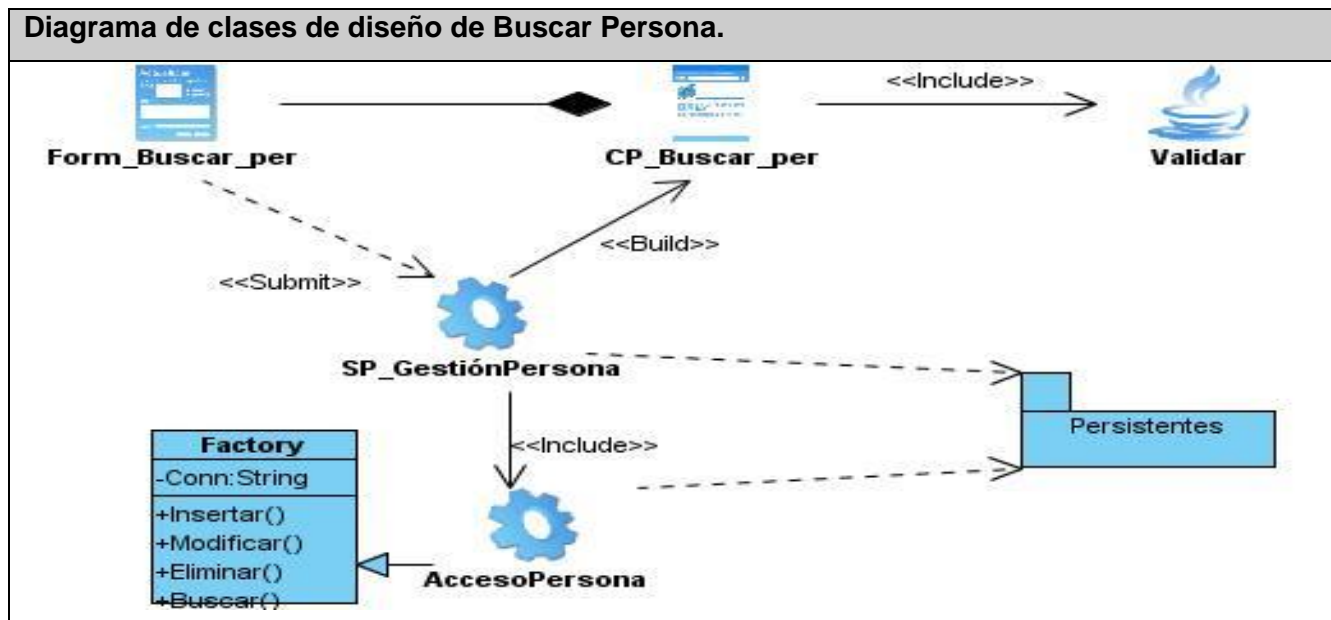


Tabla 3.17 Diagrama de clases de diseño de Buscar Persona.

3.2.2.6 Diagrama de clases de diseño de Evaluar.

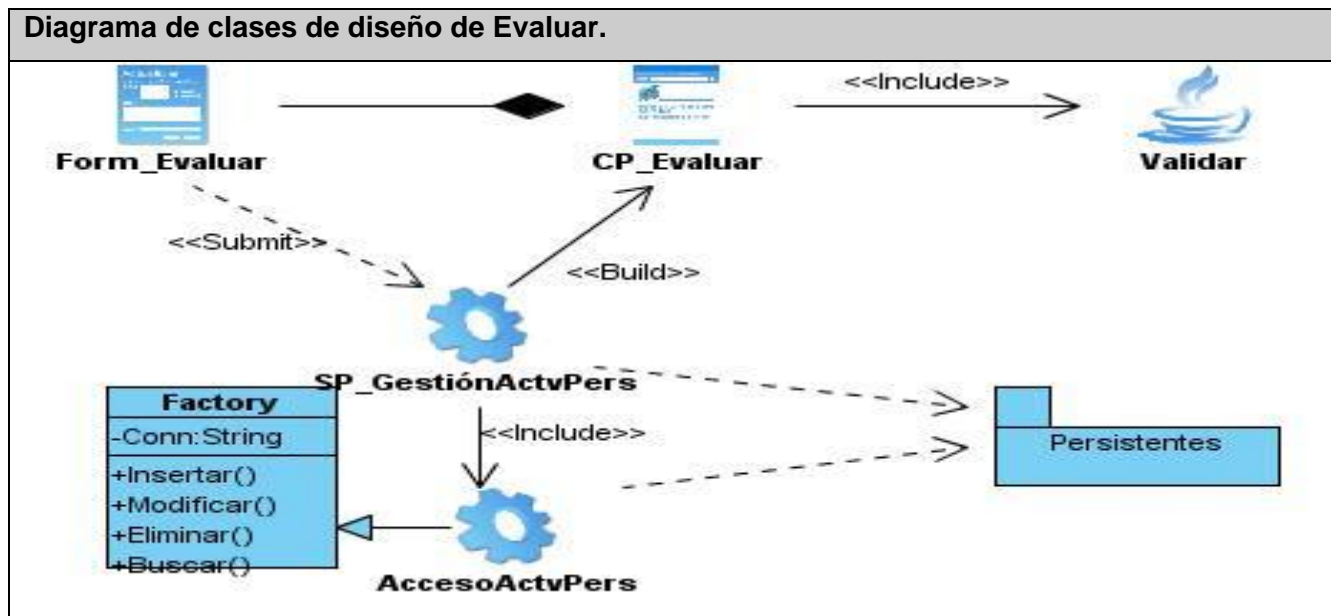


Tabla 3.18 Diagrama de clases de diseño de Evaluar.



3.2.2.7 Diagrama de clases de diseño de Generar Reportes.

Diagrama de clases de diseño de Generar Reportes.

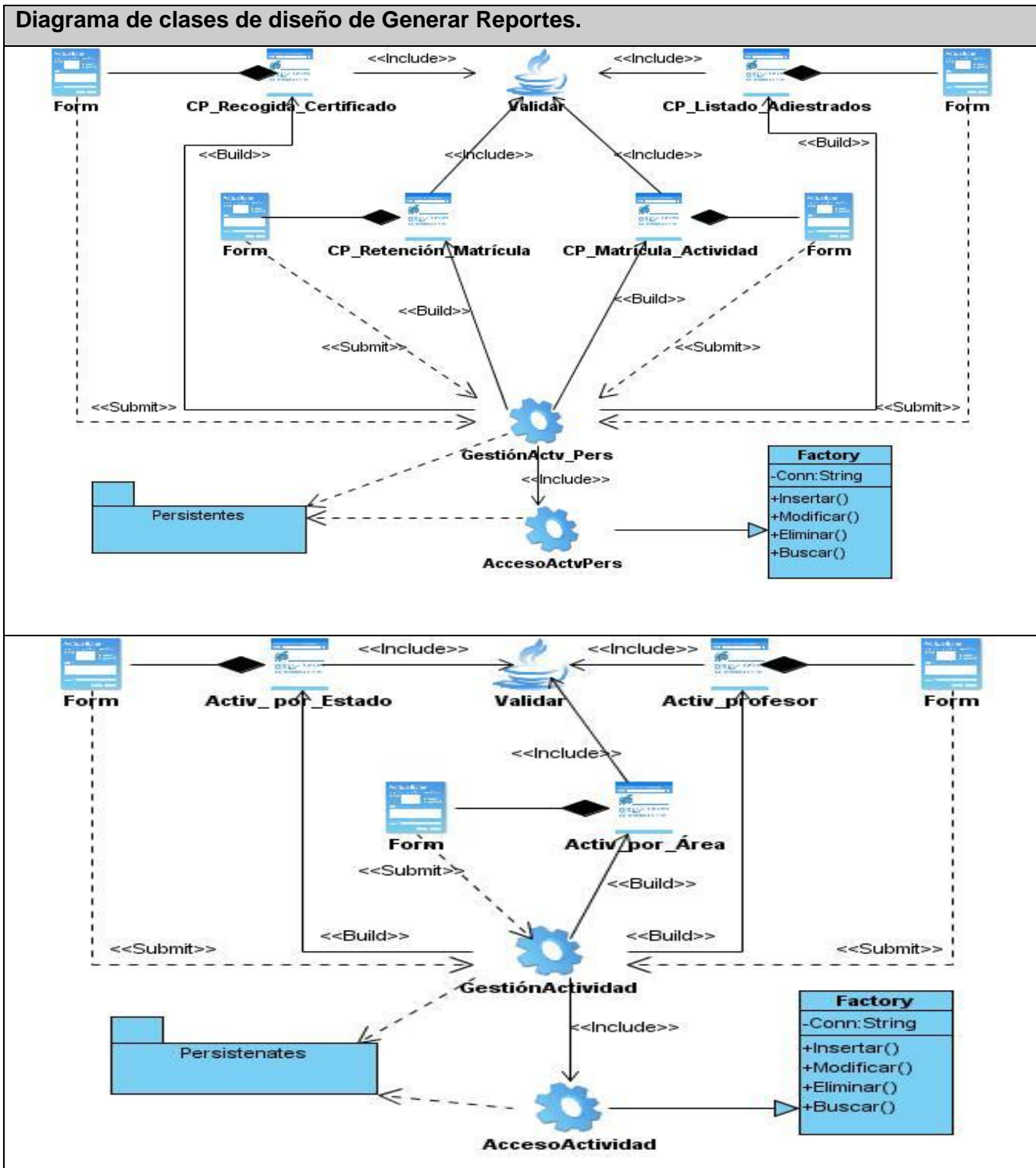


Tabla 3.19 Diagrama de clases de diseño de Generar Reportes.

3.2.2.8 Diagrama de clases de diseño de Solicitar Información.

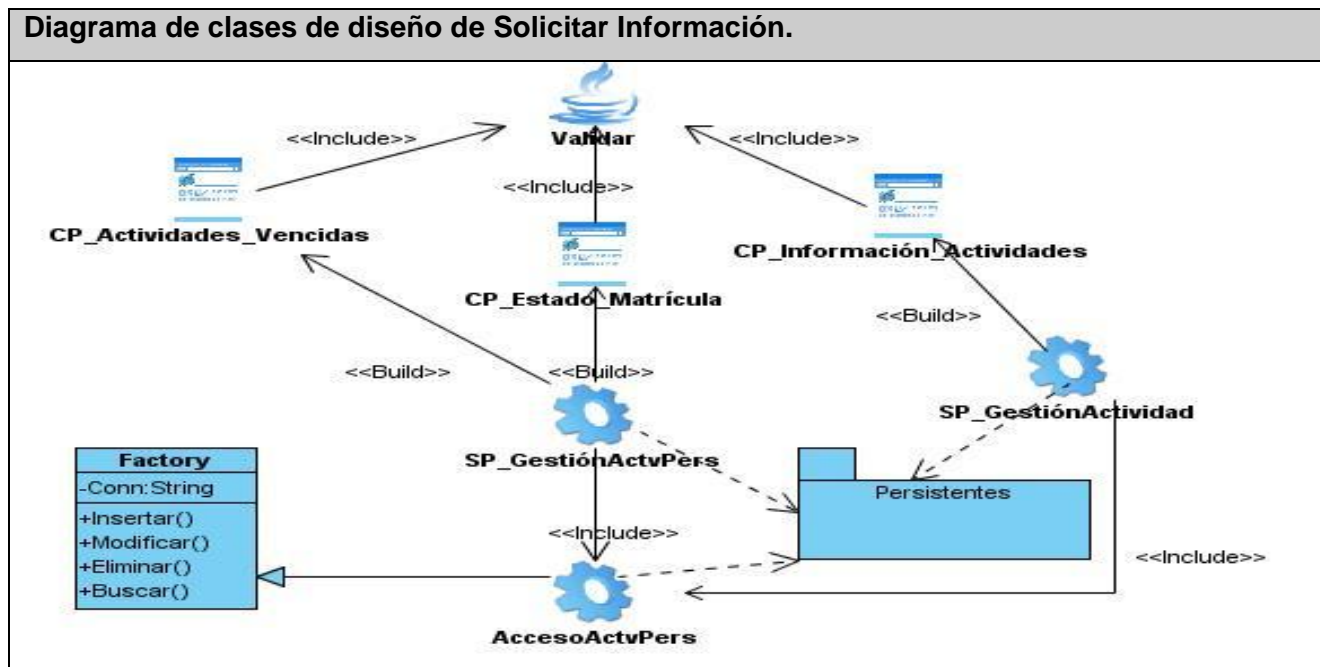


Tabla 3.20 Diagrama de clases de diseño de Solicitar Información.

