



Universidad de las Ciencias Informáticas
Facultad 8

Título: Sistema de Gestión de Información de la Facultad 8.

*Módulo para la gestión de los procesos
que se realizan en el vicedecanato de
Residencia y Extensión Universitaria.*

AUTORES:

Alien Tapia Salvador
Nelson Alvarez Castañeda

TUTOR:

Ing. Fidel Alberto Curbelo Rosell

COTUTOR:

Lic. Antonio Jesús Fernández Pérez

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma con carácter exclusivo.

Para que así conste firmo la presente a los _____ días del mes de _____ del año _____.

Firma del Autor
Alien Tapia Salvador

Firma del Autor
Nelson Alvarez Castañeda

Firma del Tutor
Fidel Alberto Curbelo Rosell

Firma del Autor
Antonio Jesús Fernandez Pérez

Pensamiento

No hay más que asomarse a las puertas de la tecnología y la ciencia contemporáneas para preguntarnos si es posible vivir y conocer ese mundo del futuro sin un enorme caudal de preparación y conocimientos.

Fidel Castro Ruz

Agradecimientos

Nelson:

A la **Revolución**, por darme la posibilidad de lograr mi primer objetivo en la vida, por permitirme ver y vivir el futuro de nuestras universidades, la UCI.

A toda **mi familia**, por la preocupación constante y el apoyo a lo largo de mi vida escolar.

A nuestro tutor **Fidel**, por toda la ayuda y confianza que nos dio durante el desarrollo de este trabajo.

A todas las personas que se preocuparon y ocuparon, en especial a Dayana, Lisandra, Yanelis, Marlon, mi primo.

A todo **mi grupo** por el apoyo y la ayuda de estos 5 años en especial a Marlon y Joanni.

A mi compañero de tesis **Tapia**.

Tapia:

A **toda mi familia** por el apoyo incondicional que me brindaron durante los 5 años de la carrera.

A **mi madre** por su ayuda, sacrificio y entrega que al final fueron los que hicieron posible este sueño.

A nuestro **tutor Fidel** por su entrega y compromiso con este trabajo.

A **todas mis amistades** que de una forma u otra ayudaron a la realización de este trabajo.

A mi antiguo **grupo 8107** y al actual **8504** especialmente a **TNT**.

Dedicatoria

Nelson:

A toda mi familia en especial a mi mamá, mi papá y mi hermana.

A todos aquellos que me dieron su confianza y apoyo en todo momento.

Tapia:

A toda mi familia especialmente a mi madre.

RESUMEN

En el presente trabajo se pretende realizar el análisis, diseño e implementación del Módulo para la gestión de los procesos que se realizan en el vicedecanato de Extensión y Residencia Estudiantil de la Facultad 8 de la Universidad de las Ciencias Informáticas (UCI), con el objetivo de automatizar varios de los procesos de esta índole que allí tienen lugar, como es el caso de: controlar los partes de la guardia estudiantil, cuartelaría y trabajos socialmente útil que se realizan a diario. Por tanto, primero se analiza cómo se llevan a cabo varias de las actividades que se desarrollan en esta importante área de la universidad y luego se exponen las características del sistema que se propone. Una vez culminado el trabajo se contará con la informatización de las principales actividades que se ejecutan en esta área, las cuales se realizan de forma un tanto engorrosas, por el gran cúmulo de información que allí se maneja.

Tabla de Contenidos

INTRODUCCIÓN	1
FUNDAMENTACIÓN TEÓRICA	6
1.1 Introducción	6
1.2 Sistemas automatizados similares existentes vinculados al campo de acción	6
1.3 Tendencias y tecnologías actuales	7
1.3.1 Lenguajes de Programación	7
1.3.2 Servidor Web Apache	14
1.3.3 Proceso de desarrollo de software	15
1.3.4 Metodologías de desarrollo de software	16
1.3.5 Herramientas CASE de Modelado con UML	19
1.3.6 IDEs a Utilizar para la Programación	23
1.3.7 Smarty	25
1.3.8 Arquitectura	25
1.4 Conclusiones	28
CARACTERÍSTICAS DEL SISTEMA	30
2.1 Introducción	30
2.2 Objeto de Estudio	30
2.2.1 Flujo actual de los procesos	30
2.2.2 Análisis crítico de la ejecución de los procesos	31
2.3 Propuesta de Sistema	32
2.4 Descripción de los procesos del negocio	32
2.5 Reglas del negocio a considerar	33
2.6 Actores del negocio	33
2.7 Diagrama de casos de uso del negocio	34
2.8 Trabajadores del negocio	34
2.9 Especificación de los Casos de Usos del Negocio	35
2.9.1 Descripción del CUN “Solicitar Reporte del Comportamiento de la Cuartelería”	35
2.9.2 Descripción del CUN “Solicitar Reporte del Comportamiento de TSU”	36
2.10 Diagrama de clases del modelo de objetos	38
2.11 Definición de los requisitos funcionales	39
2.12 Definición de los requisitos no funcionales	41
2.13 Actores del sistema	43
2.14 Diagrama de casos de uso del sistema	43

2.14.1 Descripción de los casos de uso del sistema	44
2.15 Conclusiones	65
ANÁLISIS Y DISEÑO DEL SISTEMA	66
3.1 Introducción	66
3.2 Diagrama de clases del análisis	66
3.3 Diagrama de clases del Diseño	69
3.4 Descripción textual de las clases Web	74
3.5 Diagrama de clases persistentes	83
3.6 Principios de Diseño	84
3.6.1 Interfaz de Usuario	84
3.6.2 Formato de salida de los Reportes	86
3.6.3 Ayuda	87
3.6.4 Tratamiento de errores	88
3.6.5 Seguridad	89
3.7 Conclusiones	90
IMPLEMENTACIÓN Y PRUEBAS	92
4.1 Introducción	92
4.2 Modelo de Implementación	92
4.3 Modelo de Despliegue	96
4.4 Prueba	97
4.5 Conclusiones	112
CONCLUSIONES	113
RECOMENDACIONES	114
REFERENCIAS BIBLIOGRÁFICAS	115
GLOSARIO DE TÉRMINOS Y SIGLAS	116
ANEXOS	118

INTRODUCCIÓN

En todo el sistema educacional cubano, se maneja día a día una gran cantidad de información, la cual es necesaria, tanto para conocer una serie de aspectos y características de las personas implicadas en el proceso docente-educativo como para la toma de decisiones. En la mayoría de los casos, se torna un tanto difícil el mantener dicha información organizada y almacenada de forma correcta, para poder ser utilizada de manera eficaz en el tiempo y el espacio que sea necesaria.

De igual modo las universidades forman parte de esta amplia gama de centros, en ellas no sólo se procesa información relacionada con las actividades docentes, sino además de aspectos tales como: **la extensión y residencia universitaria.**

Como parte también, de estas casas de altos estudios se encuentra La Universidad de las Ciencias Informáticas (UCI), en la que el cúmulo de información que se manipula en todo momento es considerable, ya que la misma cuenta con un total de 10 Facultades y cada una de ellas a su vez con una numerosa cantidad de estudiantes, cursando los distintos años de la carrera, lo cual acarrea que la información que se necesite obtener de cada uno de éstos y de las actividades que realizan a cada momento sea de consideración, por esto se requiere que la misma esté lo más organizada posible para ser procesada de manera eficiente.

De conjunto con estas 10 Facultades se encuentra, la Facultad 8, la cual no está exenta de ello y en la que actualmente, la gestión de toda la información, específicamente la relacionada con la Cuartelería que se realiza en cada uno de los pasos de escaleras pertenecientes a la Facultad y los Trabajo Socialmente Útil (TSU) que efectúan los estudiantes de la misma diariamente, se torna un tanto compleja, lo que trae como consecuencia que en determinados momentos se pueda ver afectada la toma de decisiones, por no contar con toda la información requerida o apropiadamente organizada para ello. Por otra parte está la Guardia Estudiantil, para la que existe ya un sistema, el cual no cuenta con todas las funcionalidades necesarias como para sustituir del todo la forma de trabajo que se llevaba a cabo antes de su existencia, por lo que se han identificado algunos problemas que persisten en la actualidad con relación a los procesos antes mencionado, como son:

- La información relacionada con la Cuartelería y el TSU, se maneja y organiza de forma manual, trayendo consigo que existan retrasos a la hora de realizar una búsqueda en la misma, teniendo en cuenta que en muchas ocasiones, la información que se requiere aparece en más de un documento y lugar de almacenamiento, por lo que si se quiere realizar un resumen de la información originaria de diferentes fuentes, el proceso se torna un tanto lento.
- Toda la información que se emite y maneja diariamente en la facultad, relacionada con la Cuartelería y el TSU que realizan los estudiantes, se almacena en la mayoría de los casos en papel, documentos Word y Excel, lo cual puede inducir con el paso del tiempo y la excesiva manipulación a: pérdidas de información, por deterioro de la documentación o por la ocurrencia de imprevistos tales como: problemas en el sistema operativo, virus, incorrecto manejo de los ficheros (documentos Word y Excel), entre otros.
- La información no está centralizada de manera que se pueda acceder a ésta en todo momento y desde cualquier lugar, lo que puede incurrir de manera notable en demoras a la hora de resumir información procedente de diversas fuentes, y para hacerle llegar la misma a la persona o las personas que la requieren, tienen que trasladarse de un lugar a otro, o en el mejor de los casos, mandarla por correo electrónico, lo que trae como consecuencia que no se cuente con la información en tiempo y forma.
- El sistema existente para gestionar la información de los partes de la Guardia Estudiantil, no cuenta con funcionalidades tales como: eliminar o modificar partes de dicha Guardia Estudiantil.

De lo anteriormente expresado se deriva el **Objeto de Estudio** de este trabajo que son los procesos que tienen lugar en el vicedecanato de residencia y extensión universitaria de la Facultad 8 de la UCI.

Esto conlleva a que el **Campo de Acción** está determinado por los procesos relacionados con las Cuartelerías y Trabajos Socialmente Útiles (TSU). Además de lo concerniente a la manipulación de la información relacionada con los partes de la guardia estudiantil, así como de las personas que conviven a diario en la facultad, específicamente, los estudiantes y profesores.

Por esta razón se plantea para este trabajo como **Idea a defender** que, si se realiza el análisis, diseño e implementación de un grupo de funcionalidades para gestionar la información que se maneja en la Facultad 8 de la UCI, con relación a las Cuarterías, TSU, Partes de la Guardia Estudiantil y las personas pertenecientes a la misma, se avala que los procesos de esta índole que allí se llevan a cabo, se realizarán con mayor organización y rapidez.

Los **aportes prácticos** que se esperan obtener son:

- Integridad de la información relacionada con los procesos del vicedecanato de extensión que tienen lugar en la Facultad.
- Mayor rapidez en la búsqueda y consulta de cualquier información, a partir de la automatización de varios de los procesos que se llevan a cabo en la Facultad en el área anteriormente mencionada, con el objetivo de humanizar el trabajo de las personas implicadas en el mismo.
- Centralización de toda la información de esta índole, con el propósito de lograr una mayor accesibilidad a la misma.
- Obtención de la información que se requiere de forma detallada y concreta.

El **Objetivo General** está centrado en la realización del: Análisis, Diseño e Implementación de una Aplicación Web que permita gestionar la información relacionada con la Cuartería, TSU, Partes de la Guardia Estudiantil, y las Personas pertenecientes a la Facultad 8 de la UCI.

Los **Objetivos Específicos** que se persiguen son:

1. Informatizar varios de los procesos que tienen lugar en el vicedecanato de residencia y extensión universitaria de la Facultad 8 de la UCI, específicamente los concernientes a la Cuartería, TSU, partes de la guardia estudiantil y a las personas pertenecientes a dicha facultad.
2. Proponer y desarrollar nuevas funcionalidades al sistema existente con el objetivo de dar solución a gran parte de los problemas y necesidades que aún persisten en la Facultad.
3. Conformar la documentación de las nuevas funcionalidades que se proponen para que sirvan de apoyo al desarrollo de posteriores versiones.

Para darle cumplimiento a los objetivos trazados se ha decidido desarrollar las siguientes **tareas**:

1. Entrevistar al Vicedecano de Extensión y Residencia para conocer en detalles cómo es que se realizan cada una de las actividades relacionadas con los procesos antes descritos.
2. Realizar un estudio de los sistemas existentes a fines con el campo de acción en cuestión, con el objetivo de analizar las deficiencias y las posibles mejoras a los mismos.
3. Determinar y conformar los requisitos que deben cumplir las nuevas funcionalidades que se pretenden agregar.
4. Proponer una solución que contenga las nuevas funcionalidades que se proponen con el objetivo de eliminar en gran medida, los problemas que aún persisten.
5. Conformar la documentación de las nuevas funcionalidades que se pretenden adicionar al sistema ya existente con el objetivo de poder contar con una documentación del mismo más acabada y que sirva de apoyo al desarrollo de posteriores versiones.

Resumen por Capítulos

Capítulo 1. Fundamentación Teórica: Este capítulo trata acerca de algunos elementos teóricos que sirven de soporte para la realización de todo el trabajo en general, tales como: algunos sistemas similares existentes vinculados al campo de acción, además de una descripción y selección de las herramientas a utilizar.

Capítulo 2. Características del Sistema: En dicho capítulo se realiza una descripción del objeto de estudio, además de que se describen todos los procesos, actores, casos de usos y trabajadores del negocio actual. También se muestra el diagrama de casos de usos del negocio y el modelo de objetos del negocio. Por otro lado, se definen cuáles son los requerimientos funcionales y no funcionales, a la vez que se presentan los actores del sistema y diagrama de casos de uso del mismo, acompañado de la descripción textual de cada uno de éstos.

Capítulo 3. Análisis y Diseño del Sistema: En este capítulo se presentan los diagramas de clases del análisis, lo cual influye notablemente a la hora de concebir el diseño del sistema, además de los diagramas de clases Web, que reflejan de una



Sistema de Gestión de Información de la Facultad 8.

forma más clara cómo va a funcionar dicho sistema y qué clases están presentes en el mismo, acompañado también de la descripción de cada una de éstas.

Capítulo 4. Implementación y Pruebas: En este capítulo se describe como está implementado el sistema, a través de los diagramas de componentes y el diagrama de despliegue, además de que se exponen y detallan las diferentes pruebas que se le realizan al mismo.

Al final se podrán observar las conclusiones generales, las recomendaciones, citas bibliográficas hechas durante el desarrollo del trabajo, un glosario de términos y siglas utilizadas. Por último, se podrán observar los anexos, los cuales contienen información de apoyo a algunos aspectos tratados durante el trabajo.

Fundamentación Teórica

1.1 Introducción

El capítulo siguiente trata temas tales como: la fundamentación teórica del Módulo para la Gestión de los Procesos que tienen lugar en el Vicedecanato de Extensión y Residencia Universitaria, correspondiente al Sistema de Gestión de Información de la Facultad 8, en el cual se realiza un análisis de sistemas similares, vinculados al campo de acción, además de que se presentan las tendencias y tecnologías actuales y se realiza una selección de aquellas que serán utilizadas para el desarrollo del proyecto.

1.2 Sistemas automatizados existentes vinculados al campo de acción

En la UCI no se cuenta con un sistema centralizado que facilite la gestión de la información no docente de cada una de las facultades y que brinde la posibilidad de contar con algunas de las funcionalidades anteriormente mencionadas. En algunas facultades se ha pensado y llegado a realizar sistemas que ofrecen la posibilidad de gestionar alguna que otra información relacionada con la extensión y la residencia estudiantil, como es el caso de la Facultad 3, dicho sistema cuenta con gran parte de las funcionalidades que se pretenden desarrollar con este trabajo, pero no brinda la posibilidad de gestionar información relacionada con el TSU, además de que está realizado en su totalidad sobre software propietario, es una aplicación de escritorio, que utiliza como lenguaje de programación C# y como gestor de Base de datos SQL Server, lo que implica que tenga que ser instalada en cada uno de los ordenadores en los que vaya a ser utilizada.

El sistema que proponemos con respecto al anterior se diferencia en: es una Aplicación Web, con lenguaje de programación PHP, gestor de Base de Datos Postgre, por lo que no necesita instalación para ser utilizada, y en caso de actualizaciones no hay que realizar esta tarea en cada uno de los ordenadores donde se utiliza, brinda privacidad con acceso (usuario y contraseña) para acceder a los datos.

En la Facultad 8 existe un sistema para la gestión de la información de la residencia, del que este trabajo pretende ser continuidad, el cual permite:

- Ubicar a los estudiantes en la residencia estudiantil.
- Controlar los apartamentos y edificios que le fueron asignados a la Facultad.
- Conocer en qué edificio, paso de escalera y apartamento de la residencia se encuentra ubicado un estudiante de la facultad o un grupo de estudiantes.
- Conocer cómo se ha comportado la guardia estudiantil de una brigada específica a través de los partes emitidos de la misma, o consultar los partes de la guardia estudiantil de cualquier día.

Pero dicho sistema no cuenta con funcionalidades como:

- Eliminar o modificar algún parte de la guardia que pueda haber tenido errores al ser insertado en el sistema por parte del personal que trabaja con el mismo.
- Gestionar postas de la guardia.
- Gestionar brigadas.
- Gestionar personas.
- Gestionar cuartería.
- Gestionar TSU.
- Gestionar los usuarios del sistema y sus privilegios.
- Restringir los accesos a las diferentes tareas según el tipo de usuario registrado.

1.3 Tendencias y tecnologías actuales

A continuación se describen algunas de las tendencias y tecnologías actuales posibles a utilizar para darle solución a los problemas planteados con anterioridad de manera eficiente, teniendo en cuenta las necesidades existentes y el entorno donde se aplicará el sistema que se va a construir.

1.3.1 Lenguajes de Programación

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

Un lenguaje de programación permite a un programador especificar de *manera precisa*: sobre qué datos una computadora debe operar, cómo deben ser estos almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar *relativamente* próximo al lenguaje humano o natural, tal como sucede con el lenguaje léxico. [1]

PHP: Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.[2]

El fácil uso y la similitud con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores experimentados crear aplicaciones complejas con una curva de aprendizaje muy suave. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas. [2]

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.[2]

Permite la conexión a diferentes tipos de servidores de base de datos tales como: MySQL, Postgre, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite; lo cual facilita la creación de aplicaciones Web muy robustas. [2]

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux), Windows y Mac, y puede interactuar con los servidores Web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI. [2]

Los principales usos del PHP son los siguientes:

- Programación de páginas Web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
 - Programación en consola, al estilo de Perl o Shell scripting.
 - Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (GIMP Tool Kit), lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado.
- [2]

Ventajas de PHP

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

Otros lenguajes como Perl (Practical Extraction and Report Language), ASP (Active Server Pages) y JSP (Java Server Pages) tienen características similares al PHP aunque poseen rasgos que los marcan y por ello los distingue, entre ellos se pueden encontrar:

- **Características multiplataformas:** Menos el ASP, que es solamente soportado por la plataforma Windows, los demás lenguajes están soportados en múltiples plataformas.

- **Velocidad de ejecución:** Es mayor en PHP, seguidos por PERL y JSP.
- **Disponibilidad de recursos:** en la actualidad los más utilizados en la Internet son el PHP y el JSP, siendo más utilizado en la publicación de artículos y códigos de ejemplos. PHP tiene una de las comunidades más grandes en Internet, al igual que la de Java.
- **Familiaridad con el lenguaje:** En las universidades los lenguajes más utilizados por los programadores es el ASP y el PHP. [4]

Java: Supone un significativo avance en el mundo de los entornos software, y esto viene avalado por tres elementos claves que diferencian a este lenguaje desde un punto de vista tecnológico:

- Es un lenguaje de programación que ofrece la potencia del diseño orientado a objetos con una sintaxis fácilmente accesible y un entorno robusto y agradable.
- Proporciona un conjunto de clases potentes y flexibles.
- Pone al alcance de cualquiera la utilización de aplicaciones que se pueden incluir directamente en páginas Web (aplicaciones denominadas *applets*).

Java aporta a la Web una interactividad que se había buscado durante mucho tiempo entre usuario y aplicación. [3]

Las principales características de Java son descritas a continuación:

Potente

Orientado a Objetos: En este aspecto Java fue diseñado partiendo de cero, no siendo derivado de otro lenguaje anterior y no tiene compatibilidad con ninguno de ellos.

En Java el concepto de objeto resulta sencillo y fácil de ampliar. Además se conservan elementos "no objetos", como números, caracteres y otros tipos de datos simples.

Riqueza Semántica: Pese a su simpleza se ha conseguido un considerable potencial, y aunque cada tarea se puede realizar de un número reducido de formas, se ha conseguido un gran potencial de expresión e innovación desde el punto de vista del programador.

Robusto: Java verifica su código al mismo tiempo que lo escribe, y una vez más antes de ejecutarse, de manera que se consigue un alto margen de codificación sin errores. Se realiza un descubrimiento de la mayor parte de los errores durante el tiempo de compilación, ya que Java es estricto en cuanto a tipos y declaraciones, y así lo que es

rigidez y falta de flexibilidad se convierte en eficacia. Respecto a la gestión de memoria, Java libera al programador del compromiso de tener que controlar especialmente la asignación que de ésta hace a sus necesidades específicas. Este lenguaje posee una gestión avanzada de memoria llamada gestión de basura, y un manejo de excepciones orientado a objetos integrados. Estos elementos realizarán muchas tareas, antes tediosas a la vez que obligadas para el programador.

Modelo de objeto rico: Existen varias clases que contienen las abstracciones básicas para facilitar a los programas una gran capacidad de representación. Para ello se contará con un conjunto de clases comunes que pueden crecer para admitir todas las necesidades del programador.

Además la biblioteca de clases de Java proporciona un conjunto único de protocolos de Internet.

El conjunto de clases más complicado de Java son sus paquetes gráficos AWT (*Abstract Window Toolkit*) y *Swing*. Estos paquetes implementan componentes de una interfaz de usuario gráfica básica común a todos los ordenadores personales modernos. [3]

Simple

Fácil Aprendizaje: El único requerimiento para aprender Java es tener una comprensión de los conceptos básicos de la programación orientada a objetos. Así se ha creado un lenguaje simple (aunque eficaz y expresivo) pudiendo mostrarse cualquier planteamiento por parte del programador sin que las interioridades del sistema subyacente sean desveladas.

Java es más complejo que un lenguaje simple, pero más sencillo que cualquier otro entorno de programación. El único obstáculo que se puede presentar es conseguir comprender la programación orientada a objetos, aspecto que, al ser independiente del lenguaje, se presenta como insalvable.

Completado con utilidades: El paquete de utilidades de Java viene con un conjunto completo de estructuras de datos complejas y sus métodos asociados, que serán de inestimable ayuda para implementar *applets* y otras aplicaciones más complejas. Se dispone también de estructuras de datos habituales, como *pilas* y *tablas hash*, como clases ya implementadas. [3]

Interactivo y orientado a red

Interactivo y animado: Uno de los requisitos de Java desde sus inicios fue la posibilidad de crear programas en red interactivos, por lo que es capaz de hacer varias cosas a la vez sin perder rastro de lo que debería suceder y cuándo. Se da soporte a la utilización de múltiples hilos de programación (*multithread*).

Las aplicaciones de Java permiten situar figuras animadas en las páginas Web, y éstas pueden concebirse con logotipos animados o con texto que se desplace por la pantalla. También pueden tratarse gráficos generados por algún proceso. Estas animaciones pueden ser interactivas, permitiendo al usuario un control sobre su apariencia.

Arquitectura neutral: Java está diseñado para que un programa escrito en este lenguaje sea ejecutado correctamente independientemente de la plataforma en la que se esté actuando (Macintosh, PC, UNIX...). Para conseguir esto utiliza una compilación en una representación intermedia que recibe el nombre de *códigos de byte*, que pueden interpretarse en cualquier sistema operativo con un intérprete de Java. La desventaja de un sistema de este tipo es el rendimiento; sin embargo, el hecho de que Java fuese diseñado para funcionar razonablemente bien en microprocesadores de escasa potencia, unido a la sencillez de traducción a código máquina hacen que Java supere esa desventaja sin problemas.

Applets: Una *applet* (mini aplicación) es un pequeño programa en Java transferido dinámicamente a través de Internet. Presentan un comportamiento inteligente, pudiendo reaccionar a la entrada de un usuario y cambiar de forma dinámica. Sin embargo, la verdadera novedad es el gran potencial que Java proporciona en este aspecto, haciendo posible que los programadores ejerzan un control sobre los programas ejecutables de Java que no es posible encontrar en otros lenguajes. [3]

Otros

Seguridad: Existe una preocupación lógica en Internet por el tema de la seguridad: virus, caballos de Troya, y programas similares que navegan de forma usual por la red, constituyendo una amenaza palpable. Java ha sido diseñado poniendo un énfasis especial en el tema de la seguridad, y se ha conseguido lograr cierta inmunidad en el aspecto de que un programa realizado en Java no puede realizar llamadas a funciones globales ni acceder a recursos arbitrarios del sistema, por lo que el control sobre los programas ejecutables no es equiparable a otros lenguajes. [3]

Los niveles de seguridad que presenta son:

- Fuertes restricciones al acceso a memoria, como son la eliminación de punteros aritméticos y de operadores ilegales de transmisión.
- Rutina de verificación de los *códigos de byte* que asegura que no se viole ninguna construcción del lenguaje.
- Verificación del nombre de clase y de restricciones de acceso durante la carga.
- Sistema de seguridad de la interfaz que refuerza las medidas de seguridad en muchos niveles.

Lenguaje basado en C++: Java fue desarrollado basándose en C++, pero eliminando rasgos del mismo poco empleados, optándose por una codificación comprensible.

Básicamente, encontramos las siguientes diferencias con C++:

- Java no soporta los tipos *struct*, *union* ni punteros.
- No soporta *typedef* ni *#define*.
- Se distingue por su forma de manejar ciertos operadores y no permite una sobrecarga de operadores.
- No soporta herencia múltiple.
- Java maneja argumentos en la línea de comandos de forma diversa a como lo hacen C o C++.
- Tiene una clase *String* que es parte del paquete *java.lang* y se diferencia de la matriz de caracteres terminada con un nulo que usan C y C++.
- Java cuenta con un sistema automático para asignar y liberar memoria, con lo que no es necesario utilizar las funciones previstas con este fin en C y C++.

Gestión de la entrada/salida: En lugar de utilizar primitivas como las de C para trabajar con ficheros, se utilizan primitivas similares a las de C++, mucho más elegantes, que permiten tratar los ficheros, sockets, teclado y monitor como flujos de datos. [3]

De este modo se pueden utilizar dichas primitivas para cualquier operación de Entrada/Salida.

Diferentes tipos de aplicaciones: En Java podemos crear los siguientes tipos de aplicaciones:

- *Aplicaciones:* Se ejecutan sin necesidad de un navegador.
- *Applets:* Se pueden descargar de Internet y se observan en un navegador.

- *JavaBeans*: Componentes software Java, que se puedan incorporar gráficamente a otros componentes.
- *JavaScript*: Conjunto del lenguaje Java que puede codificarse directamente sobre cualquier documento HTML
- *Servlets*: Módulos que permiten sustituir o utilizar el lenguaje Java en lugar de programas CGI (Common Gateway Interface) a la hora de dotar de interactividad a las páginas Web.

1.3.2 Servidor Web Apache

Es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1¹ y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf eligió ese nombre porque quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor "parcheado"). [4]

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft²). La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache. [4]

Ventajas de Apache:

- Modular.
- Open source.
- Multi-plataforma.
- Extensible.
- Popular (fácil conseguir ayuda/suporte). [4]

1.3.3 Proceso de desarrollo de software

Con el de cursar de los tiempos la producción de software ha ido en ascenso en cuanto a cantidad, calidad y complejidad, debido a que cada vez son mayores las exigencias de los usuarios, por lo que se hace mayor la necesidad de un proceso que integre todas las fases del desarrollo del software y que sea perfectamente entendible por todos aquellos que participan en su elaboración.

