

**UNIVERSIDAD DE SANCTI SPÍRITUS**

**“JOSÉ MARTÍ PÉREZ”**

**FACULTAD DE INGENIERÍA  
CARRERA INFORMÁTICA**



**APLICACIÓN WEB PARA OPTIMIZAR LA GESTIÓN DE USUARIOS DE DOMINIO DE LOS  
ESTUDIANTES EN LA UNIVERSIDAD DE SANCTI SPÍRITUS “JOSÉ MARTÍ PÉREZ”.**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN INFORMÁTICA**

**AUTOR: ORELVIS BERNAL FIALLO**

**TUTOR: BERNARDO YASER LEÓN ÁVILA**

**Sancti Spíritus, Cuba.**

**2013**

## PENSAMIENTO

*"Las organizaciones gastan millones de dólares en firewalls y dispositivos de seguridad, pero tiran el dinero porque ninguna de estas medidas cubre el eslabón más débil de la cadena de seguridad: la gente que usa y administra los ordenadores"*

Kevin Mitnick

## AGRADECIMIENTOS

Quisiera agradecer a todas las personas que una forma u otra han contribuido con su ayuda y sin la cual no hubiese sido posible la realización de este trabajo, en especial:

- A mi tutor por su ayuda, por su preocupación, y apoyo en todo momento. Gracias por todo.
- A mi compañero Ernesto, sin quien no hubiera podido ser posible la realización de esta tesis. Gracias amigo.
- A mis padres porque como alguien dijo *“somos lo que alcanzamos a ser con lo que han hecho de nosotros”*.
- A mis “tíos” Arquímedes y Gertrudis, a mis primos Solyansy, Leosvany, Ana y toda la familia que se ha preocupado a toda hora por mi carrera. A todos, Gracias.
- A mi *“Niña”*, por comprenderme y saber ocupar el lugar que le corresponde, aún cuando no ha estado muy bien de salud los últimos días.

## DEDICATORIA

A mis padres por ser tan incondicionales,  
por su paciencia, amor, dedicación y confianza.

## RESUMEN

El presente trabajo está orientado al desarrollo de una aplicación web cuya funcionalidad principal es la optimización de la gestión de los usuarios de dominio para los estudiantes en la Universidad de Sancti Spíritus “José Martí Pérez” (UNISS). Dicha gestión actualmente se realiza de forma manual y personalizada lo que dificulta la gestión eficiente del Directorio Activo y con ello de la red instalada en dicho centro. La UNISS cuenta con un sistema gestor de matrícula (SIGENU) que almacena en una base de datos de postgresQL toda la información personal de los estudiantes que matriculan cada año y desde la cual se pueden extraer los datos necesarios para la creación y actualización de usuarios de dominio evitando así la inserción manual en el Directorio Activo. Además, los estudiantes que matriculan en los cursos de postgrado que se imparten no cuentan con usuarios, por lo que también se implementará su gestión. Un análisis minucioso de las metodologías disponibles arrojó que RUP (Proceso Unificado Racional), es la más completa para el proceso de desarrollo de la aplicación web en cuestión. El lenguaje de programación multiplataforma PHP fue el seleccionado por seguir los estándares de software libre, ya que permite una mejor portabilidad y flexibilidad para el desarrollo, mantenimiento y modificación de la aplicación web, lo que permitirá a los encargados de la gestión de usuarios de la Universidad de Sancti Spíritus una fácil y rápida acogida al mantenimiento de la misma.

## ABSTRACT

The present work is aimed at developing a web application whose main functionality is to optimize the management of domain users for students at the University of Sancti Spiritus "José Martí Pérez" (UNISS). This management is currently done manually and personalized hindering the efficient management of Active Directory and thus of the net in the center. The UNISS has an enrollment management system (SIGENU) storing a PostgreSQL database all personal information of students who enroll each year and from which one can extract the data required for creating and updating users domain avoiding manual entry in Active Directory. In addition, students who enroll in graduate courses taught users do not have, so management will also be implemented. A close analysis of the available methodologies showed that RUP (Rational Unified Process), is the most complete for the process of web application development in question. The programming language PHP platform was selected for further free software standards, allowing better portability and flexibility for the development, maintenance and modification of the web application, which will allow managers of users University of Sancti Spiritus host a quick and easy maintenance of the same.

## CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICO Y METODOLÓGICA QUE SUSTENTA LA EFICIENCIA DE UN SOFTWARE PARA OPTIMIZAR LA GESTIÓN DE USUARIOS DE DOMINIO.....	5
<b>1.1    Introducción .....</b>	<b>5</b>
<b>1.2    Gestión de usuarios.....</b>	<b>5</b>
1.2.1    El Active Directory .....	6
1.2.2    La gestión de usuarios con LDAP.....	6
1.2.3    La gestión de usuarios en la UNISS.....	7
<b>1.3    SIGENU.....</b>	<b>8</b>
<b>1.4    ¿Aplicación web o de escritorio? .....</b>	<b>8</b>
1.4.1    Modelo Cliente-Servidor.....	9
1.4.2    Arquitectura en 3 capas.....	10
<b>1.5    Framework Symfony2 .....</b>	<b>10</b>
1.5.1    Modelo-Vista-Controlador.....	11
1.5.2    PHP 5.3.....	12
1.5.1.1    Funciones Anónimas .....	13
1.5.1.2    Namespaces.....	13
1.5.1.3    Anotaciones .....	13
1.5.3    Doctrine.....	13
1.5.4    Twig.....	14
<b>1.6    Metodologías utilizadas para el diseño. ....</b>	<b>14</b>
1.6.1    Lenguaje Unificado de Modelado (UML).....	15
1.6.2    Metodología RUP.....	16
<b>1.7    Otros lenguajes que serán utilizados.....</b>	<b>16</b>
1.7.1    JavaScript.....	16
1.7.2    HTML.....	17
<b>1.8    Herramientas de desarrollo.....</b>	<b>17</b>
1.8.1    NetBeans .....	17
1.8.2    Visual Paradigm.....	17
1.8.3    Rational Rose.....	18
<b>1.9    Sistemas gestores de base de datos.....</b>	<b>18</b>
1.9.1    PostgreSQL.....	18
1.9.2    MySQL.....	19
<b>1.10    Conclusiones.....</b>	<b>20</b>
CAPÍTULO 2: DESCRIPCIÓN DEL SOFTWARE PARA OPTIMIZAR LA GESTIÓN DE USUARIOS DE LOS ESTUDIANTES EN LA UNISS. ....	21
<b>1.....</b>	<b>21</b>

<b>2.1</b>	<b>Introducción .....</b>	<b>21</b>
<b>2.2</b>	<b>Identificación de los procesos del negocio .....</b>	<b>21</b>
<b>2.3</b>	<b>Reglas del negocio .....</b>	<b>23</b>
<b>2.4</b>	<b>Deficiencias en el proceso actual de gestión de usuarios para los estudiantes en la UNISS: ..</b>	<b>24</b>
<b>2.5</b>	<b>Mejoras propuestas al proceso de.....</b>	<b>24</b>
<b>2.6</b>	<b>Modelo de casos de uso del negocio.....</b>	<b>25</b>
2.6.1	Actores del negocio .....	25
2.6.2	Trabajadores del negocio .....	26
2.6.3	Caso de Uso del negocio.....	26
2.6.4	Diagrama de casos de uso del negocio.....	26
2.6.5	Descripción de los casos de uso del negocio .....	27
<b>2.7</b>	<b>Modelo de objetos.....</b>	<b>28</b>
<b>2.8</b>	<b>Requerimientos funcionales y no funcionales .....</b>	<b>28</b>
2.8.1	Requerimientos funcionales:.....	28
2.8.2	Requerimientos no funcionales:.....	30
<b>2.9</b>	<b>Modelo del sistema.....</b>	<b>32</b>
2.9.1	Actores del sistema .....	33
2.9.2	Casos de uso del sistema:.....	34
2.9.3	Paquetes y sus relaciones: .....	35
2.9.4	Diagramas de casos de uso del sistema.....	35
2.9.5	Descripción de los casos de uso del sistema y los prototipos de interfaz: .....	37
<b>2.10</b>	<b>Conclusiones.....</b>	<b>38</b>
CAPÍTULO 3: IMPLEMENTACIÓN DEL SOFTWARE PARA OPTIMIZAR LA GESTIÓN DE USUARIOS DE DOMINIO PARA LOS ESTUDIANTES EN LA UNISS. ....		39
<b>2</b>	<b>.....</b>	<b>39</b>
<b>3.1</b>	<b>Introducción: .....</b>	<b>39</b>
<b>3.2</b>	<b>Diseño.....</b>	<b>39</b>
3.2.1	Diagrama de clases.....	39
<b>3.3</b>	<b>Diseño de la base de datos .....</b>	<b>41</b>
3.3.1	Diagrama de clases persistentes .....	41
3.3.2	Modelo de datos.....	41
<b>3.4</b>	<b>Implementación.....</b>	<b>42</b>
3.4.1	Diagrama de despliegue.....	42
<b>3.5</b>	<b>Diagrama de componentes.....</b>	<b>42</b>
3.5.1	Descripción de los componentes: .....	44
<b>3.6</b>	<b>Principios de diseño.....</b>	<b>45</b>
3.6.1	Estándares en la interfaz de la aplicación .....	45



3.6.2	Tratamiento de errores.....	45
3.6.3	Concepción general de la ayuda .....	46
3.6.4	Seguridad.....	46
<b>3.7</b>	<b>Conclusiones.....</b>	<b>46</b>
	CONCLUSIONES.....	47
	RECOMENDACIONES .....	48
	BIBLIOGRAFÍA.....	1
	ANEXOS.....	1

## ÍNDICE DE FIGURAS

<b>FIGURA 1.</b>	DISTRIBUCIÓN DEL EDIFICIO DOCENTE I EN DEL ÁRBOL DE DIRECTORIO.....	7
<b>FIGURA 2.</b>	DIAGRAMA DE CASO DE USO DEL NEGOCIO RECIBIR USUARIO DE DOMINIO.....	27
<b>FIGURA 3.</b>	MODELO DE OBJETOS .....	28
<b>FIGURA 4.</b>	PAQUETES DEL SISTEMA Y SUS RELACIONES. ....	35
<b>FIGURA 5.</b>	DIAGRAMA DE CASOS DE USO DEL SISTEMA: SEGURIDAD.....	36
<b>FIGURA 6.</b>	DIAGRAMA DE CASOS DE USO DEL SISTEMA: SOLICITUD.....	36
<b>FIGURA 7.</b>	DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR ESTUDIANTE DE POSTGRADO.....	40
<b>FIGURA 8.</b>	DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR ESTUDIANTE DE PREGRADO.....	40
<b>FIGURA 9.</b>	DIAGRAMA DE CLASES PERSISTENTES.....	41
<b>FIGURA 10.</b>	DIAGRAMA DE DESPLIEGUE.....	42
<b>FIGURA 11.</b>	DIAGRAMA DE COMPONENTES.....	43
<b>FIGURA 12.</b>	DIAGRAMA DE ACTIVIDADES DEL CASO DE USO: RECIBIR USUARIO DE DOMINIO.....	1
<b>FIGURA 13.</b>	DIAGRAMA DE CASOS DE USO DEL SISTEMA: GESTIÓN.....	2
<b>FIGURA 14.</b>	DIAGRAMA DE CASOS DE USO DEL SISTEMA: CONSULTA.....	3
<b>FIGURA 15.</b>	PROTOTIPO DE INTERFAZ: INICIAR SESIÓN.....	12
<b>FIGURA 16.</b>	PROTOTIPO DE INTERFAZ: CERRAR SESIÓN.....	12
<b>FIGURA 17.</b>	PROTOTIPO DE INTERFAZ: CERRAR SESIÓN.....	13
<b>FIGURA 18.</b>	PROTOTIPO DE INTERFAZ: GESTIONAR ESTUDIANTE DE POSTGRADO.....	13
<b>FIGURA 19.</b>	PROTOTIPO DE INTERFAZ: GESTIONAR OPERADOR DE SIGENU.....	14
<b>FIGURA 20.</b>	PROTOTIPO DE INTERFAZ: GESTIONAR CURSO DE POSTGRADO.....	14
<b>FIGURA 21.</b>	PROTOTIPO DE INTERFAZ: GESTIONAR CARRERA.....	15
<b>FIGURA 22.</b>	PROTOTIPO DE INTERFAZ: GESTIONAR TIPO DE CARRERA.....	15
<b>FIGURA 23.</b>	PROTOTIPO DE INTERFAZ: ELIMINAR USUARIO DE DOMINIO.....	16

<b>FIGURA 24.</b>	PROTOTIPO DE INTERFAZ: AUTORIZAR SOLICITUD DE CURSO DE POSTGRADO. ....	16
<b>FIGURA 25.</b>	PROTOTIPO DE INTERFAZ: AUTORIZAR SOLICITUD DE ESTUDIANTE DE POSTGRADO. ....	17
<b>FIGURA 26.</b>	PROTOTIPO DE INTERFAZ: MOSTRAR ESTADO DE SOLICITUD DE CURSO DE POSTGRADO. ....	17
<b>FIGURA 27.</b>	PROTOTIPO DE INTERFAZ: MOSTRAR SOLICITUDES DE CURSO DE POSTGRADO. ....	17
<b>FIGURA 28.</b>	PROTOTIPO DE INTERFAZ: MOSTRAR SOLICITUDES DE CURSO DE POSTGRADO APROBADAS. ....	18
<b>FIGURA 29.</b>	PROTOTIPO DE INTERFAZ: MOSTRAR SOLICITUDES DE ESTUDIANTE DE POSTGRADO. ....	18
<b>FIGURA 30.</b>	PROTOTIPO DE INTERFAZ: MOSTRAR SOLICITUDES DE ESTUDIANTE DE POSTGRADO APROBADAS. ....	19
<b>FIGURA 31.</b>	PROTOTIPO DE INTERFAZ: MOSTRAR DATOS DE ESTUDIANTE. ....	19
<b>FIGURA 32.</b>	PROTOTIPO DE INTERFAZ: MOSTRAR ESTUDIANTES DE PREGRADO POR CARRERA. ....	20
<b>FIGURA 33.</b>	PROTOTIPO DE INTERFAZ: MOSTRAR ESTUDIANTES DE POSTGRADO POR CURSO. ....	20
<b>FIGURA 34.</b>	PROTOTIPO DE INTERFAZ: SOLICITAR CREACIÓN DE ESTUDIANTE DE POSTGRADO. ....	21
<b>FIGURA 35.</b>	PROTOTIPO DE INTERFAZ: SOLICITAR CREACIÓN DE CURSO DE POSTGRADO. ....	21
<b>FIGURA 36.</b>	DIAGRAMA DE CLASES DEL DISEÑO: INICIAR SESIÓN.....	22
<b>FIGURA 37.</b>	DIAGRAMA DE CLASES DEL DISEÑO: CERRAR SESIÓN.....	22
<b>FIGURA 38.</b>	DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR OPERADOR DE SIGENU.....	22
<b>FIGURA 39.</b>	DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR CURSO DE POSTGRADO. ....	23
<b>FIGURA 40.</b>	DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR CARRERA. ....	23
<b>FIGURA 41.</b>	DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR TIPO DE CARRERA. ....	24
<b>FIGURA 42.</b>	DIAGRAMA DE CLASES DEL DISEÑO: ELIMINAR USUARIO.....	24
<b>FIGURA 43.</b>	DIAGRAMA DE CLASES DEL DISEÑO: AUTORIZAR CURSO DE POSTGRADO. ....	25
<b>FIGURA 44.</b>	DIAGRAMA DE CLASES DEL DISEÑO: AUTORIZAR ESTUDIANTE DE POSTGRADO. ....	25
<b>FIGURA 45.</b>	DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR ESTADO DE SOLICITUD DE ESTUDIANTE DE POSTGRADO. ....	26
<b>FIGURA 46.</b>	DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR ESTADO DE SOLICITUD DE POSTGRADO. ....	26
<b>FIGURA 47.</b>	DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR SOLICITUDES DE CURSO DE POSTGRADO.....	27
<b>FIGURA 48.</b>	DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR SOLICITUDES DE POSTGRADO APROBADOS.....	27
<b>FIGURA 49.</b>	DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR SOLICITUDES DE ESTUDIANTE DE POSTGRADO. ....	28
<b>FIGURA 50.</b>	DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR SOLICITUDES DE ESTUDIANTE DE POSTGRADO APROBADAS.....	28
<b>FIGURA 51.</b>	DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR DATOS DE ESTUDIANTE. ....	29
<b>FIGURA 52.</b>	DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR ESTUDIANTE DE PREGRADO POR CARRERA. ....	29
<b>FIGURA 53.</b>	DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR ESTUDIANTE DE POSTGRADO POR CURSO.....	30
<b>FIGURA 54.</b>	DIAGRAMA DE CLASES DEL DISEÑO: SOLICITAR MODIFICACIÓN DE USUARIO DE DOMINIO.....	30
<b>FIGURA 55.</b>	DIAGRAMA DE CLASES DEL DISEÑO: SOLICITAR ELIMINACIÓN DE USUARIO DE DOMINIO. ....	31

<b>FIGURA 56.</b>	DIAGRAMA DE CLASES DEL DISEÑO: SOLICITAR CURSO DE POSTGRADO. ....	31
<b>FIGURA 57.</b>	DIAGRAMA DE CLASES DEL DISEÑO: SOLICITAR CURSO DE POSTGRADO. ....	32
<b>FIGURA 58.</b>	MODELO DE DATOS. ....	33

## ÍNDICE DE TABLAS

<b>TABLA 1.</b>	ACTORES DEL NEGOCIO. ....	25
<b>TABLA 2.</b>	TRABAJADORES DEL NEGOCIO. ....	26
<b>TABLA 3.</b>	DESCRIPCIÓN DEL CASO DE USO: RECIBIR USUARIO DE DOMINIO. ....	27
<b>TABLA 4.</b>	ACTORES DEL SISTEMA. ....	33
<b>TABLA 5.</b>	DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA INICIAR SESIÓN. ....	37
<b>TABLA 6.</b>	DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA CERRAR SESIÓN. ....	37
<b>TABLA 7.</b>	DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA GESTIONAR ESTUDIANTE DE PREGRADO. ....	38
<b>TABLA 8.</b>	DESCRIPCIÓN DE LOS COMPONENTES. ....	44

## INTRODUCCIÓN

La información se considera un bien de consumo y desarrollo que consecuentemente hay que gestionar, además de ser un recurso que es necesario aprovechar por contribuir al logro de los objetivos de cualquier organización, y por tanto a la elevación en el cumplimiento de sus resultados.

Los avances de la revolución informática se han convertido en una parte importante de la vida del hombre, permitiéndole lograr una mayor acumulación y organización de la información necesaria para su desarrollo. La gestión de la información une los factores que en cualquier organización interactúan con diferentes actitudes, intereses y expectativas, para acceder y usar productos y servicios de la información. Muchos de los sistemas informáticos han venido a perfeccionar procesos de gestión de la información alrededor del mundo, para de esta manera eliminar errores humanos que existen en torno a estos y permitir el almacenamiento de enormes cantidades de datos.

La principal causa de que se hiciera mejor y más rápida la gestión de la información en los últimos tiempos fue la interconexión de personas en todo el mundo permitida con la aparición de las redes informáticas y las posibilidades de comunicación que estas brindan. Con la creciente cantidad de redes informáticas, y por consiguiente de sistemas de gestión de información interconectados se generó la necesidad de la identificación de las personas que manipulan dichos sistemas, para lograr controlar el acceso y las acciones realizadas por cada una. Los usuarios vinieron a lograr la representatividad de las personas en el mundo digital con la autenticación.

La autenticación de un objeto puede significar la confirmación de su procedencia, la de una persona o usuario a menudo consiste en verificar su identidad. La autenticación de usuarios es indispensable hoy en día con el fin de evitar o minimizar las suplantaciones de identidad. (Pérez Pérez & Morales Luna, 2010)

La mayoría de las empresas preocupadas por la seguridad hoy implementan alguna forma de autenticación y autorización de acceso a la red y los recursos. Los beneficios de este enfoque son claros: los permisos de usuario se pueden verificar antes de otorgar acceso a los recursos, y la actividad del usuario puede ser monitoreada a través de diversos mecanismos de registro.

La gestión de usuarios permite a los administradores controlar la frecuencia de la autenticación de usuario, para requerir autenticación más frecuente si el usuario está accediendo recursos críticos (Blue Coat Systems, Inc, 2007).

Las redes informáticas y la gestión de usuarios se han extendido a todas las ramas de la sociedad debido a sus grandes potencialidades. En Cuba se han llevado a cabo transformaciones con la implementación de dominios en redes locales en todos los ámbitos empresariales dando respuesta al desarrollo socio-económico nacional.

Actualmente existen instaladas y configuradas en todas las universidades del país por la facilitación que aportan al proceso de enseñanza-aprendizaje, redes locales con dominios manejados por el AD (Directorio Activo, del inglés *Active Directory*) de Windows Server 2003.

“El AD utiliza servicios estándares propios de redes IP como es el caso de LDAP (Protocolo Ligero de Acceso a Directorio, del inglés *Lightweight Directory Access Protocol*), para almacenar la información referente a los recursos que controla”. (Sánchez B. &, 2010)

LDAP permite el control del acceso a la red de redes, así como la centralización de la información referente a los usuarios de dominio. Debido a esto se ha hecho necesario el completo conocimiento por parte de las personas que gestionan los usuarios de dominio en la UNISS de las características de este protocolo para así lograr la correcta gestión de la información que manipulan.

La Universidad José Martí y Pérez de Sancti Spíritus (UNISS) no ha quedado fuera de esta transformación, ya que cuenta con un dominio de red local que regula y controla el acceso a los medios informáticos, como son las computadoras, red, y toda clase de servicios implementados. Esta red instalada en el centro beneficia tanto a trabajadores como estudiantes con los servicios implementados, ya que estos permiten, no solo la comunicación entre personas de la UNISS, sino también la adquisición rápida y viable de información necesaria para realizar sus labores.

Con el desarrollo de la presente investigación se comprobó que como todos los cursos a la UNISS ingresa una gran cantidad de estudiantes de pregrado la gestión de los usuarios de dominio se convierte en un proceso largo y engorroso debido a que se realiza de forma manual y personalizada, y requiere la inserción de enormes cantidades de datos personales que por demás son importantes para realizar los documentos oficiales que en ella se emiten. Esta entrada de datos en el AD se realiza manualmente, a pesar que existe en la UNISS un sistema

gestor de matrícula (SIGENU) que registra en una base de datos toda la información personal de sus estudiantes de pregrado.

Otro de los problemas encontrados es la creación de usuarios de dominio para aquellos estudiantes que lo son temporalmente, dígame los que matriculan en los cursos de postgrado. Estos estudiantes también necesitan acceder a los servicios que se brindan en la UNISS y actualmente no cuentan con los respectivos usuarios de dominio que se lo permitan. Acceden a través de un usuario genérico que impide la correcta visualización de los sucesos o eventos desarrollados individualmente, y atenta contra la seguridad.

Tomando en cuenta lo antes expuesto se plantea el siguiente **problema de investigación**:  
¿Cómo optimizar el proceso de gestión de usuarios de dominio para los estudiantes en la “Universidad José Martí Pérez de Sancti Spíritus”?

El **objetivo general** es entonces desarrollar un software que optimice la gestión de usuarios de dominio para los estudiantes en la UNISS.

En el cumplimiento del objetivo general se realizan las siguientes **tareas de investigación**:

1. ¿Qué fundamentos teóricos y metodológicos sustentan la elaboración un software optimice la gestión de usuarios de dominio para los estudiantes en la UNISS?
2. ¿Cómo diseñar un software que optimice la gestión de usuarios de dominio para los estudiantes en la UNISS?
3. ¿Cómo implementar un software que optimice la gestión de usuarios de dominio para los estudiantes en la UNISS?

En el cumplimiento a las preguntas científicas se plantean las siguientes tareas de investigación:

1. Determinar los fundamentos teóricos y metodológicos que sustentan la elaboración de un software que optimice la gestión de usuarios de dominio.
2. Diseñar un software que optimice la gestión de usuarios de dominio para los estudiantes en la UNISS.
3. Implementar de un software que optimice la gestión de usuarios de dominio para los estudiantes en la UNISS.

El informe se estructura de la siguiente manera: introducción, desarrollo dividido en tres capítulos, conclusiones generales, recomendaciones, bibliografías y anexos.

**Capítulo I** (Fundamentación teórico y metodológica que sustenta la eficiencia de una aplicación web para optimizar la gestión de usuarios). El análisis del proceso actual en el centro y qué causas originaron la situación problemática es otro de los puntos incluidos además de la explicación de los motivos que condujeron a desarrollar la variante seleccionada, así como descripciones de las metodologías y herramientas utilizadas en las diferentes etapas de desarrollo.

**Capítulo II** (Descripción de la aplicación web para optimizar la gestión de usuarios de los estudiantes en la UNISS): se describe de modo general el funcionamiento del sistema y se definen los requerimientos funcionales y no funcionales. Con el empleo de la metodología RUP se identifican los procesos de negocio, se describe el modelo del negocio y se realiza la descripción de los casos de uso presentes en cada etapa de desarrollo.

**Capítulo III** (Implementación de la solución propuesta para optimizar la gestión de usuarios para los estudiantes en la UNISS): durante este capítulo se describe el diagrama de clases del diseño, el diagrama de las clases persistentes, el modelo de datos, los diagramas de despliegue y de componentes.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICO Y METODOLÓGICA QUE SUSTENTA LA EFICIENCIA DE UN SOFTWARE PARA OPTIMIZAR LA GESTIÓN DE USUARIOS DE DOMINIO.**

### **1.1 INTRODUCCIÓN**

En el presente capítulo se ofrece la fundamentación teórica y metodológica sobre la gestión de usuarios además de analizar a fondo las causas que dan lugar a la situación problemática y describir los conceptos asociados. También se realiza un análisis sobre las tecnologías, lenguajes y herramientas que van a ser las utilizadas en el proceso de desarrollo del sistema.