3.2.2.9 Diagrama de clases de diseño de Gestionar Noticias.

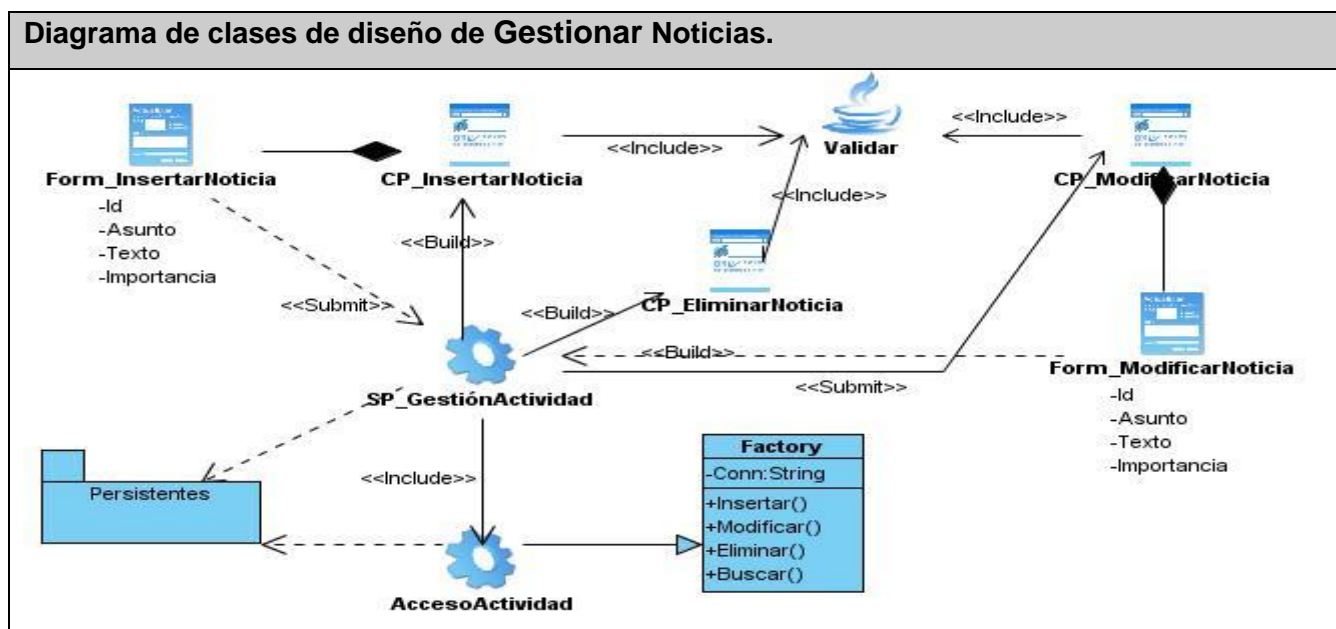


Tabla 3.21 Diagrama de clases de diseño de Gestionar Noticias.

3.2.2.10 Diagrama de clases de diseño de Solicitar Matrícula.

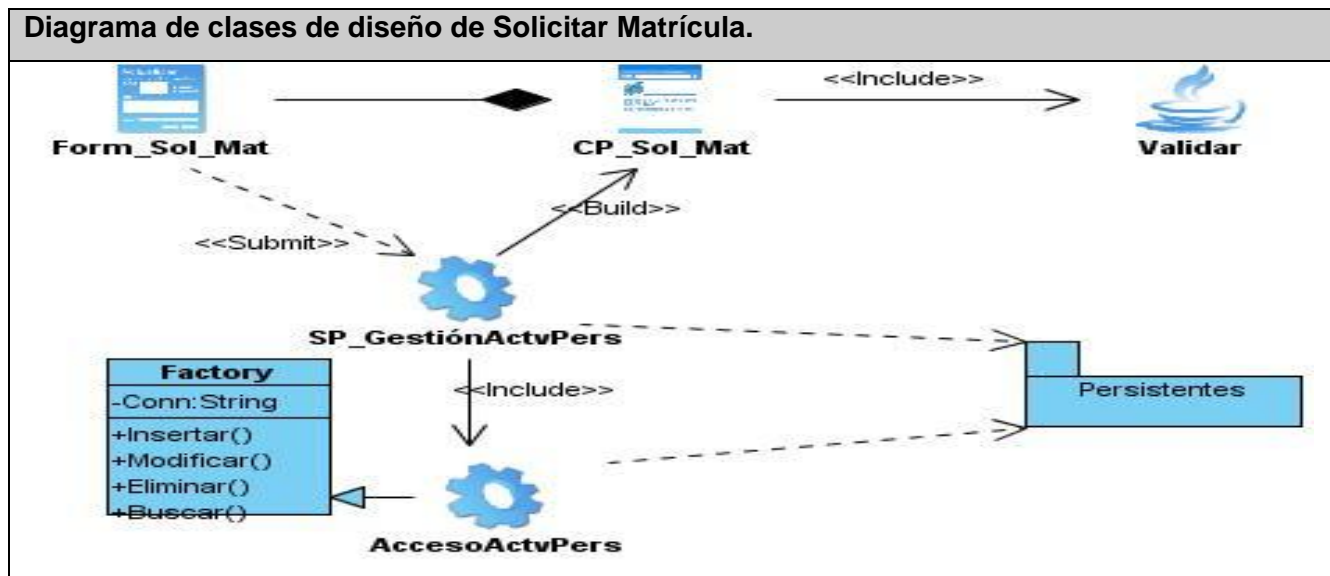


Tabla 3.22 Diagrama de clases de diseño de Solicitar Matrícula.

3.2.2.11 Diagrama de clases de diseño de Aprobar Matrícula.

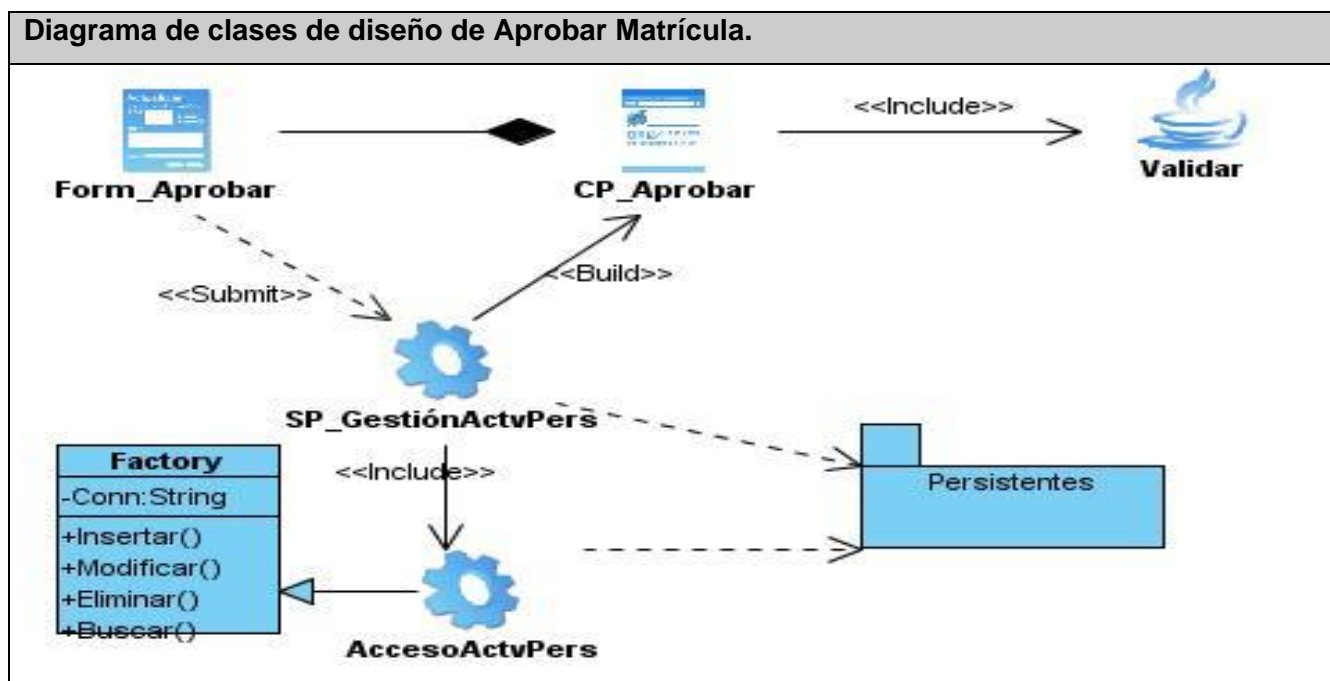


Tabla 3.23 Diagrama de clases de diseño de Aprobar Matrícula.

3.2.3 Descripción de las clases del diseño.

En los diagramas de clases del diseño se utilizó un paquete donde se relacionan las clases persistentes, a continuación se expondrán los atributos y métodos de cada una por separado.

3.2.3.1 Descripción de la clase Factory.

Nombre: Factory	
Tipo de clase: Controladora	
Atributo	Tipo
\$Conn	String
Responsabilidad:	
Nombre:	Insertar(\$pObj) Modificar(\$pObj) Eliminar(\$pObj) Buscar(\$pObj)
Descripción	Es una clase controladora abstracta para heredar con otras su conexión y redefinir sus métodos abstractos.

3.2.3.2 Descripción de las clases Controladoras.

Nombre: CGestiónActividad	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	Adicionar(\$id_subactiv,\$titulo_activ,\$sub_sist,\$fecha_inicio,\$fecha_de_fin,\$forma_org,\$univ_rectora,\$univ_colab,\$horas_semanales,\$total_de_horas,\$periodicidad,\$modalidad,\$matrícula_total,\$matrícula_ext,\$país,\$profesor_princ,\$otros_prof,\$program,\$crédito,\$estado,\$ofectar_matrícula) ModificarActividad(\$id_subactiv, ..., \$ofectar_matrícula) BuscarActividad(\$id_actv) BuscarActividadFO(\$forma_organizativa) BuscarActividadProfesor(\$profesor_principal) BuscarSubActividad(\$id_actv) BuscarActividadEstado(\$estado)

	ListarActividad() ListarActividadFO() ListarActividadM() EliminarActividad(\$id_actv)
Descripción	Esta clase permite gestionar toda información referente a las Actividades.

Nombre: CGestiónPersona	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	Adicionar(\$usuario,\$carné_id,\$primer_nomb,\$segundo_nomb,\$primer_apellido, \$segundo_apellido, \$título, \$categoría_docente, \$área, \$sub_área, \$rol) ModificarPersona(\$usuario,\$carné_id,\$primer_nomb,\$segundo_nomb,\$primer_apellid o, \$segundo_apellido, \$título, \$categoría_docente, \$área, \$sub_área, \$rol) BuscarPersonaRol(\$rol) BuscarPersona(\$usuario) EliminarPersona(\$usuario)
Descripción	Esta clase permite gestionar toda información referente a las Personas.

Nombre: CGestiónActvPers	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	AdicionarSolicitud(\$usuario, \$id_actv, \$solicitud_matrícula, \$tipo) AprobarSolicitud(\$id_actv,\$usuario, \$solicitud_matrícula, \$actividades_culminadas) EvaluarEstudiantes(\$id_actv,\$usuario,\$rendimiento,\$evaluación_estudiante,\$actividad es_culminadas, \$crédito_activo, \$fecha_vencimiento_crédito) BuscarSolicitantes(\$id_actv) BuscarMatriculados(\$id_actv) RetenciónMatrícula(\$id_actv)

	BuscarActvCulminadas(\$usuario) ListarEstudiantesNC() EliminarSolicitud(\$usuario, \$id_actv)
Descripción	Esta clase permite gestionar toda información Persona – Actividad.

Nombre: CGestiónNoticia	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	Adicionar(\$asunto, \$texto, \$importancia) ModificarNoticia(\$id, \$asunto, \$texto, \$importancia) BuscarNoticia(\$id) BuscarNoticiaM() EliminarNoticia(\$id) ListarNoticia()
Descripción	Esta clase permite gestionar toda información referente a la publicación de noticias.

Nombre: CGestiónUniversidad	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	BuscarUniversidades()
Descripción	Esta clase permite buscarlos nombres de todas las universidades del país.

3.2.3.3 Descripción de las clases de Acceso a Datos.

Nombre: AccesoActividad	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	Insertar(\$objActividad) ModificarActividad(\$objActividad, \$id_viejo) Eliminar(\$obj_actv) Buscar(\$obj_actv) BuscarfO(\$obj_actv) BuscarAP(\$obj_actv) BuscarSubA(\$obj_actv) BuscarEstado(\$obj_actv) Listar() ListarM()
Descripción	Esta clase hereda la conexión de Factory y redefine sus métodos.

Nombre: AccesoPersona	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	Adicionar(\$objpersona) ModificarPersona(\$objpersona, \$usuario) Eliminar(\$objpersona) Buscar(\$objpersona) BuscarRol(\$objpersona) Listar()
Descripción	Esta clase hereda la conexión de Factory y redefine sus métodos.

Nombre: AccesoActvPers	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	InsertarSolicitud(\$objActvPers) LlenarActvPers(\$objActvPers) EliminarS(\$objActvPers) EliminarActvPers(\$objActvPers) BuscarSolicitantes(\$obj_actv) BuscarMatriculados(\$obj_actvpers) RetenciónMatrícula(\$obj_actvpers) BuscarActvCulminadas(\$obj_actvpers) AprobarSolicitud(\$objActividad) EvaluarEstudiantes(\$objActividad) BuscarActvPers(\$objActvPers) BuscarAP(\$obj_actv) BuscarSubA(\$obj_actv) BuscarEstado(\$obj_actv) ListarE() ListarEM(\$usuario)
Descripción	Esta clase hereda la conexión de Factory y redefine sus métodos.

Nombre: AccesoNoticia	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	Insertar(\$objnot) Modificar(\$objnot) Eliminar(\$objnot) Buscar(\$objnot) BuscarNoticiaM(\$objnot)

	Listar()
Descripción	Esta clase hereda la conexión de Factory y redefine sus métodos.