Lenguaje Unificado de Modelado (UML)

UML (Unified Modeling Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software. [5]

Sus creadores pretendieron con este lenguaje, unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas en un acercamiento estándar.

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y proporciona un estándar que permite al analista de sistemas generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo. Un modelo UML indica que es lo que supuestamente hará el sistema pero no como lo hará. [5]

El UML permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas que estén involucradas en el proceso de desarrollo de los sistemas, esto se lleva a cabo mediante un conjunto de símbolos y diagramas. [6]

Existen varias herramientas CASE (Computer-Aided Systems Engineering), que dan asistencia a analistas, ingenieros de software y desarrolladores durante el ciclo de vida de desarrollo de un software, pero es Rational Rose líder en el modelado del desarrollo de los proyectos. La herramienta fue desarrollada por los creadores de UML, utilizando la notación estándar en la arquitectura de software. Esta herramienta integra todos los elementos que propone la metodología RUP para cubrir el ciclo de vida de un proyecto. [6]

1.3.4 Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y pasos a seguir para construir un software. Por otra parte, una metodología de desarrollo de software define quién debe hacer qué, cuándo y cómo para alcanzar un determinado objetivo. Se puede decir que una metodología es un proceso, y en su modelación se definen como elementos principales los siguientes:

- **Trabajadores (quién):** Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- **Actividades (cómo):** Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- **Artefactos (qué):** Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- **Flujo de actividades (cuándo):** Secuencia de actividades realizadas por trabajadores y que producen un resultado de valor observable.

Rational Unified Process (RUP)

La metodología RUP, llamada así por sus siglas en inglés (Rational Unified Process), divide en 4 fases el desarrollo del software (*Ver Anexo 1*):

- Inicio (El objetivo de esta etapa es determinar la visión del proyecto)
- Elaboración (En esta etapa el objetivo es determinar la arquitectura)
- Construcción (En esta etapa el objetivo es llegar a alcanzar una funcionalidad operativa)
- Transición (El objetivo de esta etapa es llegar a obtener el release del producto)

En cada una de estas fases tienen lugar iteraciones de varios flujos de trabajos (*Ver Anexo 1*), que son:

- **Modelo del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

- **Análisis y Diseño:** Describe cómo el sistema será construido a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo qué se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación o despliegue:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Características de RUP

- Creado por Jacobson, Rumbaugh y Booch.
- Unifica los mejores elementos de metodologías anteriores.
- Preparado para desarrollar grandes y complejos proyectos.
- Orientado a Objetos.
- Utiliza el UML como lenguaje de representación visual.

El ciclo de vida de RUP se caracteriza por:

1. **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

2. **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML (Vista de Casos de Uso, Vista Lógica, Vista de Procesos, Vista de Implementación, Vista de Despliegue).
3. **Iterativo e Incremental:** Aunque el *Anexo 1* puede sugerir que los flujos de trabajo se desarrollan en cascada, la “lectura” de este gráfico tiene que ser vertical y horizontal. RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

Extreme Programming (XP): Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. [7]

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software. [7]

Esta metodología de desarrollo de software se basa en:

- **Pruebas Unitarias:** Consiste en las pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, puede hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores.
- **Refabricación:** Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

- **Programación en pares:** Se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera el código es revisado y discutido mientras se escribe es más importante que la posible pérdida de productividad inmediata. [7]

¿Qué es lo que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustancial del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo. [7]

Derechos del Cliente

- Decidir que se implementa.
- Saber el estado real y el progreso del proyecto.
- Añadir, cambiar o quitar requerimientos en cualquier momento.
- Obtener lo máximo de cada semana de trabajo.
- Obtener un sistema funcionando cada 3 o 4 meses. [7]

Derechos del Desarrollador

- Decidir cómo se implementan los procesos.
- Crear el sistema con la mejor calidad posible.
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- Estimar el esfuerzo para implementar el sistema.
- Cambiar los requerimientos en base a nuevos descubrimientos. [7]

Lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales. [7]

1.3.5 Herramientas CASE de Modelado con UML

A medida que los sistemas que hoy se construyen se tornan más y más complejos, las herramientas de modelado con UML ofrecen muchos beneficios para todos los

involucrados en un proyecto, por ejemplo, administrador del proyecto, analistas, arquitectos, desarrolladores y otros. Las herramientas CASE (**C**omputer **A**ided **S**oftware **E**ngineering, Ingeniería de Software Asistida por Ordenador) de modelado con UML nos permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraernos del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y más fáciles de entender y modificar. Cuanto más grande es un proyecto, es más importante utilizar una herramienta CASE. Por otro lado, al usar las herramientas CASE:

- Los Analistas de Negocio/ Sistemas pueden capturar los requisitos del negocio/sistema con un modelo de casos de uso.
- Los Diseñadores/Arquitectos pueden producir el modelo de diseño para articular la interacción entre los objetos o los subsistemas de la misma o de diferentes capas (los diagramas UML típicos que se crean son los de clases y los de interacción).
- Los Desarrolladores pueden transformar rápidamente los modelos en una aplicación funcionando, y buscar un subconjunto de clases y métodos y asimilar el entendimiento de cómo lograr interfaces con ellos. [8]

El modelo actúa como el plano y guiará finalmente la construcción del sistema. De manera semejante, la administración es capaz de ver, puntualmente y desde un alto nivel, una representación del diseño y comprender lo que está sucediendo.

Por estas razones, las herramientas CASE de UML acompañadas con metodologías, nos brindan una forma de representar sistemas demasiados complejos para comprenderlos a través de su código fuente subyacente y nos permiten desarrollar la solución de software correcta, más rápido y más económicamente.

Sin embargo, las herramientas CASE varían con respecto a las capacidades de modelado con UML, el soporte del ciclo de vida del proyecto, las ingenierías directa y reversa, el modelado de datos, la performance, el precio, el soporte, la facilidad de uso, etc. [8]

Rational Rose

Rational Rose es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: Concepción y

formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

El navegador UML de Rational Rose nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable.

Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción.

[9]

Esta herramienta propone la utilización de cuatro tipos de modelos para realizar el 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 utiliza un proceso de desarrollo iterativo controlado (Controlled Iterative Process Development), donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que estos se hagan mínimos.

Cuando la implementación pasa todas las pruebas que se determinan en el proceso, esta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración.

Rational Rose permite que hayan varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades. [9]

Visual Paradigm

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. [10]

Características:

- Soporte de UML versión 2.1.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Modelado colaborativo con CVS y Subversion (nueva característica).
- Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI (nueva característica).
- Ingeniería de ida y vuelta.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Ingeniería inversa Java, C++, Esquemas XML, XML,.NET exe/dll, CORBA IDL.
- Generación de código - Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Diagramas EJB - Visualización de sistemas EJB.
- Generación de código y despliegue de EJB's - Generación de beans para el desarrollo y despliegue de aplicaciones.
- Diagramas de flujo de datos.
- Soporte ORM - Generación de objetos Java desde la base de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.

- Importación y exportación de ficheros XML.
- Integración con Visio - Dibujo de diagramas UML con plantillas (stencils) de MS Visio.
- Editor de figuras.
- SDE para Eclipse, NetBeans, Sun ONE, Oracle JDeveloper, JBuilder, IntelliJ IDEA, WebLogic Workshop. [10]

1.3.6 IDEs a Utilizar para la Programación

Los entornos de desarrollo integrados, facilitan la programación en cualquier lenguaje, en específico los que a continuación se detallan son los valorados para utilizar en este trabajo:

NuSphere PHPEd

PHPEd es un editor para programadores con soporte para múltiples formatos, similar a otras aplicaciones como PHP Coder.

PHPEd facilita el trabajo de programación con numerosas características de gran utilidad entre las que destacan: [11]

- Completo sistema de ayuda.
- Plantillas de documentos y de fragmentos de código frecuentes.
- Código de colores para comandos en PHP, Perl, Javascript, SQL, HTML y más.

Además, esta herramienta incluye un cliente de FTP y un servidor Web integrados, totalmente configurables según tus necesidades de trabajo.

Zend Studio

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientado a desarrollar aplicaciones Web, en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. [12]

El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración. [12]

Lo más destacable de este editor es que contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. Aunque esta ayuda contextual no solo se queda en las funciones definidas en el lenguaje, sino que también reporta ayudas con las funciones que vayamos creando nosotros, incluso en páginas que tengamos incluidas con la función include(). [12]

También dispone de herramientas para gestionar los proyectos, muy útiles para mejorar la productividad en la programación. Los proyectos permiten guardar mucha más información al programa sobre los archivos, discos, servidores, etc. que se gestionen en estas aplicaciones PHP. [12]

Sin duda, más de una vez los programadores de PHP se han visto en un duro problema por no encontrar un error en algún script que está dando resultados inesperados.

Zend Studio dispone de una herramienta muy interesante de debug o depuración. Gracias a ella se pueden ejecutar páginas y conocer en todo momento el contenido de las variables de la aplicación y las variables del entorno, como las cookies; las recibidas por formulario o en la sesión. Se pueden colocar puntos de parada de los scripts y realizar las acciones típicas de depuración.

Además de la ventana para visualizar el contenido de las variables, dispone de otras donde muestra la salida del script según se va generando, y otra donde se pueden ver las alertas y errores. Las posibilidades se completan con distintos tipos de depuración, en local, en remoto o a partir de una URL. [12]

1.3.7 Smarty

Smarty es un motor de plantillas para PHP, cuyo objetivo es separar el contenido de la presentación en una página Web, se encuentra bajo la licencia GPL por lo que puede ser usado libremente. [13]

Es común que en grandes proyectos el rol de diseñador gráfico y el de programador sean cubiertos por personas distintas, sin embargo la programación en PHP tiene la tendencia a combinar estas dos labores en una persona y dentro del mismo código lo que trae consigo grandes dificultades a la hora de cambiar alguna parte del diseño de la página, pues se tiene que escarbar entre los scripts para modificar la presentación del contenido, Smarty tiene como objetivo solucionar este problema. [13]

Existen más sistemas de plantillas para PHP pero éste parece ser la más avanzada y con más frecuencia de desarrollo. También hay detractores de estas técnicas que alegan que las mismas hacen en cierta medida un grado más complejo el desarrollo Web, por la necesidad de aprender un pseudo lenguaje más. [13]

1.3.8 Arquitectura

“Una Arquitectura Software, también denominada *Arquitectura lógica*, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. [4]

Se selecciona y diseña con base en unos objetivos y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura software de tres capas para implementar sistemas en tiempo real. [4]

La arquitectura software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación ente ellos. Toda arquitectura software debe ser implementable en una *arquitectura física*, que consiste

simplemente en determinar qué computadora tendrá asignada cada tarea de computación. [4]

La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad. — Kruchten, Philippe” [4]

Generalmente, no es necesario inventar una nueva arquitectura software para cada sistema de información. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto. Así, las arquitecturas más universales son: [4]

Monolítica. Donde el software se estructura en grupos funcionales muy acoplados.

Cliente-servidor. Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.

Arquitectura de tres niveles. Generalización de la arquitectura cliente-servidor donde la carga se divide en tres partes (*ver Anexo 2*) con un reparto claro de funciones: una capa para la presentación, otra para el cálculo y otra para el almacenamiento. Una capa solamente tiene relación con la siguiente.

Modelo Vista Controlador (MVC): es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página.

- **Modelo:** Esta es la representación específica del dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio. La lógica de dominio añade significado a los datos; por ejemplo, calculando si hoy es el cumpleaños del usuario o los totales impuestos en un carrito de compra.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.

- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.
- Muchas aplicaciones utilizan un mecanismo de almacenamiento persistente (como puede ser una base de datos) para almacenar los datos. MVC no menciona específicamente esta capa de acceso a datos.

Es común pensar que una aplicación tiene tres capas principales: presentación (IU), dominio, y acceso a datos. En MVC, la capa de presentación está partida en controlador y vista. La principal separación es entre presentación y dominio; la separación entre vista y controlador es menos clara.

Arquitectura n-Capas. La que más se utiliza es la de cuatro capas, la capa que se agrega es la que surge de separar definitivamente las reglas de negocio de la de Acceso a Datos. Esta arquitectura trae consigo la ventaja de aislar definitivamente la lógica de negocios de todo lo que tenga que ver con el origen de datos, ya que desde el manejo de la conexión, hasta la ejecución de una consulta, la manejará la capa de Acceso a Datos. De este modo, ante cualquier eventual cambio, solo se deberá tocar un módulo específico, así como al momento de plantear la escalabilidad de este sistema, si se han respetado las reglas básicas de diseño, entonces no se deberán afrontar grandes modificaciones. [4]

Abstracción total acerca del origen de datos. Las distintas capas se especializan absolutamente en la funcionalidad que deben brindar (procesamiento en las reglas de negocios o presentación de datos en la capa cliente) sin importar cual es el origen de los datos procesados.

Bajo costo de desarrollo y mantenimiento de las aplicaciones. Si bien al momento del diseño podemos observar una mayor carga de complejidad, la utilización de esta arquitectura brinda un control más cercano de cada componente, así como también la posibilidad de una verdadera reutilización del código.

Estandarización de las reglas de negocio. Las reglas de negocio se encuentran encapsuladas en un set de rutinas comunes y pueden ser llamadas desde diversas aplicaciones sin necesidad de saber cómo esta funciona o ha sido diseñada.

Mejor calidad en las aplicaciones. Como las aplicaciones son construidas en unidades separadas, estas pueden ser testeadas independientemente y con mucho más detalle, esto conduce a obtener un producto mucho más sólido.

Reutilización de código. La concepción natural de un sistema desarrollado con esta arquitectura, promueve la reutilización de sus componentes en varias partes del propio desarrollo y de futuros sistemas.

1.4 Conclusiones

Teniendo en cuenta las necesidades que posee la facultad 8 de la UCI, a partir de los problemas existentes en la ejecución de los procesos que tienen lugar en el vicedecanato de extensión y residencia universitaria, y habiendo hecho un previo análisis de las principales características de las tecnologías y herramientas más usadas en el mundo en la actualidad, teniendo en cuenta además que el módulo a desarrollar será la continuidad y perfeccionamiento del sistema SGIF. Módulo de Residencia (versión 1.0), se decidió seleccionar, para la construcción y desarrollo del sistema propuesto, las siguientes herramientas y tecnologías:

Como lenguaje de programación para implementar el sistema, PHP, debido a que, a pesar de poseer muchas ventajas al igual que Java, este lenguaje posee algunas características que lo distinguen, como es el caso de la simplicidad en su código. Además de que, posee una de las comunidades más grandes de desarrolladores en el mundo y en la propia UCI, lo cual es muy beneficioso a la hora de consultar alguna duda con alguien, además de que se tiene un poco más de experiencia en el uso de este lenguaje.

Por otro parte, el hecho de usar PHP, justifica el uso del servidor Web Apache, los cuales poseen una gran compatibilidad, además de que este último, brinda algunas ventajas, como son: una gran modularidad, es gratuito y multiplataforma al igual que PHP, además de que es muy configurable.

Como editor de PHP se decidió utilizar el Zend Studio, debido a que, tiene soporte para navegación en bases de datos, para control de versiones usando CVS o Subversion, phpDoc integrado, posee características semejantes al NuSphere PHPEd, pero se tiene más experiencia en el trabajo con el primero. Se decidió también, el uso de Smarty, debido a que es un motor de plantillas para PHP, que tiene como finalidad, separar la capa de presentación de la capa lógica, lo cual trae grandes ventajas, a la hora de cambiar en cualquier momento el diseño de las páginas Web sin que afecte la lógica del negocio.

En lo que respecta a la metodología de software a utilizar, se decidió escoger RUP, debido a que la misma abarca todo el ciclo de vida del software de manera



Sistema de Gestión de Información de la Facultad 8.

organizada, dividiéndolo en fases, en las cuales, se realizan varias iteraciones que traen consigo que se vaya desarrollando el producto de forma incremental, además de que utiliza UML para la representación visual, el cual constituye un estándar a nivel internacional, que trae consigo un entendimiento común del sistema por parte de los diferentes desarrolladores.

Como herramienta CASE a Rational Rose, debido a que ésta integra todos los elementos que propone dicha metodología para cubrir el ciclo de vida de un proyecto, además de que es la herramienta líder en el mundo en este sentido. En cuanto a la arquitectura, se decidió el uso de la de 3 capas, debido a que permite trabajar de una forma más organizada, lo cual trae grandes beneficios a la hora de hacer cambios y corregir errores.

Características del Sistema

2.1 Introducción

En el actual capítulo se enuncia y describe el objeto de estudio, se dan a conocer cuáles son los procesos que serán automatizados, se exponen las reglas que debe cumplir el negocio en cuestión, además de describir los actores y trabajadores que intervienen en el mismo, se muestra el modelo de objetos, se identifican cuáles son los requisitos funcionales y no funcionales del sistema, que formaran parte de la solución a los problemas existentes; a la vez que se muestran cuáles son las funcionalidades que brinda este sistema y quienes son los actores que interactúan con el mismo.

2.2 Objeto de Estudio

En la Universidad de las Ciencias Informáticas (UCI) la Facultad 8 es una de las diez facultades que compone dicha institución, en la que el objeto de estudio son los procesos que tienen lugar en el Vicedecanato de Extensión y Residencia estudiantil de la misma.

2.2.1 Flujo actual de los procesos

El análisis del flujo actual de los procesos, conlleva a que se pueda conocer como funciona en realidad el negocio y obtener posibles resultados. Estos últimos pueden ser: un servicio, una información determinada, un producto o combinaciones de ellos. El análisis de cómo se llevan a cabo estos procesos permite identificar un conjunto de problemas que tienen lugar en la entidad donde se desarrollan y que dificultan el buen desarrollo de las actividades que allí tienen lugar, como por ejemplo: pérdida de información, duplicación de información, demoras en la búsqueda de información, o incorrecto formato en que se presenta la misma, entre otras.

Los procesos no docentes que se realizan en la facultad, relacionados a la extensión y residencia estudiantil se describen a continuación:

Vicedecano de Extensión y Residencia (VDER): Es un directivo de la facultad, el cual tiene la responsabilidad entre otras cosas de llevar un control sobre la brigada que realizó la guardia estudiantil en la noche anterior, además de tener registrado cómo se

han efectuado las mismas en días pasados, en cuanto al cumplimiento o no de dicha actividad por parte de los estudiantes. De igual forma mantiene el control de los partes que a él llegan diariamente del TSU y del comportamiento en la residencia de las cuarterías de cada uno de los pasos de escaleras donde residen los estudiantes de la facultad.

Secretaria del Vicedecano: Es la persona encargada de suministrar al VDER todas las informaciones que éste requiera como por ejemplo: partes de la guardia estudiantil, TSU o cuarterías de un día determinado o de una brigada determinada. Además de encargarse de digitalizar una parte importante de las informaciones que el VDER le solicita a manera de reporte, entre las que se encuentran las mencionadas con anterioridad. Además de ayudar en alguna forma al VDER, a mantener un control sobre algunas de las tareas programadas por el mismo.

2.2.2 Análisis crítico de la ejecución de los procesos

En la actualidad, los procesos que se realizan en la facultad, específicamente en el vicedecanato de extensión y residencia estudiantil, no se realizan de forma eficiente. En ello influyen algunos factores, como son:

- Demoras en la búsqueda de información.
- Insuficiencia a la hora de recopilar información proveniente de fuentes diferentes.
- No centralización de la información.
- Pérdidas de información.
- No contar con la información que se necesita en tiempo y forma.

Estos factores provocan situaciones como las siguientes: en caso de que el Vicedecano de Extensión y Residencia necesite en un momento dado un reporte de las evaluaciones de un estudiantes o brigada en actividades tales como guardia estudiantil, TSU, cuartería o la combinación de las mismas, tiene que acudir a varios documentos donde se encuentra dicha información. Por otro lado, el Vicedecano de Extensión y Residencia necesita conocer cómo se realizó la guardia estudiantil, TSU y cuarterías no sólo del día anterior, sino de todos los días, por lo que los reportes que recibe a diario de la realización de dichas actividades son almacenados, y con el paso del tiempo, provoca que exista un cierto cúmulo de información, trayendo consigo que a la hora de realizar alguna búsqueda de cómo se llevaron a cabo alguna de estas

actividades en un determinado día o de una brigada en cuestión el proceso se dificulte en gran medida.

2.3 Propuesta de Sistema

El sistema que se propone, pretende agilizar en gran medida la realización de varios procesos que tienen lugar en el vicedecanato de extensión y residencia estudiantil de la Facultad 8 de la UCI, es decir, que se pueda tener acceso a la información que se necesite de manera casi inmediata, lo cual le reportaría grandes beneficios a aquellas personas que necesiten de la misma. Como es el caso de conformar partes de la guardia estudiantil, TSU y el resumen diario del comportamiento de la cuartería, además de almacenarlos para posteriormente hacer búsquedas de algún parte específico, ya sea en una fecha determinada o de una brigada específica, así como imprimir y salvar en formato Excel y PDF algún parte en específico o resumen de información. El sistema además pretende gestionar todas las operaciones que se deseen realizar con las personas, ya sean estudiantes o profesores, así como permitir incorporar nuevos usuarios y sus privilegios, poniendo toda la accesibilidad de los mismo de forma dinámica en función de esta última característica.

2.4 Descripción de los procesos del negocio

El **Vicedecano de Extensión y Residencia (VDER)** recibe diariamente un modelo con el parte de la guardia estudiantil realizada en la noche anterior, en el cual se refleja, entre otras cosas, quienes cumplieron o no con la misma, a qué brigada pertenecen, en cuáles postas realizaron la guardia y en qué turno, además de los responsables que estaban ese día de guardia. De la misma forma recibe diariamente los partes de la cuartería y el TSU realizados el día anterior, especificándole del primero: nombres de los estudiantes, grupos, pasos de escalera a los que pertenecen, evaluaciones de cada uno y un comentario justificando cada uno de los evaluados de M si existen, del segundo: se especifica el área donde se realizó, nombre del responsable del área, nombre de los estudiantes que asisten, brigada a la que pertenecen, evaluación individual y evaluación colectiva.

La **secretaria del VDER** recibe diariamente de manos de éste, los partes de la guardia estudiantil, cuarterías, TSU y se encarga de almacenar estos datos en la computadora para ser posteriormente utilizados por el VDER, cuando éste lo solicite.

2.5 Reglas del negocio a considerar

Reglas de Restricción:

- En un paso de escalera sólo puede realizar la actividad de cuartelería un estudiante diariamente.
- Un estudiante puede cumplir con el TSU y ser evaluado de mal.
- Los estudiantes sólo pueden realizar la cuartelería en su paso de escalera.
- El estudiante que no asista al TSU se da como ausente y no se emite evaluación.
- Una misma brigada puede realizar el TSU en más de un área.
- Un estudiante sólo puede realizar guardia en un turno y en una posta.
- En un día, solo puede realizar guardia una brigada.

Reglas de Derivación:

- Cuando un estudiante no se presenta a realizar la guardia, cuartelería o TSU se dice que éste incumplió con las mismas.
- Cuando un estudiante abandona su puesto de guardia o tarea del TSU antes de culminar la misma se convierte en ausente.
- Cuando un estudiante no está en ninguno de los casos anteriores, se dice que cumplió con la guardia, cuartelería o TSU según sea el caso.

2.6 Actores del negocio

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término *actor* significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. De acuerdo con esta idea un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, o sea, ser instancias de un mismo actor. [14]

A continuación se muestra la tabla 2.1, con los actores del negocio y la justificación de los mismos:

Actores del negocio	Justificación
Vicedecano de Extensión y Residencia (VDER)	Es uno de los directivos de la Facultad que solicita informaciones relativas a la guardia estudiantil, cuartelarias

	y TSU.
--	--------

Tabla 2.1 Descripción de los actores del negocio.

2.7 Diagrama de casos de uso del negocio

El diagrama de casos de uso del negocio representa de forma gráfica los procesos del negocio y los actores que toman parte en el mismo, así como las relaciones que se instauran entre estos.

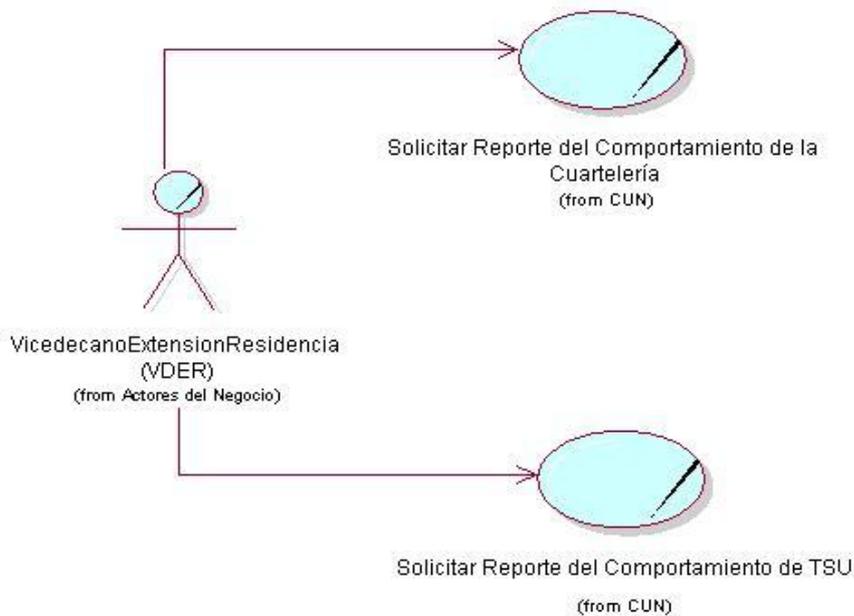


Figura 2.1 Diagrama de casos de uso del negocio

2.8 Trabajadores del negocio

A continuación se muestra la tabla 2.2, con los trabajadores del negocio y la justificación de los mismos:

Actores del negocio	Justificación
Secretaria del Vicedecano de Extensión y Residencia.	Es la encargada de conformar cada uno de los reportes en cualquiera de sus combinaciones posibles solicitadas por los actores del negocio.

Tabla 2.2 Descripción de los actores del negocio.

2.9 Especificación de los Casos de Usos del Negocio

La especificación de los CUN, facilitan la comprensión de los procesos del negocio tanto para el cliente como para los desarrolladores. Se describe de forma textual cómo se realizan las actividades del negocio y quién las realiza. Se muestra el flujo de los procesos de manera gráfica mediante los **diagramas de actividades**, en los que las actividades representan algunas actividades que los actores y trabajadores realizan y las entidades son simplemente la información con que estos interactúan.

A continuación se describen los casos de usos del negocio y se muestra el diagrama de actividades de cada uno:

2.9.1 CUN Solicitar Reporte del Comportamiento de la Cuartelería

Caso de Uso	Solicitar Reporte del Comportamiento de la Cuartelería Estudiantil	
Actores	Vicedecano de Extensión y Residencia (inicia).	
Propósito	Permite al Vicedecano de Extensión y Residencia conocer el comportamiento de la cuartelería en un periodo de tiempo determinado y ver las principales tendencias negativas que se originan.	
Resumen	El CUN se inicia cuando el Actor solicita un parte del comportamiento de la cuartelería en un periodo de tiempo determinado, entrega a la secretaria los partes necesarios para conformar su solicitud, ésta lo conforma, entrega el parte y concluye el CU.	
Curso normal de los eventos		
Acción del actor	Respuesta del negocio	
1. El caso de uso se inicia cuando el actor solicita un reporte del comportamiento de la cuartelería.		
	2. Solicita los partes necesarios para conformar el reporte de la cuartelería.	
3. Entrega los partes necesarios.		
	4. Consulta los partes recibidos.	
	5. Conformar el reporte.	
6. Recibe el reporte solicitado.		
7. Termina el CU.		