### **1.2 GESTIÓN DE USUARIOS.**

Gestión es un concepto más genérico que administración. Las personas que tienen responsabilidades de conducción, tienen que planificar y ejecutar el plan. El concepto gestión, connota tanto las acciones de planificar como las de administrar. La administración, como ejecución de las instrucciones de un plan, independientes de los contextos, no es lo que ocurre en las situaciones reales. En el objeto de la gestión se determinan otras unidades de gestión (más pequeñas que el conjunto del sistema) las que se caracterizan por tener la competencia de determinar objetivos propios, y a los cuales se pueden asignar recursos. (Casassus, 2000)

La gestión puede definirse como un proceso sistemático e integrador de coordinación de las actividades de adquisición, creación, almacenaje y difusión del conocimiento por individuos y grupos con objeto de conseguir los objetivos de la organización. (Perona & Velásquez, 2012)

Gestionar tiene que ver precisamente con diseñar, establecer y sostener pautas y reglas de juego que permitan que cada uno pueda hacer lo que le parece y que el producto sea bueno para todos. (Blejmar, 2005)

La gestión de usuarios con su característica autenticación proporciona a los administradores de las redes informáticas la capacidad de identificar y controlar el estado de los usuarios conectados a la red. La mayoría de las empresas preocupadas por la seguridad hoy implementan alguna forma de autenticación y autorización de acceso a la red y los recursos. Los beneficios de este enfoque son claros: los permisos de usuario se pueden verificar antes de otorgar acceso a los recursos, y la actividad del usuario puede ser monitoreada a través de diversos mecanismos de registro. (Blue Coat Systems, Inc, 2007)



La gestión de usuarios permite a los administradores controlar la frecuencia de la autenticación de usuario, para requerir autenticación más frecuente si el usuario está accediendo recursos críticos. Este tipo de flexibilidad permite a los administradores implementar la autenticación basada en políticas a la altura de sus políticas de seguridad de red. Otro beneficio clave de la gestión de usuarios es la visibilidad de las sesiones de usuario activas, los administradores pueden ver todos los usuarios activos y filtrar los datos de visualización del usuario como la dirección IP o dominio. Esto puede ser útil para identificar el estado de conexión general de los usuarios y para tomar decisiones en tiempo real como cerrar inmediatamente la sesión de un usuario. (Blue Coat Systems, Inc, 2007)

### 1.2.1 EL ACTIVE DIRECTORY

“*Active Directory* (AD) almacena información acerca de los objetos en una red y pone esta información a disposición de los usuarios y administradores de red. AD proporciona a los usuarios de la red el acceso a los recursos permitidos en cualquier lugar de la red mediante un proceso de inicio de sesión único. Proporciona a los administradores de red una vista jerárquica intuitiva de la red y un único punto de administración para todos los objetos.” (Microsoft Windows, 2003)

Tanto la fuente de información de AD como el servicio que hace que la información esté disponible y utilizable dependen del servicio de directorio. Un servicio de directorio permite al usuario encontrar un objeto dado cualquiera de sus atributos. El acceso al AD se realiza mediante el Protocolo de Acceso Ligero a Directorios (LDAP, del inglés *Lightweight Directory Access Protocol*). Como su nombre implica, LDAP está diseñado como un método eficiente de acceso a los servicios de directorio sin la complejidad de otros protocolos de servicio de directorio. LDAP define qué operaciones se pueden realizar para consultar y modificar la información de un directorio y cómo la información en un directorio se puede acceder con seguridad. Puede usar LDAP para encontrar o enumerar los objetos de directorio y para consultar o administrar AD. (Microsoft Windows, 2003)

### 1.2.2 LA GESTIÓN DE USUARIOS CON LDAP.

LDAP está diseñado para proveer acceso al Directorio X.500 a la vez que no utiliza todos los recursos requeridos por el Protocolo de Acceso a Directorio (DAP, por sus siglas en inglés). Este protocolo está específicamente orientado a aplicaciones simples de gestión y buscadores de aplicaciones que provean accesos sencillos de lectura/escritura interactiva al directorio X.500.

LDAP fue desarrollado como un método de consolidación de autorización, autenticación y acceso (AAA o Triple A) a la información. Por sí mismo, esto es útil, ya que usted está manteniendo toda la información en un solo lugar en vez de muchos. Esto no sólo hace que la vida de los desarrolladores de aplicaciones sea más fácil, sino que también elimina la posibilidad de aplicaciones falsas o usuarios sin pasar por las reglas de la AAA para acceder y modificar directamente el contenido del directorio, excepto, por supuesto, a través de compromisos de seguridad tradicionales.

### 1.2.3 LA GESTIÓN DE USUARIOS EN LA UNISS.

En la UNISS existe implementado un dominio de red controlado por el AD del sistema operativo Windows Server 2003. Dicho dominio de red trabaja directamente con el protocolo LDAP y se manipula con las aplicaciones de escritorio que están implementadas en ese sistema operativo desarrollado por la Microsoft. El dominio de red consta con un árbol de directorio en el que se encuentran organizados los usuarios de la red en distintas unidades organizacionales o carpetas según su categoría y jerarquía dentro de la UNISS. Por ejemplo: existe una unidad organizacional o carpeta en la que se encuentran los distintos lugares o edificios representados en el dominio de red. Dentro existen otras carpetas en las que se distingue entre los profesores, estudiantes, el personal no docente, la municipalización y el rectorado. Así sucesivamente va descendiendo el árbol hasta que un estudiante por ejemplo, se encuentra en una carpeta que representa el año en el que estudia, la que a su vez está dentro de otra que representa su carrera, y esta última en la carpeta de estudiantes.

Como ejemplo de un edificio en el dominio de red actual tenemos el **Docente I**, cuya distribución se presenta a continuación:

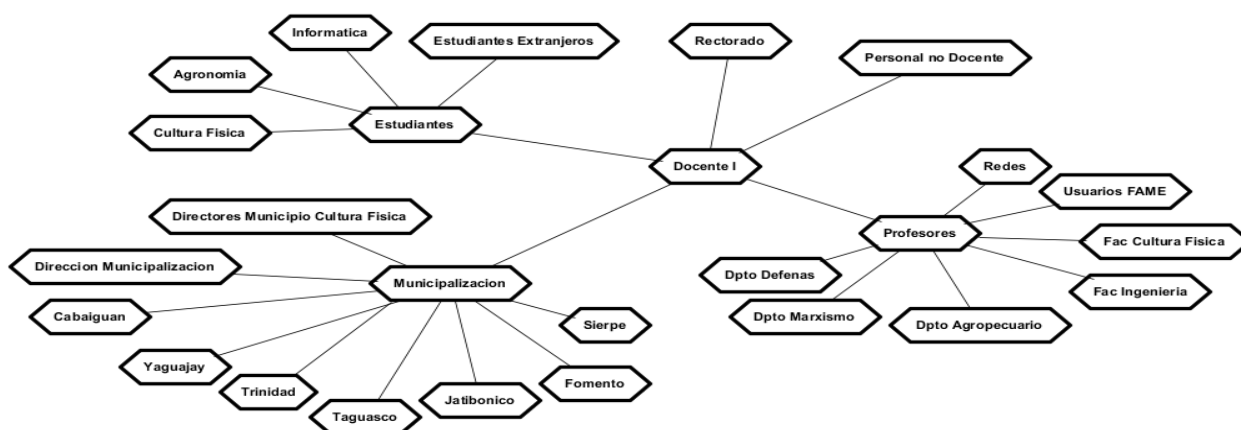


Figura 1. DISTRIBUCIÓN DEL EDIFICIO DOCENTE I EN DEL ÁRBOL DE DIRECTORIO.

### 1.3 SIGENU

En el Ministerio de educación Superior desde los finales de los años 90 se trabaja en una nueva etapa de la informatización de la gestión universitaria en general y en la gestión del proceso docente en particular. El Proyecto del Sistema de Gestión de la Nueva Universidad (SIGENU), constituye tal vez el mayor empeño en este sentido. Este proyecto ha permitido la creación de un mercado de datos sobre el cual opera un servidor OLAP que abre ante los especialistas y ejecutivos de la Organización amplísimas posibilidades. (Viña Brito, 2013)

El SIGENU se encuentra instalado en la UNISS, y entre otras cosas controla los datos de todos los estudiantes matriculados. Datos que son insertados al inicio de cada curso escolar y a partir de los cuales se crean los documentos oficiales emitidos en el centro de altos estudios. Datos que además son los utilizados para la creación de los usuarios de dominio de los estudiantes de pregrado. SIGENU trabaja con una base de datos de postgresQL, que es donde guarda los datos de la gestión de los estudiantes.

### 1.4 ¿APLICACIÓN WEB O DE ESCRITORIO?

Debido a que las utilidades para gestionar usuarios de dominio que provee Windows Server 2003 son aplicaciones de escritorio pudiera pensarse que una aplicación de escritorio sería la solución.

En el escenario de la UNISS y como ya se hizo referencia en la introducción se espera que, además de los usuarios de pregrado que se crean a partir de los datos de SIGENU, estudiantes de postgrado puedan solicitar la creación de usuarios.

Sería viable que cuando una persona solicite un usuario o su modificación pudiera tener acceso a formularios en línea para introducir sus datos. Luego, las distintas instancias de aprobación tan solo deberían leer los datos y autorizarlos. El mismo sistema que maneje estos formularios crearía los usuarios, una vez aprobados con los datos suministrados directamente por el solicitante anulando errores de transcripción.

Una aplicación de escritorio requeriría su instalación en cada una de las computadoras desde donde se pudiera hacer una solicitud, lo cual pudiera implicar todas las computadoras conectadas a la red de la UNISS.

Teniendo en cuenta que aplicación web es un software que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En

otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. En este caso tan solo se requeriría de un navegador como cliente, facilitando el acceso desde cualquier punto de la red, independientemente del sistema operativo o navegador utilizado.

Es importante mencionar que una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo, que es lo que se conoce como la Web 2.0. Con los beneficios que nos brinda la Web 2.0 se puede lograr que el usuario interactúe de manera más fácil con la aplicación.

#### 1.4.1 MODELO CLIENTE-SERVIDOR

Existen diversos puntos de vista sobre la manera en que debería efectuarse el procesamiento de datos. En este contexto, es necesario establecer una infraestructura de Procesamiento de información, que cuente con los elementos requeridos para proveer información adecuada, exacta y oportuna en la toma de decisiones y para proporcionar un mejor servicio a los clientes y ciudadanos. El modelo Cliente/Servidor reúne las características necesarias para proveer esta infraestructura, independientemente del tamaño y complejidad de las operaciones de las organizaciones públicas o privadas y, consecuentemente desempeña un papel importante en este proceso de evolución.

IBM define al modelo Cliente/Servidor de la siguiente manera: “Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores” La arquitectura tradicional de cliente/servidor también es conocida como arquitectura de dos capas. La cual facilita la integración entre sistemas diferentes y comparte información. Además de permitir un rápido mantenimiento y desarrollo de aplicaciones, contribuye a proporcionar, a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información relevante a nivel global.

### 1.4.2 ARQUITECTURA EN 3 CAPAS

Cada capa es un proceso separado y bien definido corriendo en plataformas separadas. En la arquitectura tradicional de tres capas se instala una interfaz de usuario en la computadora del usuario final (el cliente). La arquitectura basada en Web transforma la interfaz de búsqueda existente (el explorador de Web), en la interfaz del usuario final. Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario. El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular y gracias a que cada capa es independiente de la otra los cambios o actualizaciones pueden ser realizados sin afectar la aplicación como un todo. (Cordero Carrasco, 2006)

### 1.5 FRAMEWORK SYMFONY2

Un patrón es una solución común a un problema común en un contexto dado. Un mecanismo es un patrón de diseño que aplica a una sociedad de clases. Un *framework* es un modelo arquitectónico que provee una plantilla extensible para aplicaciones dentro de un dominio. Los patrones se usan para especificar los mecanismos y *frameworks* que forman la arquitectura de su sistema. (Booch, Rumbaugh, & Jacobson, 2000).

“Un *framework* es más grande que un mecanismo. De hecho, usted puede pensar en un framework como un tipo de micro-arquitectura que abarca un juego de mecanismos que trabajan juntos para resolver un problema común para un dominio común. Cuando se especifica un framework se especifica el esqueleto de una arquitectura. Los mejores *frameworks* no se diseñan en el aislamiento; hacer eso es una garantía de fallo. Más bien, los mejores *frameworks* son extraídos de arquitecturas existentes que se demuestra que funcionan y evolucionan para ser adaptables a otros dominios.” (Booch, Rumbaugh, & Jacobson, 2000)

Symfony2 es la versión más reciente de Symfony, el popular *framework* para desarrollar aplicaciones PHP. Se anunció por primera vez a principios de 2009 y supone un cambio radical tanto en arquitectura interna como en filosofía de trabajo respecto a sus versiones anteriores. Symfony2 ha sido ideado para exprimir al límite todas las nuevas características de PHP 5.3 y por eso es uno de los *frameworks* PHP con mejor rendimiento. Su arquitectura interna está completamente desacoplada, lo que permite reemplazar o eliminar fácilmente aquellas partes que no encajan en tu proyecto. Symfony2 también es el *framework* que más ideas incorpora del resto de *frameworks*, incluso de aquellos que no están programados con PHP. (Eguiluz, 2011)

Por todas las funcionalidades que brinda se seleccionó el *framework* Symfony2 para el desarrollo de la aplicación web en cuestión. A continuación una explicación más detallada de dichas funcionalidades.

### 1.5.1 MODELO-VISTA-CONTROLADOR

Symfony2 basa su funcionamiento interno en la arquitectura Modelo-Vista-Controlador (MVC).

Esta arquitectura fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas. Definición de las partes (Catalani, 2007):

- ✓ El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos.
- ✓ 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.
- ✓ El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Sirve como un intermediario entre el Modelo y la Vista.

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos. Ventajas de este patrón según (Martínez, 2007):

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

### 1.5.2 PHP 5.3

“PHP (acrónimo de *Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP de algo del lado cliente como Javascript, es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá los resultados de ejecutar el *script*, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los archivos HTML con PHP y entonces no hay manera que los usuarios puedan saber que tienes debajo de la manga.” (Manual de PHP, 2013)

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Esto incluye cualquier servidor web que pueda utilizar el binario PHP de FastCGI, como lighttpd y nginx. PHP funciona ya sea como un módulo, o como un procesador de CGI. De modo que, con PHP tiene la libertad de elegir el sistema operativo y el servidor web de su gusto. Además, tiene la posibilidad de utilizar programación por procedimientos, programación orientada a objetos (POO), o una mezcla de ambas.

Una de las características más potentes y destacables de PHP es su soporte para una gran cantidad de bases de datos. Escribir una página web con acceso habilitado a una base de datos es increíblemente simple utilizando una de las extensiones específicas (por ejemplo, para MySQL), o utilizar una capa de abstracción como PDO, o conectarse a cualquier base de datos que soporte el estándar de Conexión Abierta a Bases de Datos por medio de la extensión ODBC. PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear sockets puros e interactuar usando cualquier otro protocolo. PHP soporta WDDX para el intercambio de datos entre lenguajes de programación en web. Y hablando de interconexión, PHP puede utilizar objetos Java de forma transparente como objetos de PHP. (Manual de PHP, 2013)

De todas las novedades introducidas por PHP 5.3, las más relevantes para los programadores de Symfony2 son las funciones anónimas y los *namespaces*.

### 1.5.1.1 FUNCIONES ANÓNIMAS

Las funciones anónimas, también conocidas como *closures*, son funciones sin nombre que normalmente se utilizan para crear fácil y rápidamente un *callback*. El código fuente de Symfony2 hace un uso extensivo de estas funciones. El código interno de una función anónima no tiene acceso a ninguna variable de la aplicación. (Eguiluz, 2011)

### 1.5.1.2 NAMESPACES

En la práctica, los *namespaces* se utilizan para estructurar mejor el código fuente de la aplicación. Todas las clases de Symfony2 utilizan los *namespaces*. Antes de que existieran los *namespaces*, las aplicaciones debían ser cuidadosas al elegir el nombre de sus clases, ya que dos o más clases diferentes no podían tener el mismo nombre. Si la aplicación contenía cientos de clases, como es habitual en los *frameworks*, el resultado eran clases con nombres larguísimos para evitar colisiones. Gracias a los *namespaces* dos o más clases de una misma aplicación pueden compartir su nombre. El único requisito es que sus *namespaces* sean diferentes, de forma que la aplicación sepa en todo momento cuál se está utilizando. (Eguiluz, 2011)

### 1.5.1.3 ANOTACIONES

Las anotaciones son un mecanismo muy utilizado en lenguajes de programación como Java. Aunque la versión 5.3 de PHP todavía no soporta anotaciones, las aplicaciones Symfony2 pueden hacer uso de ellas gracias a una librería desarrollada por el proyecto Doctrine2. Técnicamente las anotaciones no son más que comentarios incluidos en el propio código fuente de la aplicación. A diferencia de los comentarios normales, las anotaciones no sólo no se ignoran, sino que se tienen en cuenta y pueden influir en la ejecución del código. (Eguiluz, 2011)

### 1.5.3 DOCTRINE

Una de las tareas más comunes y desafiantes para cualquier aplicación involucra la persistencia y lectura de información hacia y desde una base de datos. Afortunadamente, Symfony2 viene integrado con Doctrine, una biblioteca, cuyo único objetivo es dotarte de poderosas herramientas para facilitarte eso. Herramienta que te permite asociar objetos a una base de datos relacional (tal como MySQL, PostgreSQL o Microsoft SQL), además de trabajar con bases de datos de una manera mucho más interesante que solo recuperar filas de una tabla basada en columnas de un arreglo. En cambio, Doctrine te permite persistir objetos



completos a la base de datos y recuperar objetos completos desde la base de datos. (Doctrine Project Team, 2012)

Doctrine 2 es un mapeador objeto relacional (ORM) para PHP 5.3.0+ que mantiene la persistencia transparente de los objetos de PHP. Se encuentra encima de una poderosa Capa de Abstracción de Base de Datos (DBAL). La tarea primaria es la traducción transparente entre los objetos y las filas de la base de datos relacionales. Una de las claves de Doctrine es la opción de escribir las consultas a la base de datos en un dialecto SQL orientado a objetos llamado *Doctrine Query Language* (DQL por sus siglas en inglés, Lenguaje de Consultas de Doctrine), inspirado por Hibernate HQL que permite a los desarrolladores escribir poderosas consultas en un modo simple y flexible. (Doctrine Project Team, 2012)

#### 1.5.4 TWIG

Symfony2 contiene un lenguaje de plantillas aún más potente llamado Twig, que te permite escribir plantillas concisas y fáciles de leer que son más amigables para los diseñadores web y, de varias maneras, más poderosas que las plantillas PHP. Además de ser mucho más limpias y concisas, las plantillas de Twig son seguras por defecto, por lo que no debes aplicar el mecanismo de escape al valor de las variables. (Twig - The flexible, fast, and secure PHP template engine, 2013)

Twig es un motor y lenguaje de plantillas para PHP muy rápido y eficiente. Además, al ejecutar la aplicación, las plantillas de Twig se compilan a código PHP nativo, por lo que el rendimiento y el consumo de memoria es similar al de las plantillas PHP. (Eguiluz, 2011)

Las plantillas Twig están destinadas a ser simples y no procesar etiquetas PHP. Esto es por diseño: el sistema de plantillas Twig está destinado a expresar la presentación, no la lógica del programa. Twig también puede hacer cosas que PHP no puede, como controlar el espacio en blanco, cuenta con un recinto de seguridad, escape de salida automática y contextual e incluye funciones personalizadas y filtros que sólo afectan a las plantillas. (Twig - The flexible, fast, and secure PHP template engine, 2013)

## 1.6 METODOLOGÍAS UTILIZADAS PARA EL DISEÑO.

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

y herramientas de desarrollo que hagan posible su adecuada realización y que garantice el cumplimiento de todas sus funcionalidades.

### 1.6.1 LENGUAJE UNIFICADO DE MODELADO (UML).

“El Lenguaje Unificado de Modelado (UML, del inglés *Unified Modeling Language*) es un lenguaje estándar para escribir planos de software, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema. UML permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados.” (Booch, Rumbaugh, & Jacobson, 2000)

Según (Hernández Orallo, 2009) UML es “ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos.” Esto último es el objetivo de las metodologías de desarrollo. Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

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

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware. (Hernández Orallo, 2009)

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

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

### 1.6.2 METODOLOGÍA RUP.

Es un proceso basado en componentes y utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. No obstante, los verdaderos aspectos definitorios de RUP se resumen en tres frases claves: está dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental (Booch, Rumbaugh, & Jacobson, 2000)

Es un proceso de desarrollo de software, en otras palabras, es un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Es un marco de trabajo genérico que puede especializarse, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. (Booch, Rumbaugh, & Jacobson, 2000)

## 1.7 OTROS LENGUAJES QUE SERÁN UTILIZADOS.

### 1.7.1 JAVASCRIPT

JavaScript permite insertar código especial dentro del HTML de una página, su función es ampliar las posibilidades de HTML. “La ventaja fundamental de JavaScript es que su aprendizaje y uso son muy sencillos, y que permite realizar labores muy complejas en una página sin necesidad de aprender CGI” (Sánchez J. , 2003)

“Permite ejecutar secuencias de comandos en el mismo navegador del usuario. Con JavaScript se pueden realizar cálculos rápidos y complejos y hasta controlar la mayor parte de los elementos del navegador. También se pueden ejecutar acciones como abrir ventanas, verificar formularios antes de enviarlos, convertir divisas y crear calendarios.” (Hernán Ruiz, 2006)

(Álvarez, 2009) Afirma que “es un lenguaje interpretado, no requiere compilación. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias.” La mayoría de los navegadores en sus últimas versiones interpretan código Java Script. El código JavaScript puede ser integrado dentro de las

páginas web. Los script tienen capacidades limitadas, por razones de seguridad, el código visible por cualquier usuario y el código deben descargarse completamente.

### 1.7.2 HTML

“HTML significa *HiperText Mackup Language*, cuya traducción sería Lenguaje de etiquetas e hipertexto, en donde hipertexto hace referencia a la capacidad del lenguaje para la utilización de los conocidos hipervínculos. Este lenguaje está compuesto por etiquetas o marcas, y gracias a ellas es posible dar forma a todos los componentes de una página. HTML no es propiamente un lenguaje de programación sino un sistema de etiquetas, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizará en la forma como éste lo entienda. El entorno para trabajar HTML es simplemente un procesador de texto. Estos documentos pueden ser mostrados por los visores o navegadores de páginas Web en Internet, como Microsoft Internet Explorer.” (Hernán Ruiz, 2006)

## 1.8 HERRAMIENTAS DE DESARROLLO.

### 1.8.1 NETBEANS

El IDE NetBeans es un entorno de desarrollo integrado una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans.

El IDE NetBeans es un producto libre y gratuito sin restricciones de uso y de amplias comodidades para el programador. Para la realización de este proyecto se utilizó el IDE 7.0.2, una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero funciona perfectamente para la escritura de PHP ya que presenta un buen completamiento de código y reconoce todas las funciones y estructuras de este lenguaje.

### 1.8.2 VISUAL PARADIGM

*Visual Paradigm* es una herramienta CASE (Ingeniería del Software Asistida por Computadoras, del inglés *Computer Assisted Software Engineering*) para el modelado visual de sistemas software mediante UML. Permite especificar, analizar y diseñar el sistema antes de codificarlo. Mantiene la consistencia de los modelos del sistema con el software. Posibilita la generación de documentación automáticamente. Soporta Ingeniería Inversa, generación de código, importa proyectos de *Rational Rose*, generador informes, etc.

### 1.8.3 RATIONAL ROSE

Para apoyar el trabajo de la metodología RUP, ha sido desarrollada por la Compañía norteamericana *Rational Corporation* la herramienta CASE *Rational Rose* desde el año 2000. Esta herramienta integra todos los elementos que propone la metodología para cubrir el ciclo de vida de un proyecto. Esta herramienta CASE propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. “Rational Rose proporciona también mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.” (Boggs & Boggs, 2002)

## 1.9 SISTEMAS GESTORES DE BASE DE DATOS.

### 1.9.1 POSTGRESQL

PostgreSQL es un Sistema Gestor de Bases de Datos (DBMS, del inglés *Database Management System*) relacional orientado a objetos y libre. Como muchos otros proyectos de código abierto el desarrollo de PostgreSQL no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. (Worsley & Joshua D, 2002)

Según (Worsley & Joshua D, 2002) algunas de sus principales características son:

- Está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.
- Alta concurrencia: Mediante un sistema denominado Acceso Concurrente Multiversión (MVCC por sus siglas en inglés, Multiversion Concurrency Control) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit* (acción de cometer). Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases de datos, eliminando la necesidad del uso de bloqueos explícitos.

- Amplia variedad de tipos nativos: Provee nativamente soporte para números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), protocolos de internet (IP por sus siglas en inglés, Internet Protocol) (IPv4 e IPv6), bloques de direcciones al estilo de enrutamientos entre dominios sin clases (CIDR por sus siglas en inglés, *Classless Inter-Domain Routing*), direcciones de control de acceso al medio (MAC por sus siglas en inglés, *Media Access Control*), arreglos y adicionalmente los usuarios pueden crear sus propios tipos de datos.

### 1.9.2 MYSQL

“MySQL Server se desarrolló originalmente para tratar grandes bases de datos mucho más rápido que soluciones existentes y ha sido usado con éxito en entornos de producción de alto rendimiento durante varios años. MySQL Server ofrece hoy en día una gran cantidad de funciones. Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet.” (MySQL 5.0 Reference Manual, 2011)

A continuación se muestran las características (MySQL 5.0 Reference Manual, 2011) de MySQL por las que fue seleccionado como sistema gestor de bases de datos para la implementación de la aplicación web:

- Probado con un amplio rango de compiladores diferentes.
- Funciona en diferentes plataformas.
- Un sistema de reserva de memoria muy rápido basado en threads.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Tablas *hash* en memoria, que son usadas como tablas temporales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (*linkado*) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.

- Soporte completo para las cláusulas SQL GROUP BY y ORDER BY. Soporte de funciones de agrupación (COUNT(), COUNT(DISTINCT ...), AVG(), STD(), SUM(), MAX(), MIN(), y GROUP\_CONCAT()).
- Soporte para LEFT OUTER JOIN y RIGHT OUTER JOIN cumpliendo estándares de sintaxis SQL y ODBC.
- Soporte para alias en tablas y columnas como lo requiere el estándar SQL.
- DELETE, INSERT, REPLACE, y UPDATE devuelven el número de filas que han cambiado (han sido afectadas). Es posible devolver el número de filas que serían afectadas usando un *flag* al conectar con el servidor.

### 1.10 CONCLUSIONES

En el capítulo que concluye se abordaron los fundamentos teóricos necesarios para la comprensión del tema que se trata y se describieron todos los conceptos asociados al problema, así como se abordó el porqué de la solución propuesta.

Se determinó que una aplicación web cubre en forma más completa las necesidades de la gestión de usuarios que una aplicación de escritorio.

Se justificaron las razones por las cuales fue seleccionada RUP como metodología a seguir en la documentación del software propuesto y en su proceso de desarrollo y como notación el Lenguaje de Modelado Unificado (UML). El gestor de base de datos seleccionado fue MySQL y el lenguaje de programación fue PHP con el *framework* Symfony2.

## CAPÍTULO 2: DESCRIPCIÓN DEL SOFTWARE PARA OPTIMIZAR LA GESTIÓN DE USUARIOS DE LOS ESTUDIANTES EN LA UNISS.

### 2.1 INTRODUCCIÓN

Tomando la metodología RUP como guía en el presente capítulo se exponen las reglas del negocio a tener en cuenta para el diseño del sistema. Se expone además una caracterización de los actores y trabajadores del mismo, así como sus casos de uso, conformándose el diagrama de casos de uso del negocio, la descripción de estos en formato expandido, y los diagramas de actividades y el modelo objetos. Se definen además los requerimientos funcionales y no funcionales, se describe de modo general el funcionamiento del sistema diseñado y se realiza el modelo de casos de uso del sistema, basado en los actores y los casos de uso.

### 2.2 IDENTIFICACIÓN DE LOS PROCESOS DEL NEGOCIO

Un elemento clave para lograr un desarrollo exitoso de un producto de software y una buena comunicación entre los desarrolladores, los clientes y el usuario final lo constituyen la identificación y modelamiento del flujo de los procesos que serán objeto de informatización. A este flujo de trabajo se le denomina: Modelación del Negocio.

El modelado del negocio es una técnica que permite comprender los procesos del negocio de la organización y se desarrolla en dos pasos:

Diagrama de caso de uso del negocio (Booch, Rumbaugh, & Jacobson, 2000):

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

Un proceso de negocio es un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y forma, que emplean los recursos de la organización para dar resultados que apoyen sus objetivos. (Hernández Orallo, 2009)

Tomando en cuenta el planteamiento anterior y luego de realizar un estudio en la UNISS se constató que en estos momentos el proceso de gestión de los usuarios de dominio para los estudiantes se realiza de forma manual y personalizada. No existe un software que automatice



la gestión de dichos usuarios de dominio, lo que impide la actualización rápida, eficiente y viable del AD. Por lo que este proceso es lento, proclive a errores e impide la rápida obtención de la información actualizada, lo que atenta contra la seguridad informática y el control interno.