Nombre: AccesoUniversidad	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	Adicionar(\$obj_univers) Eliminar(\$obj_univers) Insertar(\$obj_univers) Modificar(\$obj_univers) Buscar(\$obj_univers) BuscarUniversidades()
Descripción	Esta clase hereda la conexión de Factory y redefine sus métodos.

3.2.3.4 Descripción de las clases Entidad.

Nombre: Actividad	
Tipo de clase: Entidad	
Atributo	Tipo
\$id_actv	String
\$id_subactividad	String
\$título_actividad	String
\$sub_sistema	String
\$fecha_inicio	Date
\$fecha_de_fin	Date
\$forma_organizativa	String
\$universidad_rectora	String
\$universidad_colaboradora	String
\$horas_semanales	Double
\$total_de_horas	Double
\$periodicidad	String

\$modalidad	String
\$matricula_total	Integer
\$matricula_externa	Integer
\$país	String
\$profesor_principal	String
\$otros_profesores	Lista
\$programa	String
\$crédito_que_otorga	Integer
\$estado	Boolean
\$ofectar_matricula	Boolean
Responsabilidad:	
Nombre:	GetId_Actv() GetId_Subactividad() GetTitulo_Actividad() GetSub_Sistema() GetFecha_Inicio() GetFecha_de_Fin() GetForma_Organizativa() GetUniversidad_Rectora() GetUniversidad_Colaboradora() GetHoras_Semanales() GetTotal_de_Horas() GetPeriodicidad() GetModalidad() GetMatrícula_Total() GetMatrícula_Externa() GetPaís() GetProfesor_Principal() GetOtros_Profesores() GetPrograma() GetCrédito_que_Otorga() GetEstado() GetOfectar_Matrícula() SetId_Actv(\$pid_actv)

	SetId_Subactividad(\$pid_subactividad) SetTítulo_Actividad(\$ptítulo_actividad) SetSub_Sistema(\$psub_sistema) SetFecha_Inicio(\$pfecha_inicio) SetFecha_de_Fin(\$pfecha_de_fin) SetForma_Organizativa(\$pforma_organizativa) SetUniversidad_Rectora(\$puniversidad_rectora) SetUniversidad_Colaboradora(\$puniversidad_colaboradora) SetHoras_Semanales(\$phoras_semanales) SetTotal_de_Horas(\$ptotal_de_horas) SetPeriodicidad(\$pperiodicidad) SetModalidad(\$pmodalidad) SetMatrícula_Total(\$pmatricula_total) SetMatricula_Externa(\$pmatricula_externa) SetPaís(\$ppaís) SetProfesor_Principal(\$pprofesor_principal) SetOtros_Profesores(\$potros_profesores) SetPrograma(\$pprograma) SetCrédito_que_Otorga(\$pcrédito_que_otorga) SetEstado(\$peestado) SetOfectar_Matrícula(\$pofectar_matrícula)
Descripción	Es una clase entidad que almacena la información de las Actividades

Nombre: Persona	
Tipo de clase: Entidad	
Atributo	Tipo
\$usuario	String
\$carné_identidad	Integer
\$primer_nombre	String
\$segundo_nombre	String
\$primer_apellido	String
\$segundo_apellido	String
\$título	String

\$categoría_docente	String
\$área	String
\$sub_área	String
\$rol	String
Responsabilidad:	
Nombre:	<p>GetUsuario() GetCarné_Identidad() GetPrimer_Nombre() GetSegundo_Nombre() GetPrimer_Apellido() GetSegundo_Apellido() GetTítulo() Getcategoría_docente() Getárea() GetSub_área() GetRol() SetUsuario(\$usuario) SetCarné_Identidad(\$pcarné_identidad) SetPrimer_Nombre(\$pprimer_nombre) SetSegundo_Nombre(\$psegundo_nombre) SetPrimer_Apellido(\$pprimer_apellido) SetSegundo_Apellido(\$psegundo_apellido) SetTítulo(\$ptítulo) SetCategoría_Docente(\$pcategoría_docente) Setárea(\$párea) SetSub_área(\$psub_área) SetRol(\$prol)</p>
Descripción	Es una clase entidad que almacena la información de las Personas.

Nombre: ActvPers	
Tipo de clase: Controladora	
Atributo	Tipo
\$usuario	String
\$id_actv	String
\$solicitud_de_matrícula	Boolean
\$evaluación_del_estudiante	Integer
\$certificado	Boolean
\$fecha_vencimiento_crédito	Date
\$actividades_culminadas	Lista
\$tipo	String
\$rendimiento	Boolean
\$crédito_activo	Integer
Responsabilidad:	
Nombre:	GetUsuario() GetId_Actv() GetSolicitud_de_Matrícula() GetEvaluación_del_Estudiante() GetCertificado() GetFecha_Vencimiento_Crédito() GetActividades_Culminadas() GetTipo() GetRendimiento() GetCrédito_Activo() SetUsuario(\$usuario) SetId_Actv(\$pid_actv) SetSolicitud_de_Matrícula(\$psolicitud_de_matrícula) SetEvaluación_del_Estudiante(\$pevaluación_del_estudiante) SetFecha_Vencimiento_Crédito(\$pfecha_vencimiento_crédito) SetActividades_Culminadas(\$pactividades_culminadas) SetTipo(\$ptipo) SetRendimiento(\$prendimiento) SetCrédito_Activo(\$pcrédito_activo)
Descripción	Clase entidad, almacena la información de las Personas vinculadas a una Actividad.

Nombre: EntidadNoticia	
Tipo de clase: Controladora	
Atributo	Tipo
\$id	int
\$asunto	String
\$texto	String
\$importancia	String
Responsabilidad:	
Nombre:	GetAsunto() GetTexto() GetImportancia() SetID(\$pid) SetAsunto(\$pasunto) SetTexto(\$ptexto) SetImportancia(\$pimportancia)
Descripción	Es una clase entidad que almacena la información de las noticias.

Nombre: EntidadUniversidad	
Tipo de clase: Controladora	
Atributo	Tipo
\$id_universidad	int
\$nombre	String
Responsabilidad:	
Nombre:	GetId_Universidad() GetNombre() SetId_Universidad(\$pid_universidad) SetNombre(\$pnombre)
Descripción	Es una clase entidad que almacena las universidades.

3.3 Diagramas de interacción.

El diagrama de interacción es una secuencia de acciones en un caso de uso, que comienza cuando el actor invoca el caso de uso mediante el envío de un mensaje al sistema.

En el diseño es preferible representar las interacciones con diagramas de secuencia pues lo principal es encontrar secuencias de interacciones detalladas y ordenadas en el tiempo.

El diagrama de secuencia contiene las instancias de los actores, los objetos del diseño y las transmisiones de mensajes entre estos.

3.3.1 Diagrama de Secuencia de Aprobar Matrícula.

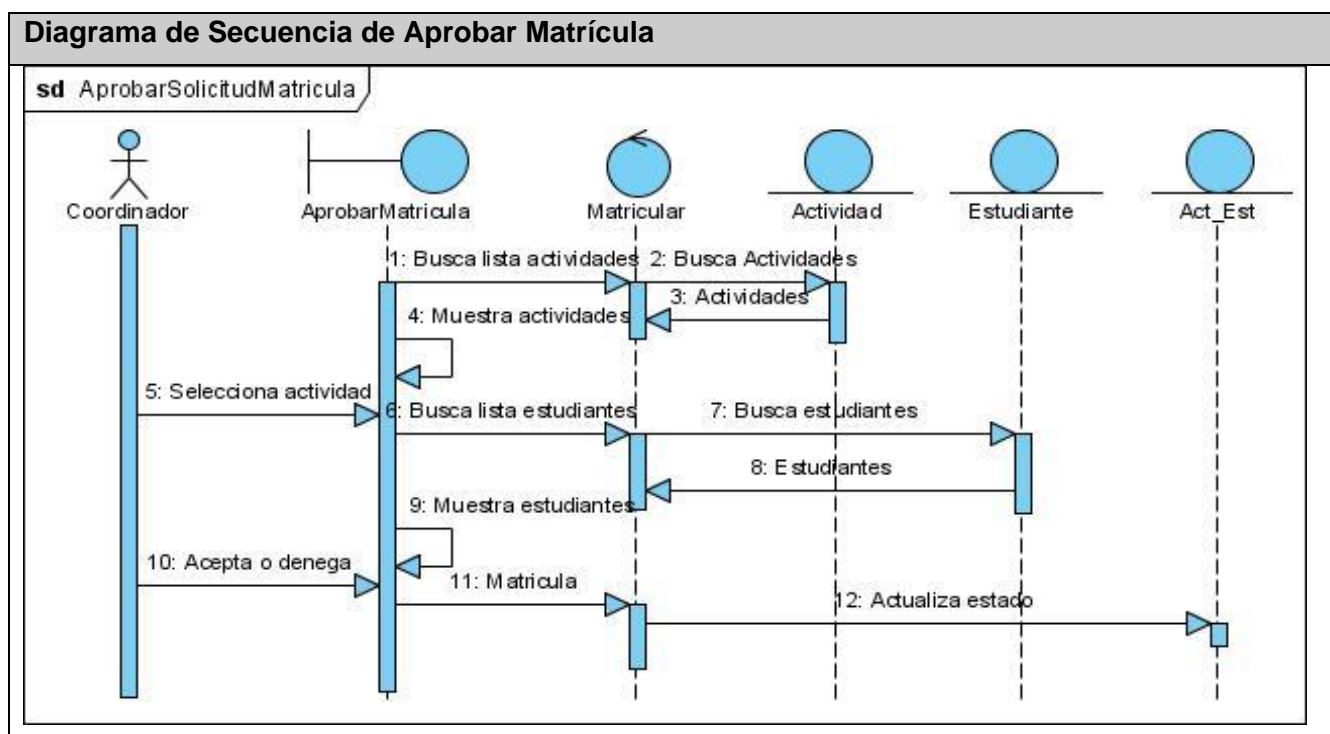


Tabla 3.23 Diagrama de Secuencia de Aprobar Matrícula.

3.3.2 Diagrama de Secuencia de Gestionar Persona.

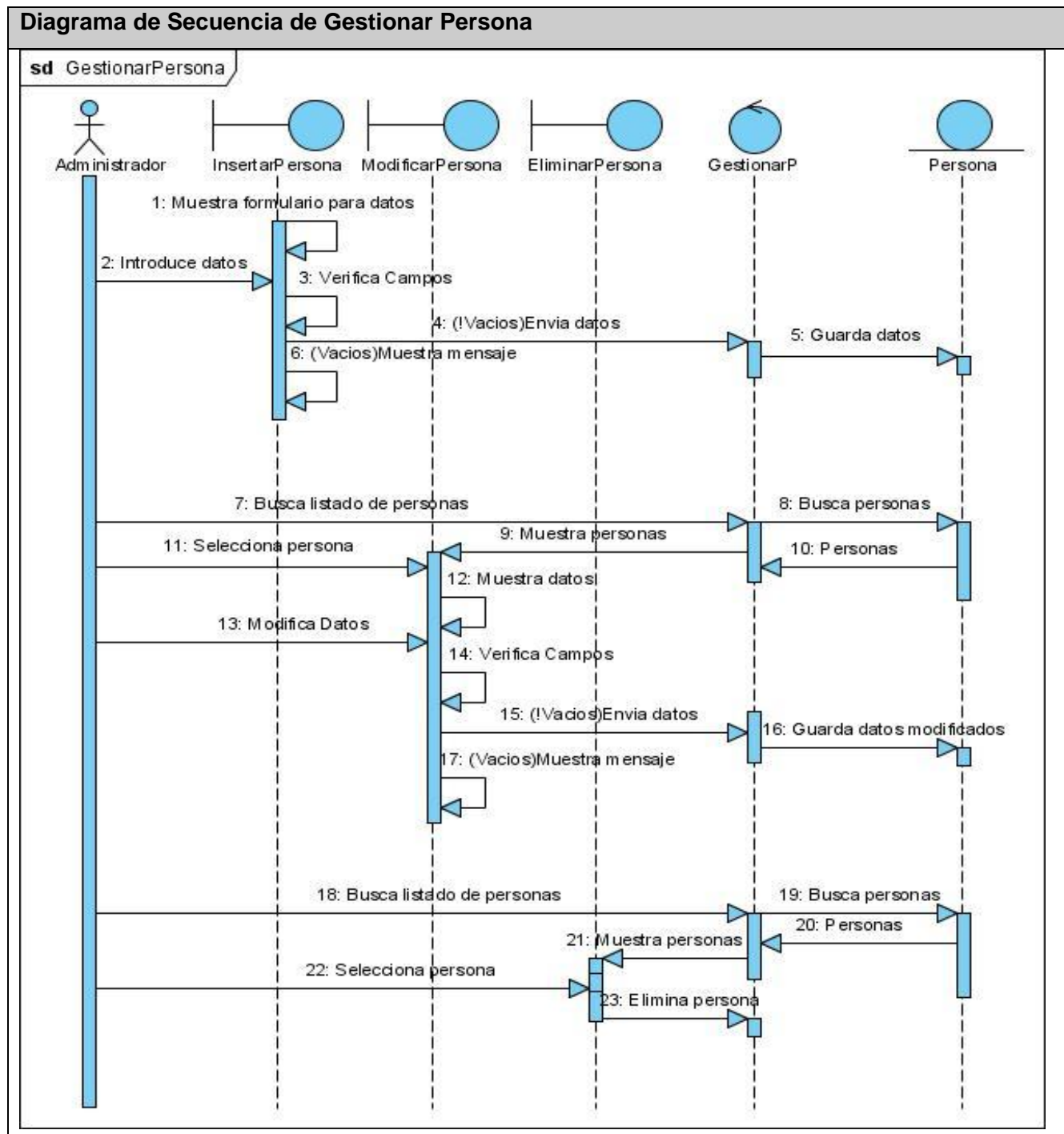


Tabla 3.24 Diagrama de Secuencia de Gestionar Persona.