Tabla 2.3 Descripción textual del caso de uso del negocio “Solicitar Reporte del Comportamiento de la Cuartelería Estudiantil”

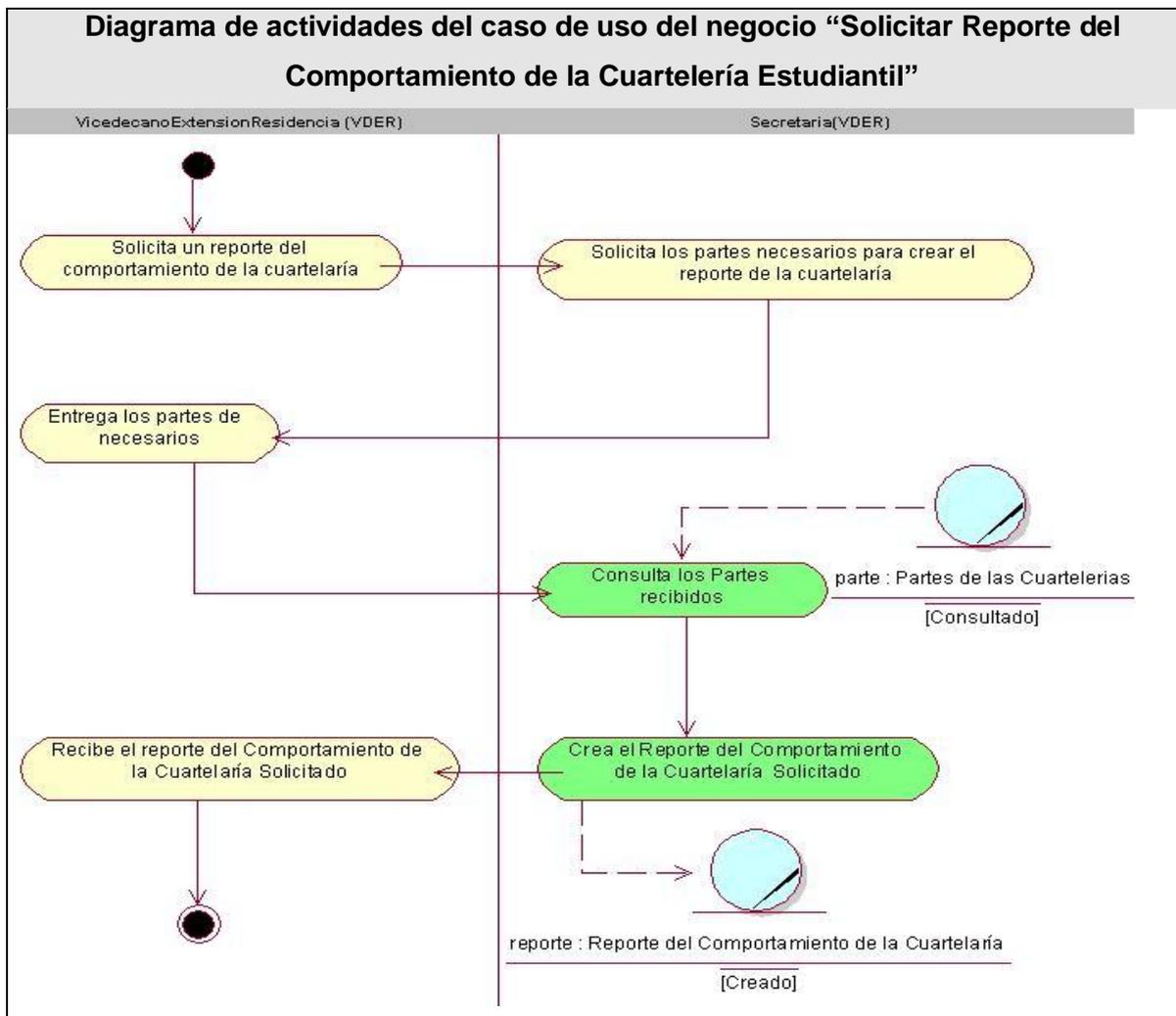


Figura 2.2 Diagrama de actividades del caso de uso del negocio “Solicitar Reporte del Comportamiento de la Cuartelería Estudiantil”

2.9.2 CUN Solicitar Reporte del Comportamiento de TSU

Caso de Uso	Solicitar Reporte del Comportamiento del TSU
Actores	Vicedecano de Extensión y Residencia (inicia).
Propósito	Permite al Vicedecano de Extensión y Residencia conocer el comportamiento del TSU en un periodo de tiempo determinado y ver las principales tendencias negativas que se originan.
Resumen	El CUN se inicia cuando el Actor solicita un parte del comportamiento del TSU en un periodo de tiempo determinado, entrega a la secretaria los

	partes necesarios para conformar su solicitud, ésta lo conforma, entrega el parte y concluye el CU.
Curso normal de los eventos	
Acción del actor	Respuesta del negocio
8. El caso de uso se inicia cuando el actor solicita un reporte del comportamiento del TSU.	
	9. Solicita los partes necesarios para conformar el reporte solicitado.
10. Entrega los partes necesarios.	
	11. Consulta los partes recibidos.
	12. Conformar el reporte.
13. Recibe el reporte solicitado.	
14. Termina el CU.	

Tabla 2.4 Descripción textual del caso de uso del negocio “Solicitar Reporte del Comportamiento del TSU”

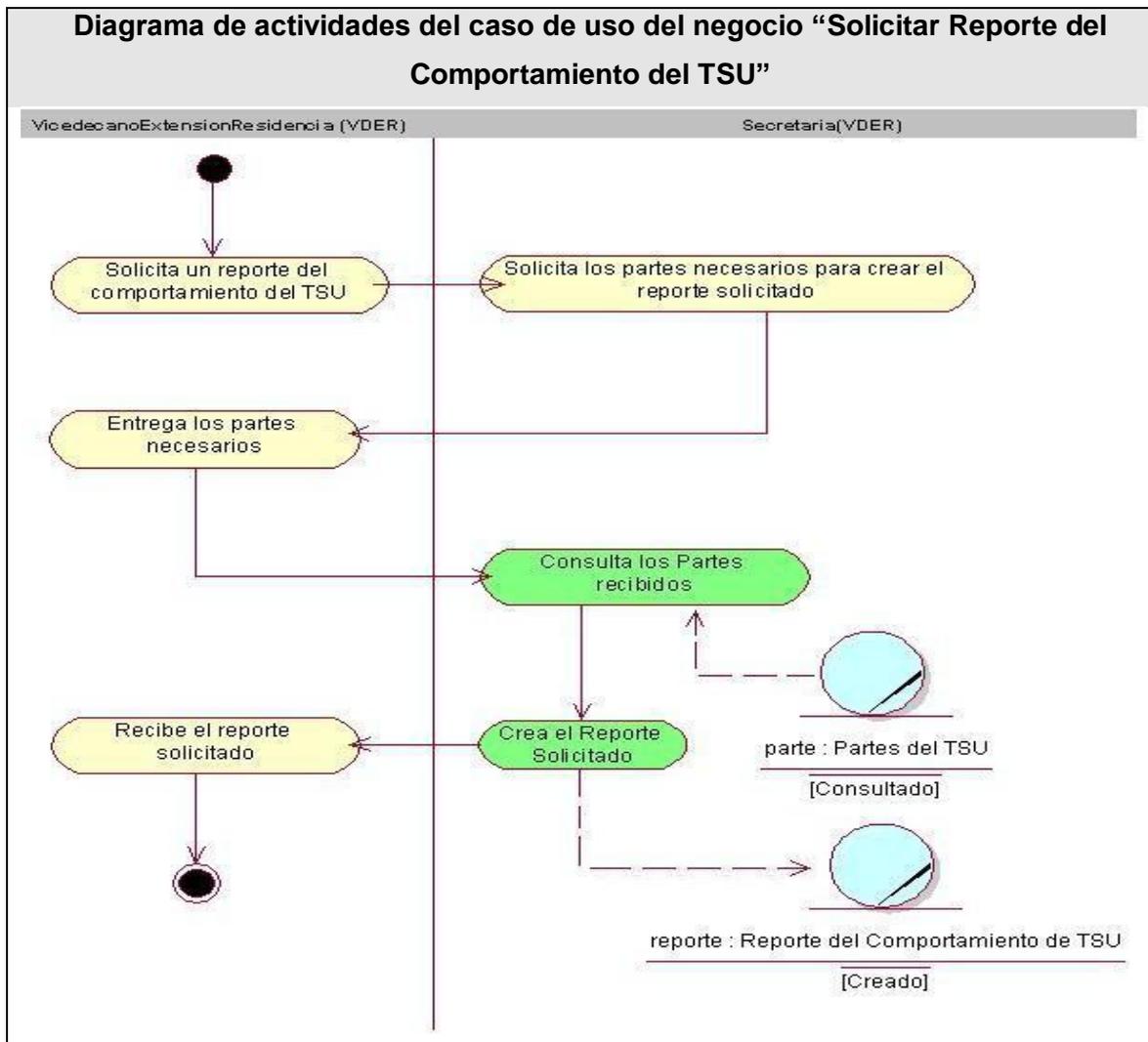


Figura 2.3 Diagrama de actividades del caso de uso del negocio “Solicitar Reporte del Comportamiento del TSU”

2.10 Diagrama de clases del negocio

Un modelo de objetos del negocio es un modelo interno a un negocio. Describe cómo cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y unidades de trabajo.

[14]

A continuación la figura 2.4 muestra el diagrama de clases del negocio:

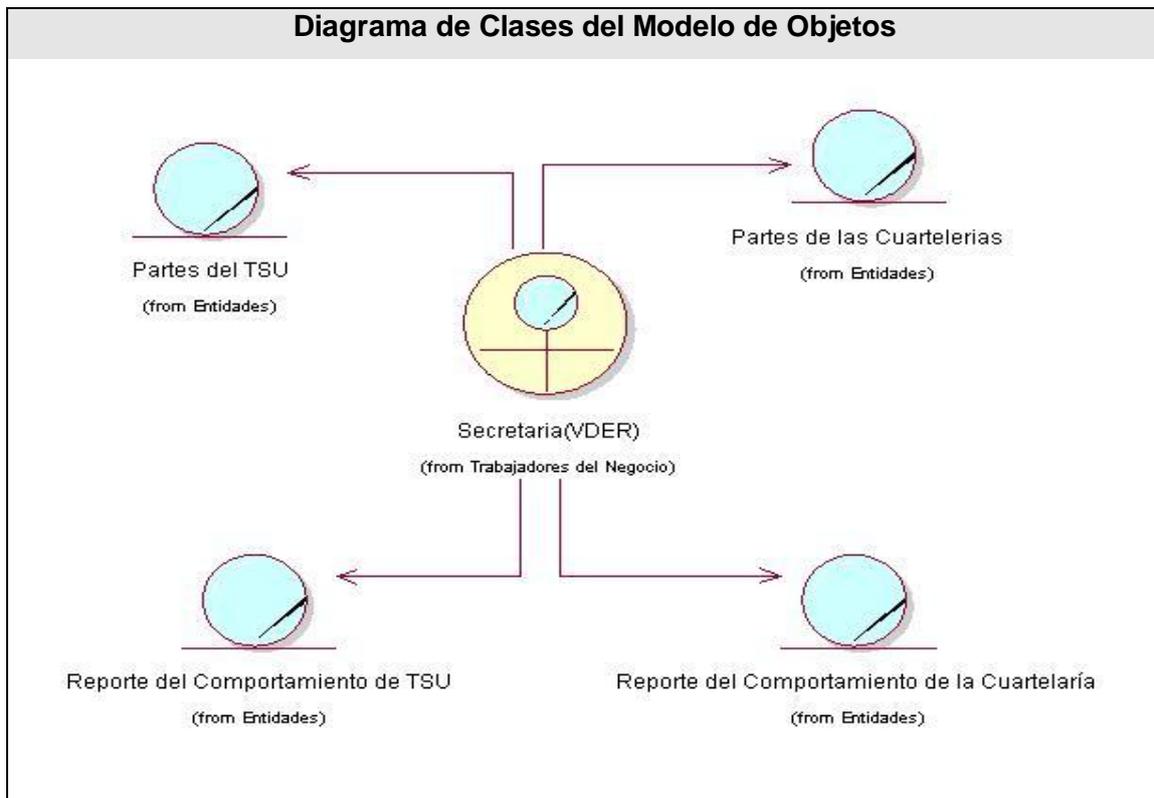


Figura 2.4 Diagrama de Clases del Modelo de Objetos

2.11 Definición de los requisitos funcionales

Los requisitos funcionales, son condiciones con las que el sistema debe cumplir, específicamente, define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas.

A continuación se presentan las funcionalidades con que debe cumplir el sistema:

RF 1 Permitir Autenticarse

- 1.1 Adicionar Usuario
- 1.2 Modificar Usuario
- 1.3 Eliminar Usuario

RF 2 Buscar Estudiante

- 2.1 Buscar estudiante por nombre y apellido
- 2.2 Buscar estudiante por brigada
- 2.3 Mostrar datos de un estudiante

RF 3 Gestionar Parte de la Guardia Estudiantil

- 3.1 Modificar Parte de la Guardia
- 3.2 Eliminar Parte de la Guardia

RF 4 Gestionar Postas de la Guardia

- 4.1 Adicionar Posta
- 4.2 Modificar Posta
- 4.3 Eliminar Posta

RF 5 Gestionar Brigada

- 5.1 Adicionar Brigada
- 5.2 Modificar Brigada
- 5.3 Eliminar Brigada

RF 6 Gestionar Persona

- 6.1 Adicionar Persona
- 6.2 Modificar Persona
- 6.3 Eliminar Persona

RF 7 Gestionar Cuartelería

- 7.1 Adicionar Cuartelería.
- 7.2 Modificar Cuartelería
- 7.3 Eliminar Cuartelería
- 7.4 Listar Cuartelería
 - 7.4.1 Listar las Cuartelerías de un día específico
 - 7.4.2 Listar las Cuartelerías que han sido evaluadas de mal en un intervalo de tiempo.

RF 8 Gestionar menú dinámico

RF 9 Gestionar TSU

- 9.1 Adicionar TSU
- 9.2 Modificar TSU
- 9.3 Eliminar TSU
- 9.4 Listar TSU
 - 9.4.1 Listar el TSU realizado en un día específico.
 - 9.4.2 Listar los TSU realizados por una brigada específica.
 - 9.4.3 Listar los estudiantes evaluados de mal en el TSU en un intervalo de tiempo determinado
 - 9.4.4 Listar los TSU que han sido evaluados de mal en un intervalo de tiempo determinado.

RF 10 Gestionar Roles

- 10.1 Adicionar Rol

- 10.2 Modificar Rol
- 10.3 Eliminar Rol
- 10.4 Listar Roles
- 10.5 Mostrar las funcionalidades de un rol determinado

2.12 Definición de los requisitos no funcionales

Los requisitos no funcionales son cualidades o propiedades con las que el producto debe contar, las mismas posibilitan que éste sea más atractivo, rápido, usable, confiable, entre otras. Asimismo, puede ser alguna restricción que éste debe tener para cumplir con una determinada funcionalidad.

A continuación se muestran los requisitos no funcionales:

Software

- RNF 1. Servidor Web Apache v1.x o superior.
- RNF 2. SGBD: PostgreSQL preferiblemente v7.x en adelante.
- RNF 3. Navegador Internet Explorer v4.0 o superior.
- RNF 4. Adobe Acrobat Reader 7.0 en adelante.
- RNF 5. Microsoft Excel.

Hardware

- RNF 6. Tarjeta de red.
- RNF 7. Para los servidores tanto Web como SGBD: PENTIUM II o superior con 256 MB de RAM o más.
- RNF 8. Capacidad de disco duro en Gigabyte, preferiblemente mayor a los 10 GB.

Apariencia o interfaz externa

- RNF 9. La interfaz no debe contener muchas imágenes que demoren las respuestas al usuario.
- RNF 10. El diseño de la interfaz debe ser sencillo y claro. Debe contener elementos visibles que identifiquen cada una de sus acciones.
- RNF 11. La navegabilidad debe ser sencilla.

Usabilidad

- RNF 12. El sistema podrá ser usado por cualquier persona que acceda a él que tenga algún conocimiento básico de computación y trabajo en la Web.
- RNF 13. Rápido acceso de búsqueda de la información, en tiempos cortos.

Rendimiento

RNF 14. El sistema deberá ser capaz de gestionar toda la información y dar respuesta a las solicitudes lo más rápido posible.

RNF 15. Debe ser eficiente a la hora de gestionar las solicitudes logrando que sin mucha navegación por el sitio se obtenga los resultados deseados.

RNF 16. Debe estar disponible las 24 horas del día.

Soporte

RNF 17. El sistema debe ser de fácil instalación y configuración, con vista a poder darle un mantenimiento asequible en caso de fallos.

Portabilidad

RNF 18. Multiplataforma. El sistema se podrá montar sobre Unix, Linux, Windows, etc. Así mismo podrá usar una serie de SGBD como PostgreSQL, MySQL, Oracle, entre otros, aunque preferiblemente se desea la portabilidad sobre software libre.

Seguridad

RNF 19. Realizar todas las validaciones pertinentes.

RNF 20. Chequeo de seguridad sobre las acciones tales como: verificación de borrado.

Políticos-culturales

RNF 21. El sistema debe tener una interfaz que esté acorde con el lugar donde se implantará, es decir, que refleje los ideales de la organización.

Legales

RNF 22. Reconocido y autorizado por instancias superiores tales como la directiva de la UCI.

RNF 23. Documentación legal de uso como Declaración de Autoría.

Ayuda y documentación en línea

RNF 24. Documentación de ayuda para uso del sistema, la cual estará asequible desde cualquier parte del mismo para satisfacer cualquier duda que el usuario presente con el manejo y uso de la aplicación.

Restricciones en el diseño y la implementación

RNF 25. Para la programación en PHP se recomienda el editor Nusphere PHPEd.

RNF 26. Se recomienda el uso de la arquitectura de n capas con el fin de tener una mayor organización del código, además de hacer el software más extensible.

2.13 Actores del sistema

Los actores del sistema definen el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que interactúan intercambiando información con éste.

A continuación se puntualizan los actores del sistema:

Actores del sistema	Justificación
Vicedecano de Extensión y Residencia	Tiene total dominio del sistema, puede ejecutar cualquiera de las acciones con que cuenta el mismo.
Usuario	Es todo aquel que hace uso del sistema.

Tabla 2.5 Descripción de los actores del sistema.

2.14 Diagrama de casos de uso del sistema

Los Casos de Uso del sistema (CUS) son un conjunto de encadenamiento de acciones que ejecuta un sistema y éstas a su vez causan un resultado observable para un actor.

A continuación se muestra la figura 2.5 correspondiente al diagrama de casos de uso del sistema:

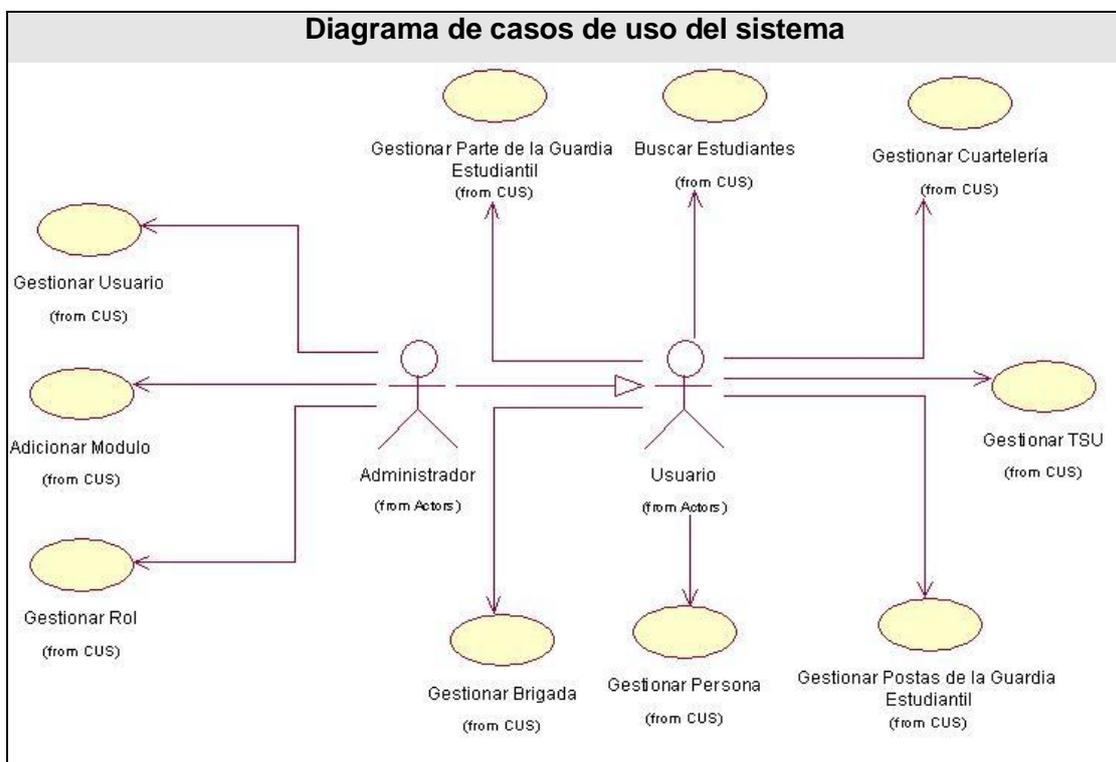


Figura 2.5 Diagrama de casos de uso del sistema

2.14.1 Descripción de los casos de uso del sistema

La descripción de los casos de uso del sistema, detallan las acciones que se suceden durante la interacción actor-sistema, con otras palabras, detalla el flujo de actividades que realiza el actor al hacer uso del sistema y las correspondientes respuestas del mismo. Por lo que, instaura un acuerdo entre clientes y desarrolladores especificando con cuales funcionalidades debe cumplir el sistema (requisitos).

A continuación se describen los casos de uso más significativos para este trabajo, el resto pueden ser consultados en el Anexo 2:

2.14.1.1 CUS Gestionar Parte de la Guardia Estudiantil

Caso de Uso	Gestionar Parte de la Guardia Estudiantil	
Actores	Vicedecano de Extensión y Residencia (inicia).	
Propósito	Permite al Vicedecano de Extensión y Residencia gestionar un nuevo parte de la guardia estudiantil.	
Resumen	El CUS se inicia cuando el Actor selecciona la opción de Gestionar parte de la guardia estudiantil, el sistema muestra las acciones Eliminar Parte y Modificar Parte. Seguidamente el Actor selecciona la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Actor y termina el caso de uso.	
Referencias	RF 3	
Precondiciones	Para modificar los datos de un parte de la guardia, el mismo tiene que estar seleccionado. Para eliminar, el parte de la guardia correspondiente debe estar seleccionado.	
Poscondiciones	Se modificó o eliminó los datos de un parte de la guardia.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El caso de uso se inicia cuando el actor selecciona la opción de Gestionar Parte de la Guardia Estudiantil.		
	2. El sistema muestra las opciones siguientes: –Modificar Parte (Ver Sección “Modificar	

	Datos de Parte de la Guardia Estudiantil”). –Eliminar Parte (Ver Sección “Eliminar Datos de Parte de la Guardia Estudiantil”).
3. Selecciona la opción Eliminar Parte.	
	4. Muestra un formulario dando la posibilidad de selecciona el intervalo de fechas en las que se encuentra el parte o los partes que se desean eliminar.
5. Selecciona la fecha o el intervalo de las mismas del parte o los partes a eliminar.	
	6. Muestra una alerta de confirmación “Es seguro que desea eliminar el parte o los partes de la guardia seleccionado.”
7. Confirma la acción.	
	8. Consulta la base de datos y elimina el parte o los partes de la fecha o el intervalo de fechas especificado.
	9. Muestra el mensaje de confirmación “Proceso Satisfactorio.”
	10. El caso de uso termina.
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
7.a Selecciona la opción cancelar.	
Acción del actor	Respuesta del sistema
	7.a.1. El caso de uso termina.
Sección 1:”Modificar Datos de Parte de la Guardia Estudiantil”	
Curso Normal de los eventos:	

Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Modificar Parte.	
	2. Muestra un calendario.
3. Selecciona la fecha en la que se realizó la guardia de la cual se desea modificar los datos de su parte.	
	4. Consulta la base de datos y obtiene los datos del parte especificado.
	5. Muestra todos los datos del parte permitiendo modificarlos.
6. Modifica los datos de su interés y selecciona la opción "Modificar".	
	7. Valida los datos.
	8. Consulta la Base de Datos y actualiza los datos del parte correspondiente.
	9. Muestra el mensaje de información "Proceso Satisfactorio."
	10. El caso de uso termina.
Curso Alternativo de los eventos:	
*a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.

Tabla 2.6 Descripción del CUS "Gestionar Parte de la Guardia Estudiantil"

2.14.1.2 CUS Gestionar Cuartelería

Caso de Uso	Gestionar Cuartelería
Actores	Vicedecano de Extensión y Residencia (inicia).
Propósito	Permite al Vicedecano de Extensión y Residencia gestionar una cuartelería dentro del sistema.
Resumen	El CUS se inicia cuando el Actor selecciona la opción de Gestionar Cuartelería, el sistema muestra las opciones Insertar Datos de Cuartelería, Modificar Datos de Cuartelería y Eliminar Datos de

	Cuartelería, Listar Cuartererías. Seguidamente el Actor selecciona la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Actor y termina el caso de uso.	
Referencias	RF 7	
Precondiciones	<p>Para modificar los datos de una cuarterería, la misma debe estar seleccionada.</p> <p>Para eliminar los datos de una cuarterería la misma debe estar seleccionada.</p> <p>Para consultar los datos de una cuarterería la misma tiene que estar seleccionada.</p>	
Poscondiciones	Se creo, modificó, eliminó o listo los datos de una cuarterería.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El caso de uso se inicia cuando el actor selecciona la opción de Gestionar Cuarterería.		
	<p>2. El sistema muestra las opciones siguientes:</p> <ul style="list-style-type: none"> – Adicionar Cuarterería (Ver Sección “Adicionar Datos de una Cuarterería”). – Modificar Cuarterería (Ver Sección “Modificar Datos de una Cuarterería”). – Eliminar Cuarterería (Ver Sección “Eliminar Datos de una Cuarterería”). – Listar Cuartererías (Ver Sección “Listar Datos de Cuartererías”) 	
3. Selecciona la opción Adicionar Cuarterería.		
	<p>4. Muestra un formulario para insertar los siguientes datos de la cuarterería:</p> <ul style="list-style-type: none"> – Nombre del Estudiante. – Edificio. – Paso de Escalera: 	

	<ul style="list-style-type: none"> - Fecha. - Cumplimiento. - Evaluación.
5. Introduce los datos y selecciona la opción Adicionar.	
	6. Valida los datos.
	7. Guarda los datos de la persona.
	8. Muestra el mensaje de información "Proceso Satisfactorio."
	9. El caso de uso termina.
Curso Alterno de los eventos:	
*a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
6.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	6.a.1. Muestra el mensaje de error "Existen datos incorrectos".
	6.a.2. Regresa al paso 5 del Curso Normal de Eventos.
Sección 1:"Modificar datos de una cuartería"	
Curso Normal de los eventos:	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Modificar Cuartería.	
	2. Muestra un calendario.
3. Selecciona la fecha en que se realizo la cuartería que se desea modificar.	
	4. Consulta la base de datos y obtiene Edificio, Paso de escalera, de cada cuartería de esa fecha.