Cada año ingresan a la UNISS varios cientos de estudiantes, a los cuales se les crea un usuario de dominio con su contraseña correspondiente para poder acceder a todos los servicios que ofrece la red del centro. Este trabajo tan complejo es elaborado por el administrador de red, quien valiéndose de un listado oficial de la matrícula emitido en la secretaría de la UNISS agrega dichos usuarios de dominio uno por uno. Teniendo en cuenta la cantidad de información manejada para cada usuario podemos constatar lo lento e ineficiente que es este proceso.

El proceso comienza cuando el operador del sistema gestor de matrículas de la UNISS (SIGENU) hace llegar al administrador de red un listado oficial con los nombres de los estudiantes de nuevo ingreso organizado por carreras. El administrador de red ejecuta el AD, el cual está montado en el sistema operativo Windows Server 2003. En este se guardan todos los usuarios existentes en la UNISS, además tiene los usuarios separados por grupos, es decir, Profesores, Estudiantes, Municipios, Personal no docente y Rectorado.

Como lo que se va a crear es un usuario de dominio para estudiantes de pregrado el administrador de red realiza los siguientes pasos:

1. Abre la carpeta Estudiantes, dentro de estas existen otras, como por ejemplo las diferentes carreras que se estudian en la UNISS.
2. Elige la carrera a la cual le va a agregar el usuario, y aparecen una serie de carpetas que indican los diferentes años de la carrera.
3. Si es un estudiante de nuevo ingreso, se escoge la carpeta de primer año y encima de esta se da clic derecho.
4. Elige la opción *new* y dentro de esta se elige *user*, entonces aparece una ventana donde el administrador de red introduce el nombre y dos apellidos del estudiante así como el nombre de usuario que le asigne. El nombre usuario es único para cada persona en el directorio activo, lo que dificulta la tarea de su selección, ya que en caso de existir el nombre de usuario seleccionado el administrador de red necesita entonces buscar una alternativa que corresponda con el nombre del estudiante y que no haya sido utilizada hasta ese momento.

5. Luego el sistema pide asignarle una contraseña a este usuario y una confirmación de la misma, seguido de esto, finaliza el proceso y el usuario quedará agregado en el directorio activo de la UNISS.

Una vez creados los usuarios de dominio para los estudiantes el administrador de red emite un listado con sus nombres y sus respectivos usuarios. Listado que utilizan los coordinadores de cada grupo de primer año para informar a sus estudiantes individualmente qué usuario les corresponde.

La contraseña que actualmente se crea es genérica para todos los estudiantes, es decir, al inicio de curso todos los estudiantes conocen la de los demás, y con solo saber el nombre de usuario de otro de ellos pueden apropiarse del usuario de dominio que le corresponda. Por esto además de ser extremadamente largo es que no es eficaz el proceso de creación de usuarios de dominio para los estudiantes de pregrado en la UNISS.

## 2.3 REGLAS DEL NEGOCIO

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. (Booch, Rumbaugh, & Jacobson, 2000)

- ✓ La persona encargada de gestionar la matrícula en la UNISS (con SIGENU) debe:
  - Emitir un listado con los nombres de los estudiantes de nuevo ingreso organizado por carreras.
- ✓ La persona encargada de gestionar los usuarios de dominio en la UNISS debe:
  - Recibir un listado con los nombres de los estudiantes de nuevo ingreso organizado por carreras para crear los respectivos usuarios de dominio.
  - Emitir un listado con los nombres de los estudiantes de nuevo ingreso y sus respectivos nombres de usuario de dominio organizado por carrera.
  - Informar cuál es la contraseña genérica que fue asignada a los usuarios creados.
  - Atender las solicitudes por parte de los usuarios de dominio de realizar cambios en sus respectivos perfiles.
  - Eliminar los usuarios de los estudiantes egresados de la UNISS.
- ✓ Los coordinadores de los grupos de primer año de la UNISS deben:

- Recibir el listado del grupo que ellos coordinan con los nombres de los estudiantes y sus respectivos nombres de usuario de dominio organizado por carrera para hacerle llegar a cada cual personalmente el suyo.

## 2.4 DEFICIENCIAS EN EL PROCESO ACTUAL DE GESTIÓN DE USUARIOS PARA LOS ESTUDIANTES EN LA UNISS:

- ✓ Lentitud en el proceso de creación de usuarios de dominio para los estudiantes debido a la gran cantidad de información manipulada por el administrador de red.
- ✓ Dificultad en la selección del nombre de usuario de dominio, ya que existen muchos nombres repetidos y eso obliga al administrador de red a tener que buscar una alternativa que se asemeje al nombre del estudiante.
- ✓ Los estudiantes de postgrado no cuentan con sus propios usuarios de dominio para acceder a la red. Realizan el acceso a través de un usuario genérico, lo que impide la correcta supervisión de las acciones realizadas individualmente y atenta contra la seguridad informática.
- ✓ Los usuarios de dominio son creados con los datos provenientes del SIGENU, datos con los que se llenarán sus títulos y toda clase de documento emitido por la UNISS. Estos datos muchas veces tienen errores y los estudiantes no pueden consultar sus datos para verificar su veracidad.
- ✓ Los usuarios de dominio se crean con una contraseña genérica, lo que implica que al inicio de curso todos los estudiantes conocen la contraseña de los demás, y con solo saber el nombre de usuario de dominio de un compañero pueden apropiarse del usuario que le corresponde.

## 2.5 MEJORAS PROPUESTAS AL PROCESO DE.

- M\_1. Automatizar el proceso de emisión del listado oficial con los nombres de los estudiantes de nuevo ingreso.
- M\_2. Automatizar el proceso de selección de nombres de usuario.
- M\_3. Automatizar el proceso inserción de datos en el directorio activo.
- M\_4. Automatizar el proceso de visualización de datos personales.
- M\_5. Creación de contraseñas aleatorias para cada usuario de dominio.
- M\_6. Creación de usuarios de dominio para los estudiantes de postgrado.

## 2.6 MODELO DE CASOS DE USO DEL NEGOCIO

Modelar es una parte central de todas las actividades que llevan al despliegue de un buen software. Construimos modelos para para comunicar la estructura y conducta deseada de nuestro sistema. Construimos a modelos visualizar y controlar la arquitectura del sistema, para entender mejor el sistema que estamos construyendo, exponiendo a menudo las oportunidades para la simplificación y reutilización. Construimos modelos para manejar el riesgo. (Booch, Rumbaugh, & Jacobson, 2000)

“Un modelo proporciona los planos detallados de un sistema, así como planos más generales del sistema en cuestión. Un buen modelo incluye esos elementos que tienen amplio efecto y omiten esos elementos menores que no son relevantes al nivel de abstracción dado. Cada sistema puede describirse de diferentes puntos de vista usando diferentes modelos, y cada modelo es por consiguiente una abstracción semánticamente cerrada del sistema. Un modelo puede ser estructural, haciendo énfasis en la organización del sistema, o puede ser conductual, dando énfasis a la dinámica del sistema.” (Booch, Rumbaugh, & Jacobson, 2000)

### 2.6.1 ACTORES DEL NEGOCIO

“Un actor representa un coherente juego de roles que los usuarios de casos de uso juegan cuando interactúan con éstos casos de uso. Típicamente, un actor representa el papel que un humano, un dispositivo del hardware, o incluso otro sistema juega con un sistema. Una instancia de un actor representa por consiguiente a un individuo que actúa recíprocamente con el sistema de una manera específica. Por ejemplo, modelando un banco, procesar un préstamo involucra, entre otras cosas, la interacción entre cliente y un funcionario del préstamo.” (Booch, Rumbaugh, & Jacobson, 2000)

Los actores del negocio se listan a continuación:

Tabla 1. ACTORES DEL NEGOCIO

Actor	Justificación
estudiante	Persona que ha aprobado los exámenes de ingreso para cursar una carrera de las que se imparten en la UNISS y está correctamente matriculado en el SIGENU.

## 2.6.2 TRABAJADORES DEL NEGOCIO

Un trabajador es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores y manipulando entidades (Booch, Rumbaugh, & Jacobson, 2000)

Tabla 2. TRABAJADORES DEL NEGOCIO

Trabajador	Justificación
Operador de SIGENU	Persona responsable de emitir el listado oficial con los estudiantes de nuevo ingreso (obtenidos del sistema gestor de matrícula SIGENU) organizado por carreras.
Administrador de Red	La persona responsable de la gestión de los usuarios de dominio en la UNISS.
Coordinador de Grupo	Persona responsable de informar a cada estudiante de su grupo personalmente el usuario de dominio que le corresponde.

## 2.6.3 CASO DE USO DEL NEGOCIO

- Recibir usuario de dominio.

## 2.6.4 DIAGRAMA DE CASOS DE USO DEL NEGOCIO

Un diagrama de caso de uso muestra un juego de casos de uso y actores (un tipo especial de clase) y sus relaciones. “Usted aplica los diagramas de casos de uso para ilustrar la vista estática de caso de uso de un sistema. Los diagramas de casos de uso son especialmente importantes organizando y modelando las conductas de un sistema.” (Booch, Rumbaugh, & Jacobson, 2000)

El flujo de trabajo de modelamiento del negocio nos enseña a describir el negocio actual y a modelar el negocio propuesto, da una visión de qué es necesario hacer para dar respuesta a la solicitud del usuario.

Con esta actividad se pretende identificar los procesos en el negocio, definir las fronteras del negocio que van a modelarse, quién y qué interactuarán con el negocio y crear diagramas del modelo de casos de uso del negocio.

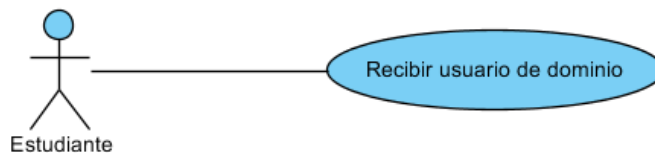


Figura 2. DIAGRAMA DE CASO DE USO DEL NEGOCIO RECIBIR USUARIO DE DOMINIO

### 2.6.5 DESCRIPCIÓN DE LOS CASOS DE USO DEL NEGOCIO

Tabla 3. DESCRIPCIÓN DEL CASO DE USO: RECIBIR USUARIO DE DOMINIO.

Caso de Uso		Recibir cuenta de usuario de dominio.	
Actores:		Estudiante	
Propósito		Obtener cuenta de usuario de dominio.	
Resumen: El caso de uso comienza cuando el Operador de SIGENU emite el listado oficial con los nombres de los estudiantes de nuevo ingreso, con el cual el Administrador de Red crea los usuarios de dominio y emite otro con los nombres de los estudiantes y sus respectivos nombres de usuario. El caso de uso finaliza cuando el coordinador de grupo informa a los estudiantes sus respectivos usuarios de dominio.			
Acción del actor		Respuesta del proceso de negocio	
		1	La Operadora de SIGENU emite el listado oficial con los nombres de los estudiantes de nuevo ingreso, organizado por carreras.
		2	El Administrador de Red recibe el listado oficial con los nombres de los estudiantes de nuevo ingreso organizado por carreras.
		3	El Administrador de Red crea los usuarios de dominio con una contraseña genérica.
		4	El Administrador de Red emite un listado con los nombres de los estudiantes de nuevo ingreso organizado por carreras, sus respectivos nombres de usuario y la contraseña genérica.
		5	El coordinador del grupo recibe el listado con los nombres de los estudiantes de su grupo, sus respectivos nombres de usuario y la contraseña genérica.
		6	El coordinador del grupo informa a cada estudiante su nombre de usuario y la contraseña genérica.
7	Recibe su nombre de usuario de dominio y la contraseña genérica.		
PRIORIDAD		Alta	
MEJORAS		M_1, M_2, M_3	

El diagrama de actividades del caso de uso Recibir usuario de dominio se encuentra en el **Anexo 1**.

## 2.7 MODELO DE OBJETOS

Un diagrama de objetos muestra un juego de objetos y sus relaciones en un momento del tiempo. Los diagramas de objetos se usan para modelar la vista estática del diseño o la vista estática del proceso de un sistema. Esto involucra modelar una instantánea del sistema a un momento en el tiempo y dando un juego de objetos, su estado, y sus relaciones.

“Los diagramas de objetos no sólo son importantes para visualizar, especificar, y documentar modelos estructurales, sino también para construir los aspectos estáticos de sistemas a través de ingeniería directa e inversa.” (Booch, Rumbaugh, & Jacobson, 2000)

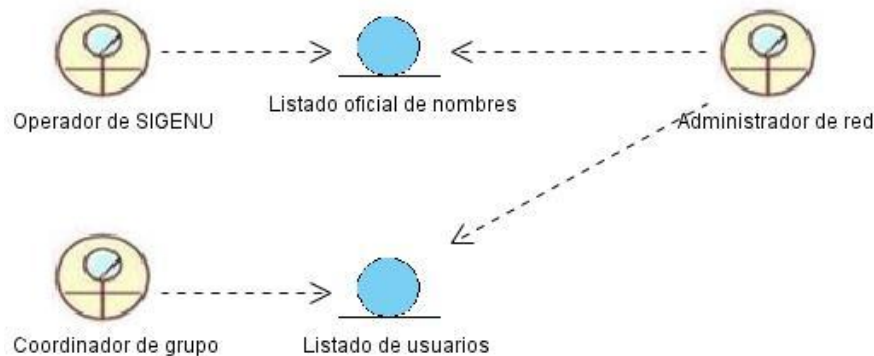


Figura 3. MODELO DE OBJETOS

## 2.8 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

### 2.8.1 REQUERIMIENTOS FUNCIONALES:

“La mayoría, si no todos los requisitos funcionales de un sistema, pueden expresarse como casos del uso, y los diagramas de casos de uso del UML son esenciales para manejar estos requisitos.” (Booch, Rumbaugh, & Jacobson, 2000)

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo.

**R\_1. Iniciar Sesión.**

**R\_2. Cerrar Sesión.**

**R\_3. Gestionar Estudiante de Pregrado.**

- 3.1 Adicionar.
- 3.2 Modificar.
- 3.3 Eliminar.
- 3.4 Mostrar Estudiante de Pregrado.

**R\_4. Gestionar Estudiante de Postgrado.**

- 4.1 Adicionar.
- 4.2 Modificar.
- 4.3 Eliminar.
- 4.4 Mostrar Estudiante de Postgrado.

**R\_5. Gestionar Operador de SIGENU.**

- 5.1 Adicionar.
- 5.2 Modificar.
- 5.3 Eliminar.
- 5.4 Mostrar Operador de SIGENU.

**R\_6. Gestionar Curso de Postgrado.**

- 6.1 Adicionar.
- 6.2 Eliminar.
- 6.3 Mostrar Curso de Postgrado.

**R\_7. Gestionar Carrera.**

- 7.1 Adicionar.
- 7.2 Modificar.
- 7.3 Eliminar.
- 7.4 Mostrar Carrera.

**R\_8. Gestionar Tipo de Carrera.**

- 8.1 Adicionar.
- 8.2 Modificar.
- 8.3 Eliminar.
- 8.4 Mostrar Tipo de Carrera.

**R\_9. Gestionar Solicitud de Curso de Postgrado.**

- 9.1 Aceptar.
- 9.2 Denegar.



9.3 Mostrar Solicitud de Curso de Postgrado.

**R\_10. Gestionar Solicitud de Estudiante de Postgrado.**

10.1 Aceptar

10.2 Denegar.

10.3 Mostrar Solicitud de Estudiante de Postgrado.

**R\_11. Eliminar Usuario de Dominio.**

**R\_12. Modificar usuario de dominio.**

**R\_13. Autorizar Solicitud de Curso de Postgrado.**

**R\_14. Autorizar Solicitud de Estudiante de Postgrado.**

**R\_15. Mostrar Estado de Solicitud de Curso de Postgrado.**

**R\_16. Mostrar Estado de Solicitud de Estudiante de Postgrado.**

**R\_17. Mostrar Estado de Solicitud de Modificación de Estudiante de Pregrado.**

**R\_18. Mostrar Solicitudes de Curso de Postgrado Aprobadas.**

**R\_19. Mostrar Solicitudes de Estudiante de Postgrado Aprobadas.**

**R\_20. Mostrar Estudiantes de Pregrado por Carrera.**

**R\_21. Mostrar Estudiantes de Postgrado por Curso de Postgrado.**

**R\_22. Mostrar Datos de Estudiante.**

**R\_23. Realizar Solicitud Modificar Usuario.**

**R\_24. Realizar Solicitud de Eliminar Usuario.**

**R\_25. Realizar Solicitud de Curso de Postgrado.**

**R\_26. Realizar Solicitud de Estudiante de Postgrado.**

### 2.8.2 REQUERIMIENTOS NO FUNCIONALES:

Los requisitos no funcionales pueden tener relación en varios casos de uso o en ninguno. Estos pueden ser:

#### ***Interfaz externa:***

- La interfaz se diseñará de tal modo que el usuario pueda tener el control de la aplicación en todo momento, lo que le permitirá ir de un punto a otro dentro de ella con gran facilidad. Además debe seguir los estándares de interfaz de la UNISS y ser semejante a la interfaz de otros sitios de la misma.

- Diseño encuadrado para la resolución 1024x768, y preparado para verse en otras resoluciones.

**Usabilidad:**

- El sistema podrá ser usado por el Administrador de red, los profesores, responsables de área y estudiantes de la UNISS.

**Rendimiento:**

- Tiempo de respuesta rápido de acuerdo a la petición realizada por el cliente y la cantidad de información a procesar. Almacenamiento de datos en caché para mejorar la velocidad de ejecución.

**Soporte:**

- Los responsables de las pruebas serán los trabajadores del departamento de Informatización de la UNISS, para evaluar en la práctica la funcionalidad y las ventajas del uso; además serán los encargados de su instalación y su mantenimiento. El sistema debe posibilitar futuras mejoras y la incorporación de nuevas opciones.

**Seguridad:**

- Establecer niveles de permisos para el acceso de los usuarios a la administración.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo.
- Proteger contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Utilizar algoritmos y protocolos para la autenticación.

**Confiabilidad:**

- La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores con el almacenamiento de los datos de todos los usuarios en una ubicación externa al AD.

**Software:**

- Directorio Activo de Windows Server 2003.
- Servidor Web Apache con PHP 5.2.3 o superior.
- Servidor de Base de Datos MySQL.
- Conexión con base de datos de SIGENU (PostgreSQL).

**Hardware:**

- Servidor con al menos 1 GB de RAM, 20 GB de disco duro y velocidad de micro de 2.5MHz.
- Las computadoras de los clientes deben tener al menos 256 MB de RAM y estar conectadas a la red local.

**Portabilidad:**

- Desde cualquier sistema operativo a través de un navegador web, debido a que en la UNISS existen laboratorios con sistemas de base Unix y otros con Microsoft Windows.

**Ayuda**

- El producto debe incluir una ayuda bien detallada sobre las principales opciones del sistema para orientar al usuario en el uso del mismo.

## 2.9 MODELO DEL SISTEMA.

“El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. Describe lo que hace el sistema para cada tipo de usuario y proporciona la entrada fundamental para el análisis, el diseño y las pruebas.” (Booch, Rumbaugh, & Jacobson, 2000)

### 2.9.1 ACTORES DEL SISTEMA

“Un actor es aquel que interactúa con el sistema, sin ser parte de él y puede asumir el rol que juega una o varias personas, un equipo o un sistema automatizado.” (Pressman, 2005) A continuación se definen y se muestran en la tabla siguiente los actores identificados en el sistema propuesto:

Tabla 4. ACTORES DEL SISTEMA

Actor del Sistema	Justificación
Administrador de red	Administrador de dominio que desde cualquier lugar de la red accede a la aplicación web y cuenta con usuario y contraseña válidos para el servidor de postgresQL donde se encuentra la base de datos de SIGENU. Es el encargado de la administración y gestión del sistema y de la creación de los usuarios con solicitudes previamente aprobadas.
Profesor	Persona que puede realizar una Solicitud de Curso de Postgrado en caso de no serlo de alguno de los que existan.
Coordinador de Postgrado	Profesor al que se le ha creado un Curso de Postgrado y que es responsable de aprobar o denegar las solicitudes de matrícula en dicho curso.
Aspirante	Usuario que solicita la creación de un usuario de dominio para estudiante de postgrado, no puede ver nada más que los datos de su propia Solicitud de Usuario de Dominio.
Responsable de Área	Es la persona encargada de aprobar las Solicitudes de Curso de Postgrado realizadas por profesores que pertenezcan a su área.
Usuario	Persona que tiene usuario de dominio y puede iniciar y cerrar sesión, así como visualizar sus datos personales y emitir una solicitud de Modificación de usuario de dominio.
Usuario avanzado	Actor genérico que hace uso de funcionalidades que son comunes a varios actores que heredan de él específicamente el Administrador de red, el Responsable de Área, Coordinador de Postgrado.
Estudiante	Persona que cursa cualquiera de los cursos de postgrado o carreras que se imparten en la UNISS y que puede visualizar sus datos y solicitar la modificación d los mismos.

## 2.9.2 CASOS DE USO DEL SISTEMA:

Se identificaron los siguientes casos de uso del sistema enmarcados en los paquetes que se muestran a continuación:

### Seguridad:

1. Autenticar usuario.
2. Cerrar sesión.

### Gestión:

3. Gestionar Estudiante de Pregrado.
4. Gestionar Estudiante de Postgrado.
5. Gestionar Operador de SIGENU.
6. Gestionar Curso de Postgrado.
7. Gestionar Carrera.
8. Gestionar Tipo de Carrera.
9. Eliminar usuario de dominio.
10. Autorizar Solicitud de Curso de Postgrado.
11. Autorizar Solicitud de Estudiante de Postgrado.
12. Modificar Usuario de Dominio.

### Consulta:

13. Mostrar Estado de Solicitud de Curso de Postgrado.
14. Mostrar Estado de Solicitud de Estudiante de Postgrado.
15. Mostrar Estado de Solicitud de Modificación de Estudiante de Pregrado.
16. Mostrar Solicitudes de Curso de Postgrado.
17. Mostrar Solicitudes de Curso de Postgrado Aprobadas.
18. Mostrar Solicitudes de Estudiante de Postgrado.
19. Mostrar Solicitudes de Estudiante de Postgrado Aprobadas.
20. Mostrar Datos de Estudiante
21. Mostrar Estudiantes de Pregrado por Carrera.
22. Mostrar Estudiantes de Postgrado por Curso de Postgrado.

### Solicitud:

23. Solicitar modificación de usuario de dominio.

24. Solicitar eliminación de usuario de dominio.
25. Solicitar creación de Curso de Postgrado.
26. Solicitar creación de usuario de dominio para Estudiante de Postgrado.

### 2.9.3 PAQUETES Y SUS RELACIONES:

Un paquete es un mecanismo de propósito general para organizar los elementos en grupos. Las partes estructurales, las partes conductuales, e incluso otros grupos pueden conformar un paquete. Gráficamente, un paquete se da como una carpeta, normalmente incluyendo sólo su nombre y, a veces, sus contenidos. Los paquetes son los grupos básicos con los cuales se puede organizar un modelo UML. (Pressman, 2005)

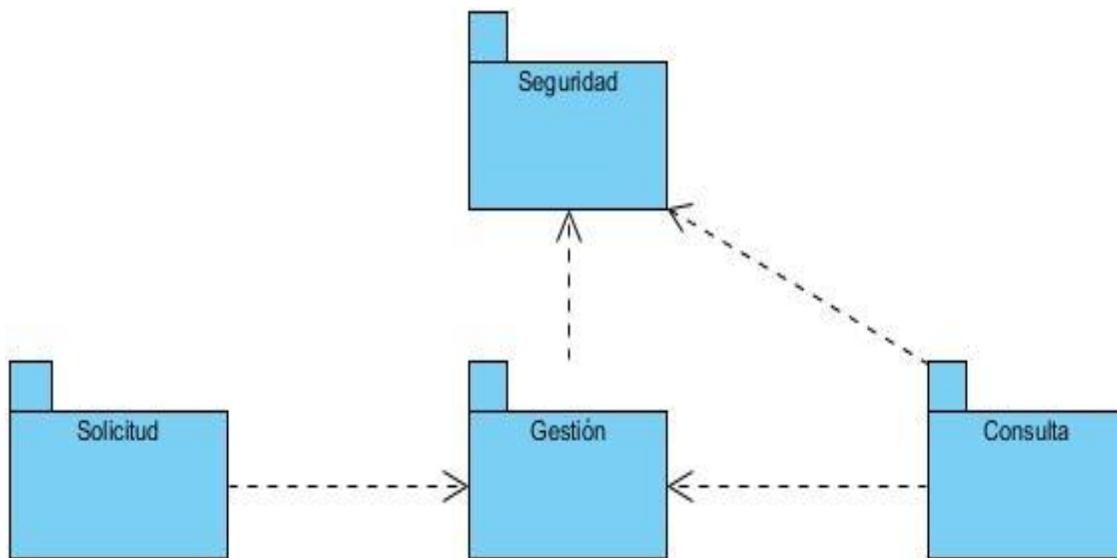


Figura 4. PAQUETES DEL SISTEMA Y SUS RELACIONES.

### 2.9.4 DIAGRAMAS DE CASOS DE USO DEL SISTEMA

Los diagramas de casos de uso proporcionan una visión externa del sistema. Un caso de uso se modela para todos los procesos que el sistema debe llevar a cabo. Los procesos se describen dentro del caso de uso por una descripción textual o una secuencia de pasos ejecutados. De manera más precisa, un Caso de Uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia.

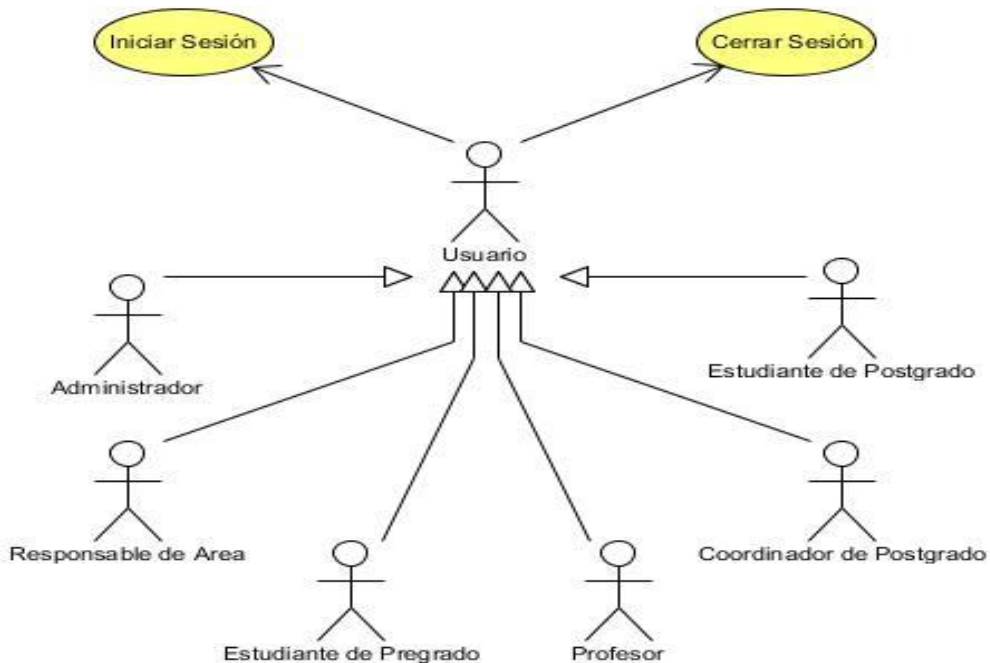


Figura 5. DIAGRAMA DE CASOS DE USO DEL SISTEMA: SEGURIDAD.