3.3.3 Diagrama de Secuencia de Gestionar Actividad.

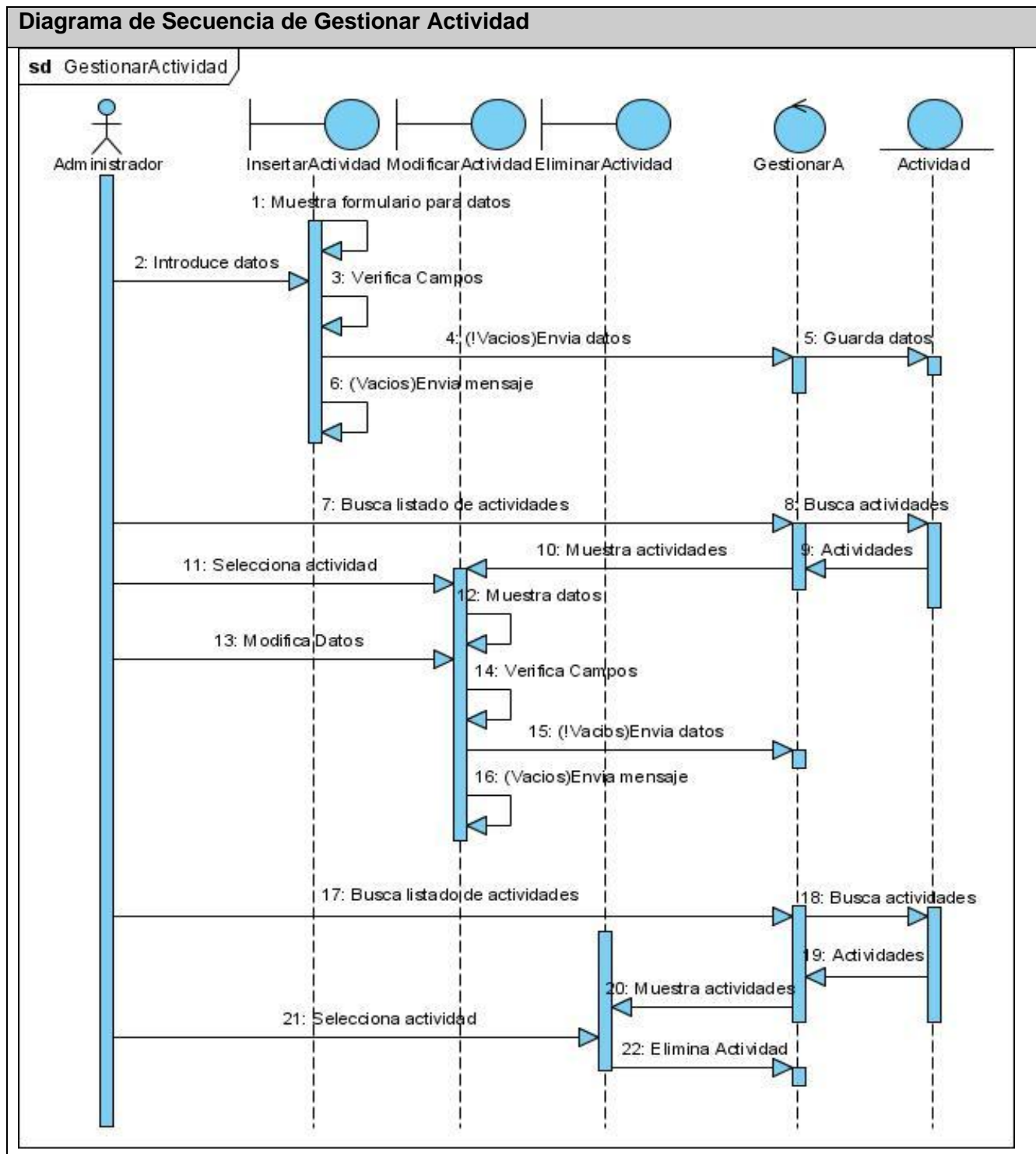


Tabla 3.25 Diagrama de Secuencia de Gestionar Actividad.

3.3.4 Diagrama de Secuencia de Evaluar Estudiante.

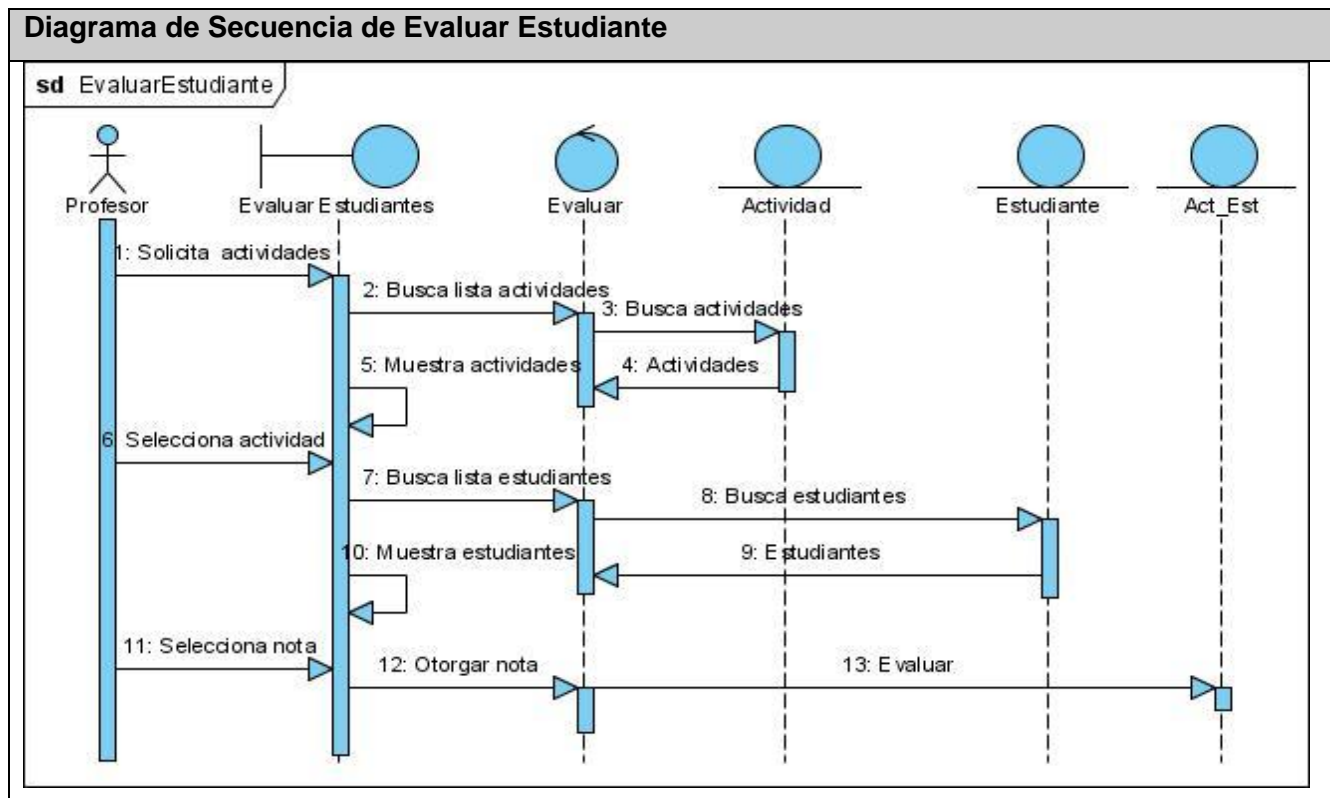


Tabla 3.26 Diagrama de Secuencia de Evaluar Estudiante

3.3.5 Diagrama de Secuencia de Solicitar Matrícula.

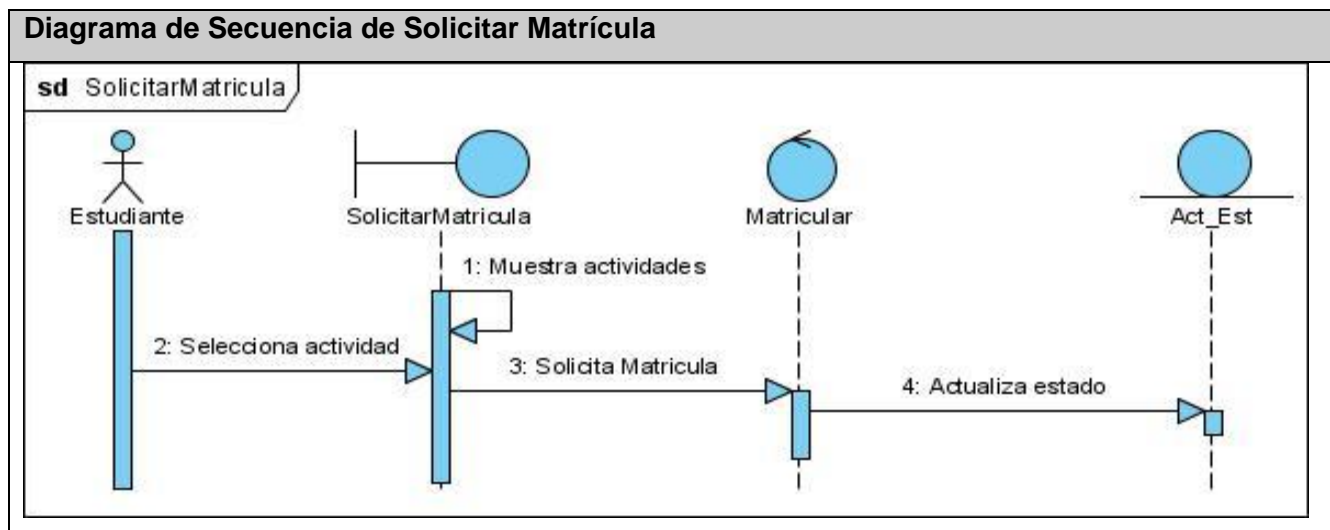
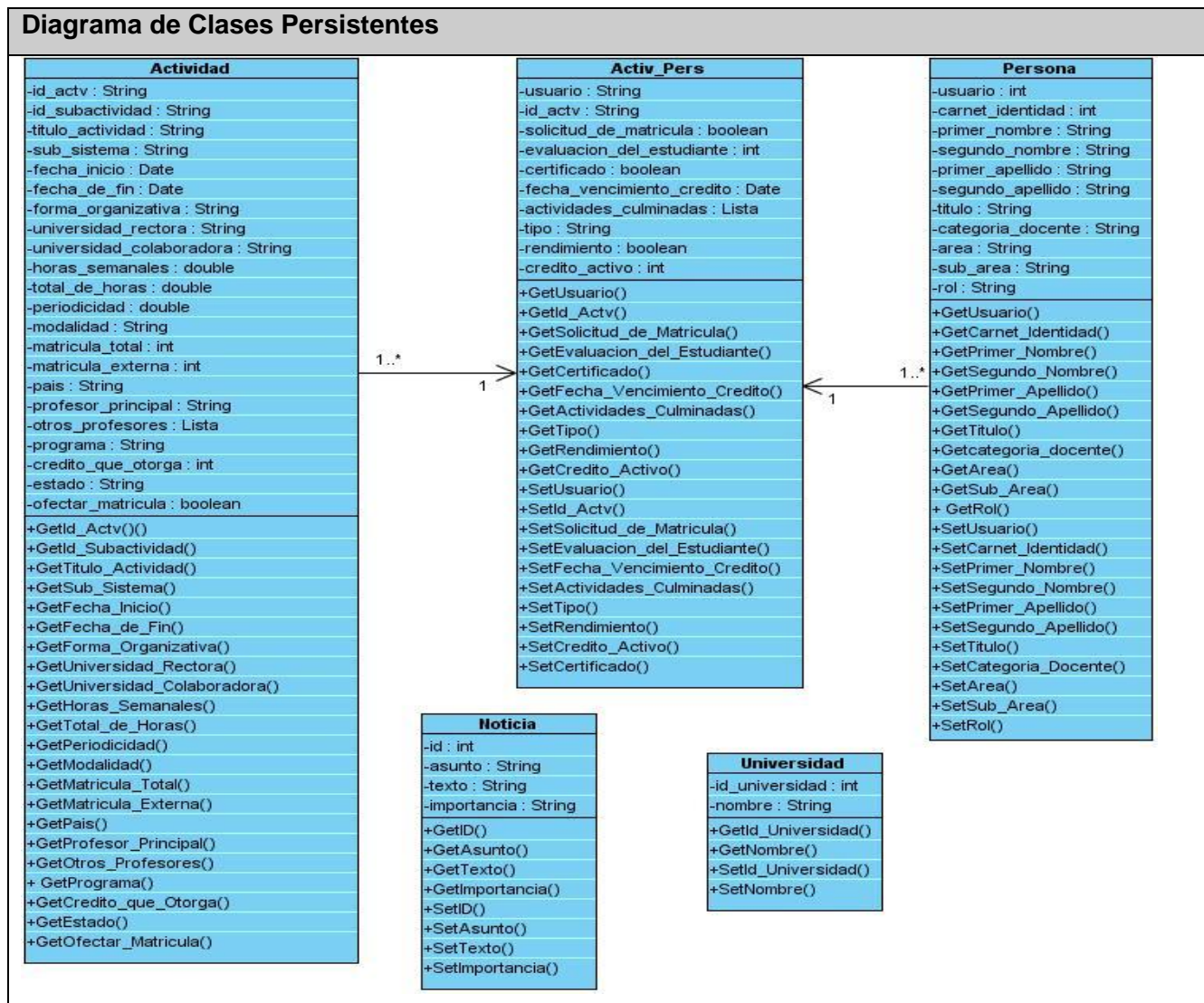


Tabla 3.27 Diagrama de Secuencia de Solicitar Matrícula.