	5. Muestra los datos obtenidos de las cuarterías y permite seleccionar una.
6. Selecciona la cuartería a la que le desea modificar los datos y la opción modificar.	
	7. Consulta la base de datos y obtiene todos os datos de la cuartería seleccionada.
	8. Muestra los datos en un formulario, permitiendo modificarlos.
9. Modifica los datos y selecciona la opción modificar datos.	
	10. Valida los datos.
	11. Actualiza los datos de la cuartería.
	12. Muestra el mensaje de información "Proceso Satisfactorio."
	13. El caso de uso termina.
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
10.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	10.a.1. Muestra el mensaje de error "Existen datos incorrectos."
	10.a.2. Regresa al paso 9 del Curso Normal de Eventos.
Sección 2: Eliminar datos de una cuartería	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema
1. Selecciona la opción eliminar los datos de una cuartería.	

	2. Muestra un calendario.
3. Selecciona la fecha en que se realizo la cuarterería que se desea eliminar.	
	4. Consulta la base de datos y obtiene Edificio, Paso de escalera, de cada cuarterería de esa fecha.
	5. Muestra los datos obtenidos de las cuartererías y permite seleccionar una.
6. Selecciona la cuarterería a la que le desea eliminar los datos.	
	7. Muestra todos los datos de la cuarterería seleccionada.
8. Selecciona la opción "Eliminar".	
	9. Muestra el mensaje de información siguiente: "Se eliminarán los datos de la cuarterería".
10. Selecciona la opción Eliminar.	
	11. Elimina los datos de la cuarterería.
	12. Muestra el mensaje de información "Proceso Satisfactorio."
	13. El caso de uso termina.
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
8. a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	8.a.1. El caso de uso termina.
Sección 3: Listar cuartererías	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema

1. Selecciona la opción listar cuartería.	
	2. El sistema muestra las opciones siguientes: –Listar por Fecha (Ver Sección “Listar Cuarterías por Fechas”). –Listar cuarterías con una evaluación determinada Evaluadas (Ver Sección “Listar Cuarterías Evaluadas de...”).
3. Selecciona la opción listar por fecha.	
	4. Muestra un calendario.
5. Selecciona la fecha.	
	6. Consulta la base de datos y obtiene todos los datos de las cuarterías de esa fecha.
	7. Lista las cuarterías especificando Edificio, Paso de Escalera, permitiendo seleccionar cualquiera que sea de interés.
8. Selecciona la cuartería a consultar y la opción “Consultar Datos”.	
	9. Muestra todos los datos de la cuartería especificada.
10. Consulta los datos.	
11. El caso de uso termina.	
Curso Alterno de los eventos:	
*a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
Sección 3.1: Listar cuarterías evaluadas de X	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema

1. Selecciona la opción listar cuarterías evaluadas de X.	
	2. Muestra un formulario con dos calendarios.
3. Selecciona una fecha inicial, una final en la que estarán enmarcadas las cuarterías, la evaluación y la opción "Consultar Datos".	
	4. Consulta la base de datos y obtiene todos los datos de las cuarterías dentro del rango especificado y evaluadas de X.
	5. Lista las cuarterías especificando Edificio, Paso de Escalera.
6. Selecciona la opción "Consultar Datos" de la cuartería de interés.	
	7. Muestra todos los datos de la cuartería especificada.
12. Consulta los datos.	
13. El caso de uso termina.	
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.

Tabla 2.7 Descripción del caso de uso "Gestionar Cuartería"

2.14.1.3 CUS Gestionar TSU

Caso de Uso	Gestionar TSU
Actores	Vicedecano de Extensión y Residencia (inicia).
Propósito	Permite al Vicedecano de Extensión y Residencia gestionar un parte de TSU dentro del sistema.
Resumen	El CUS se inicia cuando el Actor selecciona la opción de Gestionar TSU,

	el sistema muestra las opciones Insertar Datos de TSU, Modificar Datos de TSU y Eliminar Datos de TSU, Listar TSU. Seguidamente el Actor selecciona la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Actor y termina el caso de uso.
Referencias	RF 9
Precondiciones	Para modificar, eliminar o consultar los datos de un TSU, el mismo debe estar seleccionado.
Poscondiciones	Se creo, modificó, eliminó o consulto los datos de un TSU.
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Gestionar TSU.	
	2. El sistema muestra las opciones siguientes: <ul style="list-style-type: none"> – Adicionar TSU (Ver Sección “Adicionar Datos de un TSU”). – Modificar TSU (Ver Sección “Modificar Datos de un TSU”). – Eliminar TSU (Ver Sección “Eliminar Datos de un TSU”). – Listar TSU (Ver Sección “Listar Datos de TSU”)
3. Selecciona la opción Adicionar TSU.	
	4. Muestra un listado con todas las brigadas existentes en el sistema.
5. Selecciona la brigada que realizo el TSU.	
	6. Muestra un formulario especificando los nombres de los estudiantes de la brigada , profesor guía, jefe de brigada, pide introducir los datos siguientes: <ul style="list-style-type: none"> – Lugar.

	<ul style="list-style-type: none"> – Sesión del Día. – Fecha. – Evaluación General. – Evaluación de cada Estudiantes.
7. Introduce los datos y selecciona la opción “Adicionar Datos”.	
	8. Valida los datos.
	9. Muestra el mensaje de información “Proceso Satisfactorio.”
	10. El caso de uso termina.
Curso Alterno de los eventos:	
*a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
8.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	8.a.1. Muestra el mensaje de error “Existen datos incorrectos.”
	8.a.2. Regresa al paso 7 del Curso Normal de Eventos.
Sección 1:”Modificar datos de un TSU”	
Curso Normal de los eventos:	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Modificar TSU.	
	2. Pide seleccionar el criterio de búsqueda: <ul style="list-style-type: none"> – Fecha. – Brigada.
3. Selecciona el criterio de “Fecha”.	
	4. Muestra un formulario con un calendario.
5. Escoge la fecha del TSU.	
	6. Consulta la base de datos y obtiene los

	datos del TSU.
	7. Muestra los datos siguientes de cada TSU permitiendo seleccionarlos: <ul style="list-style-type: none"> - Fecha. - Brigada. - Jefe de Brigada. - Profesor Guía. - Sesión del Día. - Lugar. - Evaluación.
8. Selecciona el TSU de su interés.	
	9. Muestra todos los datos en un formulario permitiendo modificarlos.
10. Modifica los datos y selecciona la opción "Modificar".	
	11. Valida los datos.
	12. Actualiza los datos de la cuarterería.
	13. Muestra el mensaje de información "Proceso Satisfactorio."
	14. El caso de uso termina.
3.a Selecciona el criterio "Brigada"	
Acción del actor	Respuesta del sistema
3.a.1. Selecciona la opción "Brigada"	
	3.a.2. Muestra un listado con todas las brigadas existentes en el sistema.
3.a.3. Escoge la brigada y la opción "Buscar".	
3.a.4. Regresa al paso 6 del curso normal de eventos.	
Curso Alternativo de los eventos:	
*.a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema

	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
11.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	11.a.1.Muestra el mensaje de error “Existen datos incorrectos.”
	11.a.2.Regresa al paso 9 del Curso Normal de Eventos.
Sección 2: Eliminar datos de un TSU	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema
1. Selecciona la opción Eliminar TSU.	
	2. Pide especificar el criterio por el que se desea realizar la búsqueda. – Fecha. – Brigada.
3. Selecciona el criterio Fecha.	
	4. Consulta la base de datos y obtiene los datos, Brigada y Fecha de los TSU.
	5. Lista los TSU especificando los datos anteriores, permitiendo seleccionar.
6. Selecciona el TSU que desea eliminar y la opción “Eliminar”.	
	7. Muestra el mensaje de información siguiente: “Se eliminarán los datos del TSU”.
8. Selecciona la opción Eliminar.	
	9. Elimina los datos del TSU.
	10. Muestra el mensaje de información “Proceso Satisfactorio.”
	11. El caso de uso termina.
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar.	

Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
8. a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	8.a.1. El caso de uso termina.
Sección 3: Listar TSU	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema
1. Selecciona la opción listar TSU.	
	2. El sistema pide especificar el tipo de búsqueda: – Fecha. – Brigada.
3. Selecciona la opción fecha.	
	4. Muestra un calendario.
5. Selecciona la fecha.	
	6. Consulta la base de datos y obtiene todos los datos de los TSU realizados en esa fecha.
	7. Lista los TSU especificando brigada y fecha, permitiendo seleccionar un elemento de la lista.
8. Selecciona el TSU a consultar y la opción "Consultar Datos".	
	9. Muestra todos los datos del TSU especificado.
14. Consulta los datos.	
15. El caso de uso termina.	
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	*.a.3. Cancela los datos.

	*.a.4. El caso de uso termina.
Sección 3.1: Listar TSU por Brigada	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema
1. Selecciona la opción listar TSU por brigadas.	
	2. Muestra un listado con todas las brigadas existentes en el sistema.
3. Selecciona una brigada.	
	4. Consulta la base de datos y obtiene todos los datos de los TSU pertenecientes a la brigada especificada.
	5. Lista los TSU especificando nombre brigada y fecha.
6. Selecciona el TSU y la opción consultar.	
	7. Muestra todos los datos del TSU especificado.
16. Consulta los datos.	
17. El caso de uso termina.	
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	*.a.3. Cancela los datos.
	*.a.4. El caso de uso termina.
Sección 3.1: Listar Estudiantes Evaluados de Mal en un Intervalo de Tiempo	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema
1. Selecciona la opción listar estudiantes evaluadas de mal un intervalos de tiempo.	

	2. Muestra un formulario con dos calendarios.
3. Selecciona una fecha inicial, una final en la que estarán enmarcadas los TSU.	
	4. Consulta la base de datos y obtiene todos los datos de los estudiantes evaluados de mal en el rango especificado.
	5. Lista las cuarterías especificando nombre del estudiante, brigada, fecha del TSU.
6. Consulta los datos.	
7. El caso de uso termina.	
Curso Alterno de los eventos:	
*a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	*.a.5. Cancela los datos.
	*.a.6. El caso de uso termina.

Tabla 2.8 Descripción del caso de uso “Gestionar TSU”

2.14.1.4 Descripción del CUS “Gestionar Persona”

Caso de Uso	Gestionar Persona
Actores	Vicedecano de Extensión y Residencia (inicia).
Propósito	Permite al Vicedecano de Extensión y Residencia gestionar una nueva persona dentro del sistema, especificando cual es su desempeño en el mismo (Profesor o Estudiante).
Resumen	El CUS se inicia cuando el Actor selecciona la opción de Gestionar Persona, el sistema muestra las acciones Insertar Persona, Modificar Datos de Persona y Eliminar Datos de Persona. Seguidamente el Actor selecciona la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Actor y termina el caso de uso.

Referencias	RF 6
Precondiciones	Para modificar los datos de una persona, la misma debe de estar seleccionada. Para eliminar a la persona correspondiente debe estar seleccionada.
Poscondiciones	Se creo, modificó o eliminó los datos de una persona.
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Gestionar Persona.	
	2. El sistema muestra las opciones siguientes: – Adicionar Persona (Ver Sección “Adicionar Datos de una Persona”). – Modificar Persona (Ver Sección “Modificar Datos de una Persona”). – Eliminar Persona (Ver Sección “Eliminar Datos de una Persona”).
3. Selecciona la opción Adicionar Persona.	
	4. Pide escoger el tipo de persona. – Estudiante. – Profesor.
5. Selecciona la opción Estudiante.	
	6. Muestra un formulario para insertar los datos siguientes de la persona: – Primer nombre. – Segundo nombre. – Primer apellido. – Segundo apellido. – Sexo. – Carnet de Identidad. – Solapín.

	<ul style="list-style-type: none"> – Brigada. – ID.
5. Introduce los datos y selecciona la opción “Adicionar”.	
	6. Valida los datos.
	7. Guarda los datos de la persona.
	8. Muestra el mensaje de información “Proceso Satisfactorio.”
	9. El caso de uso termina.
Curso Alternativo de los eventos:	
*a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
5.a Selecciona la opción Profesor	
Acción del actor	Respuesta del sistema
	5.a.1. Muestra un formulario para insertar los datos siguientes de la persona: <ul style="list-style-type: none"> – Primer nombre. – Segundo nombre. – Primer apellido. – Segundo apellido. – Carnet de Identidad. – Solapín. – Sexo. – ID.
	5.a.2. Regresa al paso 5 del curso normal de eventos.
6.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	6.a.1. Muestra el mensaje de error “Alguno de los datos son incorrectos.”

	6.a.2. Regresa al paso 6 del Curso Normal de Eventos.
Sección 1: "Modificar los datos de una persona"	
Curso Normal de los eventos:	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Modificar Persona.	
	2. Permite escoger el tipo de persona. – Estudiante – Profesor
3. Selecciona la opción "Estudiante".	
	4. Consulta la base y obtiene un listado con todos los nombres de los estudiantes en el sistema.
	5. Muestra los nombres de los estudiantes permitiendo escoger el que se desea modificar.
6. Escoge el nombre del estudiante al que se le desea modificar los datos.	
	7. Consulta la Base de Datos y obtiene los datos del estudiante al que se le desean modificar los mismos.
	8. Muestra un formulario con los datos permitiendo modificarlos.
9. Modifica los datos de su interés y selecciona la opción actualizar datos.	
	10. Valida los datos.
	11. Actualiza los datos de la persona.
	12. Muestra el mensaje de información "Proceso Satisfactorio."
	13. El caso de uso termina.
Curso Alterno de los eventos:	

*.a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
3.a Escoge la opción Profesor	
Acción del actor	Respuesta del sistema
3.a.1. Escoge la opción "Profesor".	
	3.a.2. Consulta la base y obtiene un listado con todos los nombres de los profesores en el sistema.
	3.a.3. Muestra los nombres de los profesores permitiendo escoger el que se desea modificar.
3.a.4. Escoge el nombre del profesor al que se le desea modificar los datos.	
	3.a.5. Consulta la Base de Datos y obtiene los datos del estudiante al que se le desean modificar los mismos.
	3.a.6. Muestra un formulario con los datos permitiendo modificarlos.
	3.a.7. Regresa al paso 9 del curso normal de eventos.
10.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	10.a.1. Muestra el mensaje de error "Alguno de los datos son incorrectos."
	10.a.2. Regresa al paso 6 del Curso Normal de Eventos.
Sección 2: Eliminar datos de una persona	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema
1. Selecciona la opción eliminar los datos de una persona.	

	2. Permite escoger el tipo de persona. – Estudiante. – Profesor.
3. Escoge la opción “Estudiante”.	
	4. Consulta la Base de Datos y obtiene el nombre de todos los estudiantes en el sistema.
	5. Muestra un listado con todos los nombres, permitiendo escoger el que se desea eliminar.
6. Selecciona el nombre de la persona y la opción eliminar datos.	
	7. Muestra el mensaje de información siguiente: “Se eliminarán los datos de la persona”.
8. Selecciona la opción Eliminar.	
	9. Elimina los datos de la persona.
	10. Muestra el mensaje de información “Proceso Satisfactorio.”
	11. El caso de uso termina.
Curso Alterno de los eventos:	
*a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	3.a.1. Cancela los datos.
	3.a.2. El caso de uso termina.
3. a Escoge la opción “Profesor”	
Acción del actor	Respuesta del sistema
3.a.1. Escoge la opción “Profesor”.	
	3.a.2. Consulta la Base de Datos y obtiene el nombre de todos los profesores en el sistema.
	3.a.3. Regresa al paso 5 del curso normal de eventos.

Tabla 2.9 Descripción del caso de uso “Gestionar Personas”

2.15 Conclusiones

En este capítulo fueron detalladas las características con que contará el sistema, realizando el análisis de los procesos actuales del negocio, identificando los actores y trabajadores que intervienen en el mismo y con cuáles entidades y actividades interactúan éstos. Se identificaron, los requisitos funcionales y no funcionales a cumplir por el sistema, fueron expuestos los casos de uso a tratar durante su desarrollo, con la apropiada descripción textual de cada uno, lo cual da una visión general acerca de lo que el sistema debe hacer, por lo que se está en condiciones de pasar a ver cómo es que el mismo va a realizar los procedimientos anteriormente descritos y con ello, darle solución a los problemas identificados.

Análisis y Diseño del Sistema

3.1 Introducción

En este capítulo se presenta la idea general concerniente al análisis y diseño del sistema propuesto, donde el análisis tiene como objetivo mantener un modelo eficiente de la solución propuesta que sirva de base para el diseño. Este último tiene como objetivo, presentar cómo es que está construido el sistema, lo cual se realiza a partir de los diagramas de clases Web, éstos tienen la finalidad de describir la interacción entre las distintas páginas de la aplicación. Se presenta el diagrama de clases persistentes de la base de datos del sistema, además de los principios de diseño utilizados.

3.2 Diagrama de clases del análisis

Los diagramas de clases de análisis, expresan relación entre las clases, están compuestos por clases de tres tipos fundamentales: interfaz, controladora y entidad.

A continuación se muestran los principales diagramas de clases del análisis, el resto se pueden consultar en el Anexo 3:

3.2.1 CU Gestionar Parte de Guardia

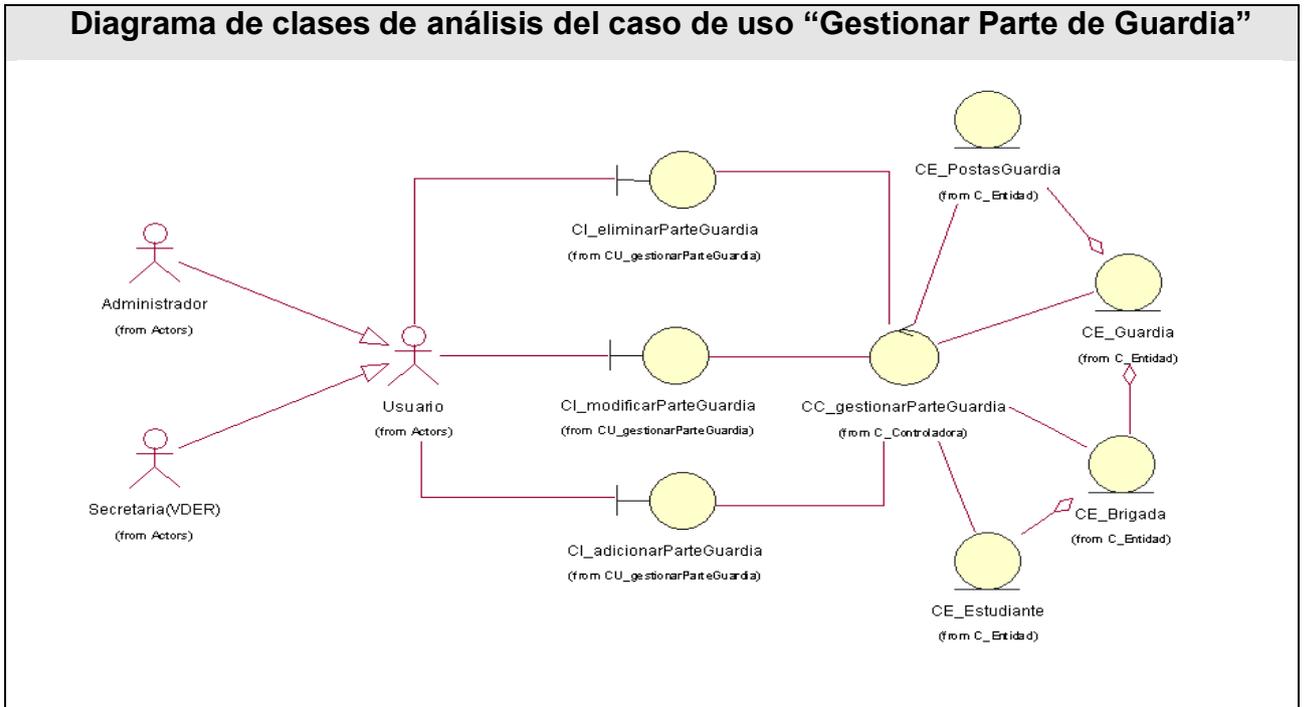


Figura 3.1 Diagrama de clases del análisis del caso de uso "Gestionar Parte de Guardia"

3.2.2 CU Gestionar Cuartelería

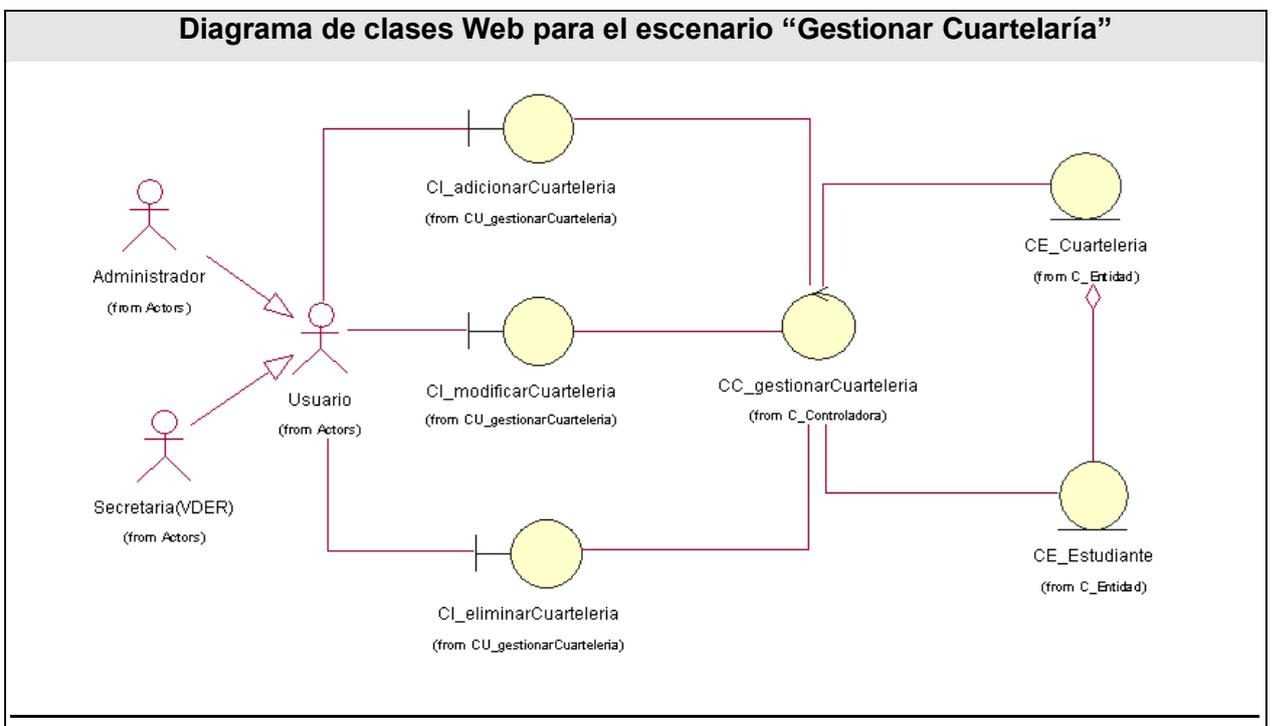


Figura 3.2 Diagrama de clases de análisis caso de uso "Gestionar Cuartelería".

3.2.3 CU Gestionar TSU

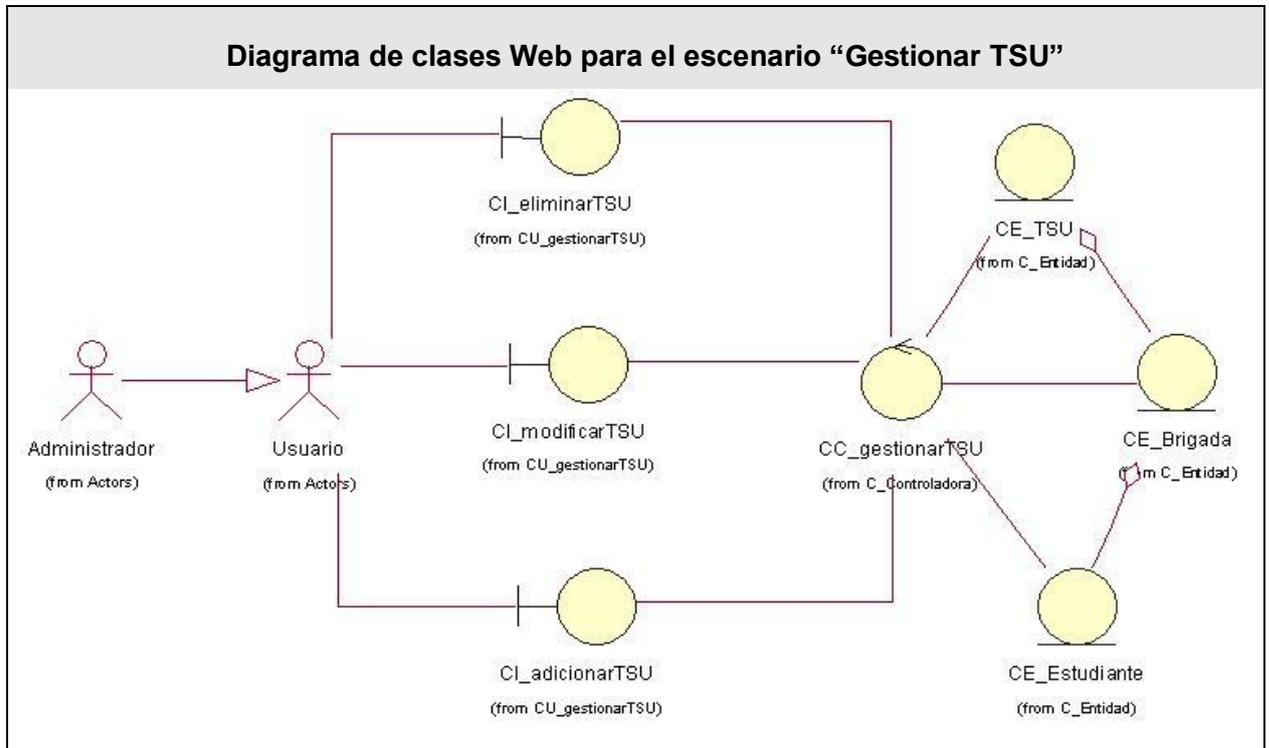


Figura 3.3 Diagrama de clases de análisis del caso de uso "Gestionar TSU"

3.2.4 CU Gestionar Persona

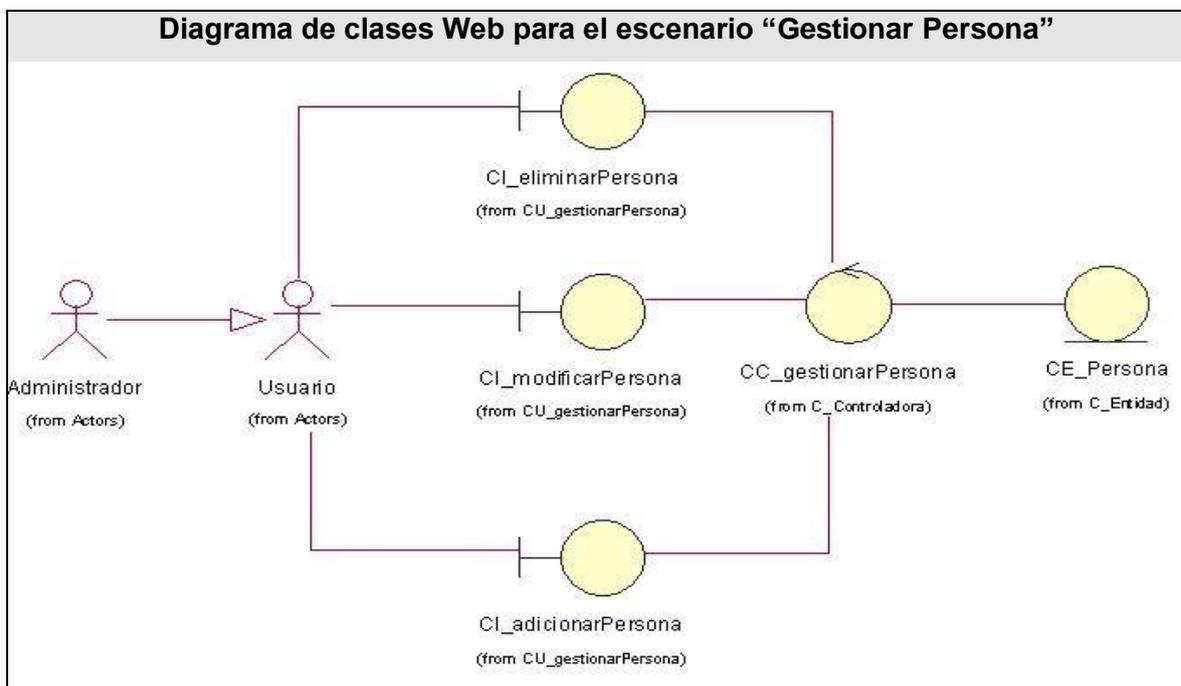


Figura 3.4 Diagrama de clases de análisis del caso de uso "Gestionar Persona"

3.3 Diagrama de clases del Diseño

Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. En el caso de las aplicaciones Web, el diagrama de clases representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase.