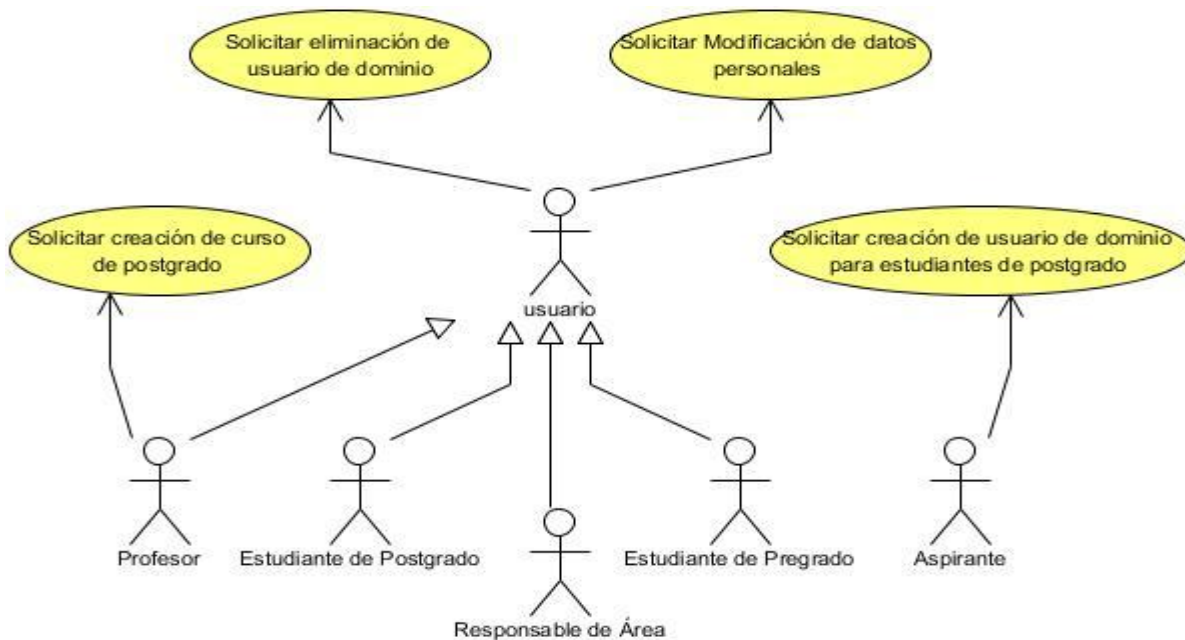


Figura 6. DIAGRAMA DE CASOS DE USO DEL SISTEMA: SOLICITUD.

El resto de los diagramas de casos de uso del sistema se encuentran en el Anexo 2.

## 2.9.5 DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA Y LOS PROTOTIPOS DE INTERFAZ:

Tabla 5. DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA INICIAR SESIÓN.

<b>CU 1</b>	<b>Iniciar sesión</b>
<b>Actores</b>	Administrador de Red, Profesor, Estudiante de Pregrado, Estudiante de Postgrado
<b>Propósito</b>	Permite al usuario avanzado iniciar sesión en el Sistema.
<b>Resumen</b>	
El caso de uso comienza cuando el usuario avanzado decide iniciar sesión en el sistema y llena el formulario de autenticación con su nombre de usuario y contraseña. El sistema comprueba la información y le da los privilegios correspondientes al usuario terminando el caso de uso.	
<b>Referencias</b>	R_1
<b>Prototipo</b>	Anexo 3

Tabla 6. DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA CERRAR SESIÓN.

<b>CU 2</b>	<b>Cerrar sesión</b>
<b>Actores</b>	Administrador de Red, Profesor, Estudiante de Pregrado, Estudiante de Postgrado
<b>Propósito</b>	Permite al usuario avanzado terminar su sesión en el Sistema.
<b>Resumen</b>	
El caso de uso comienza cuando el usuario avanzado decide terminar su sesión en el sistema y lo hace mediante el vínculo que se encuentra en el panel de Bienvenida. El sistema termina la sesión y retorna a la interfaz principal para los usuarios anónimos terminando el caso de uso.	
<b>Referencias</b>	R_2
<b>Prototipo</b>	Anexo 4



Tabla 7. DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA GESTIONAR ESTUDIANTE DE PREGRADO.

<b>CU 3</b>	<b>Gestionar Estudiante de Pregrado</b>
<b>Actores</b>	Administrador de Red
<b>Propósito</b>	Permite adicionar, eliminar, modificar o mostrar los usuarios de dominio de los estudiantes de pregrado.
<b>Resumen</b>	El caso de uso comienza cuando el Administrador de Red selecciona del menú de administración una de estas opciones y el sistema realiza lo que literalmente significa la acción seleccionada, terminando así el caso de uso.
<b>Referencias</b>	R_3
<b>Prototipo</b>	Anexo 5

El resto de las descripciones de los casos de uso se muestran en el Anexo 3.

## 2.10 CONCLUSIONES

El análisis del modelo del negocio permitió realizar un estudio profundo del proceso de creación de usuarios de dominio, se enumeraron las reglas que lo rigen y se extrajeron los actores y trabajadores que interactúan en el mismo. Para esto se elaboraron los diagramas de casos de usos y de actividades, se describieron los casos de usos del negocio y se elaboró el modelo de objetos del negocio. Se realizó una descripción general del modelo de sistema, identificando los actores, casos de uso y la relación entre estos, obteniéndose el modelo de casos de uso del sistema.

## CAPÍTULO 3: IMPLEMENTACIÓN DEL SOFTWARE PARA OPTIMIZAR LA GESTIÓN DE USUARIOS DE DOMINIO PARA LOS ESTUDIANTES EN LA UNISS.

### 3.1 INTRODUCCIÓN:

El presente capítulo abarca aspectos referentes al diseño de la solución propuesta como los diagramas de clases, de componentes y el de despliegue. Se planteará también el modelo de datos a partir del diagrama de clases persistentes.

### 3.2 DISEÑO.

Este conjunto de información describe cómo el sistema se va a construir y captura las decisiones acerca de cómo el sistema se va a construir, teniendo en cuenta todas las limitaciones de tiempo, presupuesto, el legado, la reutilización, los objetivos de calidad, y así sucesivamente. Este puede comprender un modelo de diseño, un modelo de prueba, y otras formas de expresión de la naturaleza del sistema, incluyendo pero no limitado a prototipos y arquitecturas ejecutables. (Pressman, 2005)

#### 3.2.1 DIAGRAMA DE CLASES.

El Diagrama de Clases es el diagrama principal de diseño y análisis para un sistema. Presenta las clases, junto con sus atributos, operaciones, interfaces y relaciones. Se convierte en el diagrama central del análisis del diseño orientado a objetos, y el que muestra la estructura estática del sistema. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones. Puede ser dividido en capas: aplicación, y datos, las cuales muestran las clases que intervienen con la interfaz de usuario, la lógica del software de la aplicación, y el almacenamiento de datos respectivamente. Presenta las clases del sistema con sus relaciones (estructurales y de herencia). En el caso de las aplicaciones web, el diagrama de clases representa las colaboraciones entre las páginas, donde cada página lógica puede ser representada como una clase.

Los diagramas de clases web, fueron realizados a partir de los diferentes casos de uso del sistema y empleando las extensiones de UML para web.

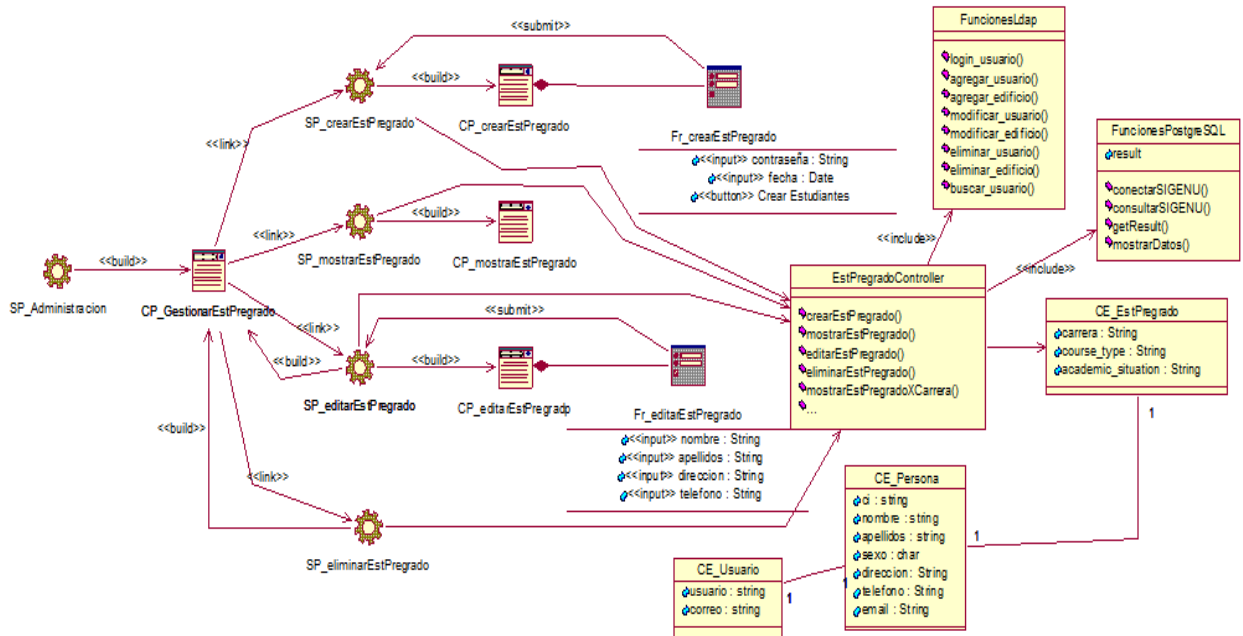


Figura 7. DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR ESTUDIANTE DE POSTGRADO.

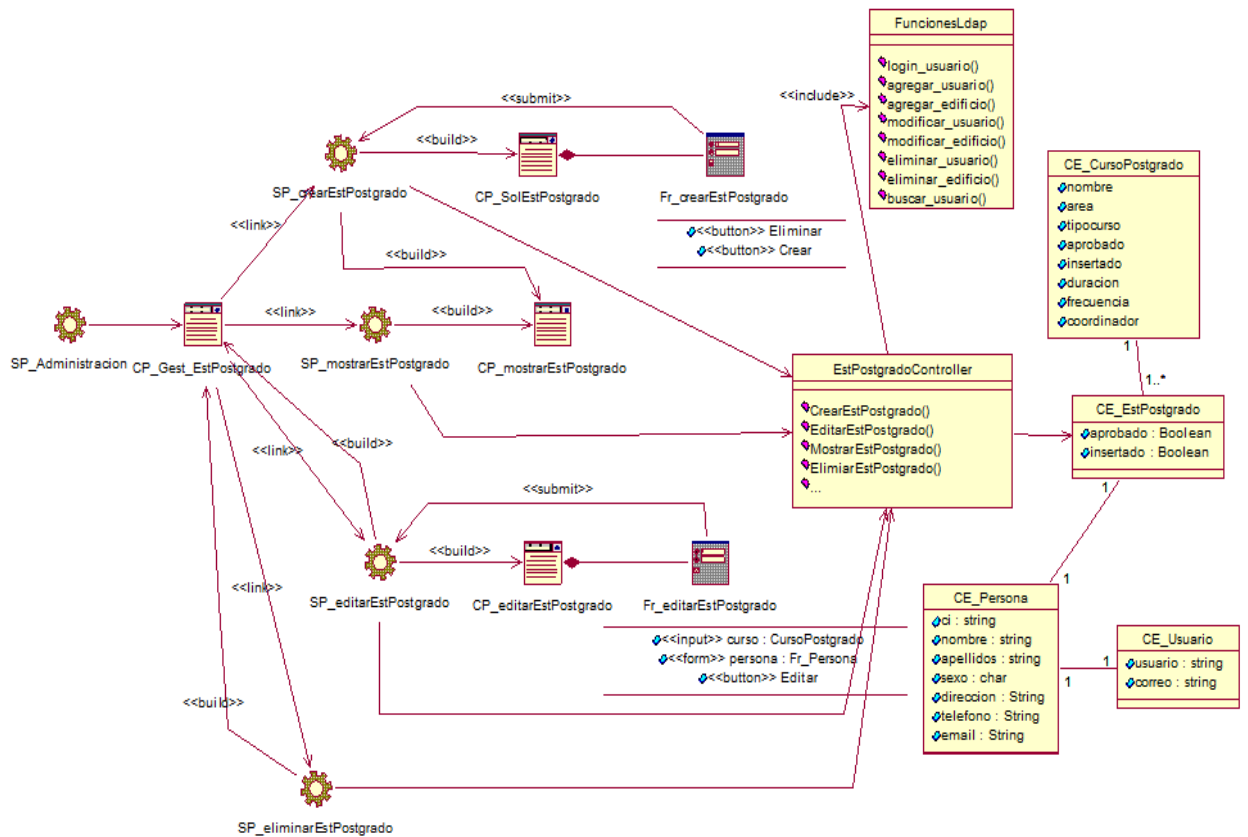


Figura 8. DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR ESTUDIANTE DE PREGRADO.

### 3.3 DISEÑO DE LA BASE DE DATOS.

#### 3.3.1 DIAGRAMA DE CLASES PERSISTENTES.

En el diagrama de clases persistentes aparecen las clases que persisten, las cuales poseen la capacidad de mantener su valor en el espacio y en el tiempo y que se convertirán en tablas de nuestra base de datos.

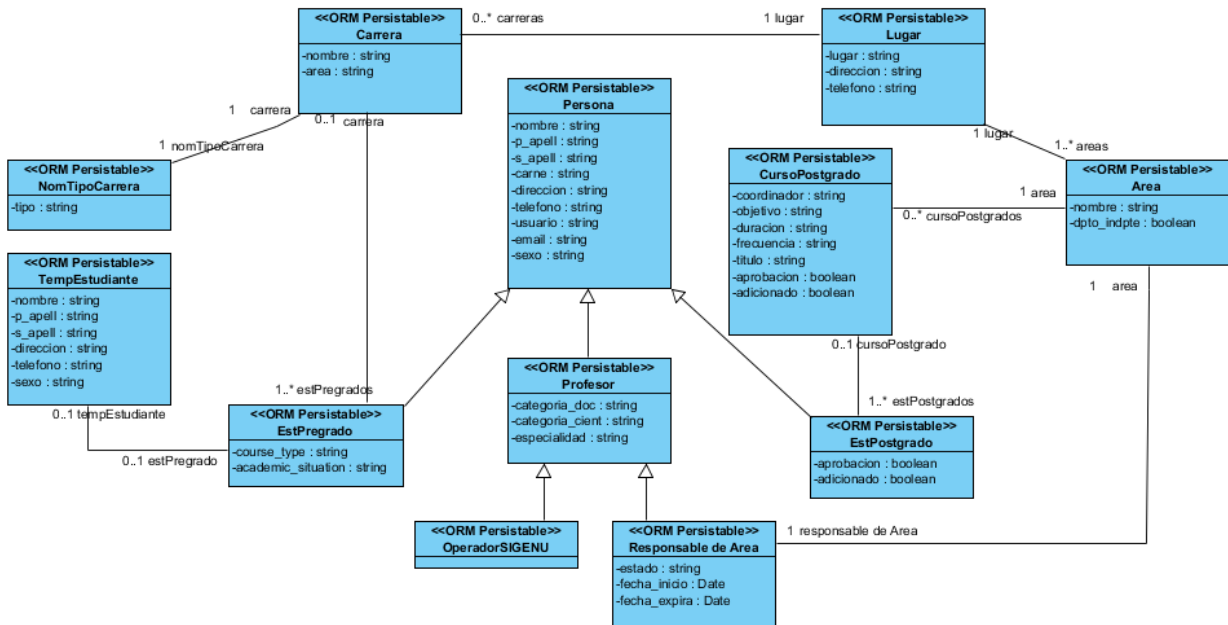


Figura 9. DIAGRAMA DE CLASES PERSISTENTES.

#### 3.3.2 MODELO DE DATOS.

El modelo físico de datos, representa la estructura o descripción física de las tablas de la base de datos, obtenido a partir del modelo lógico. Una de las características fundamentales de los sistemas de bases de datos es que proporcionan cierto nivel de abstracción de datos, al ocultar las características sobre el almacenamiento físico que la mayoría de usuarios no necesita conocer.

Los modelos de datos son el instrumento principal para ofrecer dicha abstracción. Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos.

### 3.4 IMPLEMENTACIÓN.

El modelo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes, describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en los lenguajes de programación utilizados y cómo dependen unos componentes de otros (Booch, Rumbaugh, & Jacobson, 2000)

#### 3.4.1 DIAGRAMA DE DESPLIEGUE.

El diagrama de despliegue muestra la forma en que los componentes se desplegarán en el sistema, en este se detalla las especificaciones de la red, del servidor y los requisitos de hardware. Según (Pressman, 2005) un diagrama de despliegue es un diagrama que muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos.

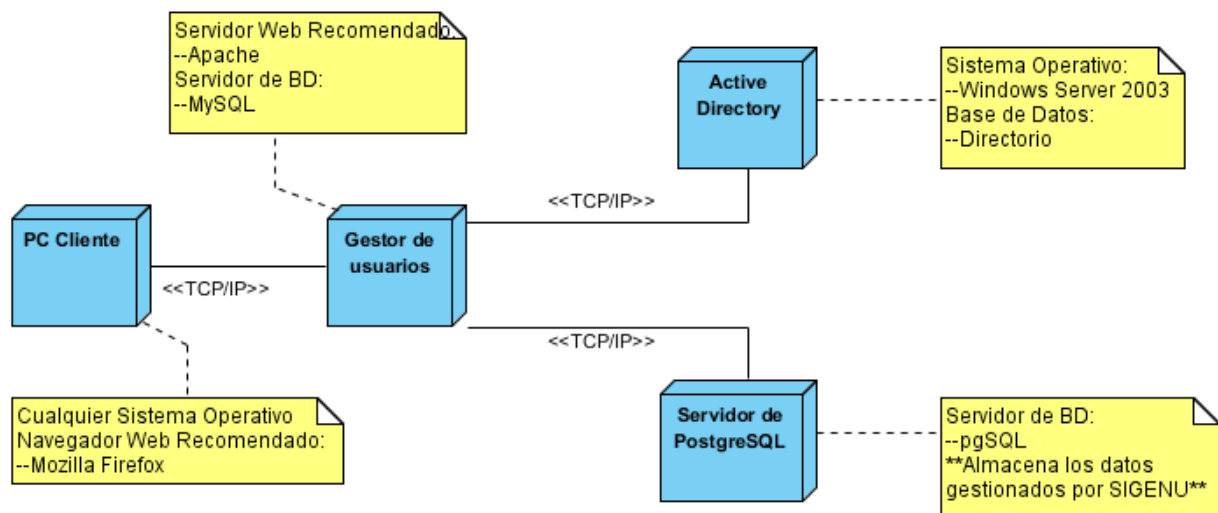


Figura 10. DIAGRAMA DE DESPLIEGUE.

### 3.5 DIAGRAMA DE COMPONENTES.

Los diagramas de componentes son uno de los dos tipos de diagramas del modelado de los aspectos físicos de sistemas orientados a objetos. Un diagrama de componentes muestra la organización y dependencias entre un juego de componentes y se usa para modelar la vista estática de la implementación de un sistema. Esto involucra modelar las partes físicas que residen en un nodo, como los ejecutables, bibliotecas, tablas, archivos, y documentos. Los diagramas de componentes son esencialmente diagramas de clase enfocados en los componentes de un sistema. No sólo son importantes para visualizar, especificar, y documentar

los sistemas basados en componentes, sino también para construir los sistemas ejecutables a través de la ingeniería directa e inversa.

Los diagramas de componentes contienen los principales elementos de implementación y las relaciones existentes entre ellos, proporcionan un alto nivel de comprensión de la arquitectura del sistema que será construido, permitiendo a los desarrolladores formalizar un plan de trabajo para la ejecución, y tomar decisiones sobre la asignación de tareas y habilidades necesarias para las mejoras futuras del software. Un diagrama de componentes muestra la organización y las dependencias entre un conjunto de componentes. Los diagramas de componentes cubren la vista de implementación estática de un sistema. Se relacionan con los diagramas de clases en que un componente se corresponde, por lo común, con una o más clases, interfaces o colaboraciones. (Booch, Rumbaugh, & Jacobson, 2000)

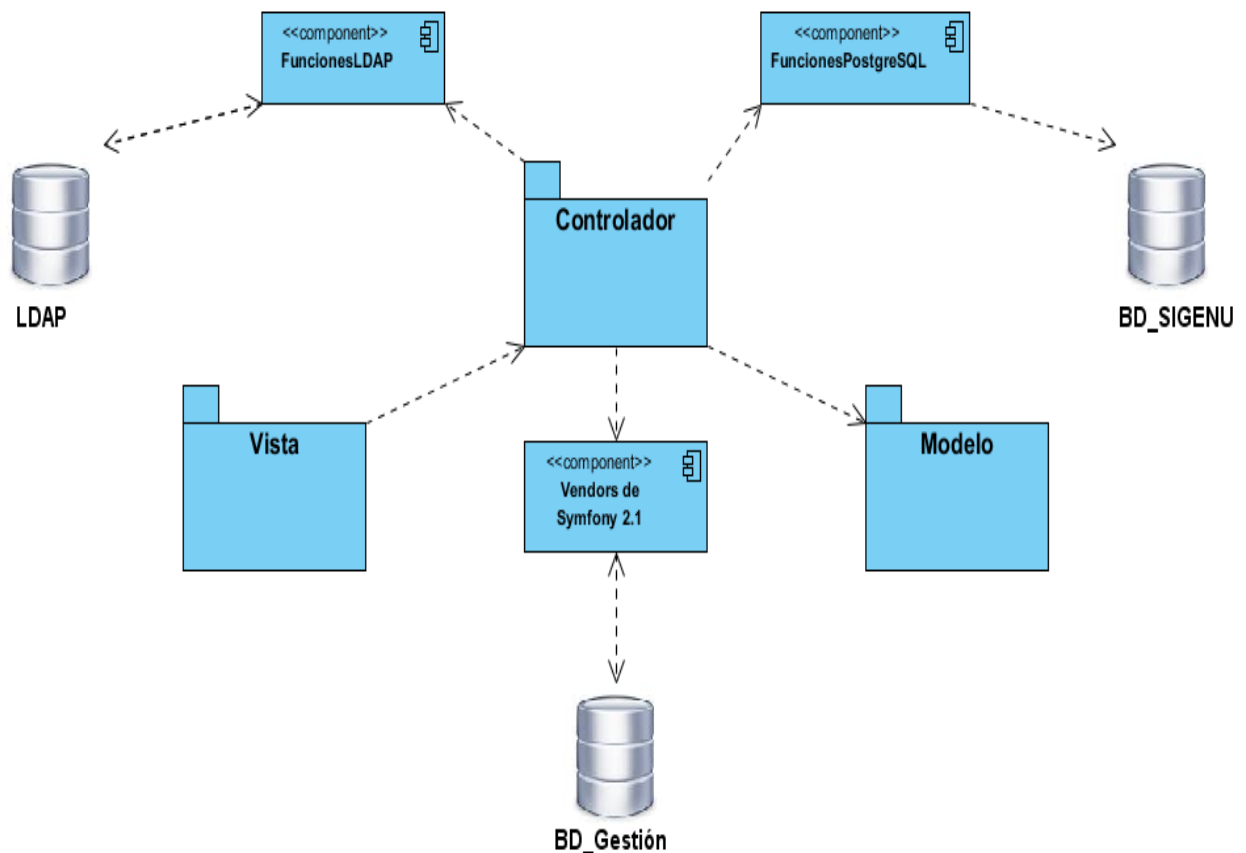


Figura 11. DIAGRAMA DE COMPONENTES.

### 3.5.1 DESCRIPCIÓN DE LOS COMPONENTES:

Tabla 8. DESCRIPCIÓN DE LOS COMPONENTES.

Componente	Descripción
Vista	Paquete que agrupa los ficheros de las clases de interfaz del usuario para los diferentes perfiles, al igual que los reportes.
Controladora	Paquete que agrupa los ficheros de las clases controladoras encargadas de manipular la lógica del negocio y que actúan de intermediaria entre las clases del paquete vista y las del modelo.
Modelo	Este paquete contiene los ficheros de las clases de acceso a datos.
FuncionesLDAP	Este paquete contiene las funciones implementadas para la manipulación del directorio activo.
FuncionesPostgreSQL	Este paquete contiene las funciones implementadas para la obtención de la información en la base de datos de SIGENU, que se encuentra en un servidor de postgresQL.
LDAP	AD ( <i>Active Directory</i> ) en el que se encuentran los usuarios de dominio.
BD_SIGENU	Sistema gestor de postgresQL en el que se encuentra la base de datos gestionada por SIGENU.
BD_Gestion	Base de datos implementada con el fin de facilitar todo el proceso realizado por la aplicación web, y crear un repositorio de todos los datos almacenados en el directorio activo.

## 3.6 PRINCIPIOS DE DISEÑO

Lograr la apariencia adecuada y que el usuario se sienta confortable es uno de los objetivos principales en la implantación de un software. Para ello se tienen en cuenta varios aspectos, sobre todo relacionados con tipografía, colores, gráficos, navegación, composición del sitio, tratamiento de errores y concepción de la ayuda, los que a continuación se detallan.

### 3.6.1 ESTÁNDARES EN LA INTERFAZ DE LA APLICACIÓN

El producto final de un proyecto de desarrollo de software debe tener una interfaz amigable y de fácil navegación. Algunos de los principios tomados en cuenta para el diseño de la aplicación web fueron:

- ✓ El tamaño de la ventana será fijo, en caso de una modificación la página no puede reajustarse al nuevo tamaño, sino que debe visualizarse a través de las barras de desplazamiento.
- ✓ Las imágenes e iconos usar deben tener un significado implícito para lograr un mejor entendimiento de las funcionalidades que realiza la aplicación.
- ✓ La interfaz diseñada para el aplicación web debe estar concebida para la resolución 800x600 píxeles o mayor.
- ✓ La navegabilidad debe estar disponible en todo momento de manera que el cliente tenga el control de la aplicación.
- ✓ Las etiquetas de los formularios deben contener un lenguaje claro y evitarse la carga visual de los mismos.
- ✓ Es aconsejable utilizar colores claros y relajantes a la vista del usuario para que este se sienta cómodo mientras interactúa con el sistema.

### 3.6.2 TRATAMIENTO DE ERRORES

En todo sistema el tratamiento de errores tiene como propósito garantizar la mínima incidencia, evitando la de entrada de datos erróneos en la aplicación. Para las validaciones del lado del cliente se utilizó JavaScript, debido que permite dar respuesta rápida, puesto que no necesariamente tiene que ir al servidor, en caso de que no pueda responder a las peticiones, entonces accede al servidor, es decir que los Scripts evitan las peticiones innecesarias al servidor. Las validaciones del lado del servidor fueron con el lenguaje PHP.



### 3.6.3 CONCEPCIÓN GENERAL DE LA AYUDA

Con el propósito de proporcionar fácil manejo de la aplicación web por parte del cliente se desarrolló una ayuda tomando en cuenta las posibles situaciones que pudieran aparecer durante el manejo del sistema. Para esto existe en cada página un menú disponible para el acceso a dicha ayuda. La ayuda está estructurada según las funcionalidades de la aplicación para facilitar la búsqueda de temas específicos.