3.4 Modelo de Clases Persistentes.



Conclusiones

En este capítulo se realizó el análisis del sistema, que proporcionó una comprensión detallada de los requisitos e impuso una estructura del sistema que se debe conservar y proporcionó una entrada esencial en el diseño, mediante el cual se comprendió en profundidad aspectos relacionados con los requisitos no funcionales y las restricciones asociadas y creó una entrada apropiada y un punto de partida para las actividades de implementación.

Capítulo 4

Implementación

Introducción

En el capítulo presente se presentan los modelos definidos en RUP como diagrama de despliegue y diagrama de componentes, como resultado principal del flujo de trabajo Implementación.

4.1 Modelo de Implementación.

El modelo de implementación describe cómo los elementos del diseño se implementan en términos de componentes, cómo se organizan de acuerdo con los mecanismos de estructuración y modularización y lenguajes de programación utilizados y cómo dependen unos de otros.

4.2 Diagrama de Despliegue.

El diagrama de despliegue describe la arquitectura física del sistema durante la ejecución en términos de procesadores, dispositivos y componentes de software. Además, describen la topología del sistema, es decir, la estructura de los elementos de hardware y software que ejecuta cada uno de ellos.

Un diagrama de despliegue es un grafo de nodos conectados por conexiones de comunicación. Los nodos son objetos físicos que existen en tiempo de ejecución y que representan algún tipo de recurso computacional (computadores con procesadores, impresoras, lectores de código barra y dispositivos de comunicación)

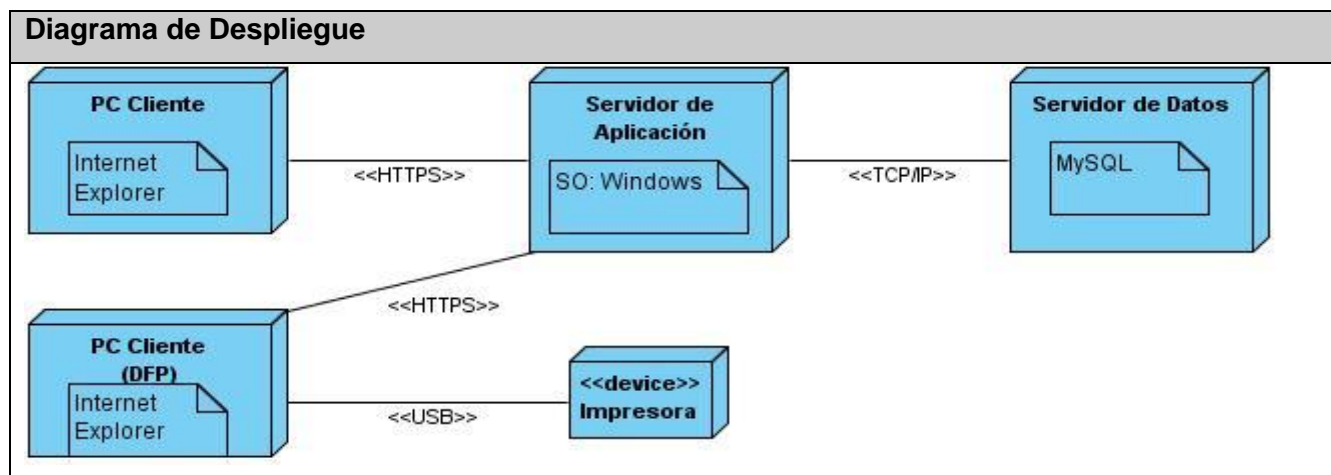


Tabla 4.1 Diagrama de Despliegue.

4.3 Diagrama de Componentes.

El diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software (componentes de código fuentes, binarios o ejecutables). Se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión de software, la reutilización y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. Un diagrama de componentes se representa como un grafo de componentes software unidos por medio de relaciones de dependencia.

4.3.1 Diagrama de Componentes.

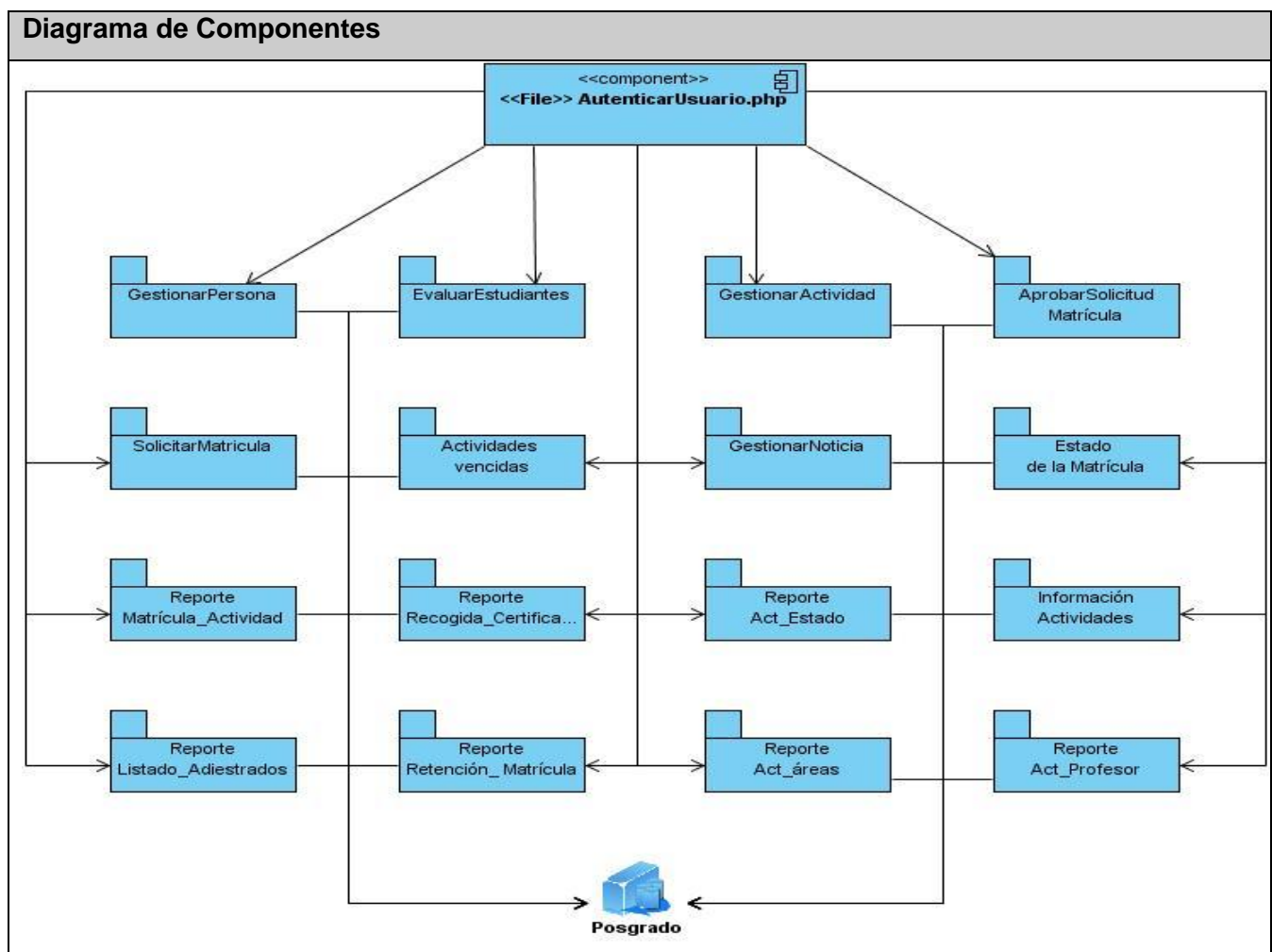


Tabla 4.2 Diagrama de Componentes.

4.3.2 Diagrama de Componentes Estado de la Matrícula.

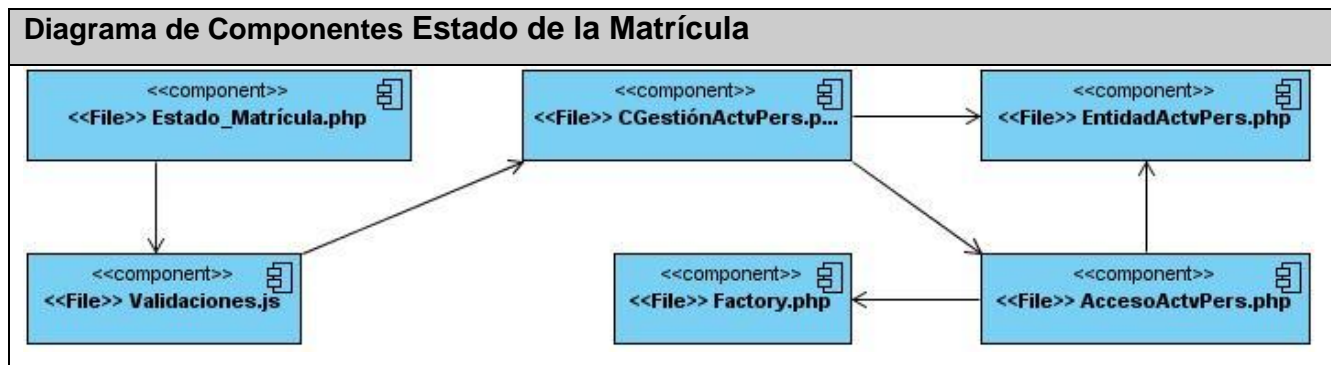


Tabla 4.3 Diagrama de Componentes Estado de la Matrícula.

4.3.3 Diagrama de Componentes Actividades por Áreas.

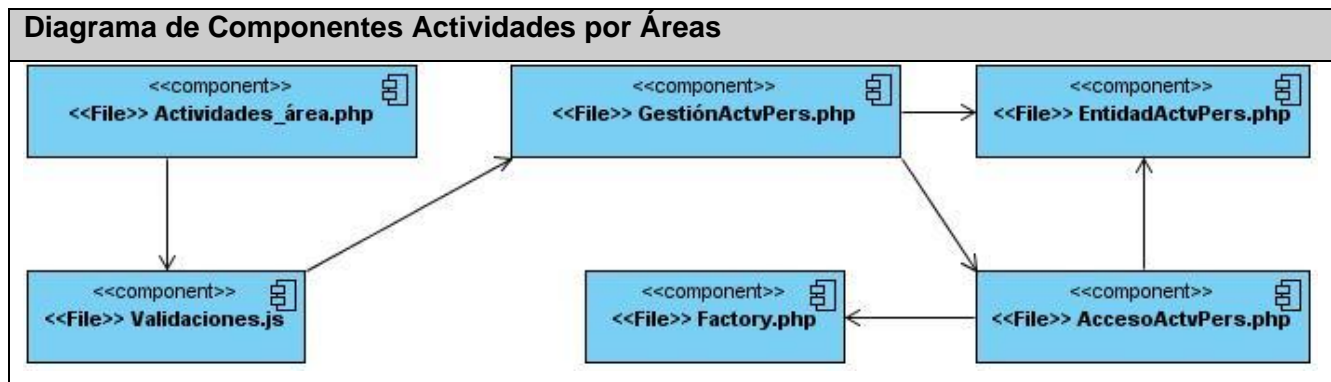


Tabla 4.4 Diagrama de Componentes Actividades por Áreas.