A continuación se muestran los diagramas de clases Web para alguna de las funcionalidades con que cuentan los principales casos de uso del sistema, el resto de los mismos se encuentran en el anexo 4:

3.3.1 Modificar Parte de la Guardia Estudiantil

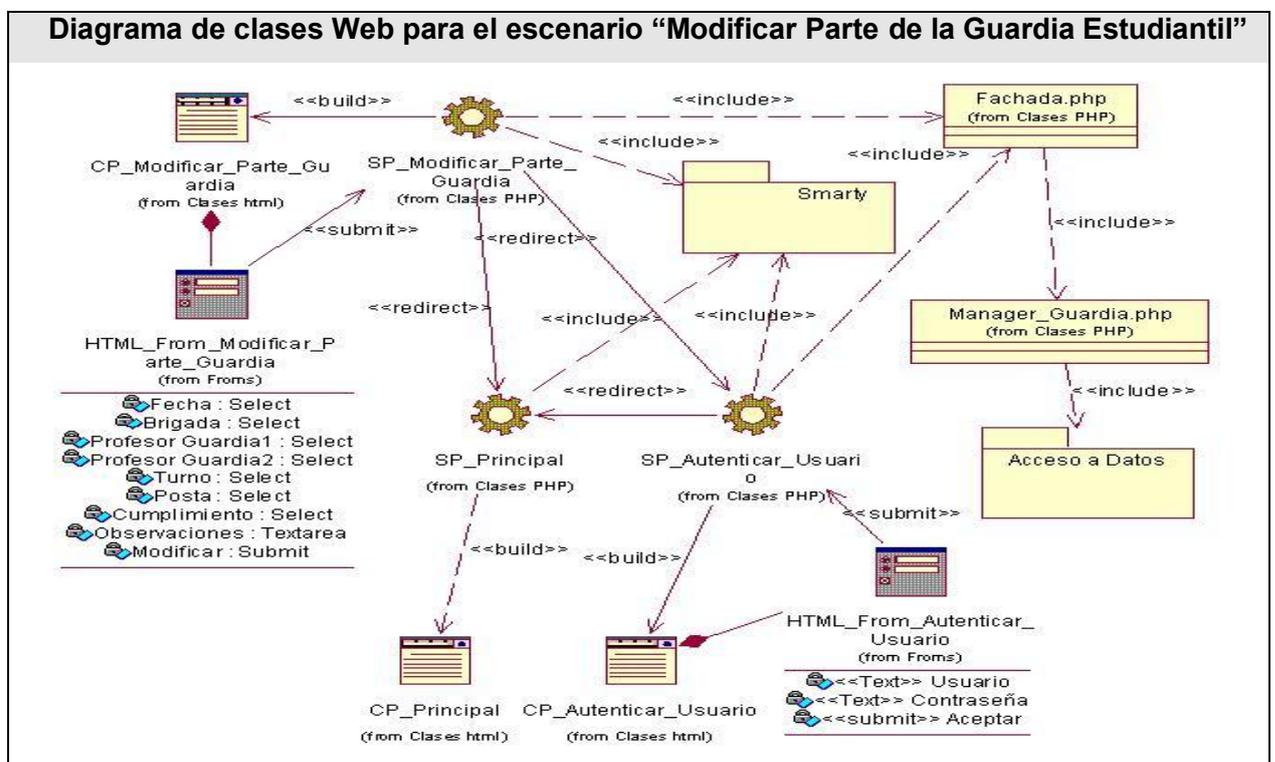


Figura 3.5 Diagrama de clases Web para el escenario "Modificar Parte de la Guardia Estudiantil"

3.3.2 Eliminar Modificar Parte de la Guardia Estudiantil

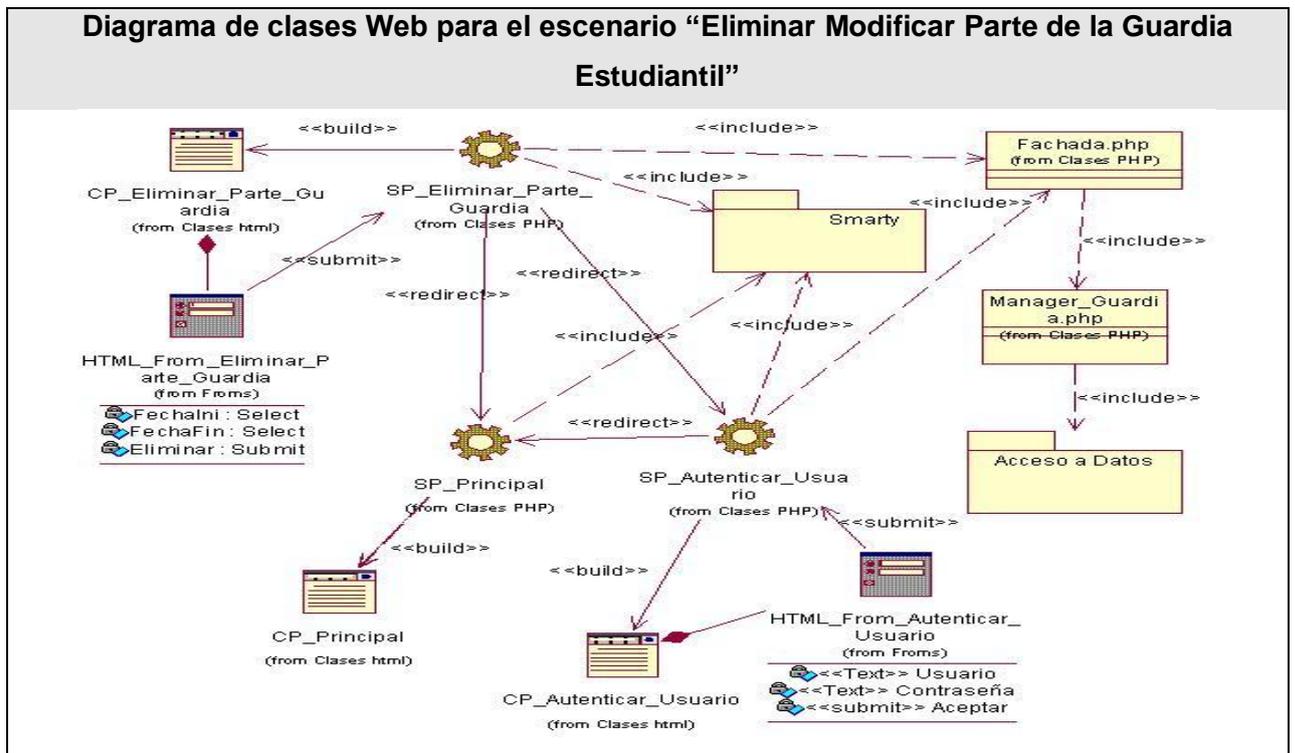


Figura 3.6 Diagrama de clases Web para el escenario “Eliminar Modificar Parte de la Guardia Estudiantil”

3.3.3 Adicionar Cuartelería

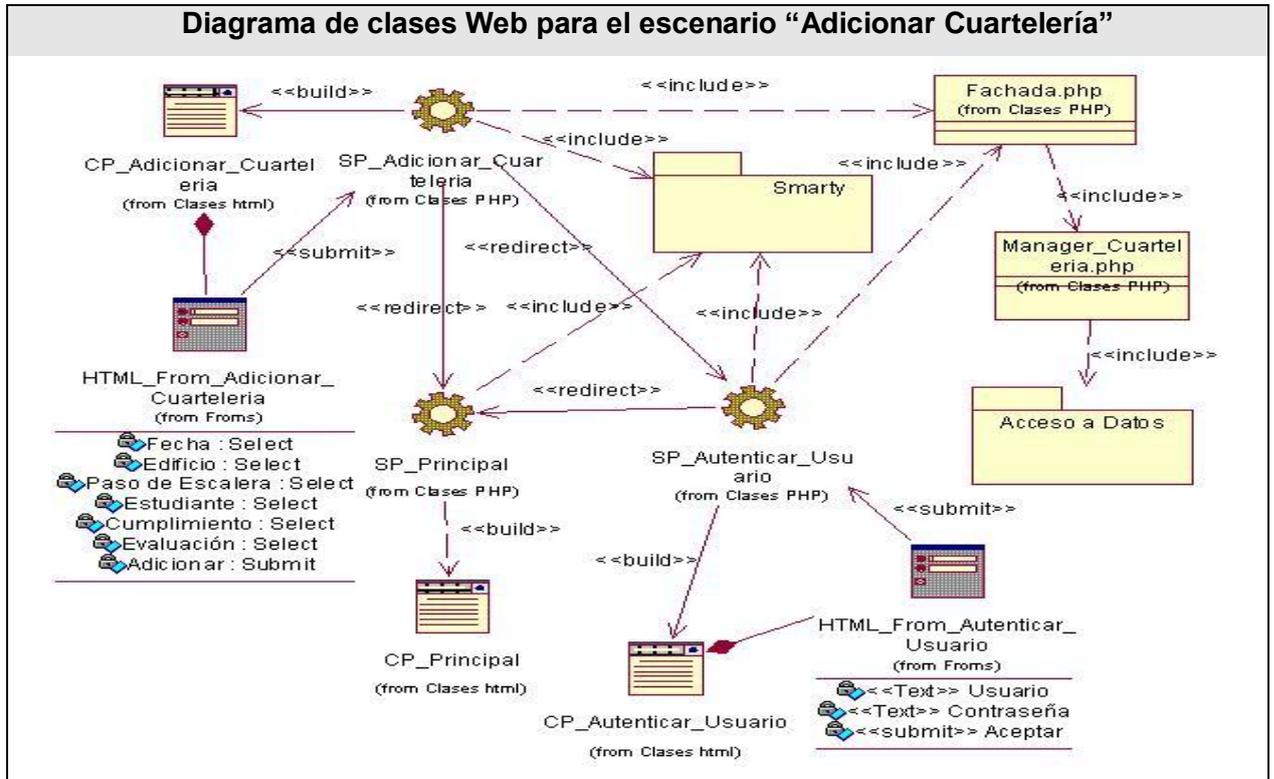


Figura 3.7 Diagrama de clases Web para el escenario "Adicionar Cuartelería"

3.3.4 Buscar TSU

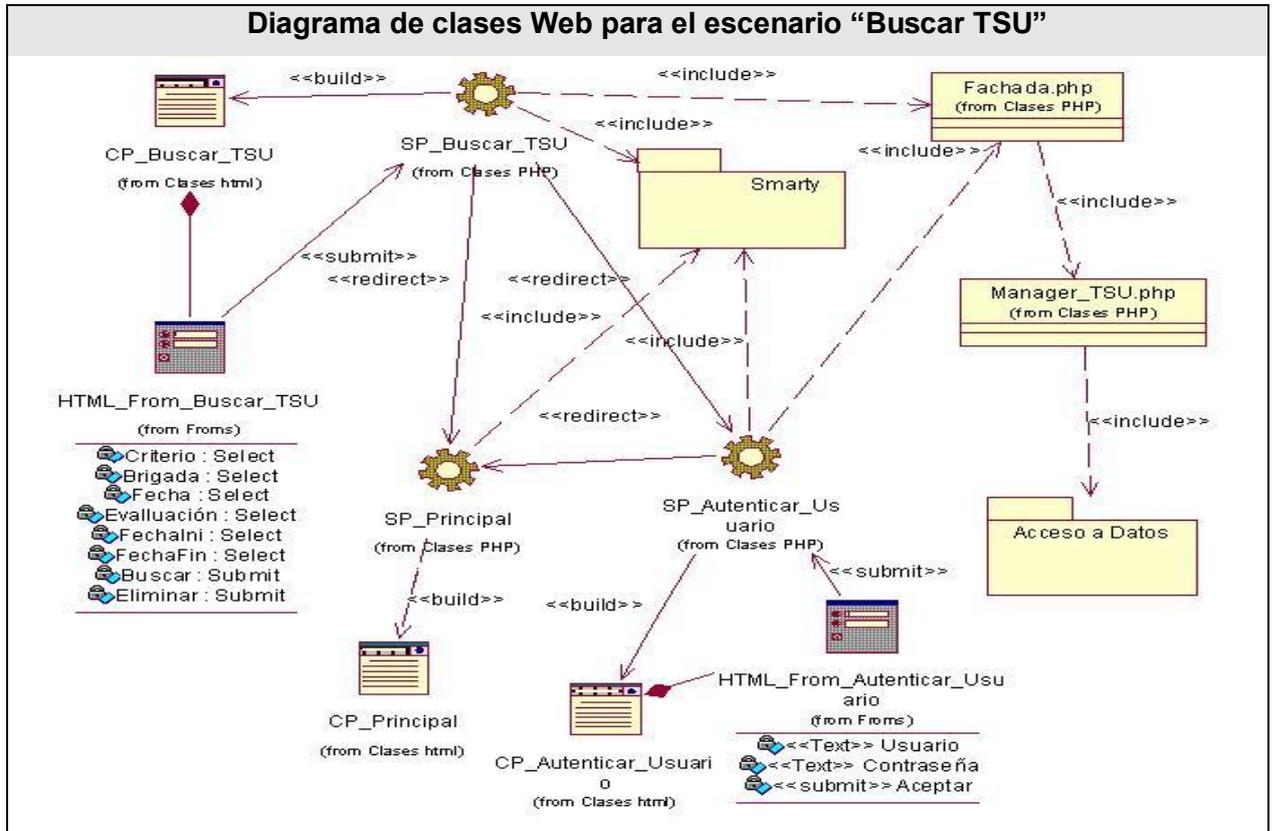


Figura 3.8 Diagrama de clases Web para el escenario "Buscar TSU"

3.3.5 Eliminar Persona

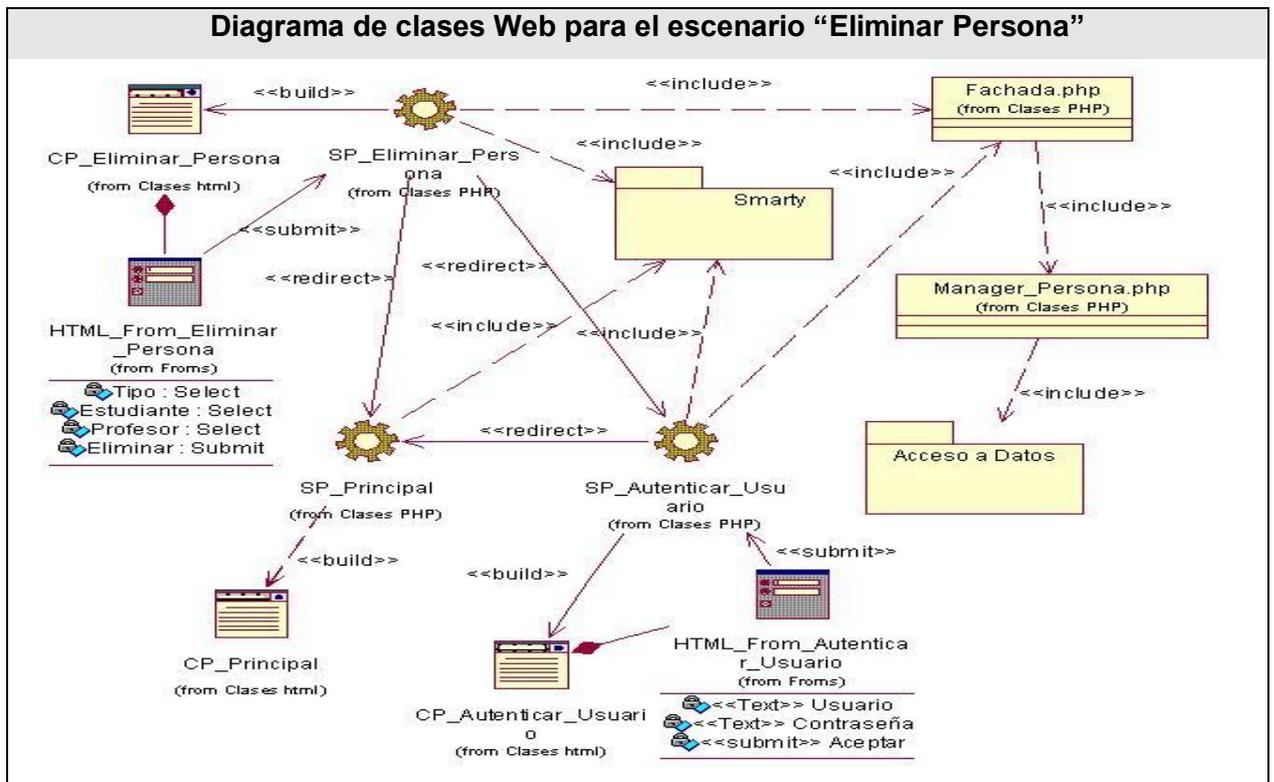


Figura 3.9 Diagrama de clases Web para el escenario "Eliminar Persona"

3.3.6 Diagrama de clases del subsistema de Acceso a Datos

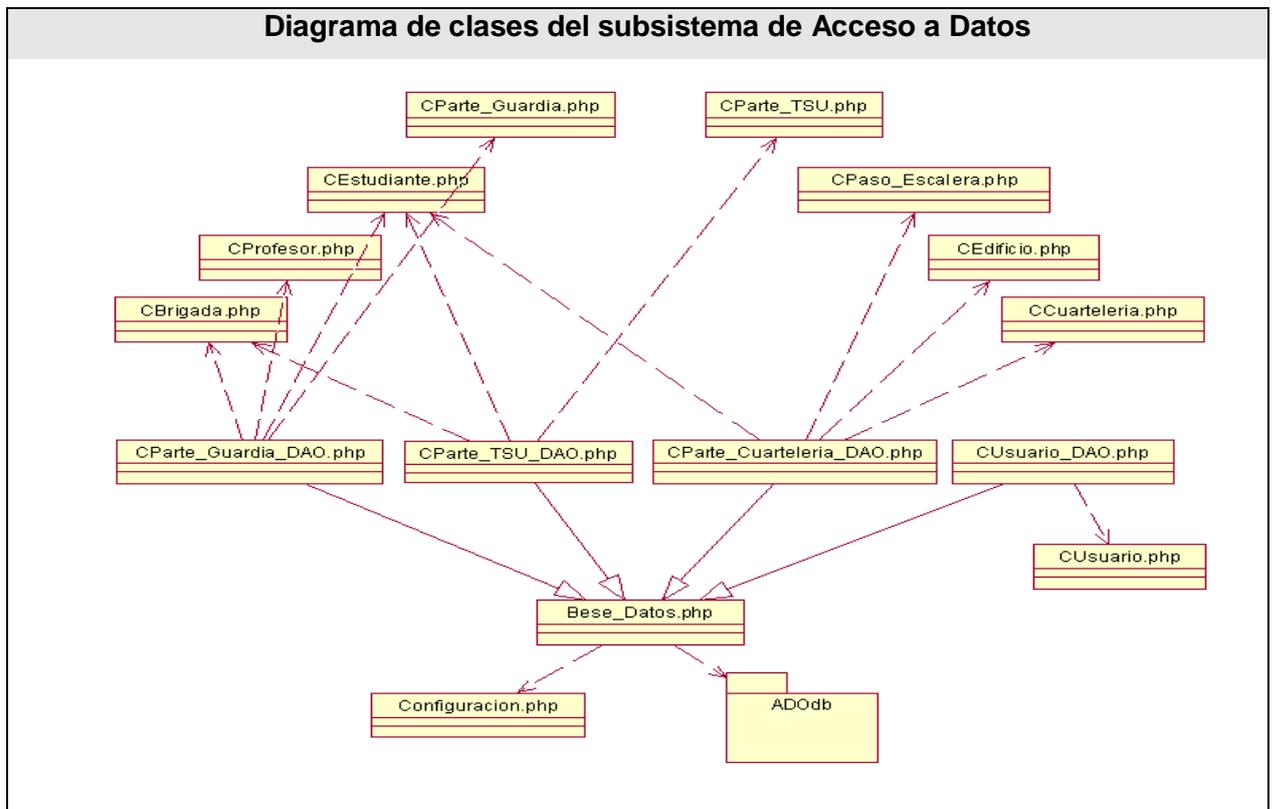


Figura 3.10 Diagrama de clases del subsistema de Acceso a Datos.

3.4 Descripción textual de las clases Web

La descripción textual de las clases del diseño ayudan a comprender la responsabilidad de cada una de estas dentro del sistema, con estas descripción se puede conocer qué funcionalidades específicas realiza cada clase, además de la información que cada una maneja.

3.4.1 Clase Controladora CManag_Parte_Guardia

Nombre	CManager_Parte_Guardia	
Tipo de Clase	Controladora	
Atributo	Tipo	
Responsabilidad		
Nombre	Listado_Brigadas()	
	Listado_Profesores()	
	Datos_JefeBrigada_ProfesorGuia(brigada)	
	Listado_Estudiantes_Brigada(brigada)	

	Listado_Postas()
	Existe_Parte_Guardia(fecha)
	Insertar_Parte_Guardia(fecha, Id_Brigada, Id_Profesor, Observaciones, Listado_Estudiantes, Listado_Turnos, Listado_Postas, Listado_Cumplimientos)
	Buscar_Guardia_Estudiantil_Fecha(fecha)
	Buscar_Guardia_Estudiantil_Brigada(brigada)
	Estudiantes_Guardia_Seleccionada(brigada, Id_guardia)
	Modificar_Guardia_Estudiantil(fecha, Id_Brigada, Id_Profesor, Observaciones, Listado_Estudiantes, Listado_Turnos, Listado_Postas, Listado_Cumplimientos)
	Eliminar_Guardia_Estudiantil(fecha)
	Eliminar_Guardia_Estudiantil(brigada)
Descripción	Controla todo lo relacionado con la adición de un nuevo parte de la guardia estudiantil, así como consultar, modificar y eliminar datos de los partes ya existentes.

Tabla 3.1 Descripción de la clase controladora "CManager_Parte_Guardia"

3.4.2 Clase Controladora CManag_Parte_Cuarteleria

Nombre	CManager_Parte_Cuarteleria	
Tipo de Clase	Controladora	
Atributo	Tipo	
Responsabilidad		
Nombre	Insertar_Cuarteleria(nombre, limpieza_areas, permanencia, control_visitantes, evaluación, fecha, observaciones)	
	Modificar_Cuarteleria(nombre, limpieza_areas, permanencia, control_visitantes, evaluación, fecha, observaciones)	
	Eliminar_Cuarteleria(fecha)	
	Buscar_Cuartelerias(paso_escalera)	
	Buscar_Cuartelerias(fecha)	
	Buscar_Cuartelerias(fecha1, fecha2)	
	Buscar_Cuartelerias(nombre)	

Descripción	Controla todo lo relacionado con la adición de un nuevo parte de la guardia estudiantil, modificar y eliminar, así como consultar datos de los partes ya existentes.
--------------------	--

Tabla 3.2 Descripción de la clase controladora “CManager_Parte_Cuarteleria”

3.4.3 Clase Controladora CManag_Parte_TSU

Nombre	CManager_Parte_TSU	
Tipo de Clase	Controladora	
Atributo	Tipo	
Responsabilidad		
Nombre	Listado_Brigadas()	
	Datos_JefeBrigada_ProfesorGuia(brigada)	
	Listado_Estudiantes_Brigada(brigada)	
	Existe_Parte_TSU(fecha)	
	Insertar_Parte_TSU(fecha, Id_Brigada, Nombre_Responsable, Observaciones, Listado_Estudiantes, Listado_Cumplimientos, Área)	
	Buscar_Parte_TSU(fecha)	
	Buscar_Parte_TSU(brigada)	
	Buscar_Estudiantes_Evaluados_Mal(fecha1, fecha2)	
	Estudiantes_TSU_Seleccionado(brigada, Id_TSU)	
	Modificar_Parte_TSU(fecha, Id_Brigada, Nombre_Responsable, Observaciones, Listado_Estudiantes, Listado_Cumplimientos, Área)	
	Eliminar_Parte_TSU(fecha, brigada)	
Descripción	Controla todo lo relacionado con la adición de un nuevo parte del TSU, así como consultar, modificar y eliminar datos de los partes ya existentes.	

Tabla 3.3 Descripción de la clase controladora “CManager_Parte_TSU”

3.4.4 Clase controladora CManager_Usuario

Nombre	CManager_Usuario	
Tipo de Clase	Controladora	
Atributo	Tipo	
Responsabilidad		

Nombre	Autenticar_Usuario(usuario, contraseña)
	Insertar_Usuario(nombre, rol, usuario, contraseña)
	Buscar_Usuario()
	Modificar_Usuario(rol, contraseña)
	Eliminar_Usuario(nombre)
Descripción	Controla lo referente a la autenticación de los usuarios del sistema, la inserción de un nuevo usuario, la modificación, eliminación y consulta de los ya existentes.