### 3.6.4 SEGURIDAD

La seguridad es una de las partes que más detalladamente se debe implementar debido a la asignación de permisos a los distintos recursos que manipula el software. Dichos permisos son concedidos sólo en el caso de que el usuario se haya identificado de manera satisfactoria. Se tuvo en cuenta además que las personas que no poseen usuario de dominio a pesar de no poder realizar la autenticación también cuentan con cierto nivel de acceso.

La seguridad y protección de la información almacenada se realiza a través del componente Security de Symfony está disponible como una biblioteca independiente de PHP.

## 3.7 CONCLUSIONES

En el presente capítulo se mostraron los resultados de la etapa de implementación del sistema. Se desarrollaron los diagramas de clases, el diseño de la base de datos, el diagrama de despliegue y el de componentes. Además, se describieron los principios de diseño seguidos, específicamente, estándares de la interfaz de usuario, el tratamiento de excepciones concepción de la ayuda y seguridad.

## CONCLUSIONES

- ✓ Para el desarrollo del sistema se utilizó como guía la metodología RUP y como lenguaje de modelación el Lenguaje Unificado de Modelado, lo cual permitió una adecuada documentación del análisis, el diseño y la implementación de la solución propuesta. Como herramienta de modelado se utilizó el Visual Paradigm y el Rational Rose.
- ✓ Se diseñó una aplicación web para optimizar la gestión de los usuarios de dominio para los estudiantes de la UNISS. El mismo se sustenta en los preceptos del software libre, la arquitectura en tres capas y la programación orientada a objeto, todo esto le brinda flexibilidad y posibilidades para futuras modificaciones y mejoras.
- ✓ Se implementó una aplicación web utilizando Symfony2 como framework de PHP incluyendo el uso de Twig como motor de plantillas y de Doctrine2 como ORM, además del uso de JavaScript para las validaciones de entradas del usuario y CSS para el estilo y apariencia. Como herramientas de desarrollo Macromedia Dreamweaver para la programación Web, MySQL como SGBD y el IDE NetBeans7.2.

## RECOMENDACIONES

Evaluar el sistema luego de su introducción en la UNISS para:

- ✓ Mejorar y actualizar el sistema de acuerdo a las necesidades del centro.
- ✓ Incluir el aviso por email a cada usuario del cambio en el estado de los eventos solicitados por él, así como las tareas pendientes que tenga en el sistema, para de esta forma evitar la demora de que los usuarios se autentiquen para saber sus tareas
- ✓ Añadir nuevas funcionalidades al sistema para su desarrollo y evolución.
- ✓ Aprovechar la generalización que brinda el sistema para extender su aplicación a centros universitarios similares.

## BIBLIOGRAFÍA

- Alcaraz, C., Roman, R., Najera, P., & Lopez, J. (2010). *Acceso seguro a redes de sensores en SCADA a través de Internet*.
- Álvarez, M. Á. (2009). *Desarrollo Web*. Obtenido de Entender el flujo de aplicación de CodeIgniter: [www.desarrolloweb.com/Manuales/Manual de CodeIgniter](http://www.desarrolloweb.com/Manuales/Manual%20de%20CodeIgniter)
- Blejmar, B. (2005). *Gestionar es hacer... que las cosas sucedan*. Noveduc Libros.
- Blue Coat Systems, Inc. (2007). *Users Management*.
- Boggs, W., & Boggs, M. (2002). *UML with Rational Rose*.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2000). *The Unified Modeling Language User Guide*. Grady Booch James Rumbaugh Ivar Jacobson Publisher: Addison Wesley First Edition October 20, 1998.
- Casassus, J. (2000). *Problemas de la gestión educativa en América Latina*.
- Doctrine Project Team. (2012). *Doctrine 2 ORM Documentation*.
- Eguiluz, J. (2011). *Desarrollo Web Ágil con Symfony2*.
- Espinosa Minguet, A. R. (2012). *Servicios de dominio de Active Directory*.
- Hernán Ruiz, M. (2006). *Programación WEB avanzada*. La Habana: Felix Varela.
- Hernández Orallo, E. (2009). *El Lenguaje Unificado de Modelado (UML)*. Obtenido de El Lenguaje Unificado de Modelado (UML)
- Manual de PHP*. (2013). Obtenido de <http://www.php.net/manual/es/introduction.php>
- Microsoft Windows. (2003). *Windows Server*.
- MySQL 5.0 Reference Manual*. (2011). Obtenido de Las principales características de MySQL.
- Pérez Pérez, S. L., & Morales Luna, G. (2010). *Autenticación de usuarios con grafos de Cayley*.
- Perona, L. A., & Velásquez, J. E. (2012). *GESTIÓN DE CONOCIMIENTO: LA SOLUCIÓN PARA DISMINUIR EL REPROCESO EN LAS PRUEBAS DE SOFTWARE*. MEDELLÍN – ANTIOQUIA .
- Pressman, R. (2005). *Ingeniería del Software Un enfoque práctico*.

- Rodríguez, M. A. (2009). *La sede electrónica y la identificación y autenticación electrónica en el proceso de reglamento de la ley de acceso.*
- Sánchez, B. &. (2010). *Gestió del LDAP.*
- Sánchez, J. (2003). *Manual de Referencia JavaScript.*
- Twig - *The flexible, fast, and secure PHP template engine.* (2013). Obtenido de <http://twig.sensiolabs.org/documentation>
- Viña Brito, S. (2013). *EXPERIENCIAS DEL USO DE HERRAMIENTA OLAP DE SIGENU EN EL SERVICIO ESTADÍSTICO DEL MINISTERIO DE EDUCACIÓN SUPERIOR.*
- Worsley, J., & Joshua D, D. (2002). *Practical PostgreSQL.*
- Tadei, L. (2009). *Frameworks en PHP...y otros lenguajes también.*
- Vedora, U. (2006). *Sistema de gestión de la información de Estudiantes Becarios Extranjeros (SIBEX).*  
Cienfuegos: Trabajo de Diploma para optar por el título de Ingeniería en Informática, Facultad de Informática, Universidad de Cienfuegos Carlos Rafael Rodríguez.
- Vidal, C. (2006). *Sistema de Envío de Correos por Lote (SECL).* Universidad de Cienfuegos Carlos Rafael Rodríguez.
- Cordero Carrasco, R. (Febrero de 2006). *Introducción al diseño y a la programación orientada a objetos.* Obtenido de <http://www.nielsoft.com/Seminario/3capas/introduccion.ppt>
- Alonso, L. C. (2006). *Modulo Admisión del Sistema Automatizado para la Gestión de Información de la Misión Milagro.* Ciudad de la Habana.
- Catalani, E. (2007). *ARQUITECTURA Modelo/Vista/Controlador.* Obtenido de <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>
- Cadenas, Y. (2010). *Aplicación Web para la gestión de la información en las Oficinas de Seguridad para las Redes Informáticas.* Universidad de Sancti-Spíritus José Martí Pérez.
- Hernández Salazar, P. (2009). *El perfil del usuario de información.* Investigación bibliotecológica.
- García Chico, J. (2011). *Estudio de viabilidad de directorio activo en linux.*

Martínez, P. (2007). Patrones. Extraído de Patrón "Modelo-Vista-Controlador": <http://www.proactiva-calidad.com/java/patrones/mvc.html>

Lago, R. (Abril de 2007). Patrones de diseño software.

## ANEXOS

### Anexo 1. Diagrama de actividades del caso de uso: Recibir usuario de dominio.

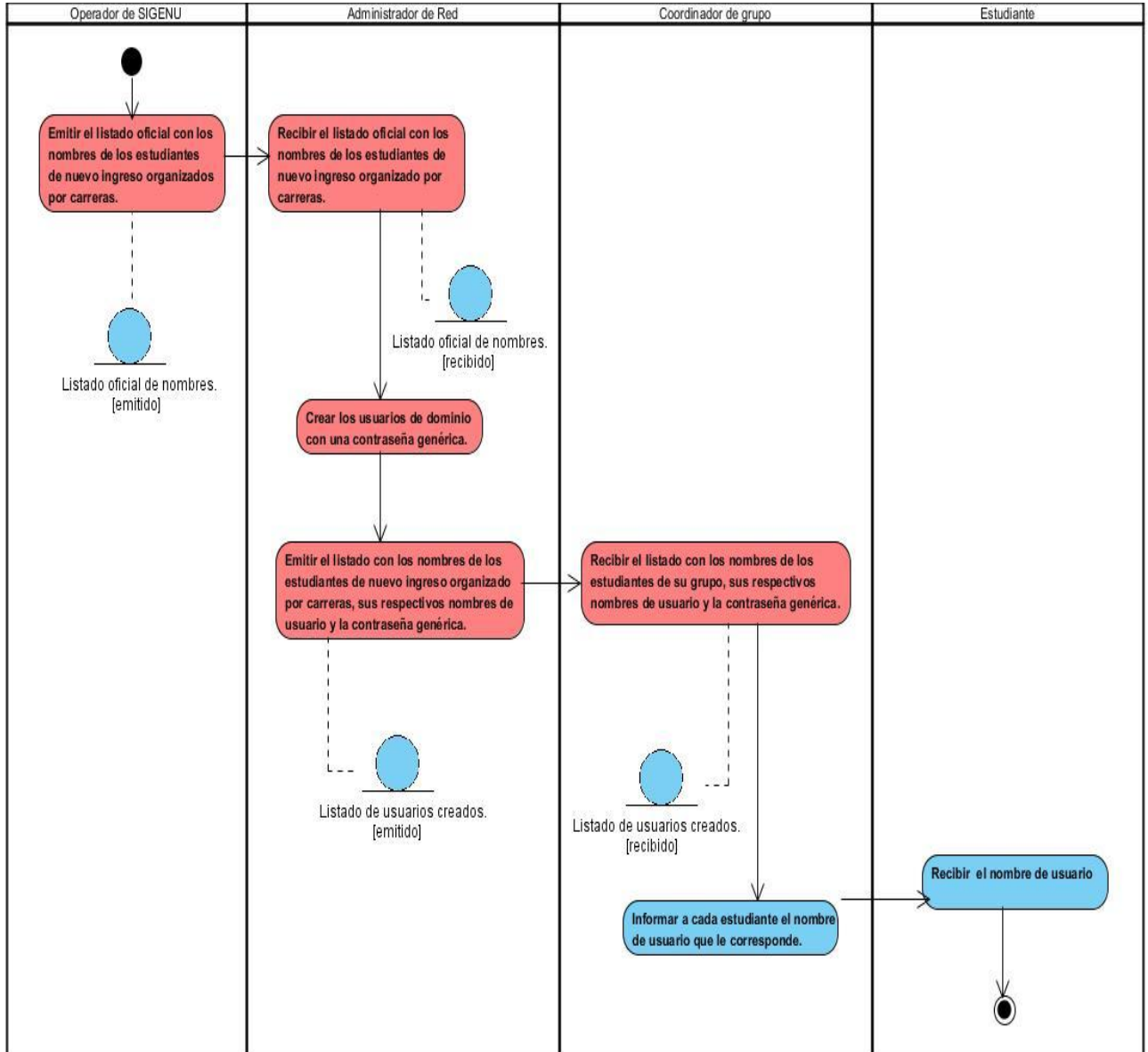


Figura 12. DIAGRAMA DE ACTIVIDADES DEL CASO DE USO: RECIBIR USUARIO DE DOMINIO.

## Anexo 2. Diagramas de casos de uso del sistema

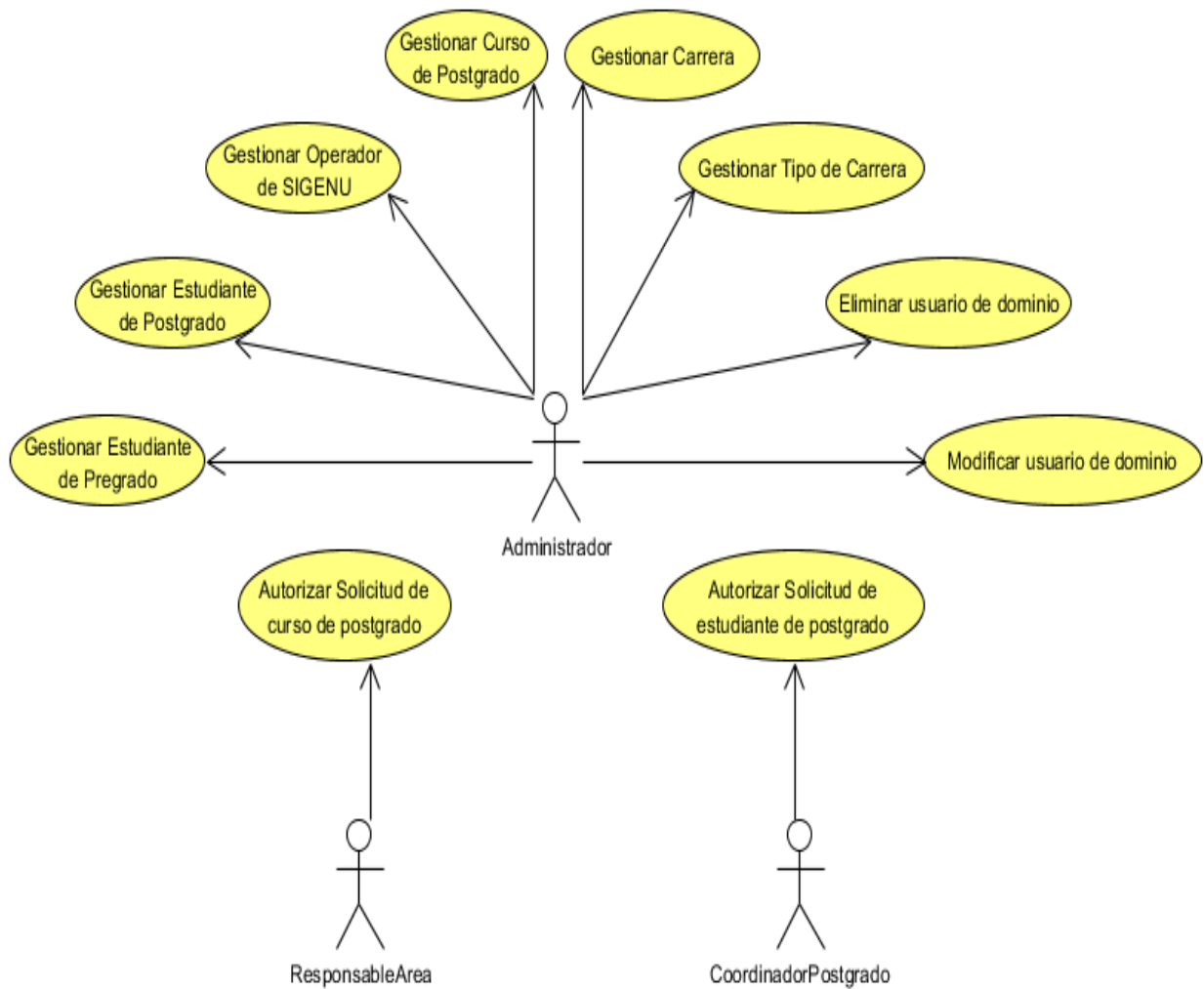


Figura 13. DIAGRAMA DE CASOS DE USO DEL SISTEMA: GESTIÓN



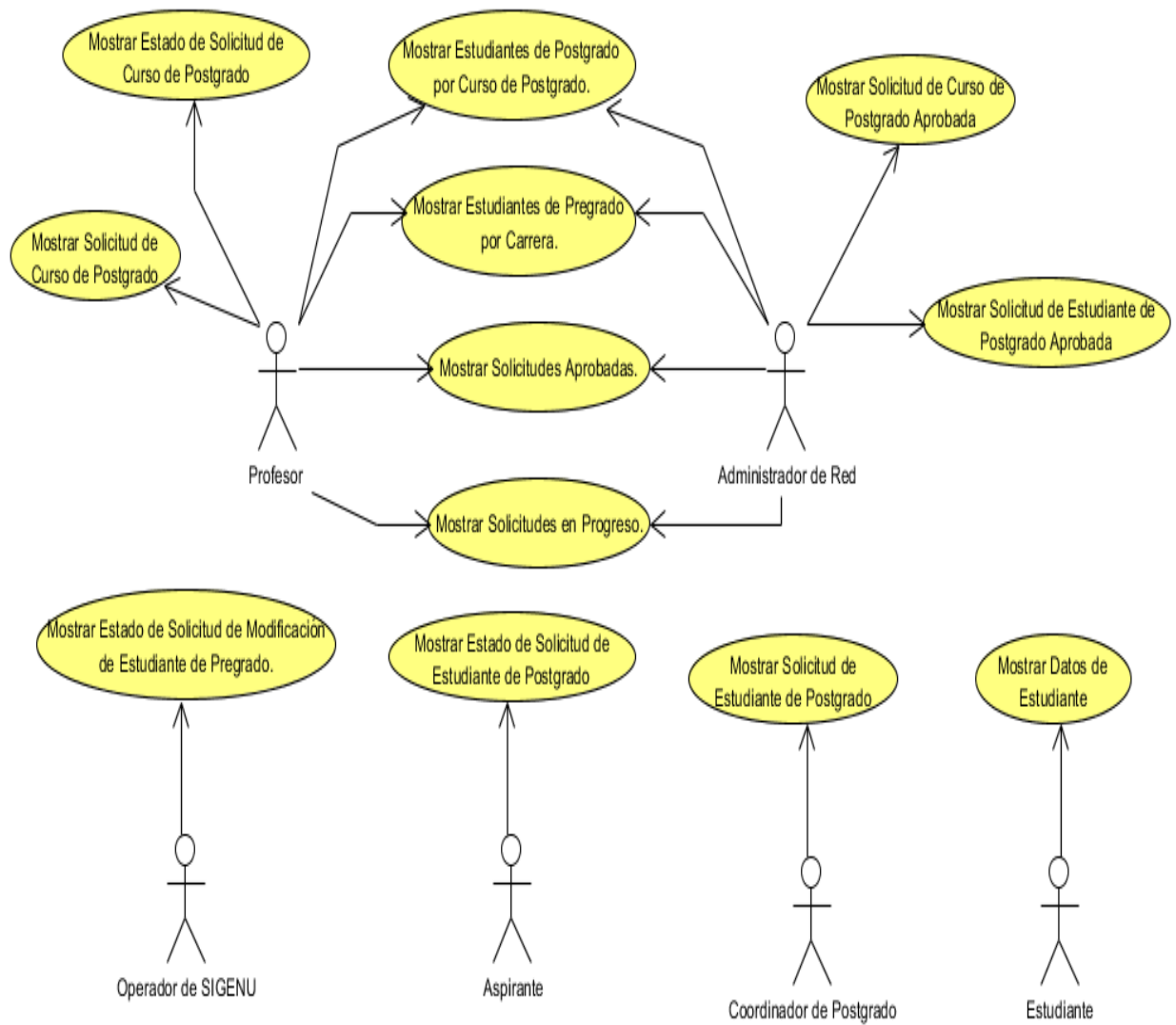


Figura 14. DIAGRAMA DE CASOS DE USO DEL SISTEMA: CONSULTA

### Anexo 3. Descripción de los casos de uso del sistema.

<b>CU 4</b>	<b>Gestionar Estudiante de Postgrado</b>
<b>Actores</b>	Administrador de Red
<b>Propósito</b>	Permite adicionar, eliminar, modificar y mostrar los Estudiantes de Postgrado.
<b>Resumen</b> El caso de uso comienza cuando el actor escoge una de estas opciones. El sistema realiza la acción seleccionada, culminando el caso de uso.	
<b>Referencias</b>	R_4
<b>Prototipo</b>	Anexo 7

<b>CU 5</b>	<b>Gestionar Operador de SIGENU</b>
<b>Actores</b>	Administrador de Red
<b>Propósito</b>	Permite adicionar, eliminar, modificar y mostrar el Operador de SIGENU.
<b>Resumen</b> El caso de uso comienza cuando el actor escoge una de estas opciones. El sistema realiza la acción seleccionada, culminando el caso de uso.	
<b>Referencias</b>	R_5
<b>Prototipo</b>	Anexo 9

<b>CU 6</b>	<b>Gestionar Curso de Postgrado</b>
<b>Actores</b>	Administrador de Red
<b>Propósito</b>	Permite adicionar, eliminar, modificar y mostrar un Curso de Postgrado.
<b>Resumen</b> El caso de uso comienza cuando el actor escoge una de estas opciones. El sistema realiza la acción seleccionada, culminando el caso de uso.	
<b>Referencias</b>	R_6
<b>Prototipo</b>	Anexo 11

<b>CU 7</b>	<b>Gestionar Carrera</b>
<b>Actores</b>	Administrador de Red
<b>Propósito</b>	Permite adicionar, eliminar, modificar y mostrar un Carrera.
<b>Resumen</b> El caso de uso comienza cuando el actor escoge una de estas opciones. El sistema realiza la acción seleccionada, culminando el caso de uso.	
<b>Referencias</b>	R_7
<b>Prototipo</b>	Anexo 13

<b>CU 8</b>	<b>Gestionar Tipo de Carrera</b>
<b>Actores</b>	Administrador de Red
<b>Propósito</b>	Permite adicionar, eliminar, modificar y mostrar un Tipo de Carrera.
<b>Resumen</b> El caso de uso comienza cuando el actor escoge una de estas opciones. El sistema realiza la acción seleccionada, culminando el caso de uso.	
<b>Referencias</b>	R_8
<b>Prototipo</b>	Anexo 15

<b>CU 9</b>	<b>Eliminar usuario de dominio</b>
<b>Actores</b>	Administrador de Red
<b>Propósito</b>	Permite eliminar un usuario de dominio.
<b>Resumen</b> El caso de uso comienza cuando el actor visualiza las solicitudes de eliminación y valida las pertinentes. El sistema realiza la acción seleccionada, culminando el caso de uso.	
<b>Referencias</b>	R_11
<b>Prototipo</b>	Anexo 17

<b>CU 10</b>	<b>Autorizar solicitud de Curso de Postgrado</b>
<b>Actores</b>	Responsable de Área
<b>Propósito</b>	Permite autorizar la creación de un Curso de Postgrado.
<b>Resumen</b> El caso de uso comienza cuando el actor visualiza las solicitudes de cursos de postgrado y valida las pertinentes. El sistema realiza la acción seleccionada, culminando el caso de uso.	
<b>Referencias</b>	R_12
<b>Prototipo</b>	Anexo 19

<b>CU 11</b>	<b>Autorizar solicitud de Estudiante de Postgrado</b>
<b>Actores</b>	Administrador de Red
<b>Propósito</b>	Permite autorizar la creación de un Estudiante de Postgrado.
<b>Resumen</b> El caso de uso comienza cuando el actor visualiza las solicitudes de estudiantes de postgrado y valida las pertinentes. El sistema realiza la acción seleccionada, culminando el caso de uso.	
<b>Referencias</b>	R_13
<b>Prototipo</b>	Anexo 21

<b>CU 12</b>	<b>Mostrar Estado de Solicitud de Curso de Postgrado</b>
<b>Actores</b>	Profesor
<b>Propósito</b>	Permite mostrar el estado de la solicitud de un Curso de Postgrado.
<b>Resumen</b> El caso de uso comienza cuando el actor inicia su sesión y visualiza el estado de su solicitud de curso de postgrado. Culminando el caso de uso.	
<b>Referencias</b>	R_14
<b>Prototipo</b>	Anexo 23

<b>CU 13</b>	<b>Mostrar Estado de Solicitud de Estudiante de Postgrado</b>
<b>Actores</b>	Aspirante
<b>Propósito</b>	Permite mostrar el estado de la solicitud de un Estudiante de Postgrado.
<b>Resumen</b> El caso de uso comienza cuando el actor se identifica con los datos de una solicitud de estudiante de postgrado. El sistema realiza la acción seleccionada, culminando el caso de uso.	
<b>Referencias</b>	R_15
<b>Prototipo</b>	Anexo 25

<b>CU 14</b>	<b>Mostrar Estado de Solicitud de Modificación de Estudiante de Pregrado</b>
<b>Actores</b>	Estudiante de Pregrado
<b>Propósito</b>	Permite mostrar el estado de la solicitud de modificación de un Estudiante de Pregrado.
<b>Resumen</b> El caso de uso comienza cuando el actor selecciona la opción. El sistema busca el estado de la solicitud y lo muestra, culminando el caso de uso.	
<b>Referencias</b>	R_16
<b>Prototipo</b>	Anexo 27

<b>CU 15</b>	<b>Mostrar Solicitudes de Curso de Postgrado</b>
<b>Actores</b>	Responsable de Área
<b>Propósito</b>	Permite mostrar las solicitudes realizadas y que no han aprobado aún.
<b>Resumen</b> El caso de uso comienza cuando el actor selecciona la opción. El sistema busca las solicitudes y las muestra, culminando el caso de uso.	
<b>Referencias</b>	R_6
<b>Prototipo</b>	Anexo 29

<b>CU 16</b>	<b>Mostrar Solicitudes de Curso de Postgrado Aprobadas</b>
<b>Actores</b>	Responsable de Área
<b>Propósito</b>	Permite mostrar las solicitudes de Curso de Postgrado aprobadas para creación.
<b>Resumen</b> El caso de uso comienza cuando el actor selecciona la opción. El sistema busca las solicitudes aprobadas y las muestra, culminando el caso de uso.	
<b>Referencias</b>	R_17
<b>Prototipo</b>	Anexo 31

<b>CU 17</b>	<b>Mostrar Solicitudes de Estudiante de Postgrado</b>
<b>Actores</b>	Profesor
<b>Propósito</b>	Permite mostrar las solicitudes realizadas y que no han sido aprobadas aún.
<b>Resumen</b> El caso de uso comienza cuando el actor selecciona la opción. El sistema busca las solicitudes y las muestra, culminando el caso de uso.	
<b>Referencias</b>	R_4
<b>Prototipo</b>	Anexo 33

<b>CU 18</b>	<b>Mostrar Solicitudes de Estudiante de Postgrado Aprobadas</b>
<b>Actores</b>	Profesor
<b>Propósito</b>	Permite mostrar las solicitudes realizadas y que no han sido aprobadas aún.
<b>Resumen</b> El caso de uso comienza cuando el actor selecciona la opción. El sistema busca las solicitudes y las muestra, culminando el caso de uso.	
<b>Referencias</b>	R_18
<b>Prototipo</b>	Anexo 35

<b>CU 19</b>	<b>Mostrar Datos de Estudiante</b>
<b>Actores</b>	Estudiante de Pregrado, Estudiante de Postgrado
<b>Propósito</b>	Permite mostrar los datos de un estudiante previamente autenticado.
<b>Resumen</b> El caso de uso comienza cuando el actor se autentica en el sistema. El sistema busca sus datos y los muestra, culminando el caso de uso.	
<b>Referencias</b>	R_21
<b>Prototipo</b>	Anexo 37

<b>CU 20</b>	<b>Mostrar Estudiante de Pregrado por Carreras</b>
<b>Actores</b>	Administrador de Red, Profesor
<b>Propósito</b>	Permite mostrar los estudiantes de pregrado organizados por carreras.
<b>Resumen</b> El caso de uso comienza cuando el actor selecciona la carrera. El sistema busca los estudiantes, culminando el caso de uso.	
<b>Referencias</b>	R_19
<b>Prototipo</b>	Anexo 39