4.3.4 Diagrama de Componentes Actividades por estado.

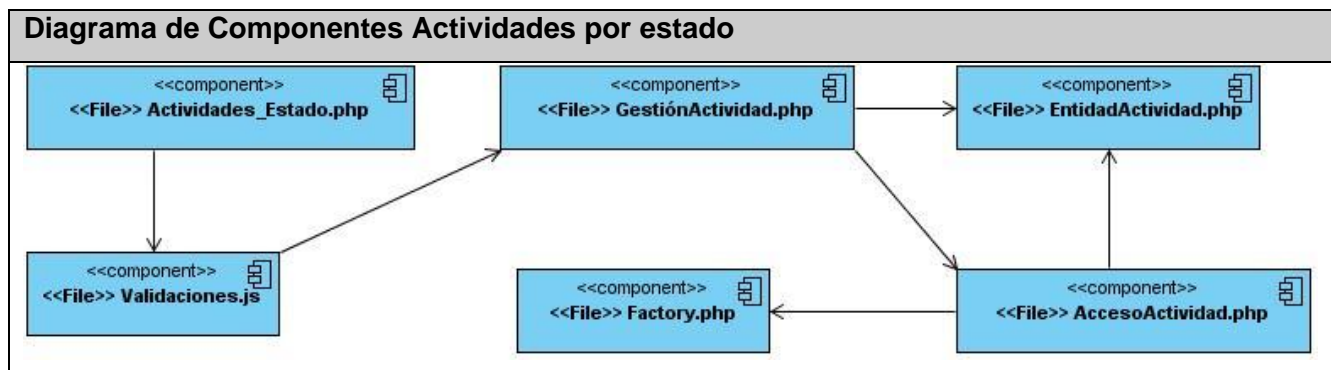


Tabla 4.5 Diagrama de Componentes Actividades por estado

4.3.5 Diagrama de Componentes Actividades por profesor.

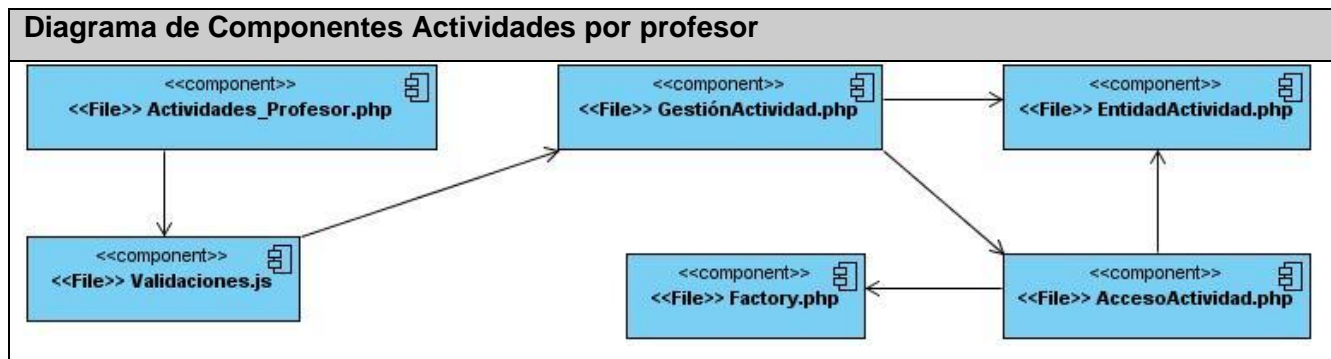


Tabla 4.6 Diagrama de Componentes Actividades por profesor.

4.3.6 Diagrama de Componentes Estudiantes adiestrados.

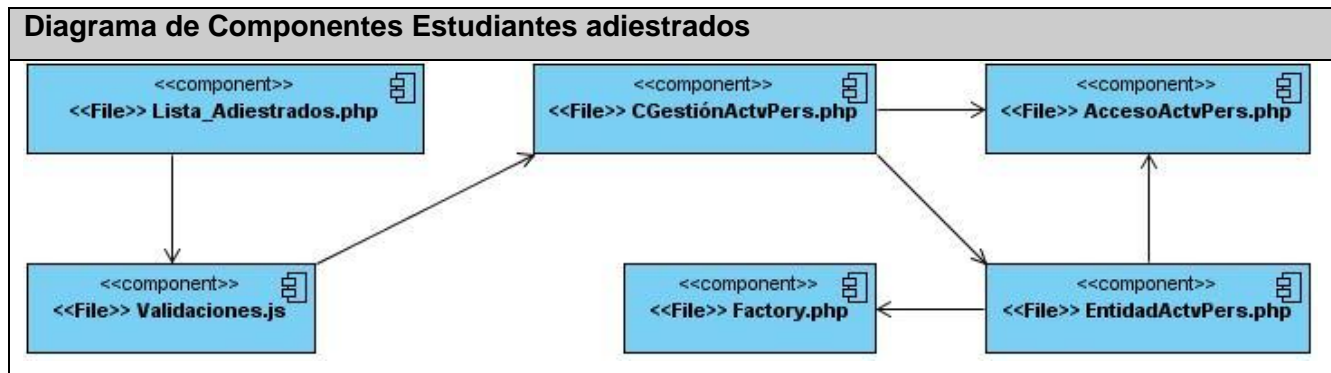


Tabla 4.7 Diagrama de Componentes Estudiantes adiestrados

4.3.7 Diagrama de Componentes Aprobar Solicitud Matrícula.

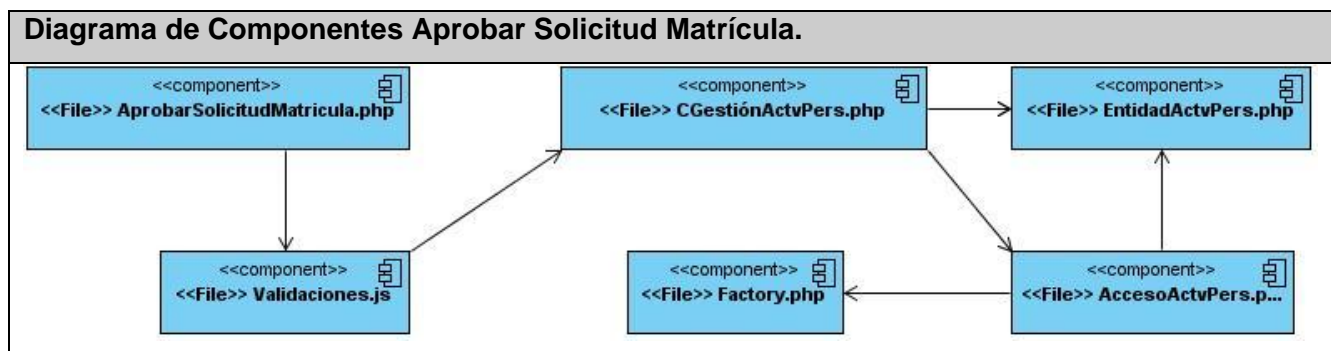


Tabla 4.8 Diagrama de Componentes Aprobar Solicitud Matrícula.

4.3.8 Diagrama de Componentes Actividades vencidas.

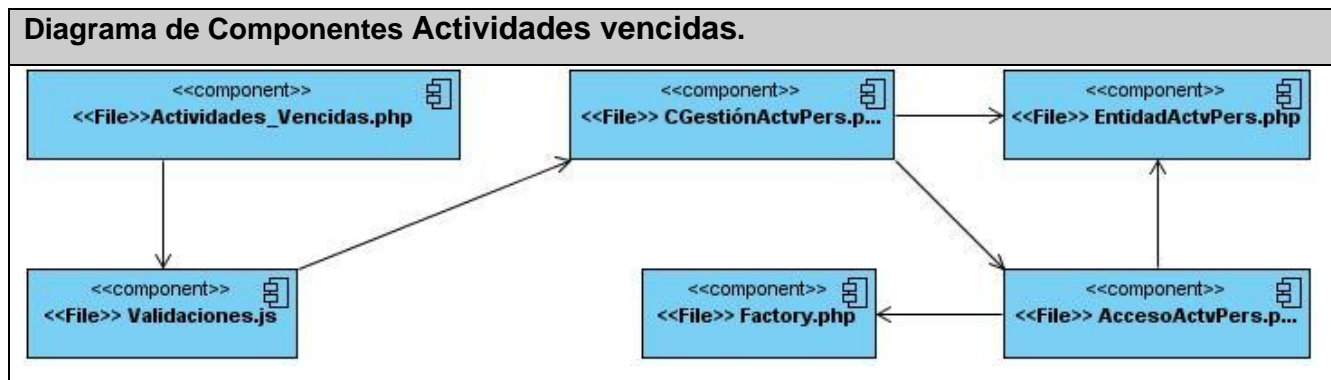


Tabla 4.9 Diagrama de Componentes Actividades vencidas.

4.3.9 Diagrama de Componentes Retención de Matrícula.

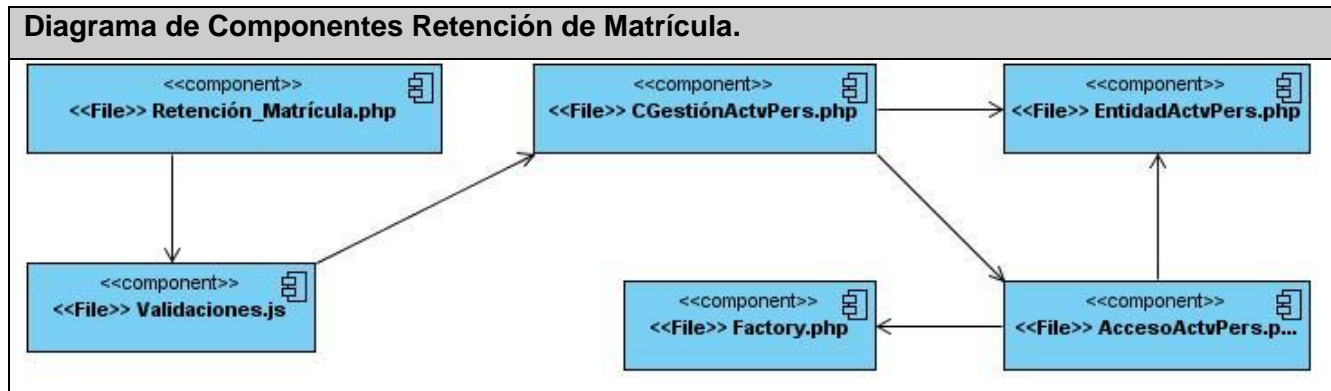


Tabla 4.10 Diagrama de Componentes Retención de Matrícula.

4.3.10 Diagrama de Componentes Matrícula por Actividad.

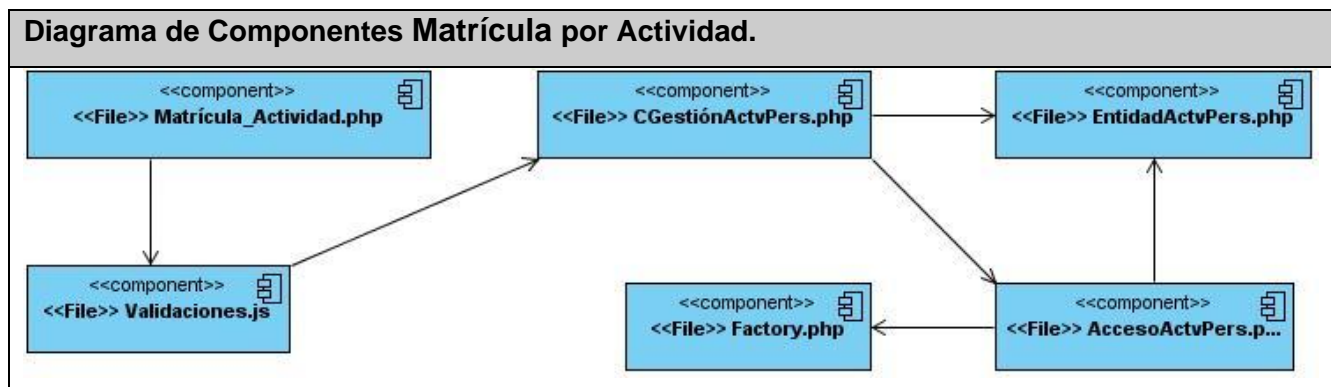


Tabla 4.11 Diagrama de Componentes Matrícula por Actividad.

4.3.11 Diagrama de Componentes Recogida de certificado.

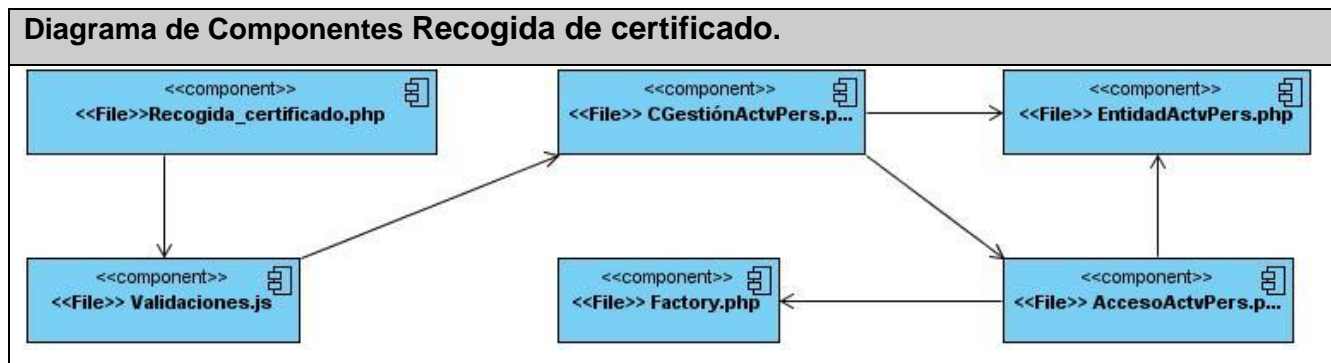


Tabla 4.12 Diagrama de Componentes Recogida de certificado.