Tabla 3.4 Descripción de la clase controladora “CManager_Usuario”

3.4.5 Clase controladora CFachada

Nombre	CFachada	
Tipo de Clase	Interfaz (PHP, servidor)	
Atributo	Tipo	
gestion_tsu	CManager_Parte_TSU	
gestion_cuarteria	CManager_Parte_Cuarteria	
gestion_parte_guardia	CManager_Parte_Guardia	
gestion_usuario	CManager_Usuario	
Responsabilidad		
Nombre	CFachada()	
	Buscar_Parte_TSU(fecha)	
	Eliminar_Parte_TSU(brigada)	
	Eliminar_Parte_TSU(fecha)	
	Buscar_Estudiantes_Evaluados_Mal(fecha1, fecha2)	
	Buscar_Parte_TSU(brigada)	
	Insertar_Usuario(nombre, rol, usuario, contraseña)	
	Buscar_Usuario()	
	Modificar_Usuario(rol, contraseña)	
	Eliminar_Usuario(nombre)	
	Modificar_Parte_TSU(fecha, Id_Brigada, Nombre_Responsable, Observaciones, Listado_Estudiantes, Listado_Cumplimientos, Área)	
	Eliminar_Parte_TSU(fecha, brigada)	
	Insertar_Cuarteria(nombre, limpieza_areas, permanencia,	

	control_visitantes, evaluación, fecha, observaciones)
	Modificar_Cuarteleria(nombre, limpieza_areas, permanencia, control_visitantes, evaluación, fecha, observaciones)
	Eliminar_Cuarteleria(fecha)
	Buscar_Cuartelerias(paso_escalera)
	Buscar_Cuartelerias(fecha)
	Buscar_Cuartelerias(fecha1, fecha2)
	Buscar_Cuartelerias(nombre)
	Buscar_Parte_TSU(fecha)
	Buscar_Parte_TSU(brigada)
	Listado_Brigadas()
	Listado_Profesores()
	Datos_JefeBrigada_ProfesorGuia(brigada)
	Listado_Estudiantes_Brigada(brigada)
	Listado_Postas()
	Existe_Parte_Guardia(fecha)
	Insertar_Parte_Guardia(fecha, Id_Brigada, Id_Profesor, Observaciones, Listado_Estudiantes, Listado_Turnos, Listado_Postas, Listado_Cumplimientos)
	Buscar_Guardia_Estudiantil_Fecha(fecha)
	Buscar_Guardia_Estudiantil_Brigada(brigada)
	Estudiantes_Guardia_Seleccionada(brigada, Id_guardia)
	Autenticar_Usuario(usuario, contrasenna)
Descripción	Clase entidad que representa la tabla Trabajador en la base de datos

Tabla 3.4 Descripción de la clase controladora “CFachada”

3.4.6 Clase de acceso a dato Base_Datos

Nombre	Base_Datos	
Tipo de Clase	DAO	
Atributo	Tipo	
Servidor	String	
Usuario	String	
Password	String	

BaseDatos	String
Conexion	String
Error	String
Ado_DB	ADONewConnection
Responsabilidad	
Nombre	Base_Datos(s=SERVIDOR, u=USUARIO, p=PASSWORD, bd=BASEDATOS, mt_bd=MOTOR_BD)
	Abrir_Conexion ()
	Cerrar_Conexion ()
	Insertar (sql)
	Modificar (sql)
	Eliminar (sql)
	Ejecutar_Consulta(sql)
	Ejecutar_Insercion(sql)
	Cargar_Registro (sql)
	Empezar_Transaccion ()
	Terminar_Transaccion ()
	Ocurrio_Error_Transaccion ()
	Forzar_Transaccion ()
Descripción	Clase encargada de realizar la conexión a la base de datos y ejecutar las consultas, haciendo uso del paquete ADOdb.

Tabla 3.5 Descripción de la clase DAO "Base_Datos"

3.4.7 Clase de acceso a dato CParte_Guardia_DAO

Nombre	CParte_Guardia_DAO	
Tipo de Clase	DAO	
Atributo	Tipo	
Responsabilidad		
Nombre	Listado_Brigadas()	
	Listado_Profesores()	
	Datos_JefeBrigada_ProfesorGuia(BrigadaSeleccionada)	
	Listado_Estudiantes_Brigada(BrigadaSeleccionada)	

	Listado_Postas()
	Existe_Parte_Guardia(parte)
	Insertar_Parte_Guardia(Parte_Guardia, Brigada)
	Modificar_Parte_Guardia(Parte_Guardia, Brigada)
	Buscar_Guardia_Estudiantil_Fecha(parte)
	Buscar_Guardia_Estudiantil_Brigada(BrigadaSeleccionada)
	Eliminar_Parte_Guardia(parte)
	Estudiantes_Guardia_Seleccionada(BrigadaSeleccionada, Id_guardia)
Descripción	Clase encargada de realizar todas las consultas en la base de datos para el manejo de los partes de la guardia.

Tabla 3.6 Descripción de la clase DAO “CParte_Guardia_DAO”

3.4.8 Clase de acceso a dato CUsuario_DAO

Nombre	CUsuario_DAO	
Tipo de Clase	DAO	
Atributo	Tipo	
Responsabilidad		
	Autenticar_Usuario(user, contrasenna)	
	Insertar_Usuario(usuario)	
	Buscar_Usuario()	
	Modificar_Usuario(usuario)	
	Eliminar_Usuario(usuario)	
Descripción	Clase encargada de verificar en la base de datos si un usuario está autenticado o no, además de insertar un nuevo usuario, modificar y eliminar usuarios ya existentes.	

Tabla 3.7 Descripción de la clase DAO “CUsuario_DAO”

3.4.9 Clase entidad CProfesor

Nombre	CProfesor	
Tipo de Clase	Entidad	
Atributo	Tipo	
Identificador	String	
Responsabilidad		

Nombre	CProfesor (aIdentificador, aCumplimiento)
	Get_Identificador ()
Descripción	Clase entidad que representa la tabla Profesor en la base de datos

Tabla 3.8 Descripción de la clase entidad “CProfesor”

3.4.10 Clase entidad CEstudiante

Nombre	CEstudiante	
Tipo de Clase	Entidad	
Atributo	Tipo	
Identificador	String	
TurnoGuardia	int	
Posta	int	
Cumplimiento	String	
Responsabilidad		
Nombre	CEstudiante (aIdentificador, aTurnoGuardia, aPosta, aCumplimiento)	
	Get_Identificador ()	
	Get_TurnoGuardia ()	
	Get_Posta ()	
	Get_Cumplimiento ()	
Descripción	Clase entidad que representa la tabla Guardia_Estudiante en la base de datos	

Tabla 3.9 Descripción de la clase entidad “CEstudiante”

3.4.11 Clase entidad CBrigada

Nombre	CBrigada	
Tipo de Clase	Entidad	
Atributo	Tipo	
Identificador	String	
ListaEstudiantes	CEstudiante	
Responsabilidad		
Nombre	CBrigada (aIdentificador)	
	Get_Identificador ()	
	Add_Estudiante (CEstudiante)	

	Obtener_Estudiante (int)
	Get_ListadoEstudiantes()
Descripción	Clase entidad que contiene un listado con los estudiantes que pertenecen a una brigada.

Tabla 3.10 Descripción de la clase entidad “CBrigada”

3.4.12 Clase entidad CParteGuardia

Nombre	CParteGuardia	
Tipo de Clase	Entidad	
Atributo	Tipo	
Fecha	Date	
Brigada	CBrigada	
Profesor	CProfesor	
Observaciones	String	
Responsabilidad		
Nombre	CParteGuardia(aFecha, aBrigada, aProfesor, aObservaciones)	
	Get_Fecha ()	
	Get_Brigada ()	
	Get_Profesor ()	
	Get_Observaciones ()	
Descripción	Clase entidad que contiene toda la información referente a un parte de la guardia estudiantil.	

Tabla 3.11 Descripción de la clase entidad “CParteGuardia”

3.4.13 Clase entidad CParteTSU

Nombre	CParteTSU	
Tipo de Clase	Entidad	
Atributo	Tipo	
Fecha	Date	
Brigada	CBrigada	
Área	String	
Responsable_Area	String	
Observaciones	String	

Responsabilidad	
Nombre	CParteGuardia(aFecha, aBrigada, aProfesor, aObservaciones)
	Get_Fecha ()
	Get_Brigada ()
	Get_Area()
	Get_Responsable_Area()
	Get_Profesor ()
	Get_Observaciones ()
Descripción	Clase entidad que contiene toda la información referente a un parte del TSU.

Tabla 3.12 Descripción de la clase entidad "CParteTSU"

3.5 Diagrama de clases persistentes

Las clases persistentes son aquellas que cuentan con atributos o características particulares, la información de las misma debe perdurar en todo momento dentro de la base de datos, de no ser así se afecta el correcto funcionamiento del sistema.

A continuación se muestra el diagrama de clases persistentes del sistema:

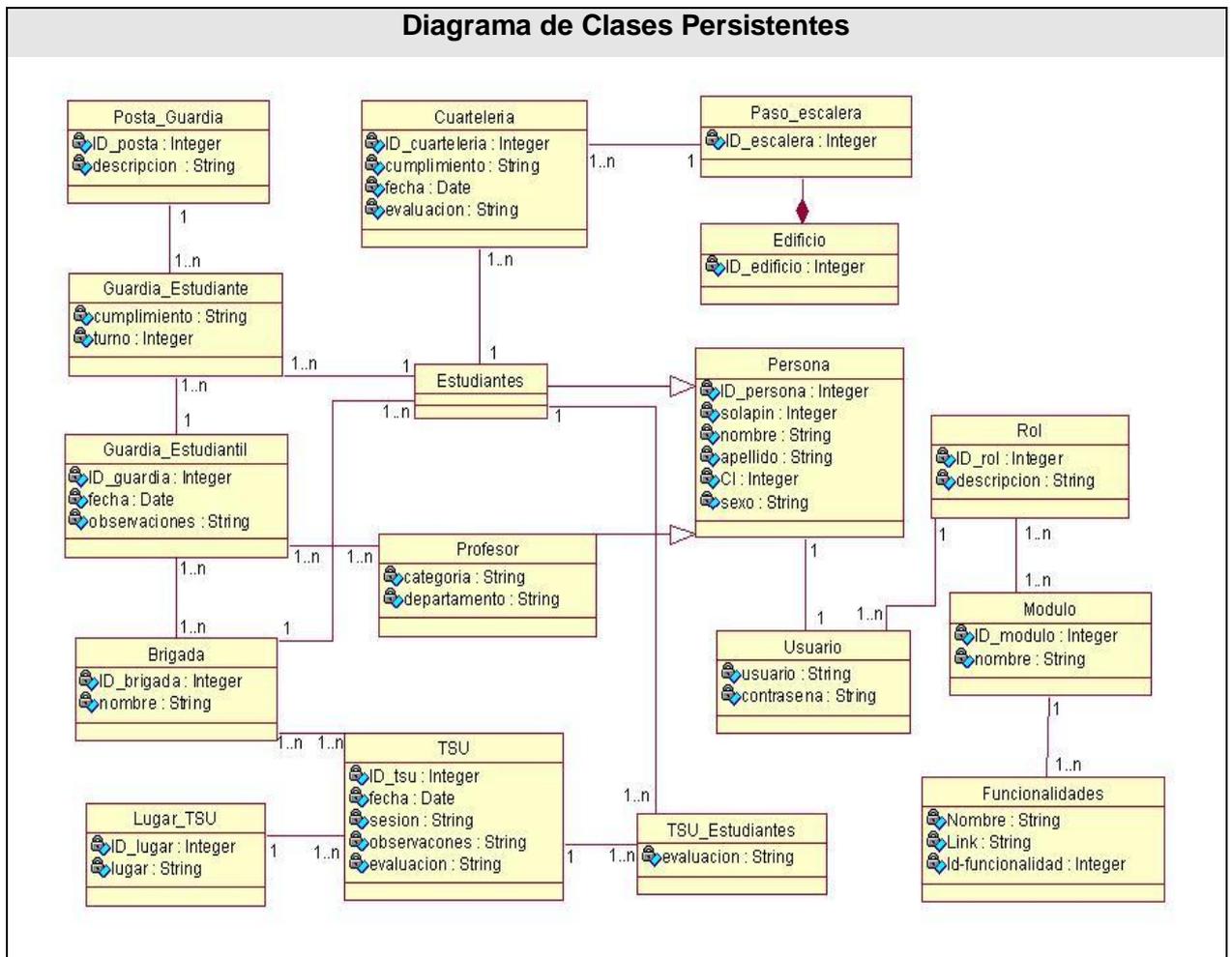


Figura 3.11 Diagrama de Clases Persistentes.

3.6 Principios de Diseño

La interfaz de un sistema, así como el formato de los reportes, la concepción de la ayuda y el tratamiento de excepciones juegan un papel fundamental en el éxito o fracaso del mismo.

A continuación se describen los principios de diseño seguidos para el desarrollo de dicho sistema.

3.6.1 Interfaz de Usuario

El diseño de la interfaz de usuario puede definirse como: “el conjunto de trabajos y pasos que seguirá el usuario, durante todo el tiempo que se relacione con el programa, detallando lo que verá y escuchará en cada momento, y las acciones que realizará, así como las respuestas que el sistema dará”. [4]

Para el diseño de la interfaz se tuvieron en cuenta los ocho principios básicos existentes, los que garantizan la usabilidad de los diseños de aplicaciones Web:

1. Proporcionar atajos a usuarios expertos.
2. Búsqueda estable.
3. Ofrecer información de retroalimentación.
4. Diseñar diálogos que conduzcan a una conclusión.
5. Permitir deshacer acciones fácilmente.
6. Prever errores y manejar errores simples.
7. Favorecer la sensación de control.
8. Reducir la carga a la memoria de corto plazo.

Por otra parte, el diseño de todo sistema, debe centrarse en el usuario que va a hacer uso del mismo, que es quien decide sobre el éxito o fracaso de éste, por lo que, además de los principios antes referidos, también se tuvo en cuenta lo siguiente:

- La utilización de un mismo formato y estilo en cada página.
- Proporcionarle al usuario una navegabilidad plena dentro de la aplicación.
- Evitar sobrecarga de colores e imágenes.
- Proporcionar un ambiente amigable.

El sistema utiliza, una hoja de estilo para guardar la configuración del diseño de todas las páginas, tanto para el tipo y tamaño de letra como para los botones, logrando uniformidad en todos estos elementos. Además, todas las páginas cuentan con el mismo menú en la parte izquierda garantizando con ello que se pueda acceder en todo momento a cada una de las opciones, lo que propicia una elevada flexibilidad en cuanto a la navegabilidad. Los formularios de entrada están centrados en una región específica concebida para ello dentro de cada página, además de estar organizados según la prioridad de los datos que se solicitan, ver figura 3.12.

Interfaz de Usuario para Adicionar Persona



SISTEMA DE GESTIÓN DE INFORMACIÓN DE LA FACULTAD 8

Bienvenido(a): nelson Lunes 11 de Mayo de 2009 Cerrar Sesión Ayuda

Gestionar Brigadas

Buscar Estudiante

Adicionar Brigada

Modificar Brigada

Eliminar Brigada

Gestionar Cuartelerías

Listar Cuartelerías

Adicionar Cuartelería

Gestionar Edificios

Adicionar Paso de Escalera

Adicionar Edificio

Eliminar Edificio

Actualizar Jefe de Apartamento

Actualizar Jefe de Paso de Escalera

Modificar Datos de un Apartamento

Actualizar Jefe de Edificio

Adicionar Apartamento

Gestionar Módulos

Adicionar Módulo

Editar Módulo

Adicionar Persona

Tipo de Persona

Seleccionar Tipo: *

Datos del Estudiante

Primer Nombre: *

Segundo Nombre:

Primer Apellido: *

Segundo Apellido: *

Carnet de Identidad: *

Sexo: *

Figura 3.12 Interfaz de usuario para Adicionar un Apartamento.

3.6.2 Formato de salida de los Reportes

Los reportes generados cuentan con un modelo estándar, estos son visualizados a través de un documento Excel generado por el sistema, donde la información que está contenida en los mismos se encuentra organizada, facilitando la búsqueda de algún dato específico, ver figura 3.13.

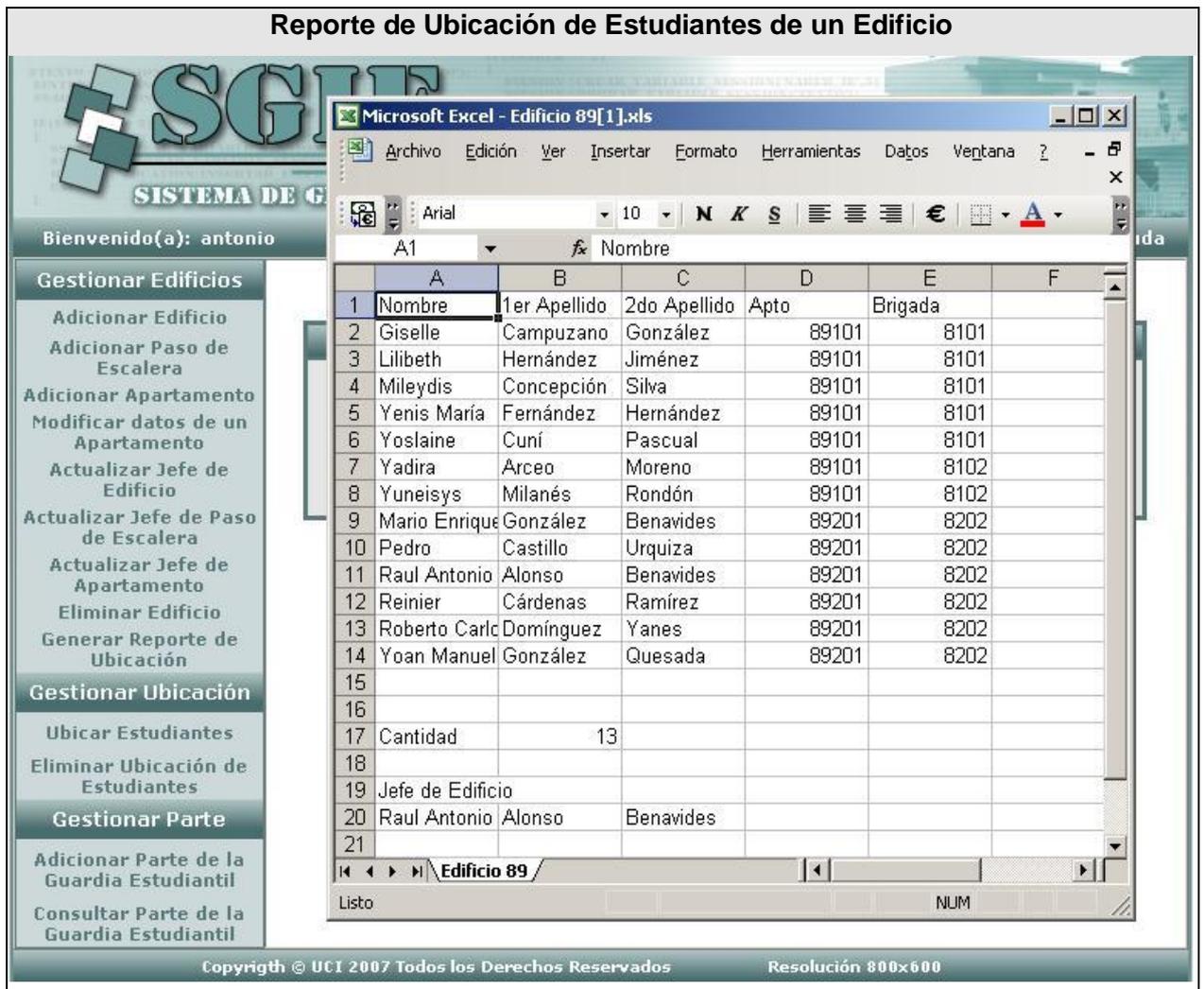


Figura 3.13 Reporte de Ubicación de Estudiantes de un Edificio.

3.6.3 Ayuda

La ayuda estará accesible siempre en cada una de las páginas de la aplicación y mostrará solo aquella información que es de interés para el usuario, según la página en la que se encuentre el mismo en ese momento. En cada página de ayuda se ofrece información sobre qué es lo que se debe hacer para realizar cada una de las acciones con las que el usuario esté interactuando en ese momento, así como una explicación de los posibles mensajes de error que puede darle el sistema ante cualquier acción que se ejecute, ver figura 3.14

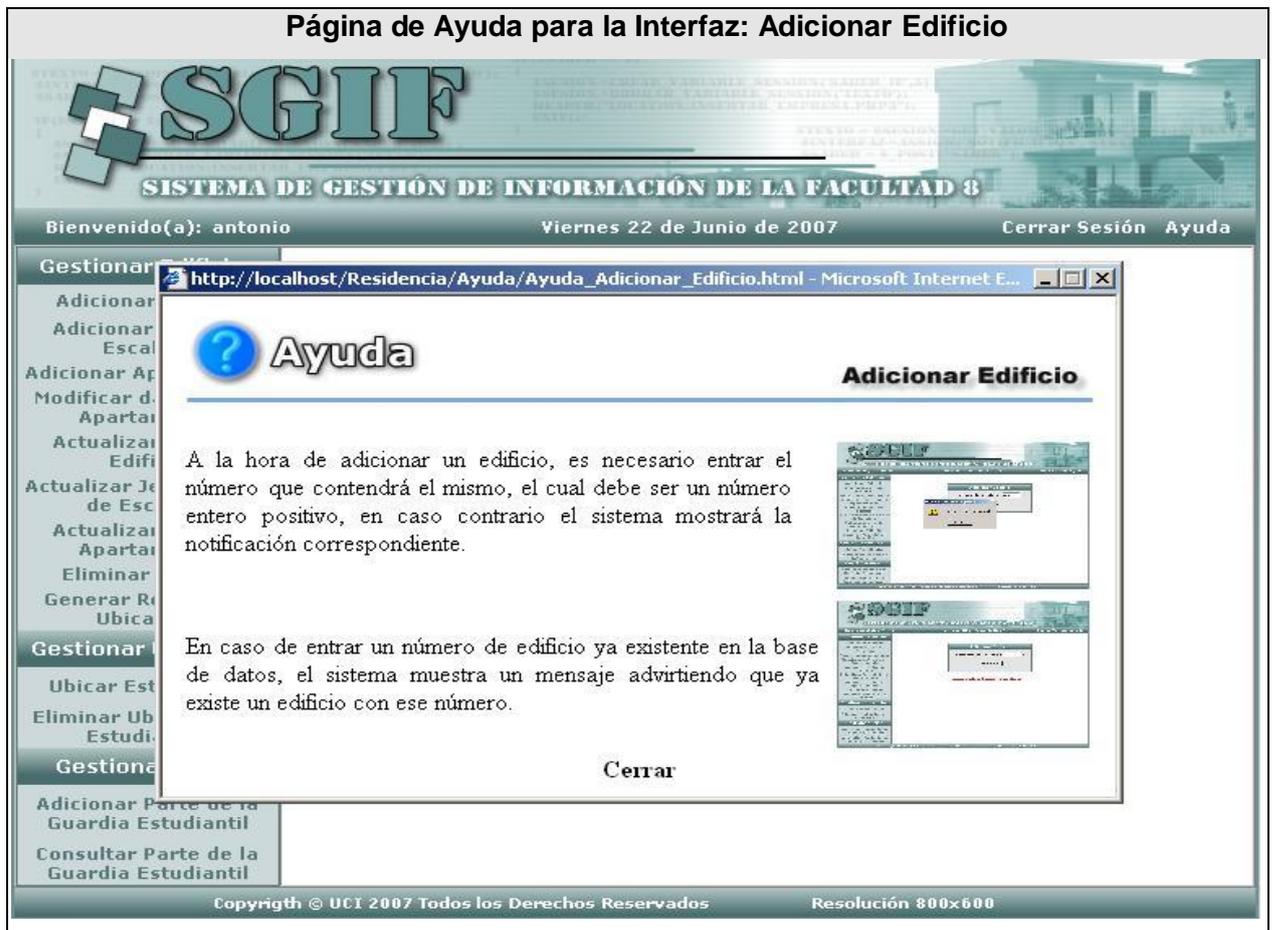


Figura 3.14 Página de ayuda para la interfaz: Adicionar Edificio.

3.6.4 Tratamiento de errores

El correcto tratamiento de los errores en un sistema, influye notablemente en su buen funcionamiento, garantizando la integridad de la información. Para lograr este último aspecto, fueron analizados todos los posibles errores que pudiera generar el sistema a partir de su interacción con el usuario, así como de problemas inesperados que pudieran surgir, como es el caso de la pérdida de conexión con la base de datos por falta del fluido eléctrico, etc. Muchos de estos posibles errores son tratados a partir de funciones JavaScript, y otros se obtienen a partir de la ejecución de la página en el servidor de la aplicación, estos últimos aparecen de forma textual en la misma página en la que se encuentra el usuario en ese momento. En cualquiera de los dos casos, los mensajes de error son mostrados con un texto claro, que señala de forma explícita y legible la respuesta del sistema ante cualquier acción que se ejecute. Además de esto, se muestran mensajes de confirmación ante acciones que son irreversibles,

como es el caso de la eliminación de datos, a la vez que se muestran mensajes para indicar cuándo una acción fue realizada con éxito, ver figura 3.15



Figura 3.15 Mensaje de error durante la acción de Adicionar Parte de la Guardia Estudiantil.

3.6.5 Seguridad

La seguridad del sistema, se basa principalmente en el empleo de la autenticación como una acción de obligatorio cumplimiento para el posterior uso del mismo, por lo que cada usuario del sistema tendrá un nombre para identificarse y una contraseña, los cuales serán verificados antes de darle acceso a las funcionalidades de sistema y si alguno de estos datos son incorrectos, se denegará dicho acceso. La contraseña pasa por un proceso de encriptación en el cliente, siguiendo el algoritmo MD5 y posteriormente se verifica en el servidor, si la misma ha sido encriptada o no, para en caso negativo encriptarla y manejarla de esta forma. Por otra parte, cada usuario posee un rol, a partir del cual el sistema construye un menú específico para cada uno,

garantizando con esto que el usuario tenga acceso sólo a las tareas que su rol especifica, evitando así que se tenga que chequear en cada una de las páginas si el usuario tiene acceso o no a ella. Como parte de la seguridad de la aplicación se a implementado la conexión por validación de dirección IP, cada usuario tiene una dirección IP especifica para conectarse a la aplicación, si el usuario intenta realizar esta acción desde otro IP diferente la conexión será denegada, sucede de igual forma si una PC con un IP no autorizado intenta acceder a la aplicación. También, son validados cada uno de los campos de los formularios con el objetivo de evitar que se introduzcan o seleccionen datos no permisibles por el sistema. Además, se prevén acciones tales como la pérdida de conexión con la base de datos, ver figura 3.16



Figura 3.16 Mensaje de error durante la acción de Autenticarse.

3.7 Conclusiones

En este capítulo se abordaron diferentes elementos que dan una idea general de cómo está construido el sistema, en términos de clases del análisis y del diseño. Se describieron textualmente las clases Web, lo que posibilita que se conozca la responsabilidad de cada una de ellas y qué funcionalidad específica realizan, además de contar con el diagrama de clases persistentes de la base de datos, éste cuenta con



Sistema de Gestión de Información de la Facultad 8.

la información física que se utilizó para el desarrollo de la aplicación. Por último fueron expuestos los principios de diseño seguidos durante la implementación del sistema, entre los que se encuentran: Interfaz de usuario, Formato de salida de los reportes, Tratamiento de errores, Ayuda y Seguridad.

Implementación y Pruebas

4.1 Introducción

Este capítulo describe cómo los elementos del modelo de diseño son implementados en términos de componentes y cómo los mismos se organizan de acuerdo a los nodos referidos en el modelo de despliegue. Se exponen las distintas pruebas realizadas a cada caso de uso, siguiendo particularmente, el método de pruebas de caja negra.

4.2 Modelo de Implementación

En el modelo de implementación se especifica cómo los elementos del diseño, específicamente las clases, se implementan en términos de componentes, como son: ejecutables, ficheros de código fuente, etc. También describe como están organizados los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje de programación utilizado, así como la dependencia existente entre estos componentes.

A continuación se muestran los diferentes diagramas de componentes, asociados al sistema en general y a los casos de usos más significativos, el resto de los mismos se pueden encontrar en el Anexo 5:

4.2.1 Diagrama de componentes de la Base de Datos

En el siguiente diagrama se muestra la relación del sistema completo como un componente y la base de datos.