<b>CU 21</b>	<b>Mostrar Estudiante de Postgrado por Curso de Postgrado</b>
<b>Actores</b>	Administrador de Red, Profesor
<b>Propósito</b>	Permite mostrar los Estudiantes de Postgrado organizados por Curso de Postgrado.
<b>Resumen</b> El caso de uso comienza cuando el actor selecciona el Curso de Postgrado. El sistema busca los estudiantes, culminando el caso de uso.	
<b>Referencias</b>	R_20
<b>Prototipo</b>	Anexo 41

<b>CU 22</b>	<b>Solicitar Modificación de usuario de dominio</b>
<b>Actores</b>	usuario
<b>Propósito</b>	Permite solicitar un cambio en el rol desempeñado por el actor en el sistema.
<b>Resumen</b> El caso de uso comienza cuando el actor selecciona la opción porque ha cambiado su estatus en la UNISS. El sistema muestra el formulario y procesa los datos que se hayan introducido, culminando el caso de uso.	
<b>Referencias</b>	R_22
<b>Prototipo</b>	

<b>CU 23</b>	<b>Solicitar Eliminación de usuario de dominio</b>
<b>Actores</b>	usuario
<b>Propósito</b>	Permite solicitar la eliminación de un usuario del directorio activo.
<b>Resumen</b> El caso de uso comienza cuando el actor selecciona la opción porque ha causado baja de la UNISS. El sistema muestra el formulario y procesa los datos que se hayan introducido, culminando el caso de uso.	
<b>Referencias</b>	R_23
<b>Prototipo</b>	

<b>CU 24</b>	<b>Solicitar creación de Curso de Postgrado</b>
<b>Actores</b>	Profesor
<b>Propósito</b>	Permite solicitar la creación de un Curso de Postgrado
<b>Resumen</b> El caso de uso comienza cuando el actor selecciona la opción. El sistema muestra el formulario para introducir los datos, los procesa y culmina el caso de uso.	
<b>Referencias</b>	R_22
<b>Prototipo</b>	Anexo 45



<b>CU 25</b>	<b>Solicitar creación de usuario de dominio para Estudiante de Postgrado</b>
<b>Actores</b>	Aspirante
<b>Propósito</b>	Permite solicitar la creación de un usuario de dominio para un Estudiante de Postgrado.
<b>Resumen</b>	
El caso de uso comienza cuando el actor selecciona la opción. El sistema muestra el formulario para introducir los datos, los procesa y culmina el caso de uso.	
<b>Referencias</b>	R_23
<b>Prototipo</b>	Anexo 43

## Anexo 4. Prototipos de Interfaz



Figura 15. *PROTOTIPO DE INTERFAZ: INICIAR SESIÓN.*

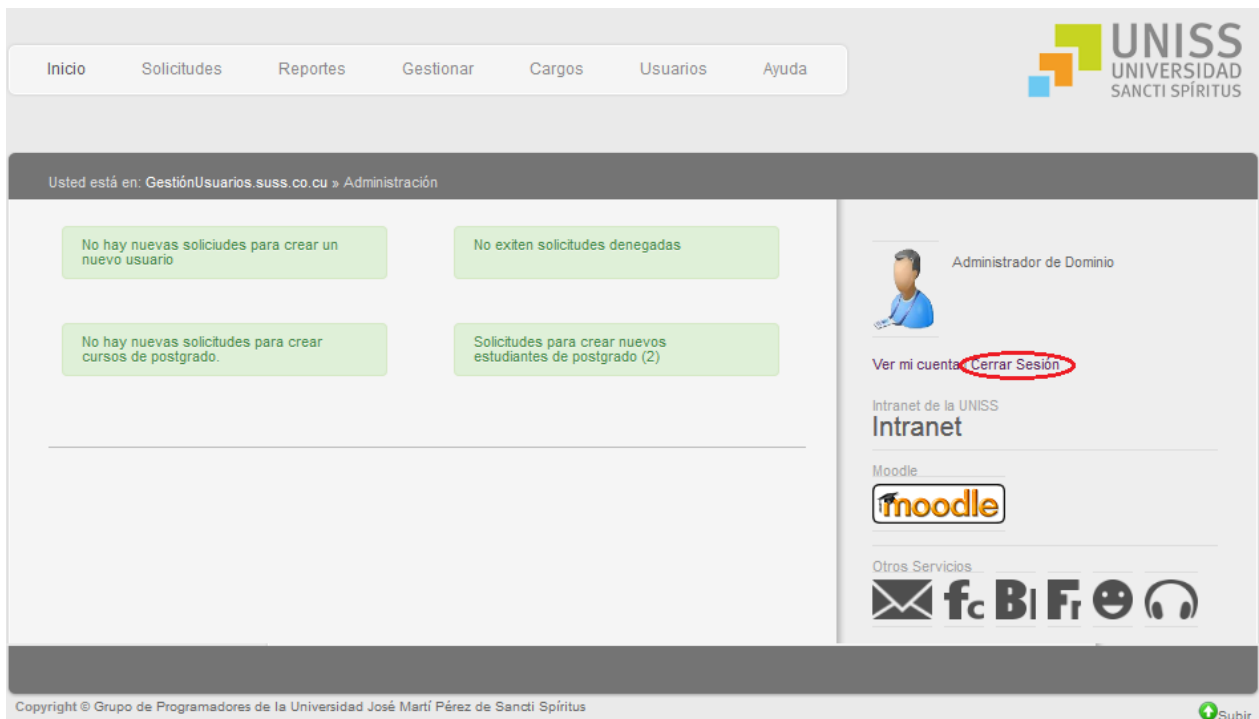


Figura 16. *PROTOTIPO DE INTERFAZ: CERRAR SESIÓN.*



Figura 17. *PROTOTIPO DE INTERFAZ: CERRAR SESIÓN.*



Figura 18. *PROTOTIPO DE INTERFAZ: GESTIONAR ESTUDIANTE DE POSTGRADO.*

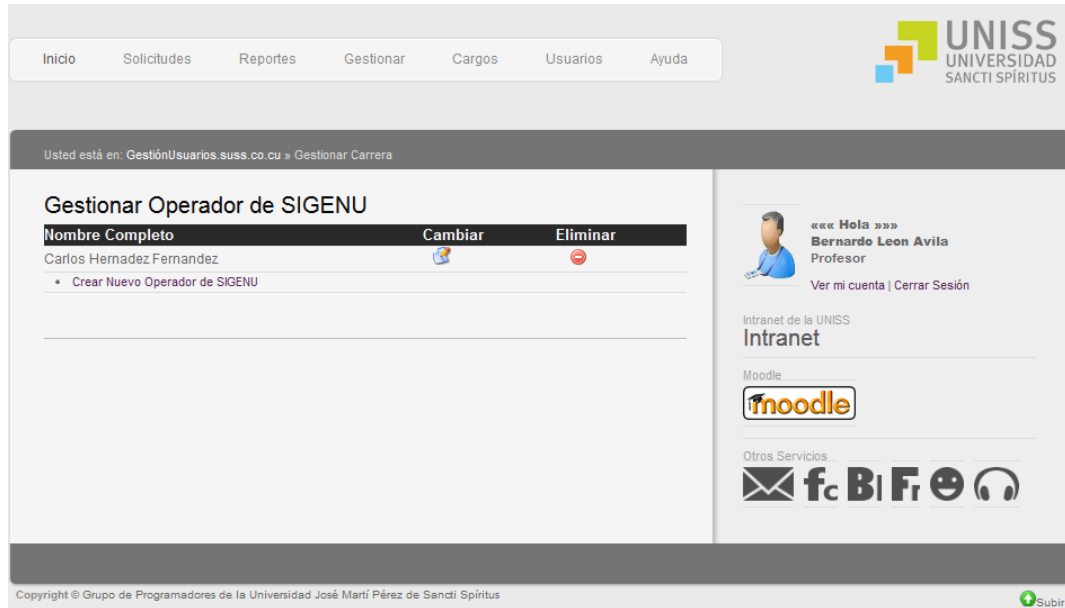


Figura 19. PROTOTIPO DE INTERFAZ: GESTIONAR OPERADOR DE SIGENU.

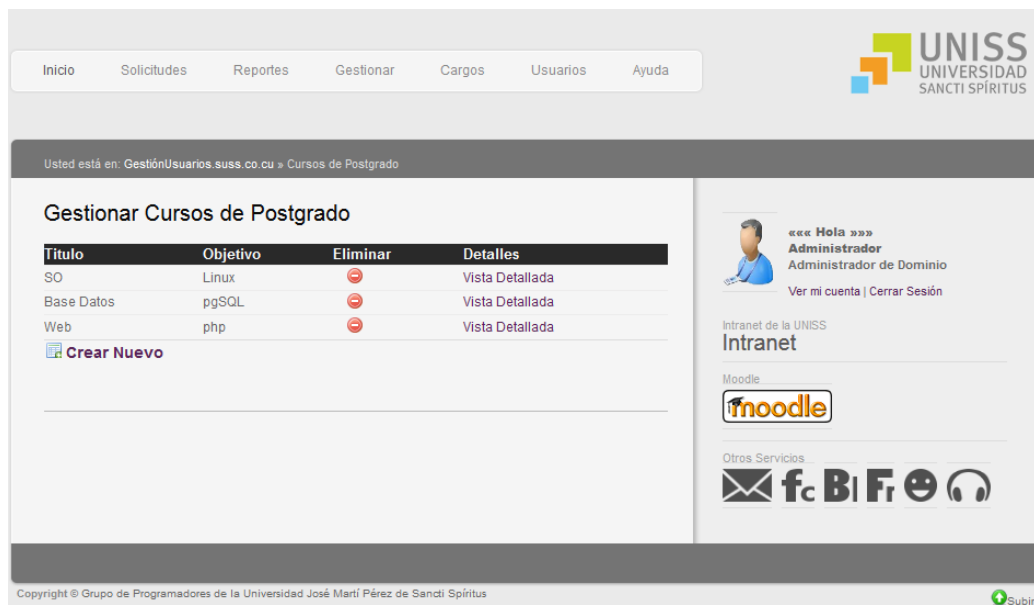


Figura 20. PROTOTIPO DE INTERFAZ: GESTIONAR CURSO DE POSTGRADO.

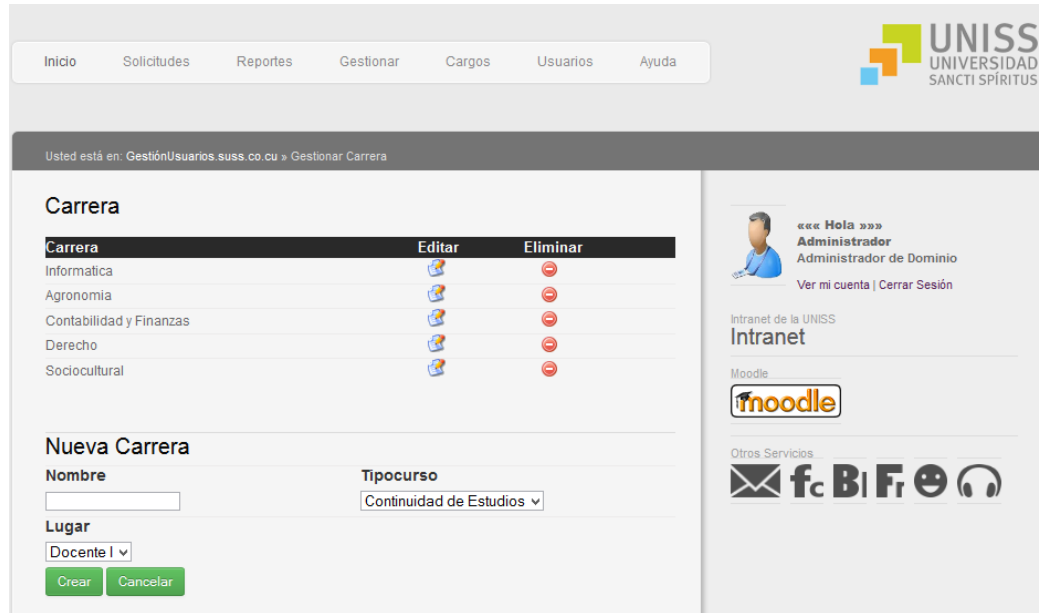


Figura 21. *PROTOTIPO DE INTERFAZ: GESTIONAR CARRERA.*

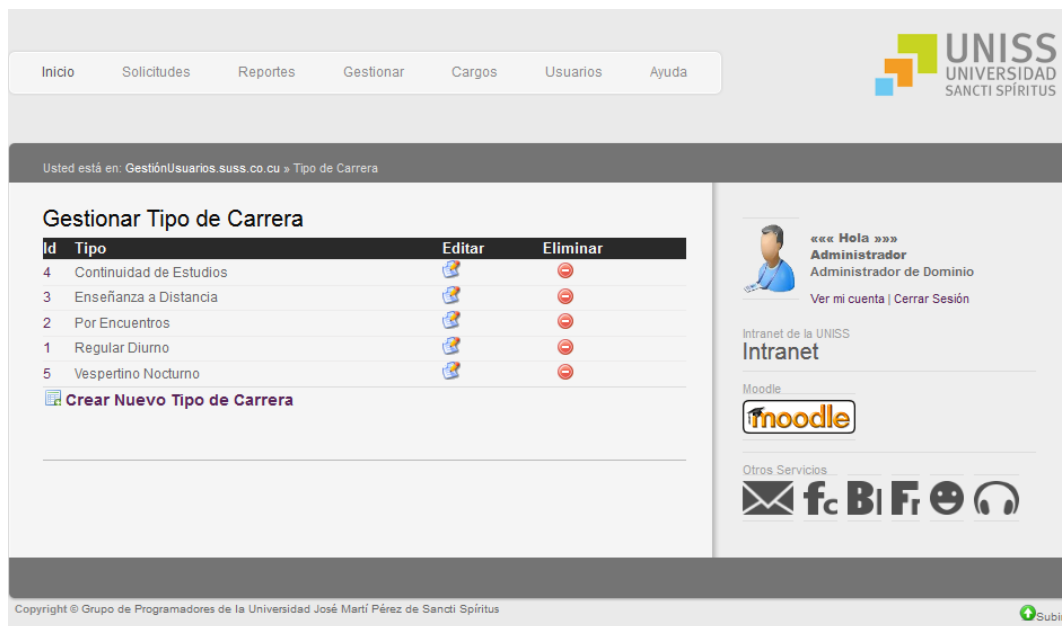


Figura 22. *PROTOTIPO DE INTERFAZ: GESTIONAR TIPO DE CARRERA.*

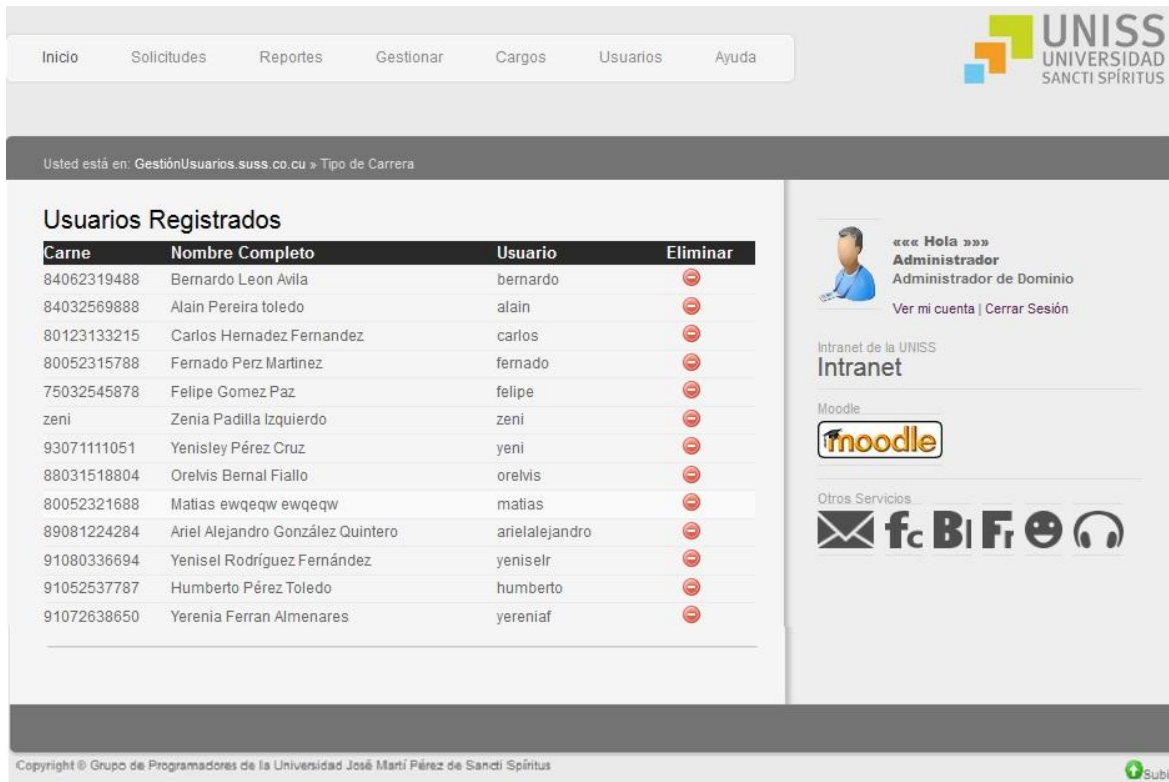


Figura 23. PROTOTIPO DE INTERFAZ: ELIMINAR USUARIO DE DOMINIO.

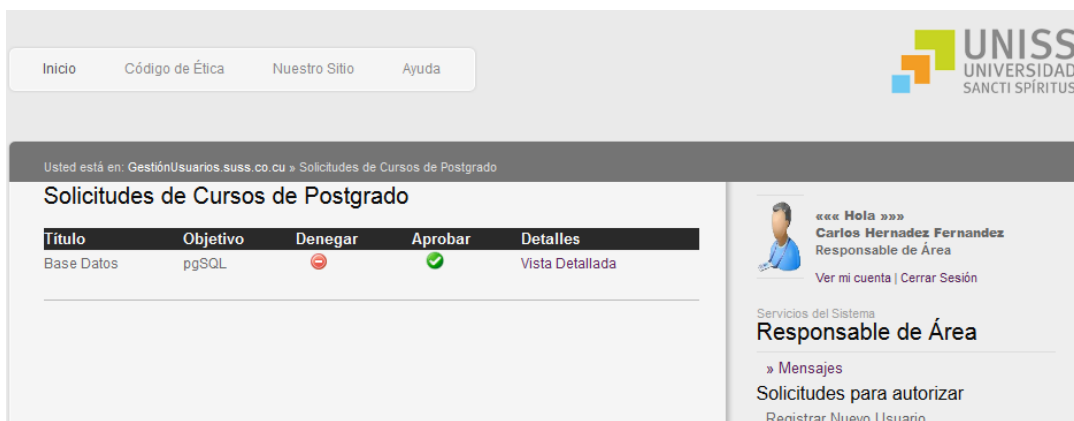


Figura 24. PROTOTIPO DE INTERFAZ: AUTORIZAR SOLICITUD DE CURSO DE POSTGRADO.



Figura 25. *PROTOTIPO DE INTERFAZ: AUTORIZAR SOLICITUD DE ESTUDIANTE DE POSTGRADO.*



Figura 26. *PROTOTIPO DE INTERFAZ: MOSTRAR ESTADO DE SOLICITUD DE CURSO DE POSTGRADO.*



Figura 27. *PROTOTIPO DE INTERFAZ: MOSTRAR SOLICITUDES DE CURSO DE POSTGRADO.*

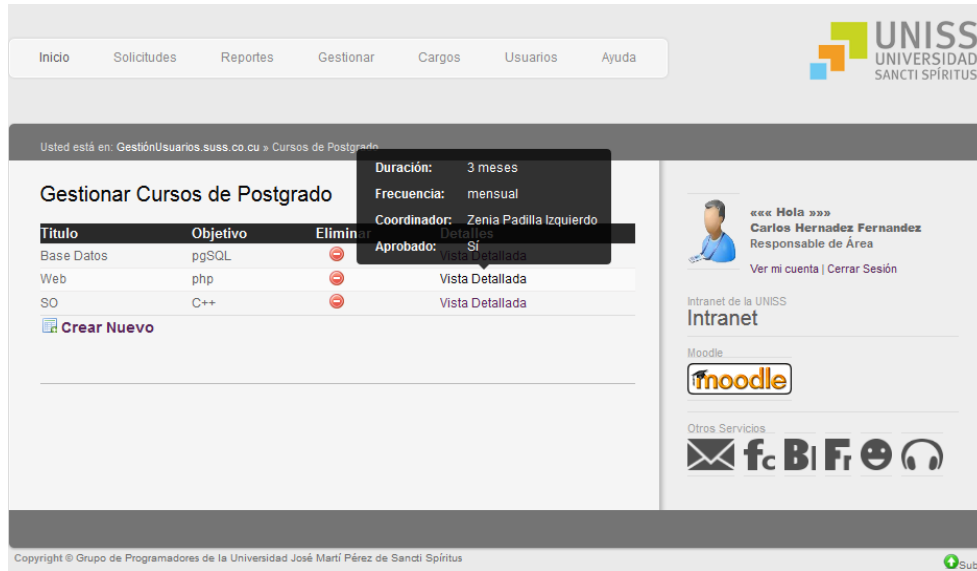


Figura 28. *PROTOTIPO DE INTERFAZ: MOSTRAR SOLICITUDES DE CURSO DE POSTGRADO APROBADAS.*



Figura 29. *PROTOTIPO DE INTERFAZ: MOSTRAR SOLICITUDES DE ESTUDIANTE DE POSTGRADO.*



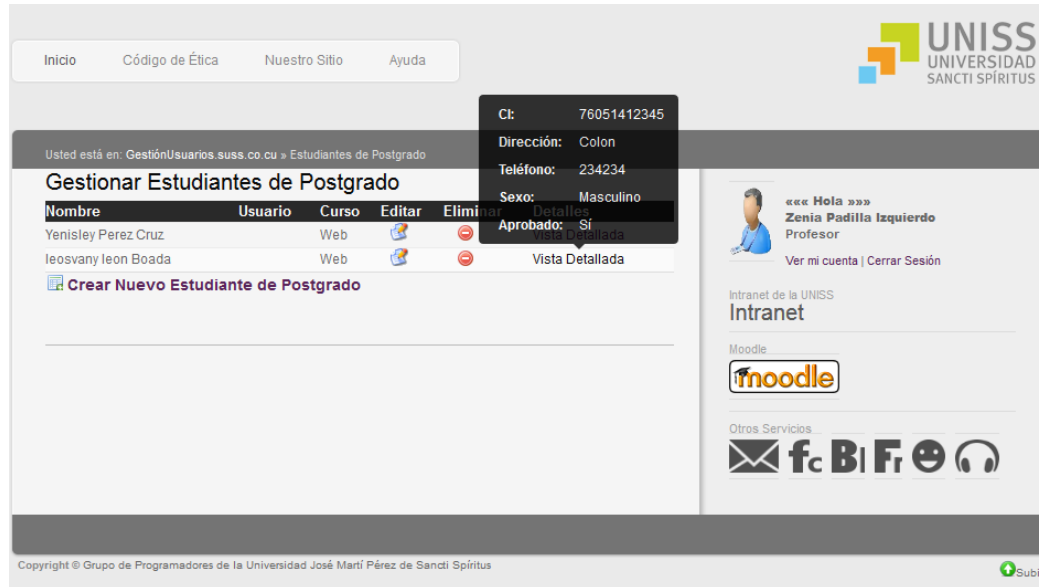


Figura 30. *PROTOTIPO DE INTERFAZ: MOSTRAR SOLICITUDES DE ESTUDIANTE DE POSTGRADO APROBADAS.*

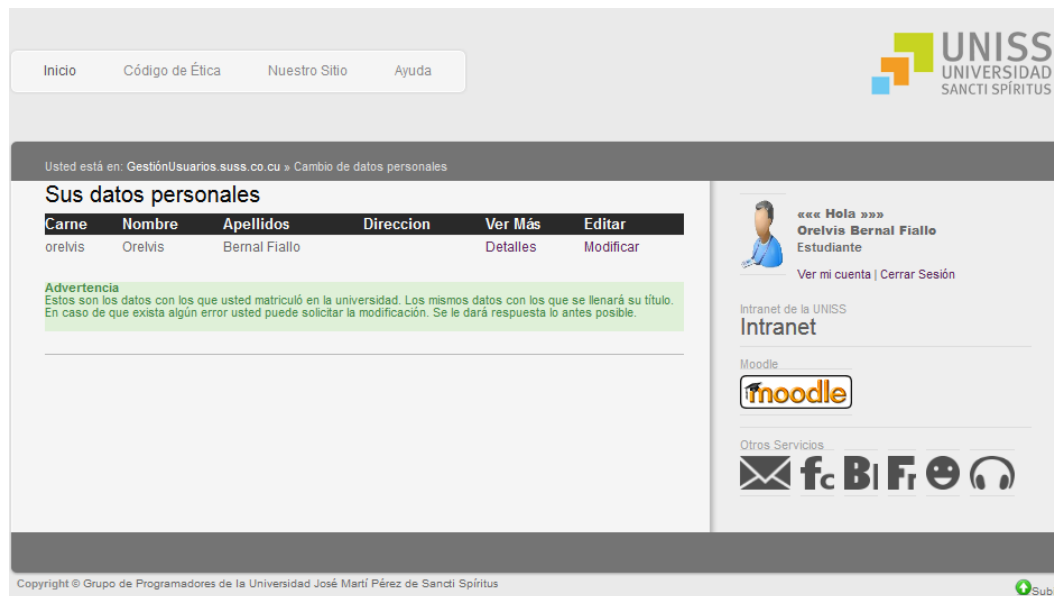


Figura 31. *PROTOTIPO DE INTERFAZ: MOSTRAR DATOS DE ESTUDIANTE.*



Figura 32. PROTOTIPO DE INTERFAZ: MOSTRAR ESTUDIANTES DE PREGRADO POR CARRERA.



Figura 33. PROTOTIPO DE INTERFAZ: MOSTRAR ESTUDIANTES DE POSTRADO POR CURSO.

The screenshot shows a web interface for requesting a postgraduate student. At the top, there is a navigation menu with 'Inicio', 'Código de Ética', 'Nuestro Sitio', and 'Ayuda'. The UNISS logo (UNIVERSIDAD SANCTI SPIRITUS) is in the top right. Below the navigation is a breadcrumb trail: 'Usted está en: GestiónUsuarios.suss.co.cu » Estudiantes de Postgrado'. The main heading is 'Solicitar estudiante de postgrado'. The form contains several fields: 'Nombre' (filled with 'José'), 'Carné de Identidad' (filled with '88021518804'), 'Primer Apellido' (filled with 'Pérez'), 'Segundo Apellido' (filled with 'Cancio'), 'Dirección' (filled with 'Olivos'), 'Sexo' (dropdown menu with 'Masculino' selected), 'Teléfono' (empty), 'Curso de Postgrado' (dropdown menu with 'Web' selected), and 'Correo Electrónico' (empty). A green 'Registrar' button is at the bottom left. On the right side, there is a sidebar with 'Intranet de la UNISS', 'Intranet', 'Moodle' (with a Moodle logo), and 'Otros Servicios' (with icons for email, Facebook, Blogger, Flickr, and a smiley face).

Figura 34. PROTOTIPO DE INTERFAZ: SOLICITAR CREACIÓN DE ESTUDIANTE DE POSTGRADO.