4.3.12 Diagrama de Componentes Evaluar estudiantes.

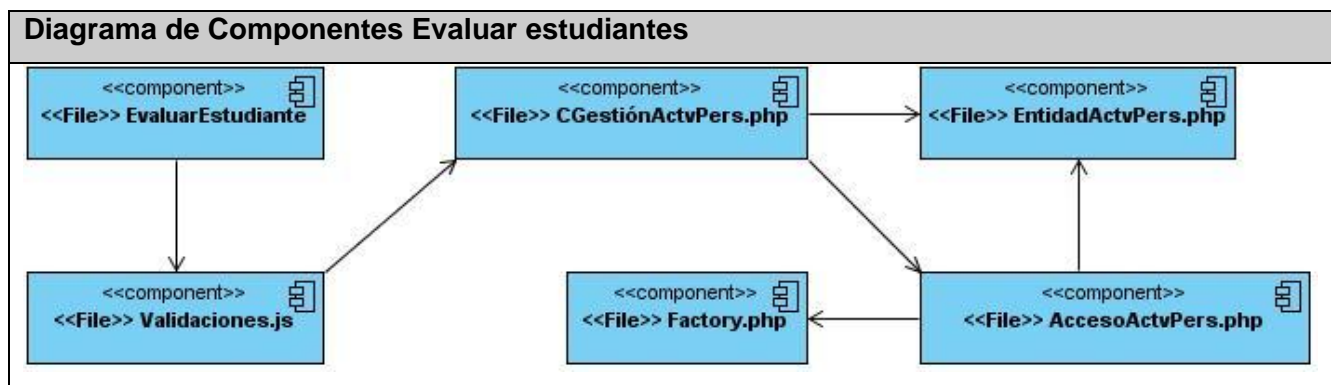


Tabla 4.13 Diagrama de Componentes Evaluar estudiantes.

4.3.13 Diagrama de Componentes Información Actividades.

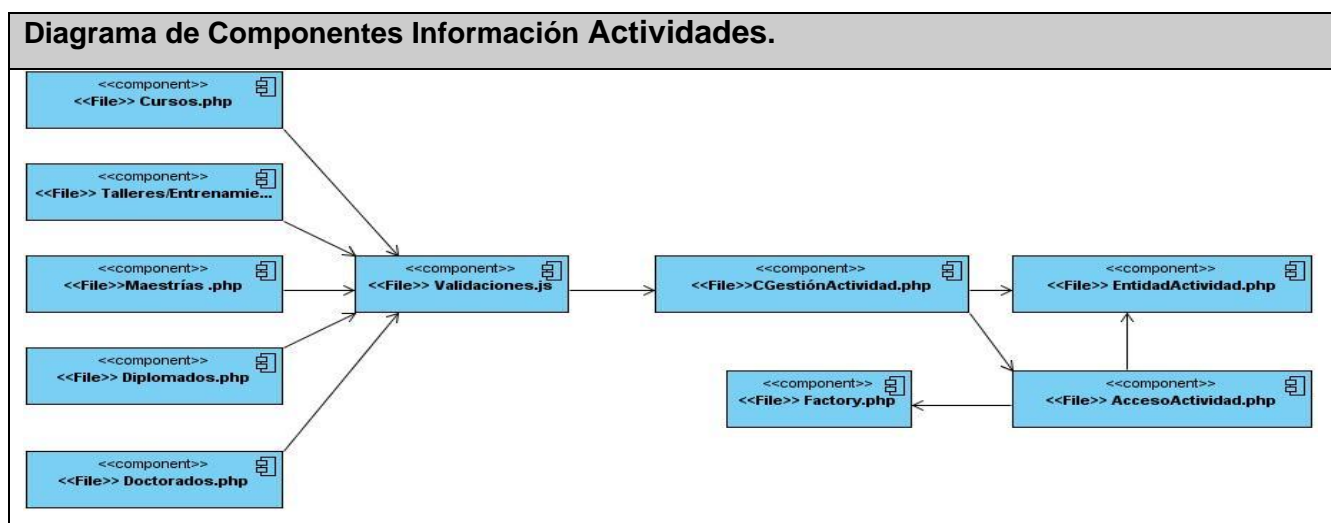


Tabla 4.14 Diagrama de Componentes Información Actividades.

4.3.14 Diagrama de Componentes Gestionar Actividad.

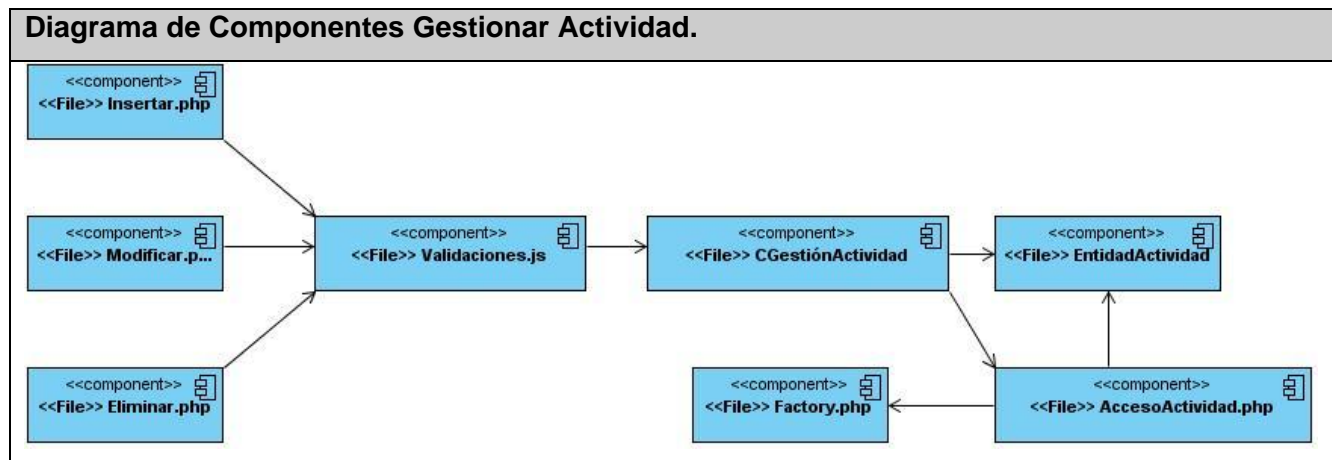


Tabla 4.15 Diagrama de Componentes Gestionar Actividad.

4.3.15 Diagrama de Componentes Gestionar Persona.

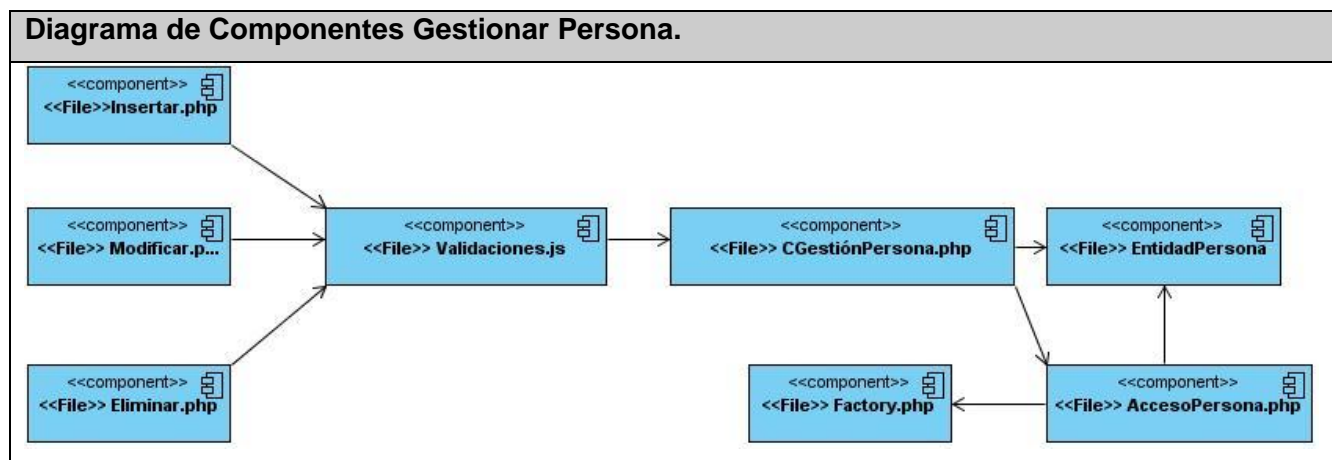


Tabla 4.16 Diagrama de Componentes Gestionar Persona.

4.3.16 Diagrama de Componentes Solicitar Matrícula.

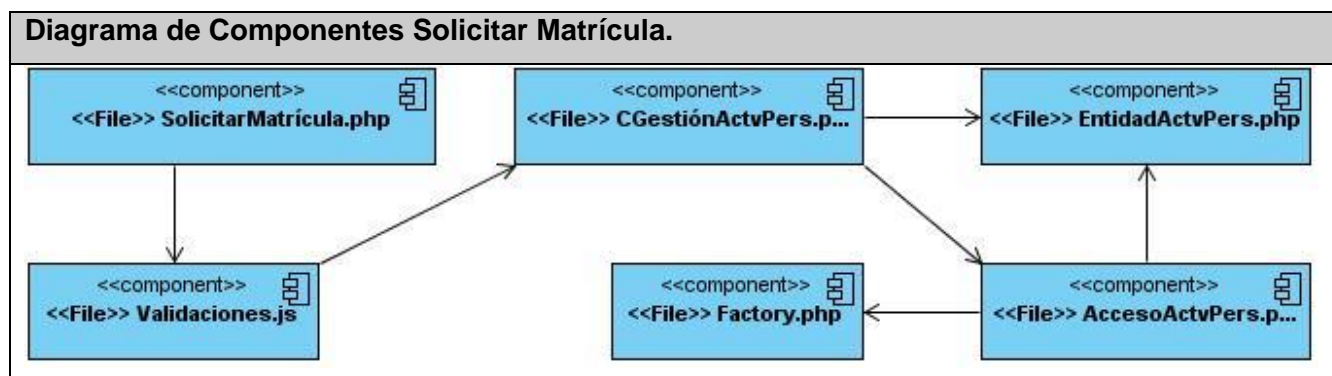


Tabla 4.19 Diagrama de Componentes Solicitar Matrícula.

4.3.17 Diagrama de Componentes Gestionar Noticia.

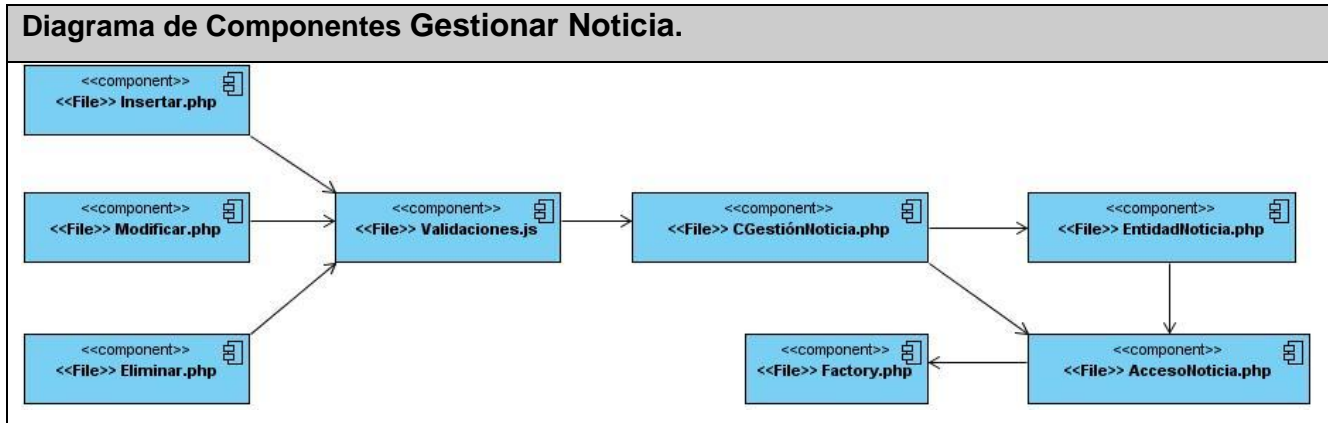


Tabla 4.18 Diagrama de Componentes Gestionar Noticia.

Conclusiones

En este capítulo fueron expuestos los objetivos de la fase Implementación del sistema verificando que el mismo cumpla los requisitos establecidos anteriormente por los clientes, a través de los diagramas de componentes que muestra la organización y dependencia entre un conjunto de componentes y el diagrama de despliegue, que juntos proveen la vista de implementación.

Conclusiones

Este trabajo estuvo guiado por las tareas de la investigación para dar cumplimiento al objetivo general y los objetivos específicos antes expuestos con lo cual se dará solución a la situación problemática pues mejorará la gestión de las actividades que se realizan en la Dirección de Formación Postgraduada en la UCI, brindando facilidades a los trabajadores y estudiantes de postgrado.

Se analizaron algunos sistemas de gestión de postgrados y las tendencias, tecnologías y metodologías actuales necesarias para la implementación del sistema, que por sus características particulares, responden a los requerimientos. Se logró establecer seguridad y protección de los datos de acuerdo con el nivel de seguridad requerido.

Se modeló el sistema obteniendo los artefactos de ingeniería necesarios para el desarrollo efectivo del proceso de Implementación utilizando como proceso de desarrollo del software, RUP.