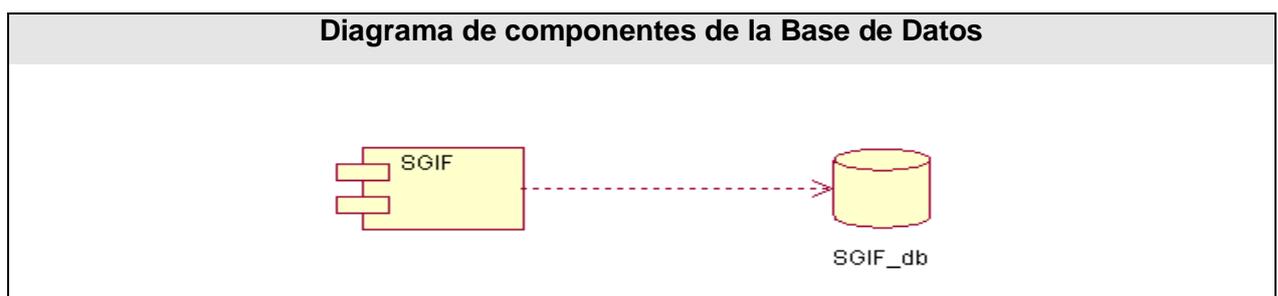


Figura 4.1 Diagrama de componentes de la base de datos

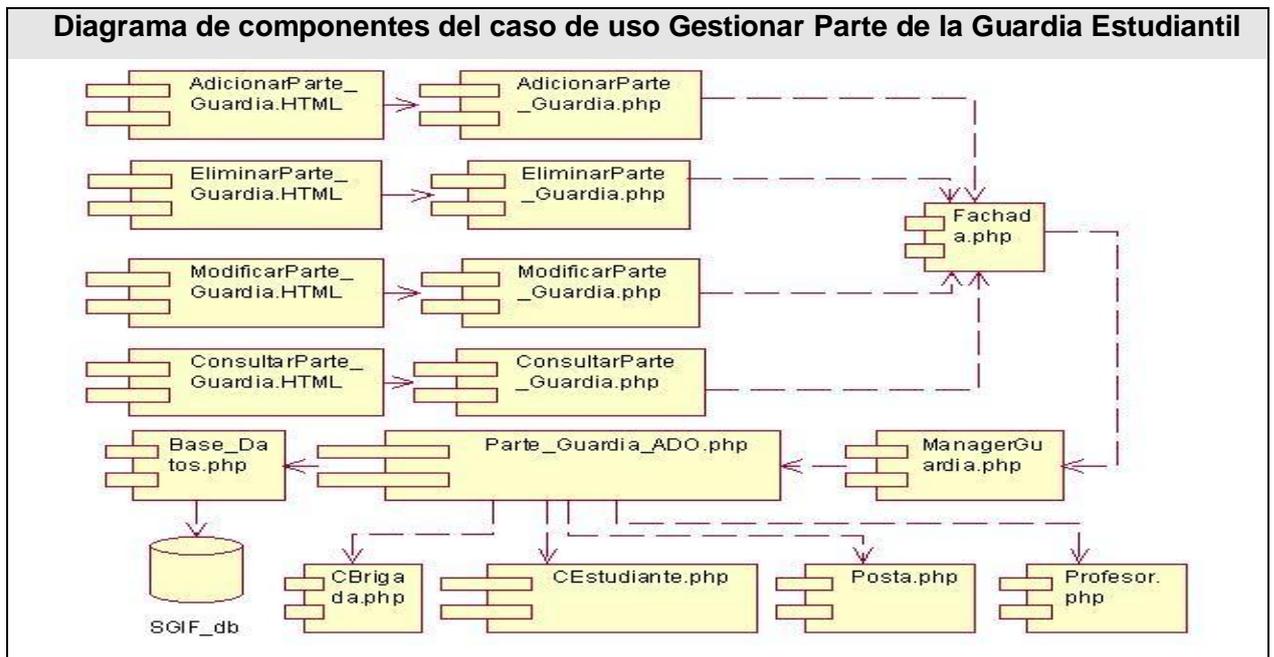


Figura 4.3 Diagrama de componentes del caso de uso Gestionar Parte de la Guardia Estudiantil.

4.2.4 Gestionar Cuarterería

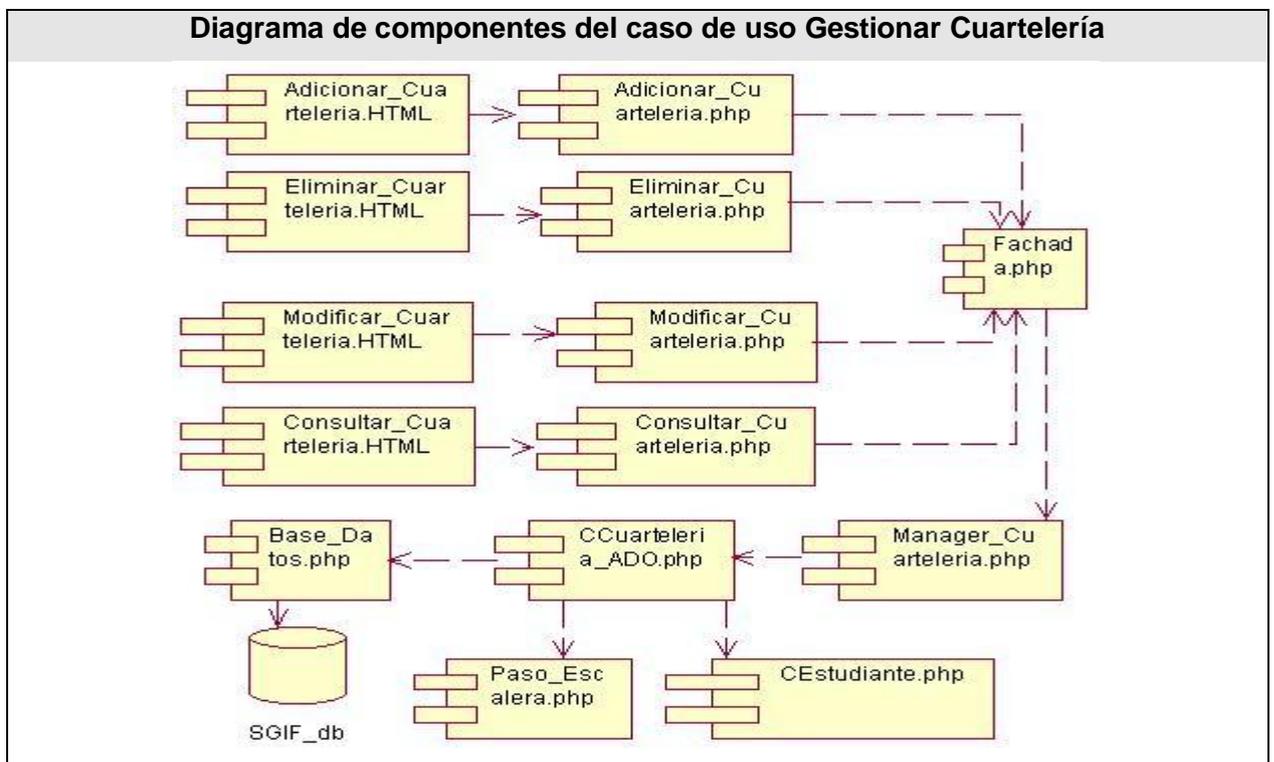


Figura 4.4 Diagrama de componentes del caso de uso Gestionar Cuarterería.

4.2.5 Gestionar TSU

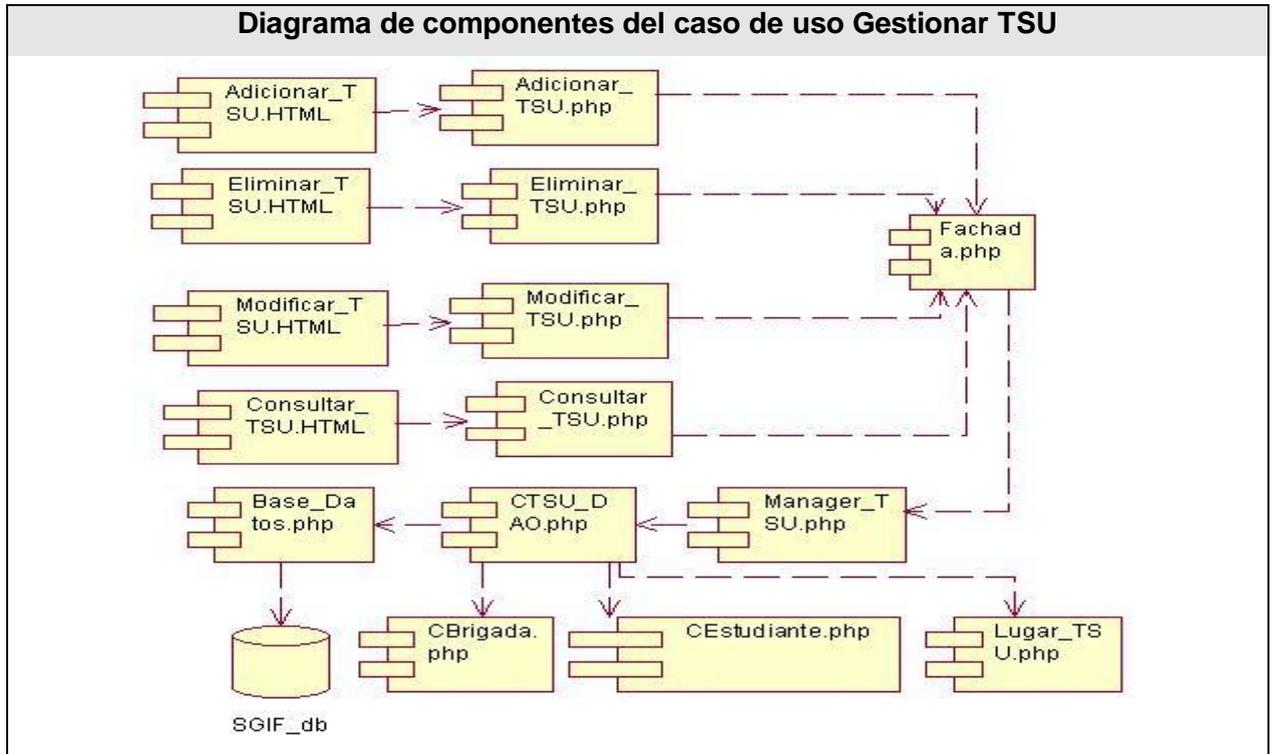


Figura 4.5 Diagrama de componentes del caso de uso Gestionar TSU.

4.2.6 Gestionar Persona

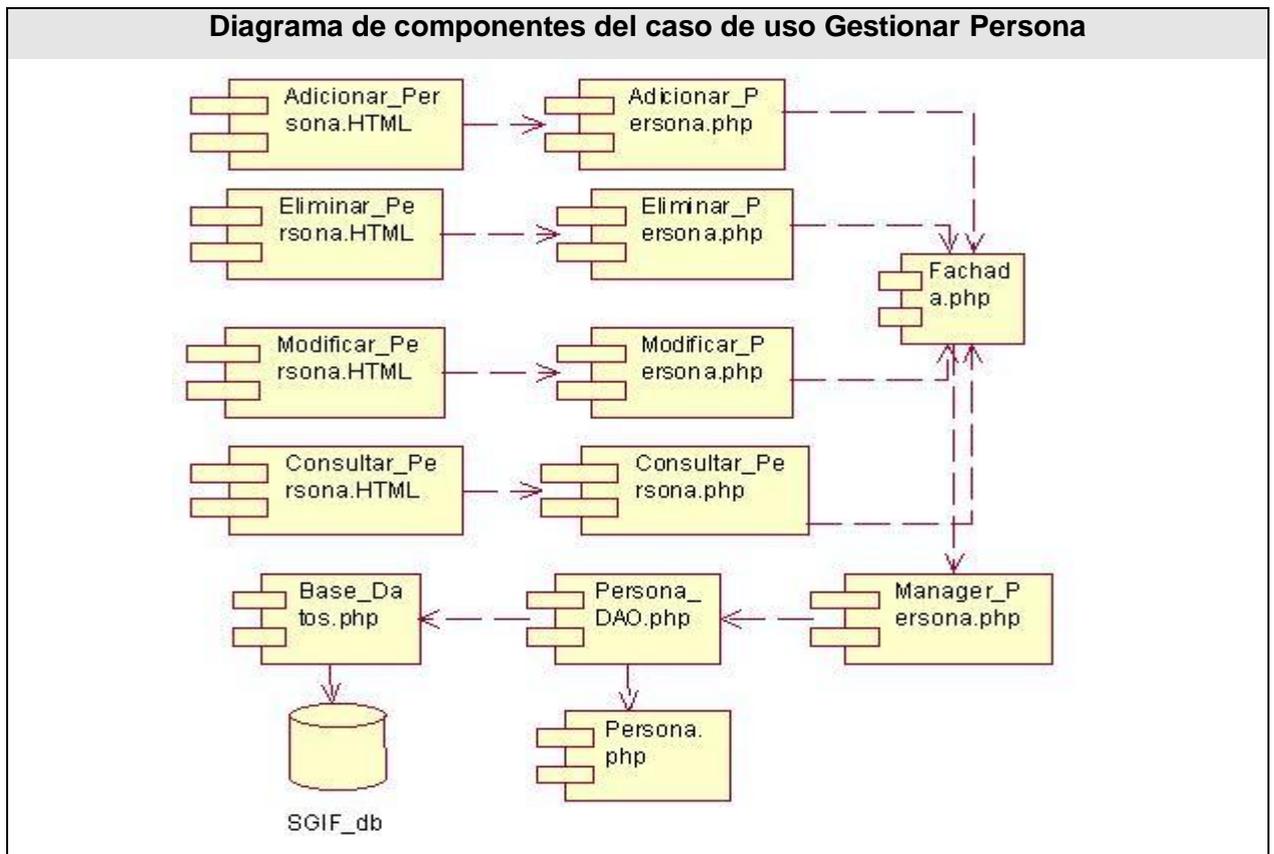


Figura 4.5 Diagrama de componentes del caso de uso Gestionar Persona.

4.3 Modelo de Despliegue

En un diagrama de despliegue se muestran las relaciones físicas entre los componentes de hardware y de software, específicamente la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes de software, que no son más, que los procesos que se ejecutan en ellos. Se puede decir también que, un diagrama de despliegue es un grafo de nodos, unidos por conexiones de comunicación, donde un nodo puede contener instancias de componentes. En general un nodo, se entiende como una unidad de computación de algún tipo como es el caso de impresoras o la misma computadora.

A continuación se muestra el diagrama de despliegue del sistema:

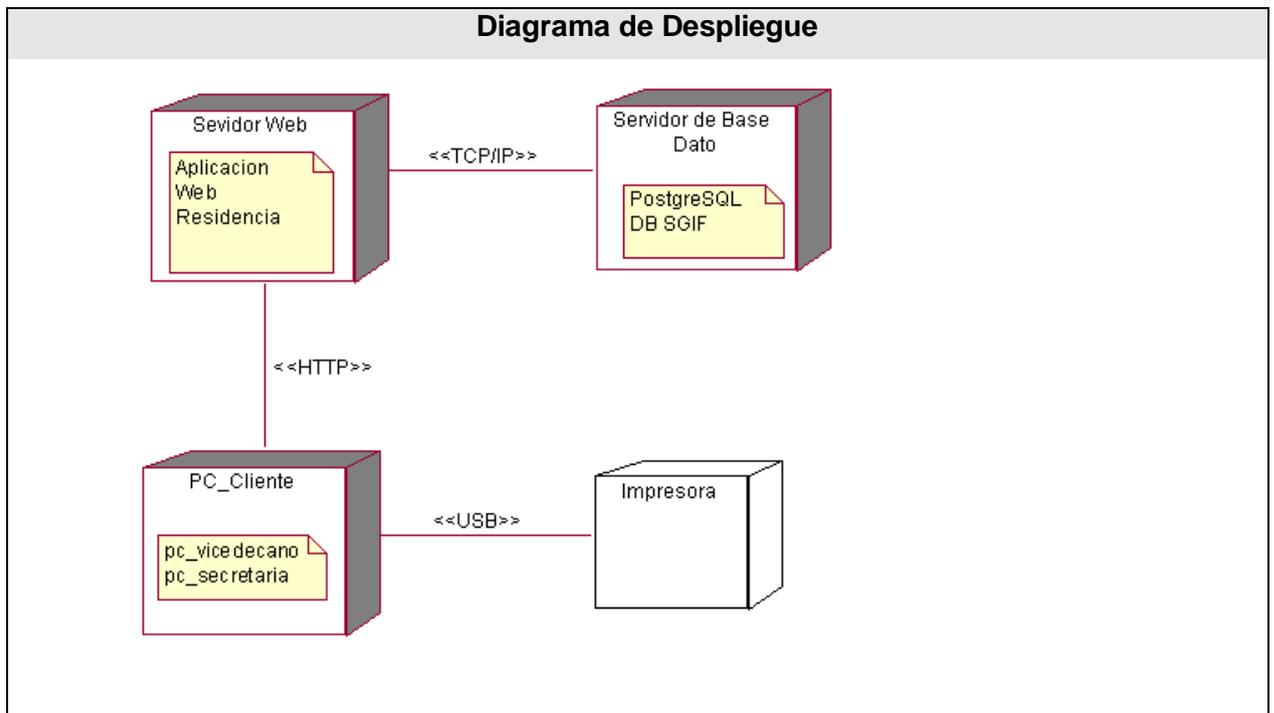


Figura 4.6 Diagrama de despliegue

4.4 Prueba

Las pruebas constituyen una actividad en la cual un sistema, es ejecutado bajo ciertas condiciones, en las que los resultados obtenidos son vistos y registrados, para la realización posterior de alguna evaluación de dicho sistema.

Entre los varios métodos de pruebas que existen, se encuentran las conocidas como **Pruebas de Caja Negra**, las que poseen las siguientes características:

- Verifican las especificaciones funcionales y no consideran la estructura interna del programa.
- No validan funciones ocultas, como es el caso de las funciones que son implementadas pero no descritas durante la fase de diseño, por lo que los errores asociados a ellas, no serán encontrados.
- Son realizadas sin el conocimiento interno del producto.

Además, las pruebas de caja negra son realizadas sobre la interfaz del sistema, controlando así tanto los datos de entrada como los de salida.

A continuación se presentan las pruebas realizadas al sistema por cada caso de uso, y siguiendo el método de caja negra descrito anteriormente.

4.4.1 Pruebas de Caja Negra para el Caso de uso “Gestionar Parte de la Guardia Estudiantil”

El caso de uso “Gestionar Parte de la guardia Estudiantil”, ha sido subdividido en dos escenarios. A continuación se presentan las pruebas de caja negra realizadas a cada uno de los mismos.

4.4.1.1 Escenario Modificar Parte de la Guardia Estudiantil

Condición de Entrada	Casos Válidos	Casos no Válidos
Selección de una Fecha.	Seleccionar una fecha.	No seleccionar ninguna fecha.
Selección de un Trabajador.	Seleccionar un trabajador.	No seleccionar ningún trabajador.
Selección de una brigada.	Seleccionar una brigada.	No seleccionar ninguna brigada.

Caso de Uso:	Gestionar Parte de la Guardia Estudiantil (Modificar parte de la guardia estudiantil).
Caso de Prueba:	Modificar parte de la guardia estudiantil habiendo seleccionado correctamente los datos necesarios para ello.
Entrada:	
El Vicedecano selecciona correctamente los datos necesarios para modificar un parte de la guardia estudiantil. Fecha de Realización: “12/10/2007” Trabajador de Guardia: “José Carlos Alfonso Roque” Brigada de Guardia: “8502”	
Resultado:	El sistema modifica el parte de la guardia estudiantil en la base de datos y muestra un mensaje notificando la acción realizada (“Proceso Satisfactorio”).
Condiciones:	Haber seleccionado correctamente los datos necesarios para ello

Caso de Uso:	Gestionar Parte de la Guardia Estudiantil (Modificar parte de la guardia estudiantil)
Caso de Prueba:	Modificar parte de la guardia estudiantil sin haber seleccionado correctamente alguno de los datos necesarios para ello.
Entrada:	
El Vicedecano no selecciona correctamente alguno de los datos necesarios para adicionar un	

parte de la guardia estudiantil. Fecha: "19/5/2007" Profesor de Guardia: "Fidel A. Curbelo Rosell" Brigada de Guardia: "Campo Omitido"	
Resultado:	El sistema muestra un mensaje de alerta notificando el error ("Debe seleccionar una Brigada").
Condiciones:	No haber seleccionado alguno de los datos necesarios para modificar un parte de la guardia estudiantil.

4.4.1.2 Escenario Eliminar Parte de la Guardia Estudiantil

Condición de Entrada	Casos Válidos	Casos no Válidos
Selección de una Fecha Inicial.	Seleccionar una fecha.	No seleccionar ninguna fecha.
Selección de una Fecha Final.	Seleccionar una fecha.	No seleccionar ninguna fecha.

Caso de Uso:	Gestionar Parte de la Guardia Estudiantil (Eliminar parte de la guardia estudiantil)
Caso de Prueba:	Eliminar parte de la guardia estudiantil habiendo seleccionado correctamente los datos necesarios para ello.
Entrada:	
El Vicedecano selecciona correctamente los datos necesarios para eliminar un parte de la guardia estudiantil. Fecha Inicial: "12/4/2009" Fecha Final: "10/05/2009"	
Resultado:	El sistema elimina los partes de la guardia estudiantil de la base de datos comprendidos entre las fechas especificadas, muestra un mensaje notificando la acción realizada ("Proceso Satisfactorio").
Condiciones:	Haber seleccionado correctamente los datos necesarios para ello

Caso de Uso:	Gestionar Parte de la Guardia Estudiantil (Eliminar parte de la guardia estudiantil)
Caso de Prueba:	Eliminar parte de la guardia estudiantil sin haber seleccionado

	correctamente alguno de los datos necesarios para ello.
Entrada:	
El Vicedecano no selecciona correctamente alguno de los datos necesarios para adicionar un parte de la guardia estudiantil. Fecha Inicial: "12/4/2009" Fecha Final: "Campo Omitido"	
Resultado:	El sistema muestra un mensaje de alerta notificando el error ("Debe seleccionar la fecha final").
Condiciones:	No haber seleccionado alguno de los datos necesarios para eliminar un parte de la guardia estudiantil.

4.4.2 Pruebas de Caja Negra para el Caso de uso "Gestionar Cuartelería"

El caso de uso "Gestionar Cuartelería", debido a su magnitud, ha sido subdividido en varios escenarios. A continuación se presentan las pruebas de caja negra realizadas al escenario "Adicionar Cuartelería", el resto de las mismas se pueden encontrar en el *Anexo*.

4.4.2.1 Escenario Adicionar Cuartelería

Condición de Entrada	Casos Válidos	Casos no Válidos
Selección de una Fecha.	Seleccionar una Fecha.	No seleccionar ninguna fecha.
Selección de un Edificio.	Seleccionar un Edificio.	No seleccionar ningún edificio.
Selección de un Paso de Escalera.	Seleccionar un Paso de Escalera.	No seleccionar ningún paso de escalera. Seleccionar un paso de escalera en el que en la fecha seleccionada con anterioridad ya exista una cuartelería.
Selección de un Estudiante.	Seleccionar un Estudiante.	No seleccionar ningún estudiante.
Selección de un valor para el cumplimiento.	Seleccionar un valor para el cumplimiento.	No seleccionar ningún valor para el cumplimiento.
Selección de un valor para la evaluación.	Seleccionar un valor para la evaluación.	No seleccionar ningún valor para la evaluación.

Caso de Uso:	Gestionar Cuartelería (Adicionar Cuartelería)
---------------------	---

Caso de Prueba:	Adicionar una nueva cuartería entrando correctamente los datos.
Entrada:	
<p>La secretaria introduce correctamente los datos necesarios para adicionar una cuartería.</p> <p>Fecha: "07/05/2009"</p> <p>Edificio: "141"</p> <p>Paso de Escalera: "2"</p> <p>Estudiante: "Nelson Alvarez Castañeda"</p> <p>Cumplimiento: "Cumplió"</p> <p>Evaluación: "Bien"</p>	
Resultado:	El sistema adiciona la cuartería a la base de datos y muestra un mensaje notificando la acción realizada ("Proceso Satisfactorio").
Condiciones:	Introducir los datos correctamente.

Caso de Uso:	Gestionar Cuartería (Adicionar Cuartería)
Caso de Prueba:	Adicionar una nueva cuartería entrando dato erróneo.
Entrada:	
<p>La secretaria introduce incorrectamente alguno de los datos necesarios para adicionar una cuartería.</p> <p>Fecha: "07/05/2009"</p> <p>Edificio: "141"</p> <p>Paso de Escalera: "2"</p> <p>Estudiante: "Nelson Alvarez Castañeda"</p> <p>Cumplimiento: "Campo Omitido"</p> <p>Evaluación: "Bien"</p>	
Resultado:	El sistema muestra un mensaje de alerta notificando el error ("Seleccione un valor para el campo "Cumplimiento"").
Condiciones:	Dato introducido incorrectamente.

4.4.2.2 Escenario Seleccionar Cuartería

Condición de Entrada	Casos Válidos	Casos no Válidos
Selección de una Fecha.	Seleccionar una Fecha.	No seleccionar ninguna fecha.

Caso de Uso:	Gestionar Cuartería (Seleccionar Cuartería)
Caso de Prueba:	Seleccionar cuarterías entrando correctamente los datos.

Entrada:	
La secretaria introduce correctamente el dato necesario para seleccionar una cuarterería. Fecha: "07/05/2009"	
Resultado:	El sistema muestra las cuartererías de la fecha seleccionada especificando Edificio y Paso de Escalera, además de un vínculo para ampliar los datos de cada una.
Condiciones:	Introducir los datos correctamente.

Caso de Uso:	Gestionar Cuarterería (Seleccionar Cuarterería)
Caso de Prueba:	Seleccionar cuartererías entrando los datos erróneos.
Entrada:	
La secretaria introduce incorrectamente el dato necesario para seleccionar una cuarterería. Fecha: " Campo Omitido "	
Resultado:	El sistema muestra un mensaje de alerta notificando el error ("Seleccione la fecha de la o las cuartererías a consultar").
Condiciones:	Dato introducido incorrectamente

4.4.2.3 Escenario Listar Cuartererías

Condición de Entrada	Casos Válidos	Casos no Válidos
Selección de Fecha Inicial.	Seleccionar una Fecha.	No seleccionar ninguna fecha.
Selección de Fecha Final.	Seleccionar una Fecha.	No seleccionar ninguna fecha.
Selección de un valor para la evaluación.	Seleccionar un valor para la evaluación.	No seleccionar un valor para la evaluación.

Caso de Uso:	Gestionar Cuarterería (Listar Cuartererías)
Caso de Prueba:	Listar cuartererías entrando correctamente los datos.
Entrada:	
La secretaria introduce correctamente el dato necesario para listar cuartererías. Desde: "01/05/2009" Hasta: "30/05/2009" Evaluación: "Mal"	
Resultado:	El sistema muestra las cuartererías comprendidas en el rango de fechas

	seleccionado, con la evaluación elegida, especificando Edificio y Paso de Escalera, además de un vínculo para ampliar los datos de cada una.
Condiciones:	Introducir los datos correctamente.

Caso de Uso:	Gestionar Cuartelería (Listar Cuartelerías)
Caso de Prueba:	Listar cuartelerías entrando los datos erróneos.
Entrada:	
La secretaria introduce incorrectamente los datos necesarios para listar cuartelerías. Desde: "01/05/2009" Hasta: "30/05/2009" Evaluación: "Campo Omitido"	
Resultado:	El sistema muestra un mensaje de alerta notificando el error ("Seleccione el tipo de evaluación de las cuartelerías a listar").
Condiciones:	Dato introducido incorrectamente

4.4.2.4 Escenario Modificar Cuartelería

Condición de Entrada	Casos Válidos	Casos no Válidos
Selección de una Fecha.	Seleccionar una Fecha.	No seleccionar ninguna fecha.
Selección de un Edificio.	Seleccionar un Edificio.	No seleccionar ningún edificio.
Selección de un Paso de Escalera.	Seleccionar un Paso de Escalera.	No seleccionar ningún paso de escalera. Seleccionar un paso de escalera en el que en la fecha seleccionada con anterioridad ya exista una cuartelería.
Selección de un Estudiante.	Seleccionar un Estudiante.	No seleccionar ningún estudiante.
Selección de un valor para el cumplimiento.	Seleccionar un valor para el cumplimiento.	No seleccionar ningún valor para el cumplimiento.
Selección de un valor para la evaluación.	Seleccionar un valor para la evaluación.	No seleccionar ningún valor para la evaluación.