The screenshot shows a web interface for requesting the creation of a postgraduate course. At the top, there is a navigation menu with 'Inicio', 'Código de Ética', 'Nuestro Sitio', and 'Ayuda'. The UNISS logo (UNIVERSIDAD SANCTI SPIRITUS) is in the top right. Below the navigation is a breadcrumb trail: 'Usted está en: GestiónUsuarios.suss.co.cu » Registrar Curso de Postgrado'. The main heading is 'Solicitar creación de Postgrado'. The form contains several fields: 'Título' (empty), 'Objetivo' (empty), 'Duración' (empty), 'Frecuencia' (empty), 'Area' (dropdown menu with 'Rectorado' selected), and 'Coordinador' (filled with 'Bernardo Leon Avila'). A green 'Registrar' button is at the bottom left. Below the form is a green warning box: 'Advertencia Al solicitar la inserción de un curso de postgrado usted se pondrá en la solicitud como el coordinador del curso.' On the right side, there is a sidebar with a user profile for 'Bernardo Leon Avila' (Profesor) with a 'Ver mi cuenta | Cerrar Sesión' link. Below that are 'Servicios del Sistema' (Profesor), 'Inicio', and 'Reportes' (with links for 'Solicitudes de matrícula' and 'Solicitar curso de postgrado'). At the bottom of the sidebar are 'Intranet de la UNISS', 'Intranet', and 'Moodle' (with a Moodle logo).

Figura 35. PROTOTIPO DE INTERFAZ: SOLICITAR CREACIÓN DE CURSO DE POSTGRADO.

### Anexo 5. Diagramas de clases

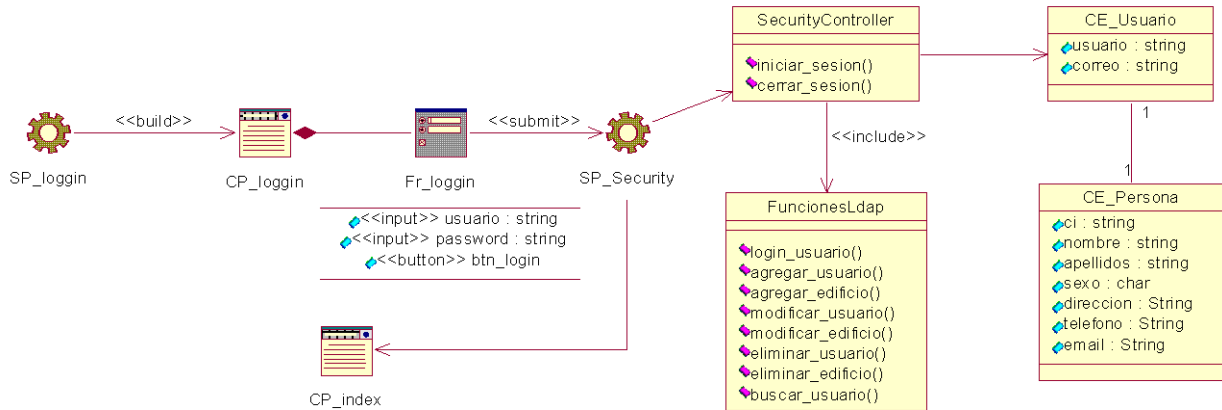


Figura 36. DIAGRAMA DE CLASES DEL DISEÑO: INICIAR SESIÓN.

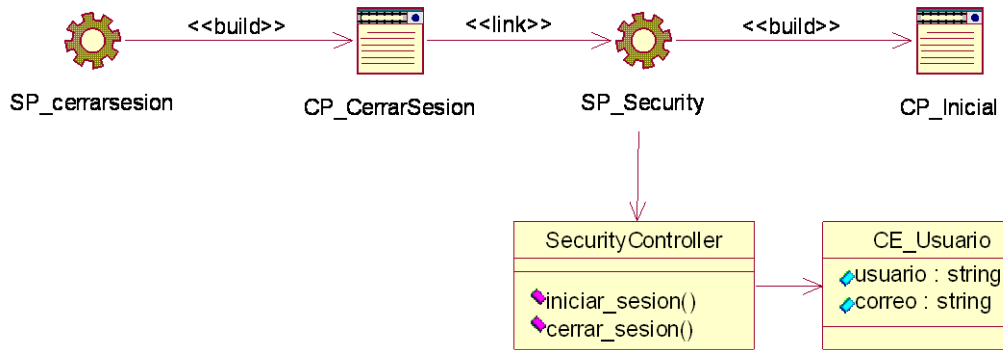


Figura 37. DIAGRAMA DE CLASES DEL DISEÑO: CERRAR SESIÓN.

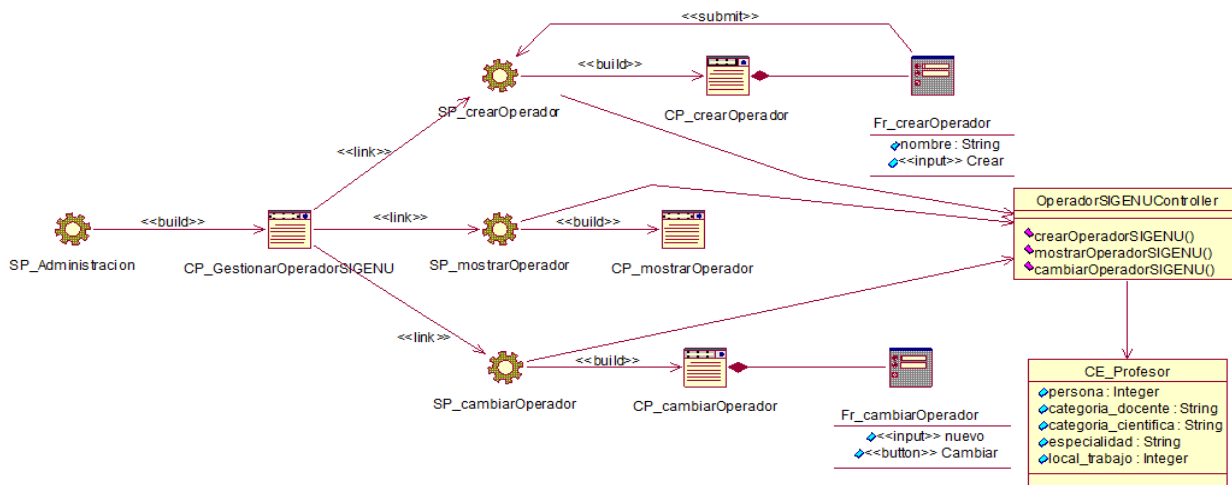


Figura 38. DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR OPERADOR DE SIGENU.

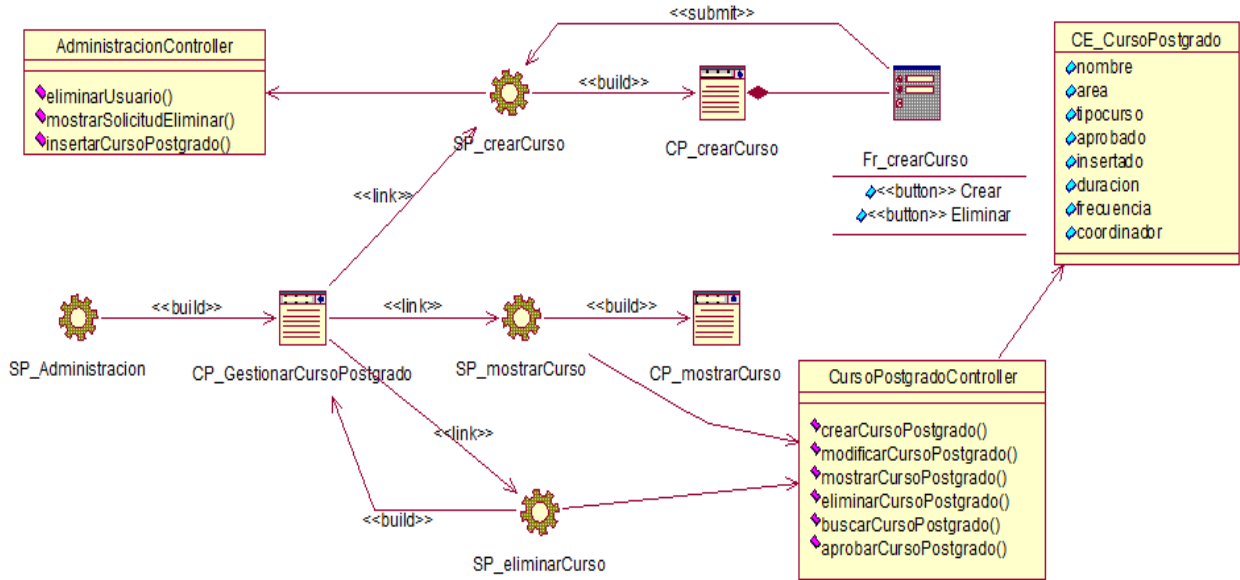


Figura 39. DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR CURSO DE POSTGRADO.

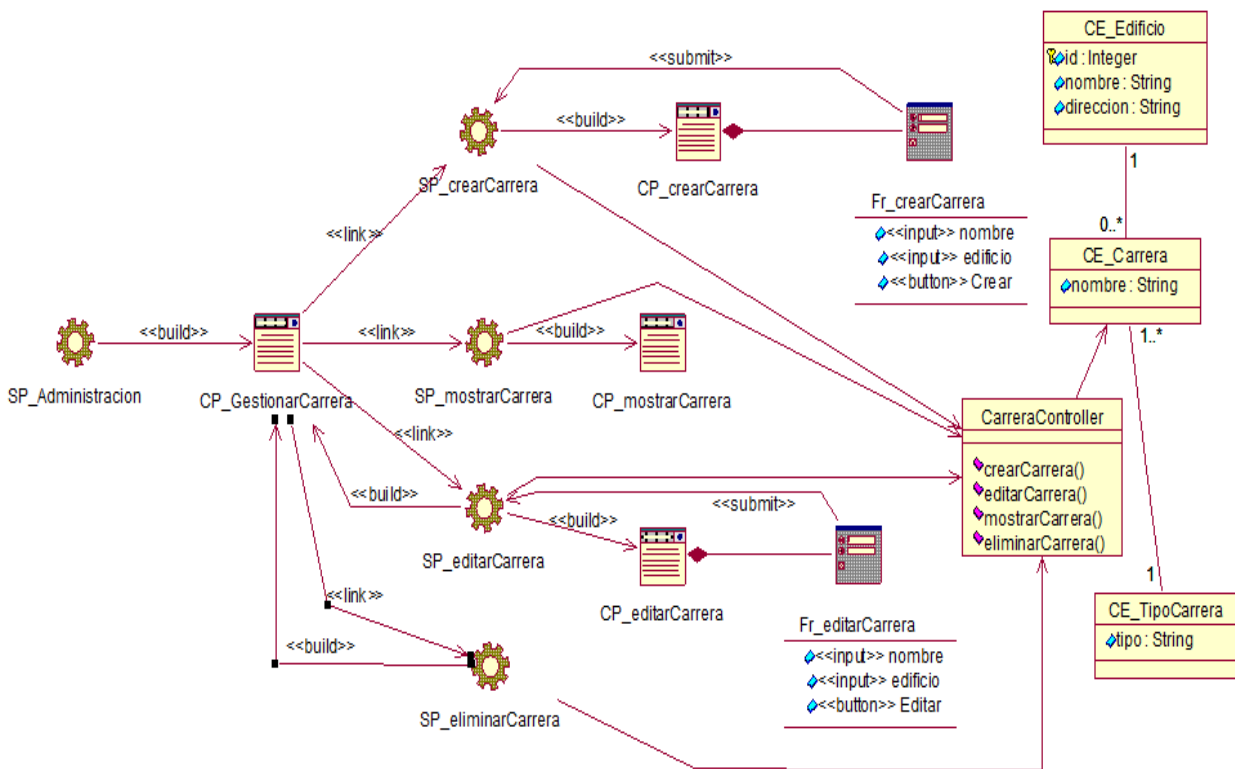


Figura 40. DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR CARRERA.

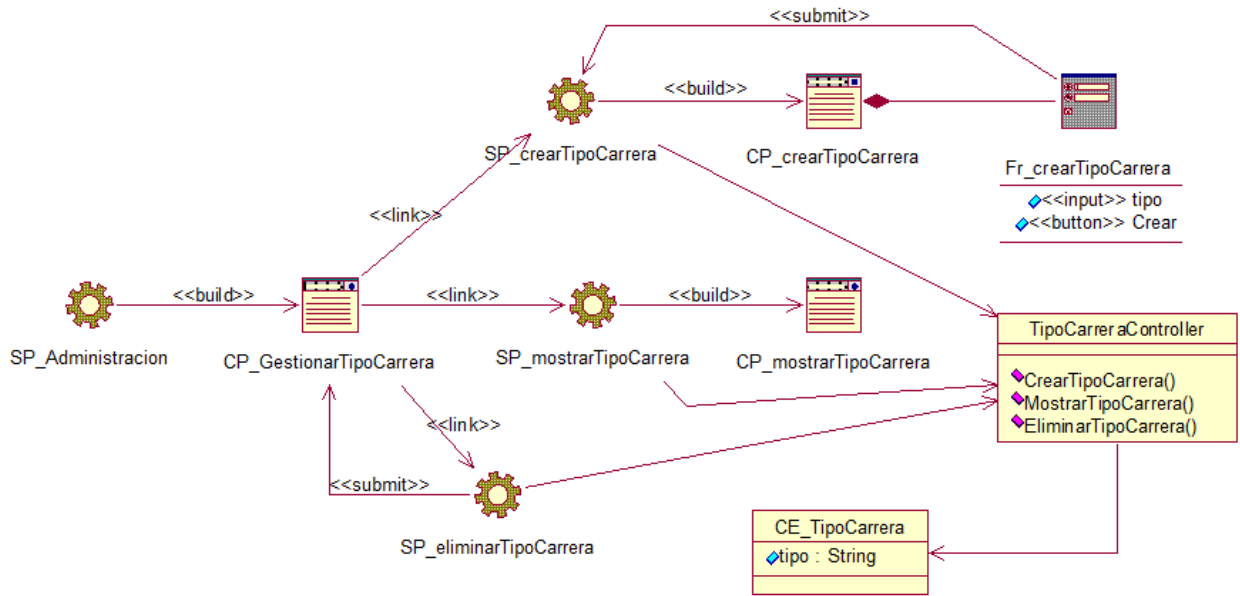


Figura 41. DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR TIPO DE CARRERA.

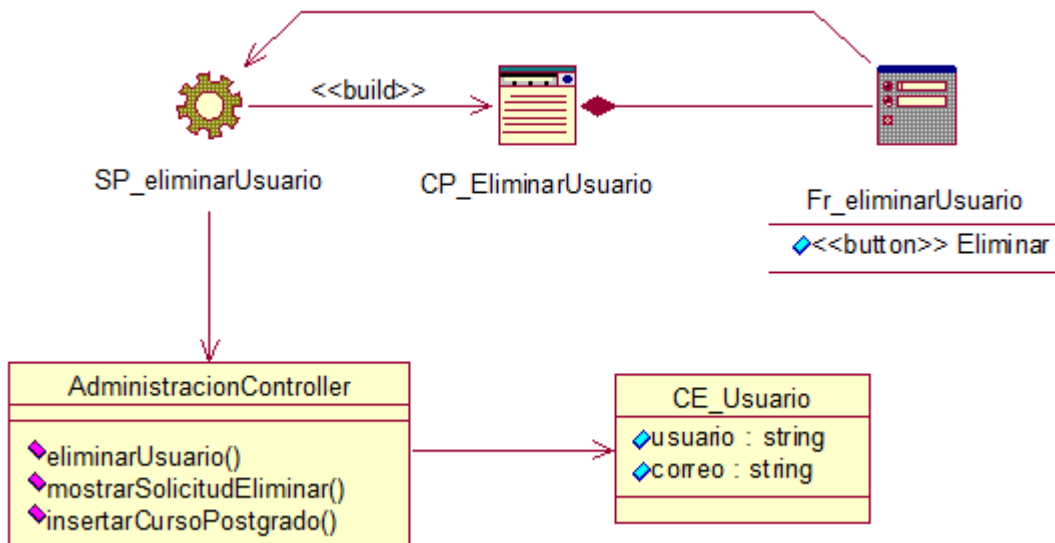


Figura 42. DIAGRAMA DE CLASES DEL DISEÑO: ELIMINAR USUARIO.

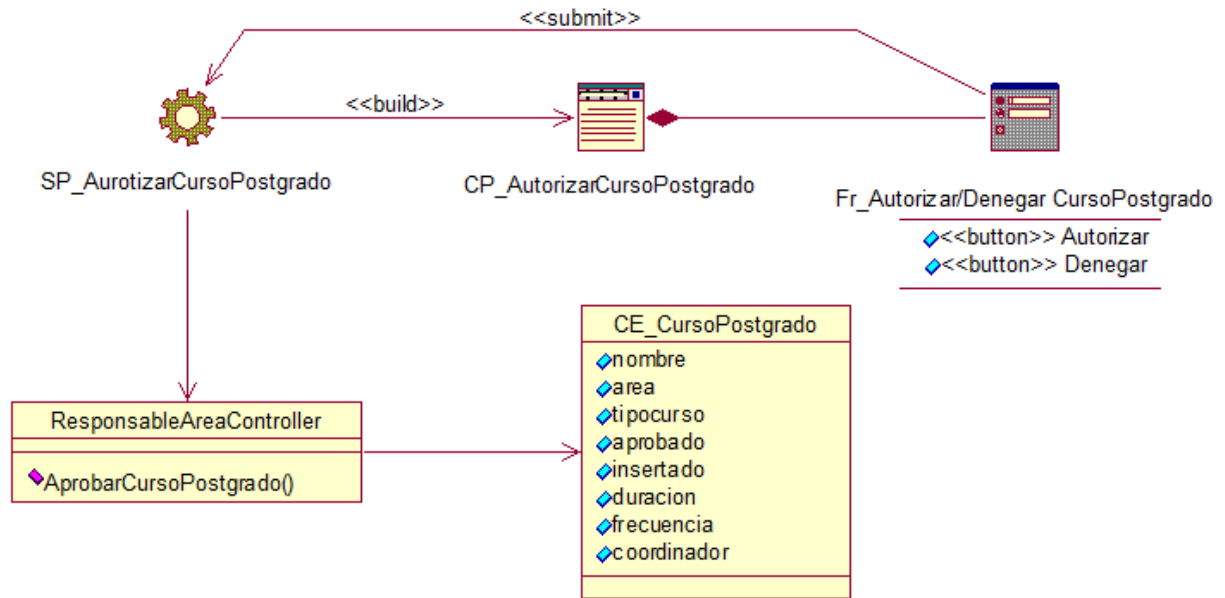


Figura 43. DIAGRAMA DE CLASES DEL DISEÑO: AUTORIZAR CURSO DE POSTGRADO.

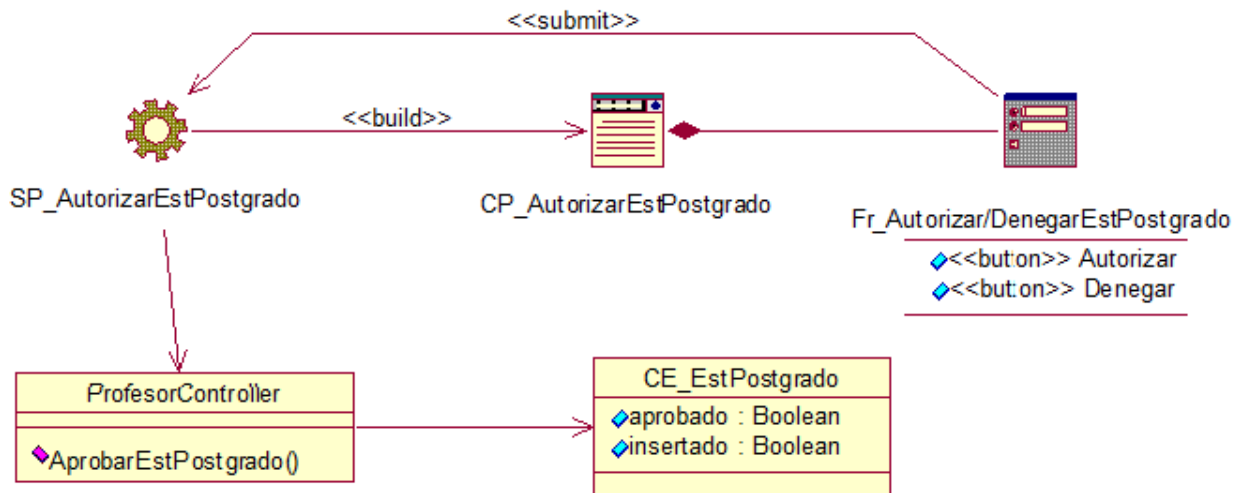


Figura 44. DIAGRAMA DE CLASES DEL DISEÑO: AUTORIZAR ESTUDIANTE DE POSTGRADO.

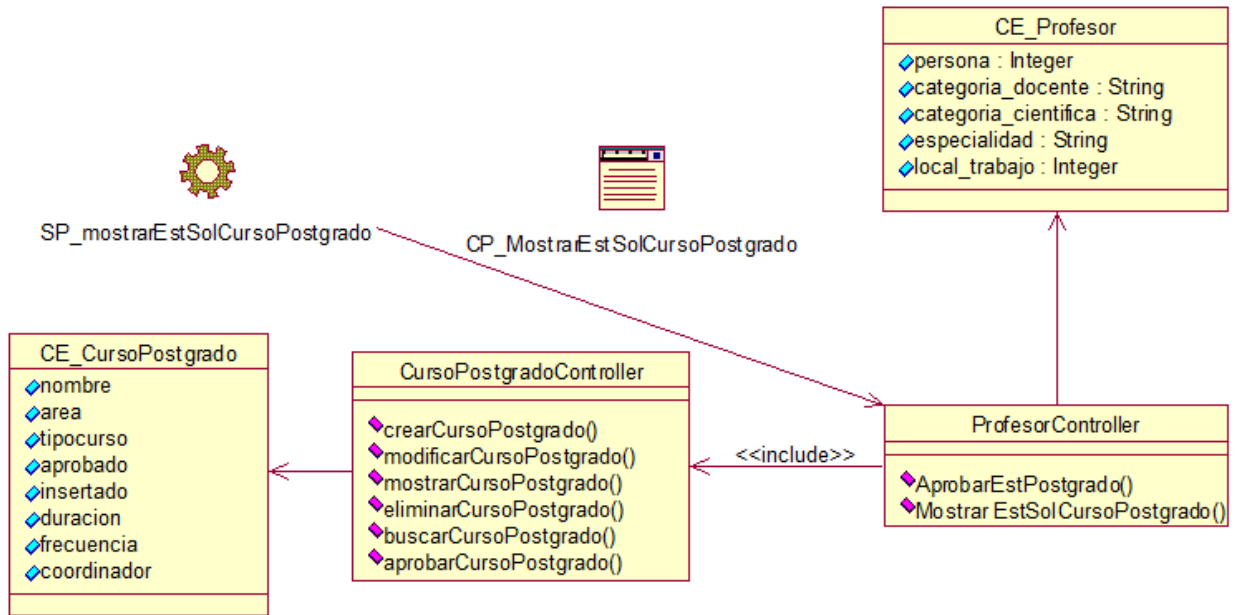


Figura 45. DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR ESTADO DE SOLICITUD DE ESTUDIANTE DE POSTGRADO.

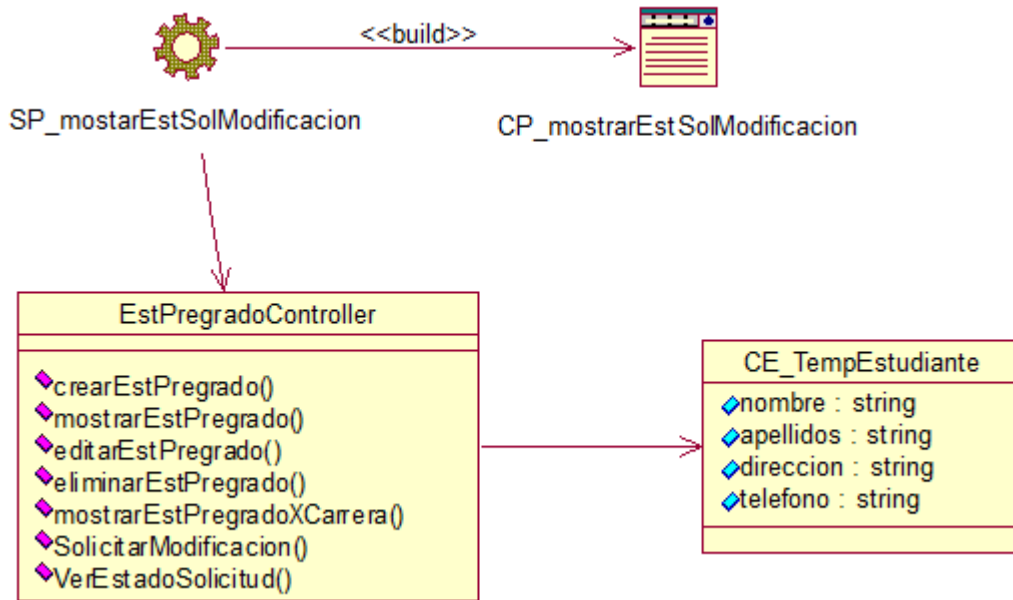


Figura 46. DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR ESTADO DE SOLICITUD DE POSTGRADO.



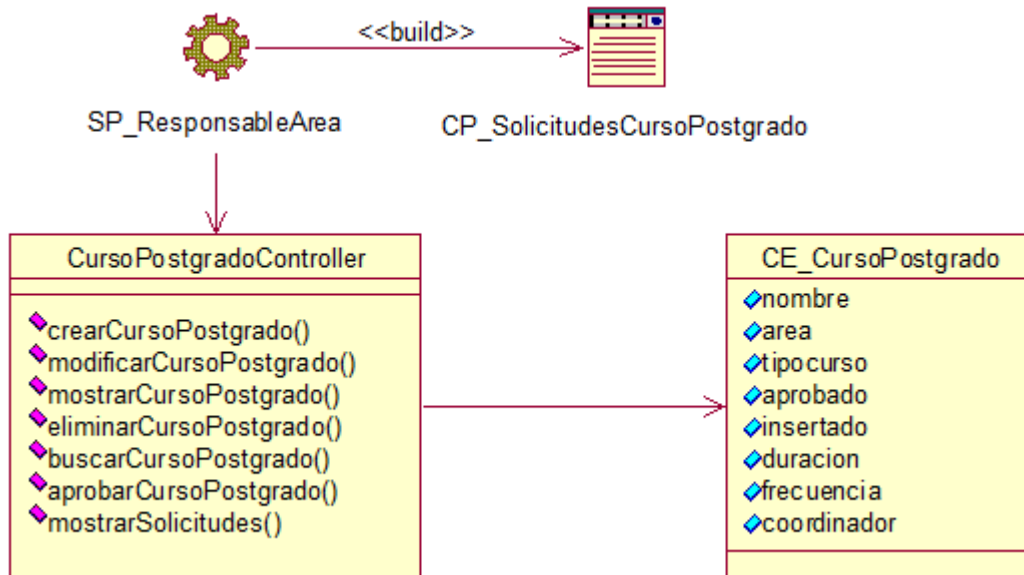


Figura 47. DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR SOLICITUDES DE CURSO DE POSTGRADO.