Recomendaciones

Para lograr un sistema informático con más capacidades y utilidad los autores recomiendan:

- Dar mantenimiento a la aplicación.
- Realizar pruebas al sistema.
- Continuar con el estudio de las nuevas tecnologías y tendencias para futuras mejoras de la aplicación.
- Brindar capacitación al personal que utilizará la aplicación.
- Dotar al sistema de nuevas funcionalidades que se requieran con el tiempo y documentarlas.

Referencias Bibliográficas

1. **Superior, ministerio de educación.** *Reglamento de la Educación de Posgrado de la República de Cuba.* 2005. 350.
2. Información detallada módulos SIGA. [En línea] [Citado el: 2 de Mayo de 2009.] <http://www.dara.es/siga/>.
3. [En línea] [Citado el: 2 de Mayo de 2009.] <http://www.um.es/academic/>.
4. [En línea] [Citado el: 2 de Mayo de 2009.] cantabria: <http://www.unican.es/infoacademica>.
5. **Gutiérrez, Jorge A. Saavedra.** El Mundo Informático y tú en que mundo vives? [En línea] [Citado el: 28 de Enero de 2009.] <http://jorgesaaavedra.wordpress.com/about/>.
6. Breve historia. [En línea] [Citado el: 9 de Febrero de 2009.] http://www.librosweb.es/javascript/capitulo1/breve_historia.html.
7. [En línea] [Citado el: 9 de Febrero de 2009.] <http://www.desarrolloweb.com/articulos/491.php>.
8. [En línea] [Citado el: 20 de Enero de 2009.] <http://www.desarrolloweb.com/articulos/25.php>.
9. maestros del web. [En línea] 31 de Agosto de 2006. [Citado el: 9 de Febrero de 2009.] <http://www.maestrosdelweb.com/editorial/htmlhis/>.
10. [En línea] 9 de Mayo de 2001. [Citado el: 20 de Enero de 2009.] <http://www.desarrolloweb.com/articulos/392.php>.
11. [En línea] 1 de Noviembre de 2004. [Citado el: 27 de Enero de 2009.] http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.
12. [En línea] [Citado el: 27 de Enero de 2009.] http://www.netpecos.org/docs/mysql_postgres/x57.html.
13. [En línea] 5 de Mayo de 2007. [Citado el: 27 de Enero de 2009.] <http://jackopc.blogspot.com/2007/05/metodologias-rup-y-xp-procesos-de.html>.
14. Reynox, Servicios Informáticos. [En línea] [Citado el: 27 de Enero de 2009.] <http://www.reynox.com/sap/metodologia.php>.
15. **Orallo, Enrique Hernández.** El Lenguaje Unificado de Modelado (UML). [Online] [Cited: Enero 28, 2009.] <http://74.125.47.132/search?q=cache:xHa5jlFN1ocJ:www.disca.upv.es/enheror/pdf/ActaUML.PDF+ingenieria+%2B+lenguaje+unificado+de+modelado+%2B+enrique&cd=1&hl=es&ct=clnk&gl=cu>.
16. [En línea] 15 de Abril de 2002. [Citado el: 28 de Enero de 2009.] <http://www.ibit.org/home/bitacora/bitacora.php?id=26..>
17. **Cornejo, José Enrique González.** [En línea] 25 de Marzo de 2001. [Citado el: 28 de Enero de 2009.] http://www.docirs.cl/arquitectura_tres_capas.htm.

18. [En línea] [Citado el: 28 de Enero de 2009.] <http://html.rincondelvago.com/desarrollo-de-n-capas.html>.
19. Arquitecturas Cliente Servidor . [Online] [Cited: Enero 20, 2009.] <http://www.telefonica.net/web2/mikeferrer/TICB1%20-%20Arquitecturas%20cliente%20servidor.pdf>.
20. Arquitectura del Software. [En línea] [Citado el: 30 de Enero de 2009.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_1_a/capitulo2.pdf.
21. Arquitectura Modelo/Vista/Controlador. [Online] [Cited: Enero 30, 2009.] http://www.cica.es/formacion/JavaTut/Apendice/arg_mvc.html.
22. [En línea] [Citado el: 20 de Enero de 2009.] <http://www.raona.com/Not%C3%ADcias/Opini%C3%B3/SOA/tabid/150/Default.aspx>.
23. InfoWorld. [En línea] 27 de Julio de 2006. [Citado el: 29 de Enero de 2009.] http://iworld.com.mx/iw_news_read.asp?iwid=4567.
24. [En línea] [Citado el: 29 de Enero de 2009.] <http://www.desarrolloweb.com/articulos/1178.php>.
25. [En línea] [Citado el: 29 de Enero de 2009.] <http://www.osalt.com/es/dreamweaver>.
26. **Yanover, David Alejandro.** [En línea] [Citado el: 29 de Enero de 2009.] <http://www.desarrolloweb.com/articulos/1178.php>.
27. **Peña, ing. Dennys J. Hernández.** SACCEM: Sistema Automatizado CUBANO para el control de Equipos Médicos. [En línea] [Citado el: 30 de Enero de 2009.] http://74.125.47.132/search?q=cache:rkDB5soh8ZMJ:informatica2009.sld.cu/Members/dhernandez00242/saccem-sistema-automatizado-cubano-para-el-control-de-equipos-medicos-1/at_download/trabajo+visual+paradigm+%2B+ventajas&hl=es&ct=clnk&cd=7&gl=cu&lr=lang_es.
28. **Saha, Amit Kumar.** [En línea] [Citado el: 30 de Enero de 2009.] <http://74.125.47.132/search?q=cache:iNpMQcxpWdkJ:amitsaha.in.googlepages.com/MySQL-Intro-features-benefits-SPANIS.ppt+caracteristicas+%2B+MySQL&hl=es&ct=clnk&cd=6&gl=cu>.
29. [En línea] [Citado el: 30 de Enero de 2009.] <http://www.maestrosdelweb.com/editorial/zendstudio/>.
30. [Online] [Cited: Enero 30, 2009.] [http://www.taringa.net/posts/downloads/891980/WAMP-5-v-1_7_3-\(Muy-util-para-desarrolladores\).html](http://www.taringa.net/posts/downloads/891980/WAMP-5-v-1_7_3-(Muy-util-para-desarrolladores).html).

Bibliografía

1. Alonso, E. M. (s.f.). *Tutorial de HTML*. Recuperado el 28 de Enero de 2009, de http://gias720.dis.ulpgc.es/Gias/Cursos/Tutorial_html/indice.htm
2. Arenas, M. I. (s.f.). *GENEURA*. Recuperado el 10 de Febrero de 2009, de <http://geneura.ugr.es/~maribel/php/>
3. BSI. (s.f.). Recuperado el 12 de Enero de 2009, de <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/>
4. Castro, I. A. (2007). *TutorialPHP.net*. Recuperado el 25 de Enero de 2009, de <http://tutorialphp.net/>
5. García, J. J. (6 de Junio de 1996). *Tutorial HTML*. Recuperado el 21 de Enero de 2009, de <http://www.um.es/psibm/tutorial/>
6. Ivar Jacobson, G. B. *El Proceso Unificado de Desarrollo*.
7. *Ministerio de Educación Superior República de Cuba*. (s.f.). Recuperado el 15 de Enero de 2009, de http://www.mes.edu.cu/index.php?option=com_content&task=view&id=12&Itemid=28
8. *MySQL*. (s.f.). Recuperado el 10 de Febrero de 2009, de <http://dev.mysql.com/doc/refman/5.0/es/index.html>
9. *MySQL*. (s.f.). Recuperado el 10 de Febrero de 2009, de <http://dev.mysql.com/doc/refman/5.0/es/tutorial.html>
10. Ocegueda, E. L. (s.f.). *Gestión de Información y Fortalecimiento Académico*. Recuperado el 15 de Enero de 2009, de [http://74.125.47.132/search?q=cache:wA9g3hpfW40J:www.congresoretosyexpectativas.udg.mx/Congreso%25202/Mesa%25205/a\)%2520Dise%25F1o%2520de%2520los%2520procesos%2520educativos/5.a.8..pdf+gestores+de+informacion+academica&hl=es&ct=clnk&cd=1&gl=cu](http://74.125.47.132/search?q=cache:wA9g3hpfW40J:www.congresoretosyexpectativas.udg.mx/Congreso%25202/Mesa%25205/a)%2520Dise%25F1o%2520de%2520los%2520procesos%2520educativos/5.a.8..pdf+gestores+de+informacion+academica&hl=es&ct=clnk&cd=1&gl=cu)
11. Pozo, J. R. (4 de Agosto de 2003). *HTML con clase*. Recuperado el 4 de Febrero de 2009, de <http://html.conclase.net/tutorial/html/>
12. Rodríguez, M. L. (3 de Abril de 2006). *La educación de posgrado y su repercusión en la formación del profesional iberoamericano*. Recuperado el 10 de Diciembre de 2008, de http://bvs.sld.cu/revistas/ems/vol20_03_06/ems09306.htm
13. *Tutorial d Javascript*. (s.f.). Recuperado el 20 de Enero de 2009, de <http://www.lcc.uma.es/~eat/services/html-js/manual13.html>
14. Valente, M. (s.f.). *HTMLPOINT.com*. Recuperado el 15 de Enero de 2009, de <http://www.htmlpoint.com/javascript/tutorial/>
15. *WebEstilo*. (s.f.). Recuperado el 20 de Enero de 2009, de <http://www.webestilo.com/javascript/>

Glosario de Términos

API (La Interfaz de Programación de Aplicaciones): un conjunto de funciones residentes en bibliotecas, que permiten que una aplicación corra bajo un determinado sistema operativo.

CERN (Centro Europeo para la investigación Nuclear): es el mayor laboratorio de investigación en física de partículas a nivel mundial. Está situado en la frontera entre Francia y Suiza.

Créditos: es una unidad de expresión cuantitativa y cualitativa que valora los resultados alcanzados teniendo en cuenta la profundidad, el volumen y la intensidad del trabajo que realiza el estudiante para lograr las metas trazadas en los programas. Se otorgan al considerar cumplidos los objetivos de las actividades planificadas. El comité académico tiene la facultad de estimar el periodo de vigencia o caducidad de los créditos, previamente establecido en el programa que dirige, siempre que no rebasen los cinco años después de haber sido otorgados.

GPL de la GNU: es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Hipertexto: es el nombre que recibe el texto que en la pantalla de una computadora te conduce a otro texto relacionado. La forma más habitual de hipertexto en documentos es la de hipervínculos o referencias cruzadas automáticas que van a otros documentos.

Informix: compañía fundada en 1980, salió a bolsa en 1986 y durante parte de los años 1990 fue el segundo sistema de bases de datos más popular después de Oracle. Sin embargo, su éxito no duró mucho y para el año 2000 una serie de tropiezos en su gestión había debilitado seriamente a la compañía desde el punto de vista financiero.

Lenguaje interpretado: lenguaje que está diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados. También se les conoce como lenguajes de script.

Microsoft: empresa multinacional estadounidense, fundada en 1975 por Bill Gates y Paul Allen. Dedicada al sector de la informática, con sede en Redmond, Washington, Estados Unidos. Microsoft desarrolla, fabrica, licencia y produce software y equipos electrónicos. Siendo sus productos más usados el Sistema operativo Microsoft Windows y la suite Microsoft Office, estos productos tienen una importante posición entre los ordenadores personales.

Módems: acrónimo de MODulador-DEModulador; es decir, que es un dispositivo que transforma las señales digitales del ordenador en señal telefónica analógica y viceversa, con lo que permite al ordenador transmitir y recibir información por la línea telefónica.

MS SQL Server: es un sistema de gestión de bases de datos relacionales, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Navegador: herramienta que permite a los usuarios de Internet visitar sus sitios Web favoritos. Es un tipo de software con una interfaz gráfica que incluye botones de navegación, una barra de direcciones y una barra de estado (en general debajo de la ventana). La mayor parte de su superficie se usa para mostrar páginas Web.

Netscape: una empresa fundada en 1994 por Jim Clark y Mark Andreessen, desarrolló el primer navegador de éxito comercial, Netscape Navigator, que alentó al crecimiento explosivo del World Wide Web.

Oracle: es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general.

REYNOX: empresa, que comenzó a operar el 1 de julio de 2002, es de capitales argentinos y se especializa en el desarrollo de un amplio espectro de soluciones para diversos mercados, tanto en el país como en el exterior, desde sus inicios, ha experimentado un progreso sostenido. Para acompañar su expansión con mejores estándares de servicios, inició el proceso de certificación de la norma ISO 9001, Versión 2000- Sistemas de Gestión de Calidad, siendo seleccionada para participar de la tercera edición del PRO ISO-Programa de Apoyo a la Certificación ISO 9000, a cargo de la Dirección General de Tecnología de la Ciudad de Buenos Aires.

Sun Microsystems: fundada en 1982 por Andreas Bechtolsheim, Vinod Khosla, Scott McNealy y Bill Joy como la concreción de un proyecto de Bechtolsheim que se había originado un año antes. Su intención era crear una estación de trabajo en red que fuera más funcional y más barata que las existentes. Este proyecto es, curiosamente, el que le da el nombre a la compañía: Stanford University Network (SUN).

World Wide Web: o Red Global Mundial es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet. Con un navegador Web, un usuario visualiza páginas Web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

XHTML (lenguaje extensible de marcado de hipertexto): lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas Web.