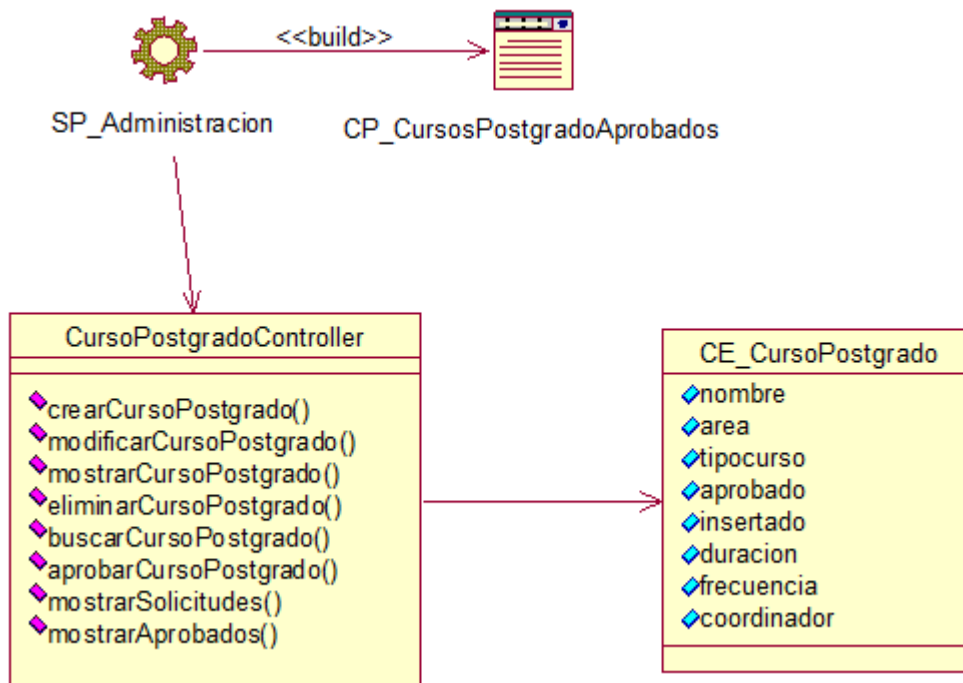


Figura 48. DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR SOLICITUDES DE POSTGRADO APROBADOS.

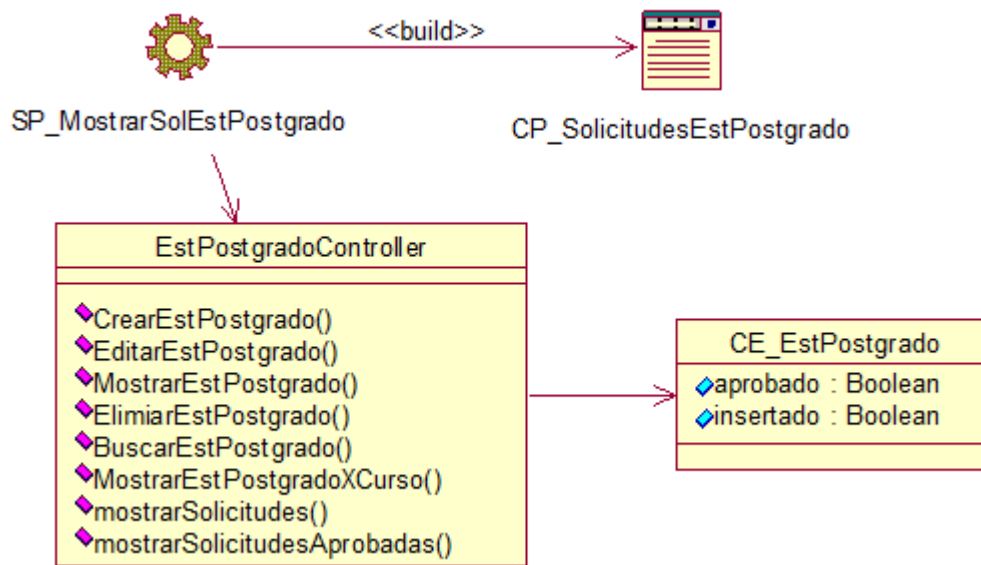


Figura 49. DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR SOLICITUDES DE ESTUDIANTE DE POSTGRADO.

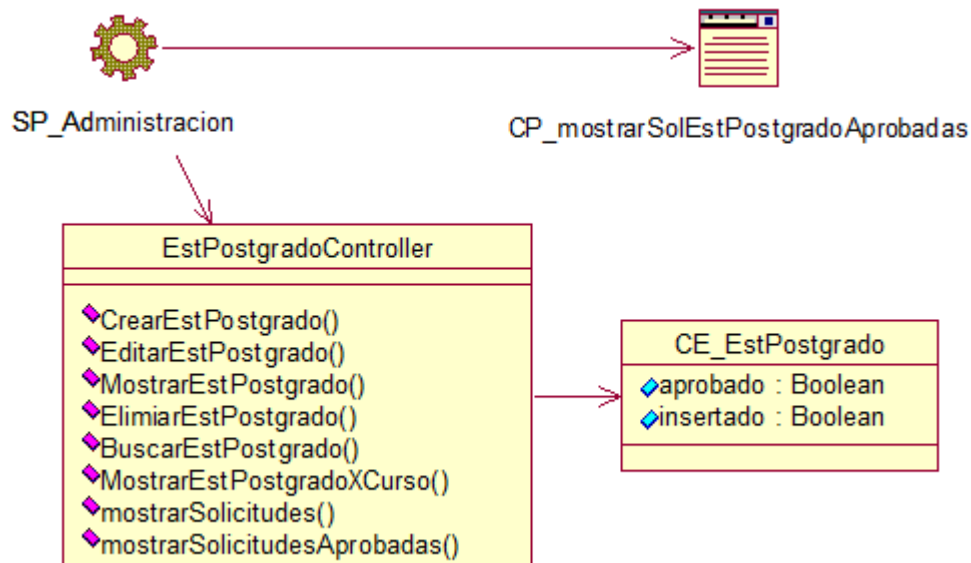


Figura 50. DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR SOLICITUDES DE ESTUDIANTE DE POSTGRADO APROBADAS.

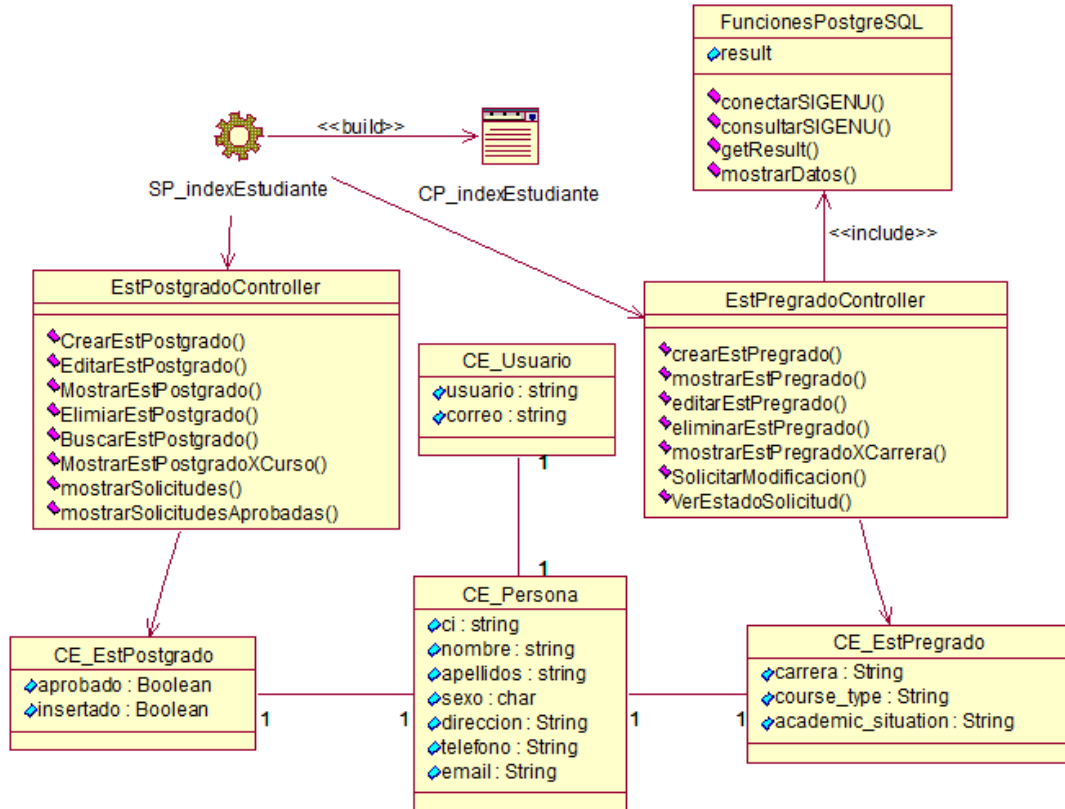


Figura 51. DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR DATOS DE ESTUDIANTE.

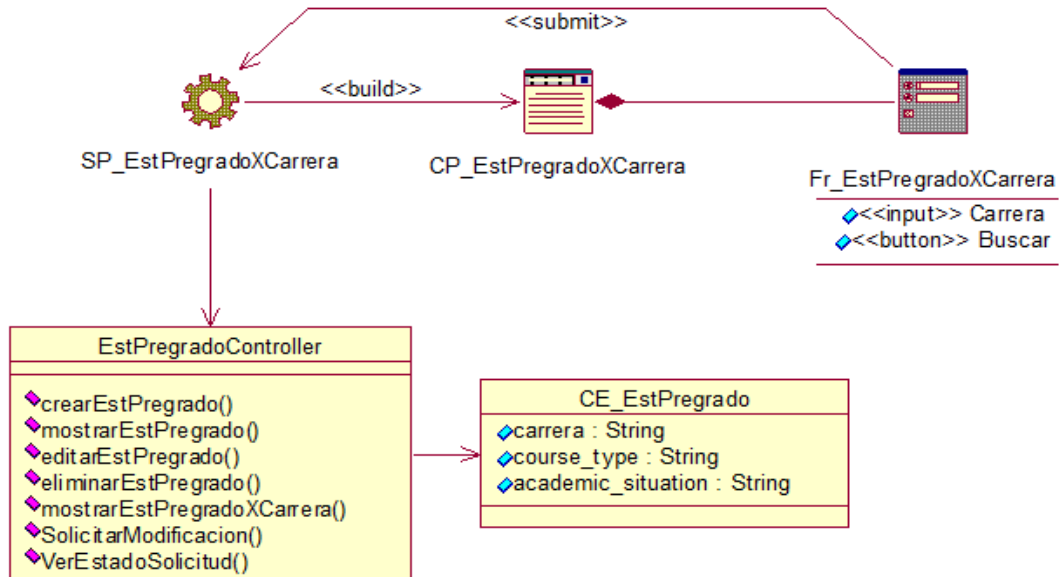


Figura 52. DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR ESTUDIANTE DE PREGRADO POR CARRERA.

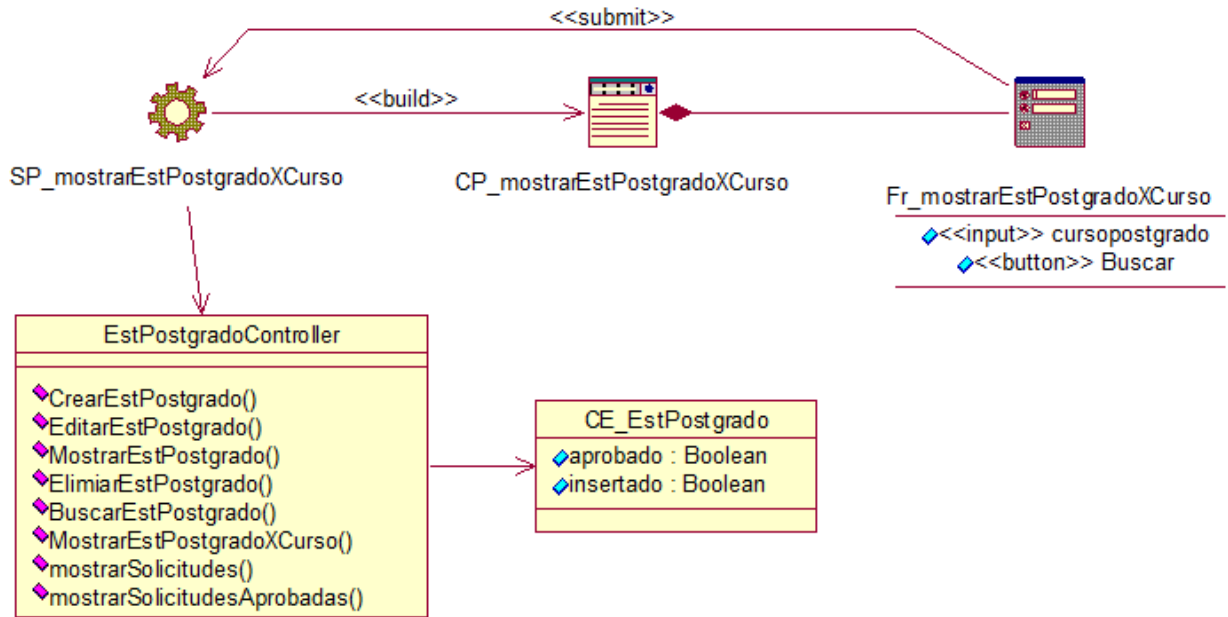


Figura 53. DIAGRAMA DE CLASES DEL DISEÑO: MOSTRAR ESTUDIANTE DE POSTGRADO POR CURSO.

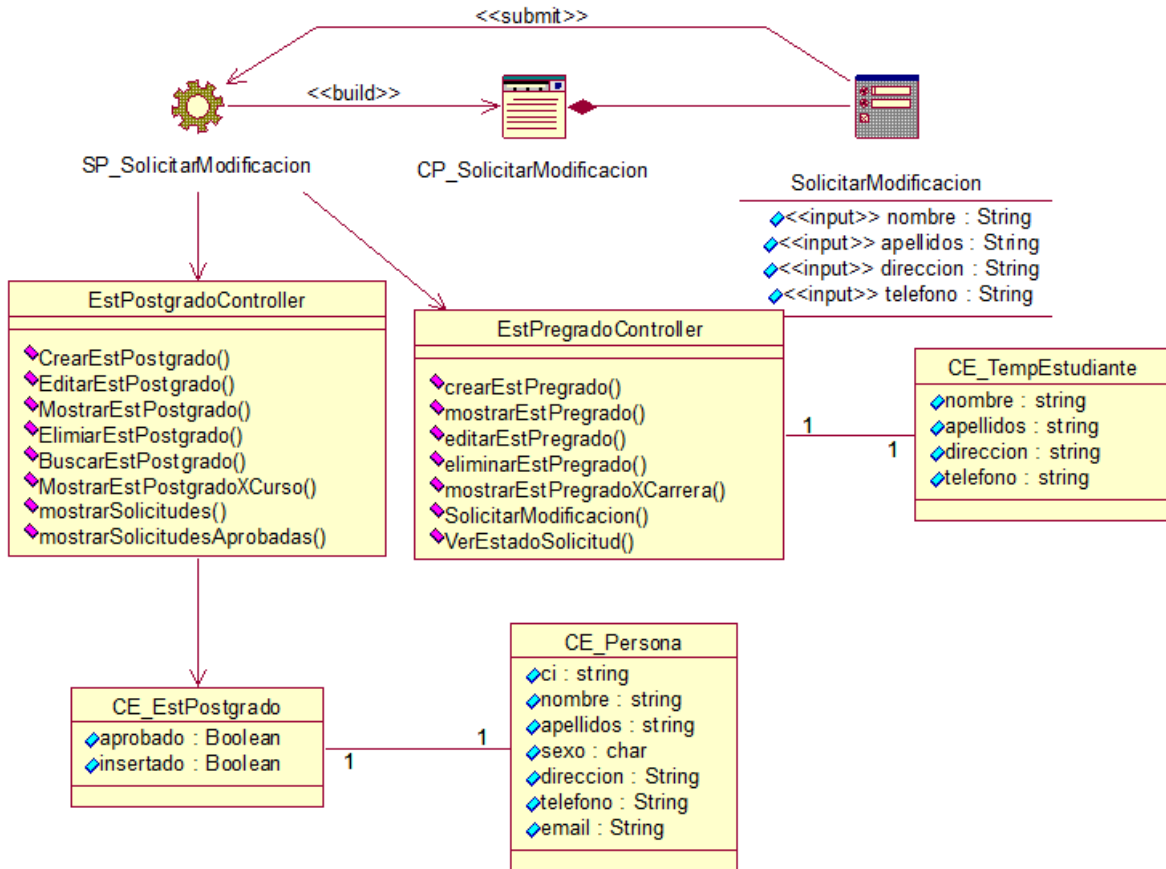


Figura 54. DIAGRAMA DE CLASES DEL DISEÑO: SOLICITAR MODIFICACIÓN DE USUARIO DE DOMINIO.

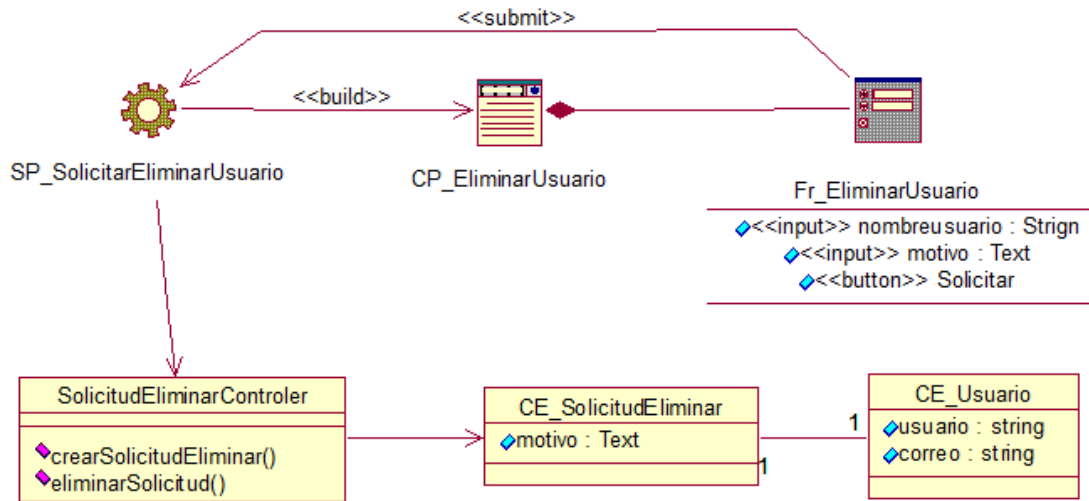


Figura 55. DIAGRAMA DE CLASES DEL DISEÑO: SOLICITAR ELIMINACIÓN DE USUARIO DE DOMINIO.

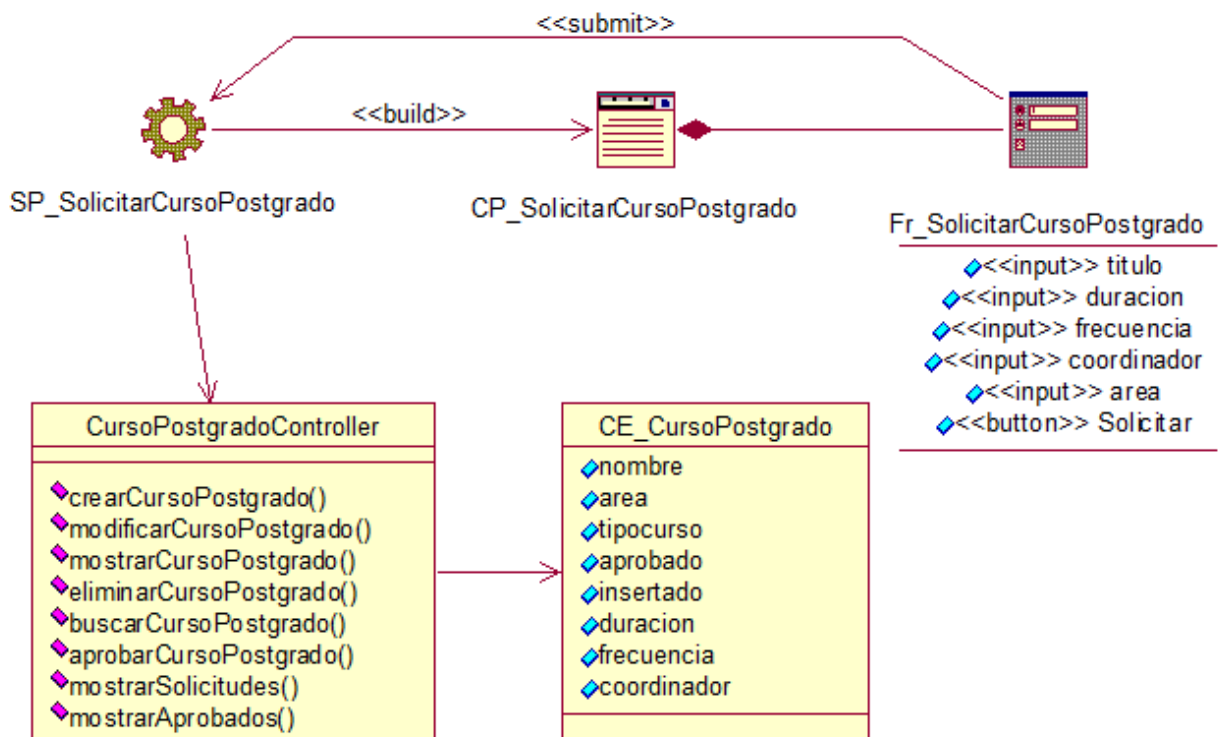


Figura 56. DIAGRAMA DE CLASES DEL DISEÑO: SOLICITAR CURSO DE POSTGRADO.

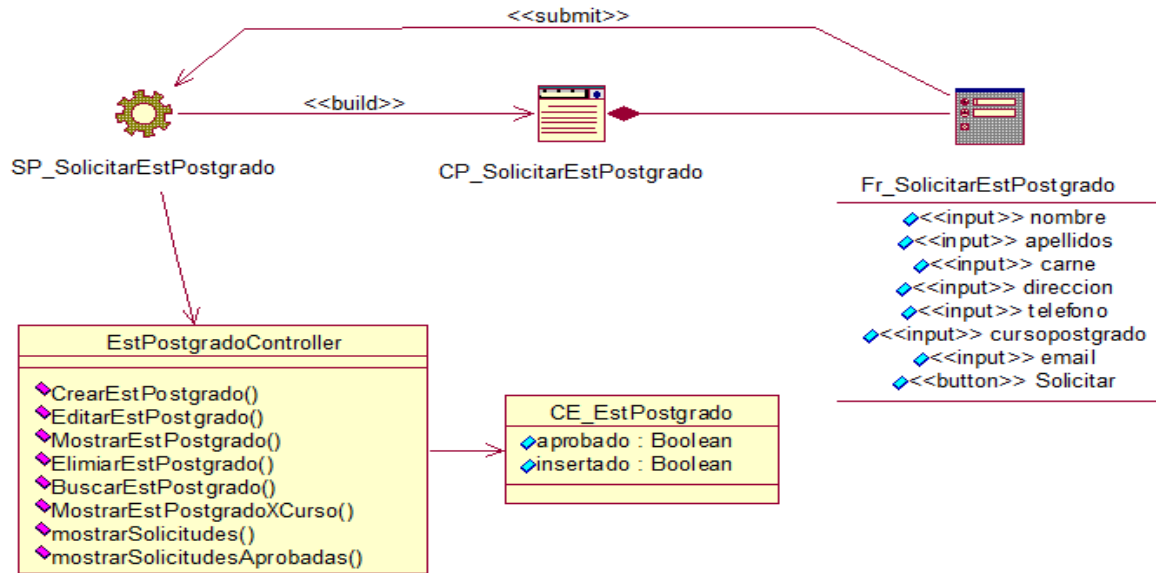


Figura 57. DIAGRAMA DE CLASES DEL DISEÑO: SOLICITAR CURSO DE POSTGRADO.

## Anexo 6. Modelo de Objetos

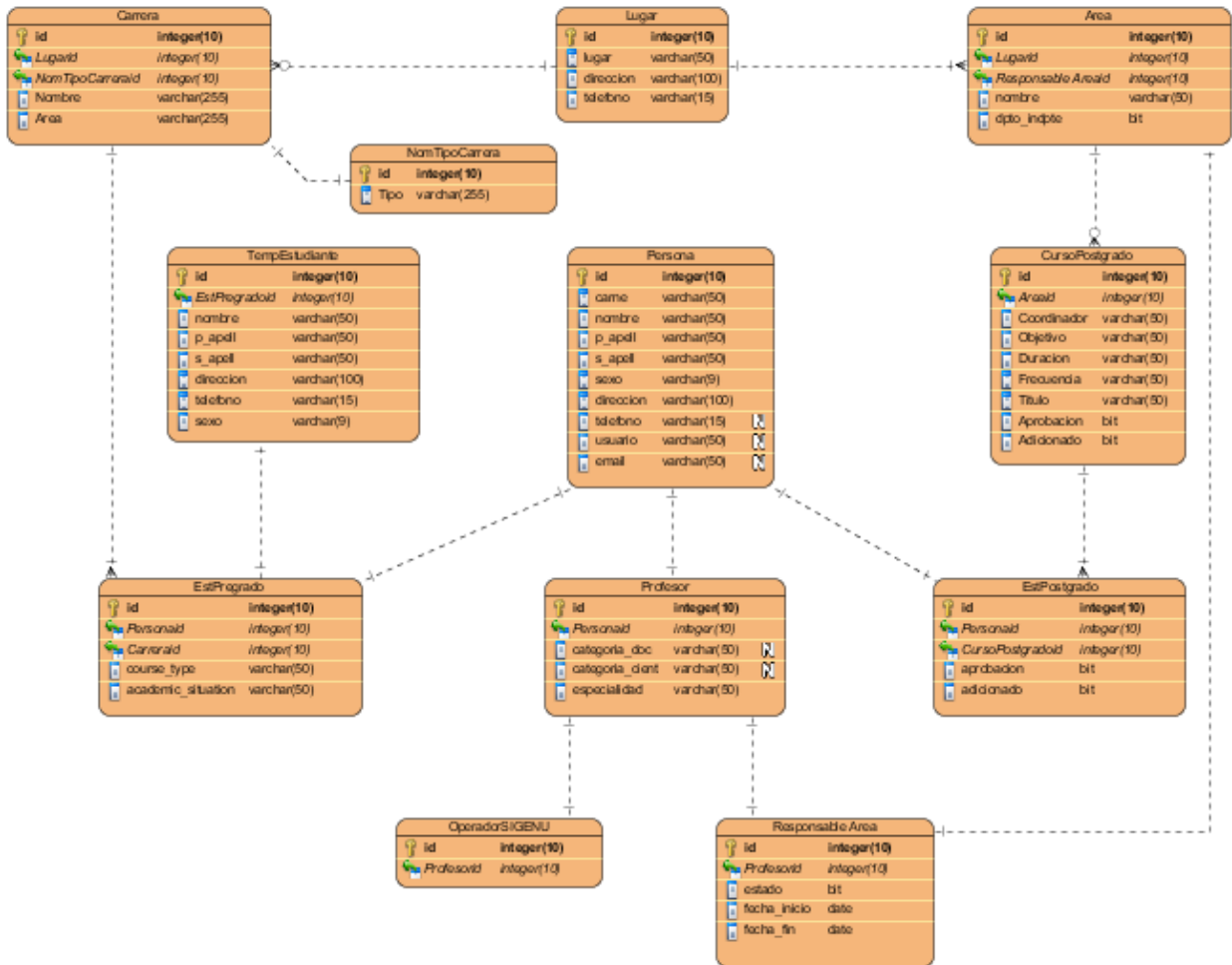


Figura 58. MODELO DE DATOS.