Caso de Uso:	Gestionar Cuartelería (Modificar Cuartelería)
Caso de Prueba:	Modificar una cuartelería entrando correctamente los datos.

Entrada:	
La secretaria introduce correctamente los datos necesarios para modificar una cuarterería. Fecha: "07/05/2009" Edificio: "141" Paso de Escalera: "2" Estudiante: "Nelson Alvarez Castañeda" Cumplimiento: "Cumplió" Evaluación: "Bien"	
Resultado:	El sistema modifica la cuarterería en la base de datos y muestra un mensaje notificando la acción realizada ("Proceso Satisfactorio").
Condiciones:	Introducir los datos correctamente.

Caso de Uso:	Gestionar Cuarterería (Modificar Cuarterería)
Caso de Prueba:	Modificar una nueva cuarterería entrando dato erróneo.
Entrada:	
La secretaria introduce incorrectamente alguno de los datos necesarios para modificar una cuarterería. Fecha: "07/05/2009" Edificio: "141" Paso de Escalera: "2" Estudiante: "Nelson Alvarez Castañeda" Cumplimiento: "Incumplió " Evaluación: "Campo Omitido"	
Resultado:	El sistema muestra un mensaje de alerta notificando el error ("Seleccione un valor para el campo "Evaluación" ").
Condiciones:	Dato introducido incorrectamente.

4.4.3 Pruebas de Caja Negra para el Caso de uso "Gestionar TSU"

El caso de uso "Gestionar TSU", ha sido subdividido varios escenarios. A continuación se presentan las pruebas de caja negra para el escenario Modificar TSU, el resto se encuentra en el Anexo.

4.4.3.1 Escenario Adicionar TSU

Condición de Entrada	Casos Válidos	Casos no Válidos
----------------------	---------------	------------------

Selección de una Fecha.	Seleccionar una fecha	No seleccionar ninguna fecha.
Selección de una Sesión.	Seleccionar una sesión.	No seleccionar ninguna sesión.
Selección de una brigada	Seleccionar una brigada	No seleccionar ninguna brigada.
Selección de un Lugar.	Seleccionar un lugar.	No seleccionar un lugar.
Selección de una Evaluación.	Seleccionar una evaluación.	No seleccionar una evaluación.

Caso de Uso:	Gestionar TSU (Adicionar TSU)
Caso de Prueba:	Adicionar TSU habiendo seleccionado correctamente los datos necesarios para ello.
Entrada:	
El Vicedecano selecciona correctamente los datos necesarios para adicionar TSU. Fecha de Realización: "12/10/2007" Sesión: "Mañana" Brigada de Guardia: "8504" Lugar: "Docente 5" Evaluación: "Bien"	
Resultado:	El sistema adiciona el TSU en la base de datos y muestra un mensaje notificando la acción realizada ("Proceso Satisfactorio").
Condiciones:	Haber seleccionado correctamente los datos necesarios para ello.

Caso de Uso:	Gestionar TSU (Adicionar TSU)
Caso de Prueba:	Adicionar TSU sin haber seleccionado correctamente alguno de los datos necesarios para ello.
Entrada:	
El Vicedecano no selecciona correctamente alguno de los datos necesarios para adicionar TSU. Fecha de Realización: "12/10/2007" Sesión: "Campo Omitido" Brigada de Guardia: "8504" Lugar: "Docente 5" Evaluación: "Bien"	
Resultado:	El sistema muestra un mensaje de alerta notificando el error ("Debe

	seleccionar la sesión en la que se realizó el TSU”).
Condiciones:	No haber seleccionado alguno de los datos necesarios para adicionar un TSU.

4.4.3.2 Escenario Buscar TSU

Condición de Entrada	Casos Válidos	Casos no Válidos
Selección de un Criterio.	Seleccionar un criterio.	No seleccionar ningún criterio.
Selección de una Brigada.	Seleccionar una brigada.	No seleccionar ninguna brigada.

Caso de Uso:	Gestionar TSU (Buscar TSU)
Caso de Prueba:	Buscar TSU habiendo seleccionado correctamente los datos necesarios para ello.
Entrada:	
El Vicedecano selecciona correctamente los datos necesarios para buscar TSU. Criterio: “Brigada” Brigada: “8504”	
Resultado:	El sistema consulta la base de datos y muestra todos los TSU realizados por la brigada, especificando de cada uno: Brigada, Fecha y un vínculo para ampliar los datos.
Condiciones:	Haber seleccionado correctamente los datos necesarios para ello.

Caso de Uso:	Gestionar TSU (Buscar TSU)
Caso de Prueba:	Buscar TSU sin haber seleccionado correctamente alguno de los datos necesarios para ello.
Entrada:	
El Vicedecano no selecciona correctamente alguno de los datos necesarios para buscar TSU. Criterio: “Brigada” Brigada: “ Campo Omitido ”	
Resultado:	El sistema muestra un mensaje de alerta notificando el error (“Debe seleccionar la brigada a la que corresponde el TSU”).
Condiciones:	No haber seleccionado alguno de los datos necesarios para buscar un TSU.

4.4.3.3 Escenario Modificar TSU

Condición de Entrada	Casos Válidos	Casos no Válidos
Selección de una Fecha.	Seleccionar una fecha	No seleccionar ninguna fecha.
Selección de una Sesión.	Seleccionar una sesión.	No seleccionar ninguna sesión.
Selección de una brigada	Seleccionar una brigada	No seleccionar ninguna brigada.
Selección de un Lugar.	Seleccionar un lugar.	No seleccionar un lugar.
Selección de una Evaluación.	Seleccionar una evaluación.	No seleccionar una evaluación.

Caso de Uso:	Gestionar TSU (Modificar TSU)
Caso de Prueba:	Modificar TSU habiendo seleccionado correctamente los datos necesarios para ello.
Entrada:	
El Vicedecano selecciona correctamente los datos necesarios para modificar TSU. Fecha de Realización: "12/10/2007" Sesión: "Mañana" Brigada de Guardia: "8504" Lugar: "Docente 5" Evaluación: "Bien"	
Resultado:	El sistema modifica los datos del TSU especificado en la base de datos y muestra un mensaje notificando la acción realizada ("Proceso Satisfactorio").
Condiciones:	Haber seleccionado correctamente los datos necesarios para ello.

Caso de Uso:	Gestionar TSU (Modificar TSU)
Caso de Prueba:	Modificar TSU sin haber seleccionado correctamente alguno de los datos necesarios para ello.
Entrada:	
El Vicedecano no selecciona correctamente alguno de los datos necesarios para adicionar TSU. Fecha de Realización: "12/10/2007" Sesión: "Mañana" Brigada de Guardia: "8504" Lugar: "Campo Omitido" Evaluación: "Bien"	

Resultado:	El sistema muestra un mensaje de alerta notificando el error (“Debe seleccionar el lugar en el que se realizó el TSU”).
Condiciones:	No haber seleccionado alguno de los datos necesarios para adicionar un TSU.

4.4.5 Pruebas de Caja Negra para el Caso de uso “Gestionar Persona”

El caso de uso “Gestionar Persona”, ha sido subdividido en varios escenarios. A continuación se presentan las pruebas de caja negra realizadas a cada uno de los mismos.

4.4.5.1 Escenario Adicionar Persona

Condición de Entrada	Casos Válidos	Casos no Válidos
Tipo de Persona.	Seleccionar un tipo de persona.	No seleccionar ninguno de los tipos de personas.
pnombre.	Cadena de caracteres de solo letras.	No entrar nada. Entrar una cadena que no contenga solo letras.
snombre.	Cadena de caracteres de solo letras.	No entrar nada. Entrar una cadena que no contenga solo letras.
papellido.	Cadena de caracteres de solo letras.	No entrar nada. Entrar una cadena que no contenga solo letras.
sapellido.	Cadena de caracteres de solo letras.	No entrar nada. Entrar una cadena que no contenga solo letras.
Cl.	Número real de 11 dígitos.	No entrar nada. Un número menor o mayor de 11 dígitos.
Solapin.	Cadena de caracteres entre 4 y 7 caracteres, compuesta por números y letras.	No entrar nada. Entrar una cadena mayor de 7 o menor que 4 caracteres.

Selección del Sexo.	Seleccionar el sexo.	No seleccionar el sexo.
Selección de la Brigada.	Seleccionar la Brigada.	No seleccionar la brigada.

Caso de Uso:	Gestionar Persona (Adicionar Persona).
Caso de Prueba:	Adicionar Persona habiendo seleccionado correctamente los datos necesarios para ello.
Entrada:	
<p>El Vicedecano selecciona correctamente los datos necesarios para adicionar persona.</p> <p>Tipo de Persona: "Estudiante"</p> <p>pnombre: "Nelson"</p> <p>snombre: " "</p> <p>papellido: "Alvarez"</p> <p>sapellido: "Castañeda"</p> <p>CI: "85040718167"</p> <p>Solapin: "59549"</p> <p>Sexo: "M"</p> <p>Brigada: "8408"</p>	
Resultado:	El sistema adiciona la persona a la base de datos y muestra un mensaje notificando la acción realizada ("Proceso Satisfactorio").
Condiciones:	Haber seleccionado correctamente los datos necesarios para ello

Caso de Uso:	Gestionar Persona (Adicionar Persona)
Caso de Prueba:	Adicionar persona sin haber seleccionado correctamente alguno de los datos necesarios para ello.
Entrada:	
<p>El Vicedecano no selecciona correctamente alguno de los datos necesarios para adicionar persona.</p> <p>Tipo de Persona: "Estudiante"</p> <p>pnombre: "Nelson"</p> <p>snombre: " "</p> <p>papellido: "Alvarez"</p> <p>sapellido: "Castañeda"</p> <p>CI: "85040718167"</p>	

Solapín: "59549"	
Sexo: "M"	
Brigada: "Campo Omitido"	
Resultado:	El sistema muestra un mensaje de alerta notificando el error ("Brigada incorrecta").
Condiciones:	No haber seleccionado alguno de los datos necesarios para adicionar persona.

4.4.5.2 Escenario Modificar Persona

Condición de Entrada	Casos Válidos	Casos no Válidos
Tipo de Persona.	Seleccionar un tipo de persona.	No seleccionar ninguno de los tipos de personas.
Selección de una Persona.	Seleccionar una Persona.	No seleccionar una persona.
pnombre.	Cadena de caracteres de solo letras.	No entrar nada. Entrar una cadena que no contenga solo letras.
snombre.	Cadena de caracteres de solo letras.	No entrar nada. Entrar una cadena que no contenga solo letras.
papellido.	Cadena de caracteres de solo letras.	No entrar nada. Entrar una cadena que no contenga solo letras.
sapellido.	Cadena de caracteres de solo letras.	No entrar nada. Entrar una cadena que no contenga solo letras.
Cl.	Número real de 11 dígitos.	No entrar nada. Un número menor o mayor de 11 dígitos.
Solapín.	Cadena de caracteres entre 4 y 7 caracteres, compuesta por números y letras.	No entrar nada. Entrar una cadena mayor de 7 o menor que 4 caracteres.

Selección del Sexo.	Seleccionar el sexo.	No seleccionar el sexo.
Selección de la Brigada.	Seleccionar la Brigada.	No seleccionar la brigada.

Caso de Uso:	Gestionar Persona (Modificar Persona).
Caso de Prueba:	Modificar Persona habiendo seleccionado correctamente los datos necesarios para ello.
Entrada:	
<p>El Vicedecano selecciona correctamente los datos necesarios para modificar persona.</p> <p>Tipo de Persona: "Estudiante"</p> <p>Persona: "Nelson Alvarez Castañeda"</p> <p>pnombre: "Nelson"</p> <p>snombre: " "</p> <p>papellido: "Alvarez"</p> <p>sapellido: "Castañeda"</p> <p>CI: "85040718167"</p> <p>Solapin: "59549"</p> <p>Sexo: "M"</p> <p>Brigada: "8504"</p>	
Resultado:	El sistema modifica los datos de la persona en la base de datos y muestra un mensaje notificando la acción realizada ("Proceso Satisfactorio").
Condiciones:	Haber seleccionado correctamente los datos necesarios para ello

Caso de Uso:	Gestionar Persona (Modificar Persona)
Caso de Prueba:	Modificar persona sin haber seleccionado correctamente alguno de los datos necesarios para ello.
Entrada:	
<p>El Vicedecano no selecciona correctamente alguno de los datos necesarios para modificar persona.</p> <p>Tipo de Persona: "Estudiante"</p> <p>pnombre: "Nelson"</p> <p>snombre: " "</p> <p>papellido: "Alvarez"</p> <p>sapellido: "Castañeda"</p>	

CI: "85040718167" Solapín: "59549" Sexo: "Campo Omitido" Brigada: "8504"	
Resultado:	El sistema muestra un mensaje de alerta notificando el error ("Sexo incorrecta").
Condiciones:	No haber seleccionado alguno de los datos necesarios para modificar persona.

4.5 Conclusiones

En el presente capítulo se presentado cómo está constituido el sistema a partir de los diagramas de componentes de los principales caso de uso, el diagramas de componentes de todo el sistema en general, se mostrado el diagrama de despliegue, el que ilustra los nodos que serán usados para darle uso a la aplicación y por último fueron especificadas un conjunto de pruebas que se le realizaron al sistema, donde se especifica qué parámetros fueron utilizados durante la realización de las mismas, además del resultado obtenido a partir de la ejecución de cada una de ellas.

Conclusiones

Durante el desarrollo de este trabajo se le dio cumplimiento a cada uno de los objetivos trazados para el mismo, logrando con esto, el análisis, diseño e implementación de un sistema del cual se espera mejore en gran medida alguno de los procesos que tienen lugar en el Vicedecanato de Extensión y Residencia Estudiantil de la Facultad 8 perteneciente a la Universidad de las Ciencias Informáticas (UCI), tal es el caso de, mantener un control sobre los partes de la guardia estudiantil, cuarterías, TSU y el personal en general perteneciente a dicha facultad, garantizando de esta forma la disponibilidad e integridad de la información en todo momento. El sistema está acorde tanto a los requisitos no funcionales que soportan el sistema como a los funcionales que determinan cada una de las tareas que pueden ser gestionadas mediante la aplicación.

Recomendaciones

Luego de haber cumplido con los objetivos propuestos y dado el alcance del trabajo realizado, se recomienda:

- Continuar con el estudio de los procesos que tienen lugar en el Vicedecanato de Extensión y Residencia Estudiantil de la Facultad 8 que aún no han sido automatizadas, con el objetivo de integrar las mismas a este sistema en futuras versiones.
- Agregar nuevas funcionalidades con el objetivo de que el usuario pueda obtener otros tipos de reportes o resúmenes de información.

Referencias Bibliográficas

1. Anónimo. *Lenguaje de programación*. 2009 [cited febrero 2009]; Available from: http://es.wikipedia.org/wiki/Lenguaje_de_programacion#Concepto
2. Anónimo. *PHP*. 2009 [cited febrero 2009]; Available from: http://es.wikipedia.org/wiki/PHP#Visi.C3.B3n_general
3. Anónimo. *Características de Java*. 2006 [cited febrero 2009]; Available from: http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto/I_3.htm.
4. Anónimo. *Servidor HTTP Apache* 2009 [cited febrero 2009]; Available from: <http://es.wikipedia.org/wiki/apache>
5. *OMG Unified Modeling Language Specification*. 2003, OMG.
6. Schuller, J., *Aprendiendo UML en 24 horas*: Prentice Hall.
7. Anónimo. *Metodologías de Desarrollo de Software*. 2009 [cited febrero 2009]; Available from: http://es.wikipedia.org/wiki/Programaci%C3%B3n_Extrema
8. Anónimo. *Comparación de Herramientas de Modelado UML: Enterprise Architect y Rational Rose*. 2006 [cited febrero 2009]; Available from: <http://www.apexnet.com.ar/index.php/news/main/38/event=view>.
9. Anónimo. *Rational Rose: Procedimientos básicos para desarrollar un proyecto con UML*. 2006 [cited febrero 2007]; Available from: <http://www.vico.org/TallerRationalRose.pdf>.
10. Anónimo. *Visual Paradigm* 2007 [cited febrero 2009]; Available from: [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/)
11. Anónimo. *NuSphere PHPEd*. 2005 [cited febrero 2009]; Available from: <http://nusphere-phped.softonic.com>.
12. Anónimo. *Editores para PHP, Zend Studio*. 2007 [cited febrero 2009]; Available from: <http://www.adrformacion.com/cursos/php/leccion1/tutorial3.html>.
13. Anónimo. *Smarty*. 2006 [cited febrero 2009]; Available from: <http://es.wikipedia.org/wiki/Smarty>.
14. Jacobson, I., G. Booch, and J. Rumbaugh, *El Proceso Unificado de Desarrollo de software*. 2000: Addison-Wesley.

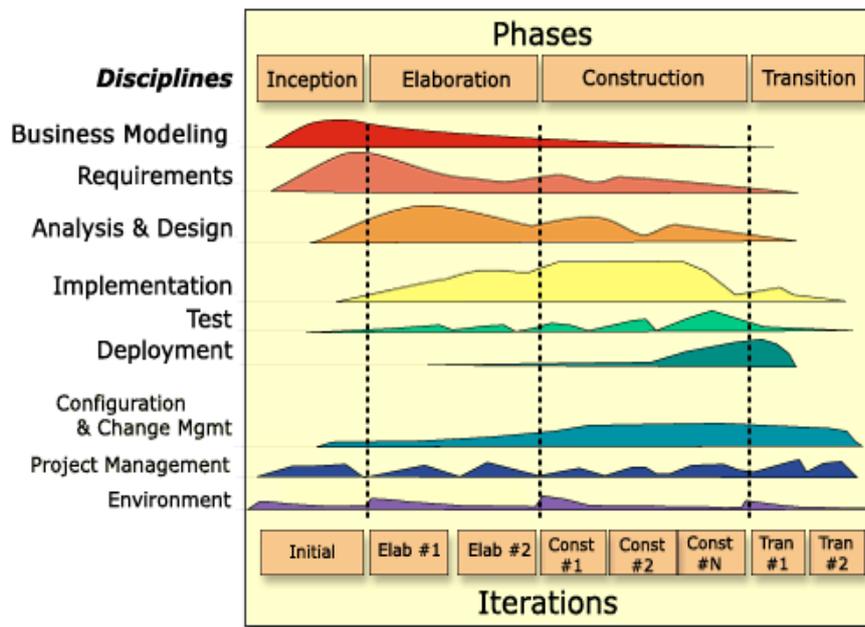
Glosario de Términos y Síglas

- **UCI**: *Universidad de las Ciencias Informáticas.*
- **TSU**: *Trabajo Socialmente Útil.*
- **ASP**: *Active Server Pages.* Es una tecnología del lado servidor de Microsoft para páginas Web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS).
- **CASE**: *Computer Aided Software Engineering.*
- **CUN**: *Caso de uso del negocio.*
- **CUS**: *Caso de uso del sistema.*
- **HTML**: *Hypertext Markup Language.* Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.
- **HTTP**: *HyperText Transfer Protocol.* Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.
- **Herramientas CASE**: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.
- **Hardware**: Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.
- **Internet**: Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.
- **Java**: Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los 90.
- **Microsoft**: Compañía que manufactura los sistemas de operación DOS y Windows.

- **MySQL**: Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.
- **PHP**: *PHP: Hypertext Preprocessor*. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.
- **PostgreSQL**: es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.
- **RUP**: *Rational Unified Process* (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.
- **Samrty**: Es un motor de plantillas para PHP, cuyo objetivo es separar el contenido de la presentación en una página Web.
- **Software**: Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.
- **SQL**: *Structured Query Language*. Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos.
- **SGBD**: *Sistema de Gestión de Bases de Datos*. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.
- **UML**: *Unified Modeling Language*. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.
- **XML**: *Extensible Markup Language*. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Orientado principalmente al almacenamiento, procesamiento y transmisión de mensajes.

Anexos

Anexo 1: Fases y Flujos de trabajos de RUP



Fases

- Inicio
- Elaboración
- Construcción.
- Transición.

Flujos de trabajo

- Modelamiento del negocio.
- Requerimientos.
- Análisis y diseño.
- Implementación.
- Prueba (Testeo).
- Instalación.

- Administración del proyecto.
- Administración de configuración y cambios.
- Ambiente.

Anexo 2: Descripción del resto de los casos de uso del sistema.

Descripción del CUS “Gestionar Usuario”

Caso de Uso	Gestionar Usuario
Actores	Vicedecano de Extensión y Residencia (inicia).
Propósito	Permite al Vicedecano de Extensión y Residencia gestionar un nuevo usuario del sistema, especificando cual es su rol en el mismo y con este los permisos que tendrá en la aplicación.
Resumen	El CUS se inicia cuando el Actor selecciona la opción de Gestionar Usuario, el sistema muestra las acciones Adicionar Usuario, Modificar Usuario y Eliminar Usuarios. Seguidamente el Actor selecciona la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Actor y termina el caso de uso.
Referencias	RF 1
Precondiciones	Para modificar los datos de un usuario, el mismo debe estar seleccionado. Para eliminar, el usuario correspondiente debe estar seleccionado.
Poscondiciones	Se creó, modificó o eliminó los datos de un usuario.
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Gestionar Usuario.	
	2. El sistema muestra las opciones siguientes: – Adicionar Usuario. – Modificar Usuario (Ver Sección “Modificar Datos de Usuario”). – Eliminar Usuario (Ver Sección “Eliminar

	Datos de Usuario”).
3. Selecciona la opción Adicionar Datos.	
	4. Muestra un formulario para insertar los datos siguientes del Usuario: – Nombre y Apellidos. – Usuario. – Contraseña. – Rol.
5. Introduce los datos de su interés y selecciona la opción Adicionar.	
	6. Valida los datos.
	7. Guarda los datos del Usuario.
	8. Muestra el mensaje de información “Proceso Satisfactorio.”
	9. El caso de uso termina.
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
6.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	6.a.1. Muestra el mensaje de error “Alguno de los datos son incorrectos.”
	6.a.2. Regresa al paso 5 del Curso Normal de Eventos.
Sección 1:”Modificar datos de Usuario”	
Curso Normal de los eventos:	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Modificar Usuario.	
	2. Muestra un listado con todos los

	usuarios del sistema.
3. Selecciona el usuario al cual le modificara los datos.	
	4. Consulta la base de datos y obtiene todos los datos del usuario.
	5. Muestra los datos del usuario y permite modificarlos.
6. Modifica los datos de su interés y selecciona la opción "Modificar".	
	7. Valida los datos.
	8. Actualiza los datos del usuario.
	9. Muestra el mensaje de información "Proceso Satisfactorio."
	10.El caso de uso termina.
Curso Alterno de los eventos:	
*a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
7.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	7.a.1. Muestra el mensaje de error "Alguno de los datos son incorrectos."
	7.a.2. Regresa al paso 6 del Curso Normal de Eventos.
Sección 2: Eliminar datos de Usuario	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema
1.Selecciona la opción Eliminar Usuario.	
	2.Muestra un listado con todos los usuarios del sistema.
3.Selecciona el usuario a eliminar y la opción eliminar usuario.	

	4.Muestra el mensaje de información siguiente: “Se eliminarán los datos del usuario”.
5.Selecciona la opción Eliminar.	
	6.Elimina los datos del usuario.
	7.Muestra el mensaje de información “Proceso Satisfactorio.”
	8.El caso de uso termina.
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
4. a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	5.a.1. El caso de uso termina.

Tabla 2.6 Descripción del caso de uso “Gestionar Usuario”

Descripción del CUS “Buscar Estudiantes”

Caso de Uso	Buscar Estudiantes
Actores	Vicedecano de Extensión y Residencia (inicia).
Propósito	Permite al Vicedecano de Extensión y Residencia buscar y consultar los datos con que cuenta el sistema de un estudiante.
Resumen	El CUS se inicia cuando el Actor selecciona la opción de Buscar Estudiante, siguiendo los pasos que le impone el sistema llega hasta el estudiante del cual desea conocer sus datos, pide que esta acción se concrete, consulta los datos y termina el caso de uso.
Referencias	RF 2
Precondiciones	Para consultar los datos de un Estudiante el mismo debe estar dentro del sistema.
Poscondiciones	Se encontraron y consultaron los datos de algún o algunos estudiantes.
Curso normal de los eventos	

Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Buscar Estudiante.	
	2. Consulta la base de datos y obtiene el identificador de cada brigada.
	3. Muestra un listado con todos los identificadores de las brigadas, además de campos para entrar: <ul style="list-style-type: none"> – Primer Nombre. – Segundo Nombre. – Primer Apellido. – Segundo Apellido.
4. Selecciona la brigada a la que pertenece el estudiante que se esta buscando, selecciona la opción “Buscar”.	
	5. Consulta la base de datos y obtiene el nombre de todos los estudiantes de la brigada seleccionada.
	6. Muestra un listado con todos los nombres de los estudiantes que pertenecen a la brigada, donde especifica: <ul style="list-style-type: none"> – Nombre y Apellidos. – Brigada. – Apto. – Teléfono.
7. Selecciona la opción “Consultar Datos” del estudiante que le interese.	
	8. Consulta la base de datos y obtiene todos los datos del estudiante.
	9. Muestra un resumen del estudiante e tareas coma:

	<ul style="list-style-type: none"> – Guardia Estudiantil. – TSU. – Cuartelarías.
	10. Termina el caso de uso.
Curso Alternativo de los eventos:	
4.a Realizar la búsqueda por datos del nombre y apellidos del estudiante.	
Acción del actor	Respuesta del sistema
4.a.1. Introduce los datos de su conocimiento del Nombre y Apellidos, selecciona la opción “Buscar”.	
	4.a.2. Consulta la Base de Datos y muestra un listado con todos los estudiantes que su nombre y apellido coincida con los datos especificados.
	4.a.3. Muestra un listado con los resultados de la consulta, especificando de cada uno: <ul style="list-style-type: none"> – Nombre y Apellidos. – Brigada. – Apto. – Teléfono.
4.a.4. Regresa al paso 7 del curso normal de eventos.	
Curso Alternativo de los eventos:	
*.a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	4.a.1. Cancela los datos.
	4.a.2. El caso de uso termina.

Tabla 2.7 Descripción del CUS “Buscar Estudiantes”

Descripción del CUS “Gestionar Postas de la Guardia Estudiantil”

Caso de Uso	Gestionar Postas de la Guardia Estudiantil
Actores	Vicedecano de Extensión y Residencia (inicia).

Propósito	
Resumen	El CUS se inicia cuando el Actor selecciona la opción de Gestionar posta de la guardia estudiantil, el sistema muestra las opciones Insertar Posta, Eliminar Posta y Modificar Posta. Seguidamente el actor selecciona la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Actor y termina el caso de uso.
Referencias	RF 4
Precondiciones	Para modificar los datos de una posta de la guardia, la misma tiene que estar seleccionada. Para eliminar, la posta de la guardia correspondiente debe estar seleccionada.
Poscondiciones	Se creó, modificó o eliminó los datos de una posta de la guardia.
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Gestionar postas de la guardia estudiantil.	
	2. El sistema muestra las opciones siguientes: – Adicionar Posta (Ver Sección “Adicionar Datos de una Posta de la Guardia Estudiantil”). – Modificar Posta (Ver Sección “Modificar Datos de una Posta de la Guardia Estudiantil”). – Eliminar Posta (Ver Sección “Eliminar Datos de una Posta de la Guardia Estudiantil”).
3. Selecciona la opción Adicionar Posta.	
	4. Muestra un formulario para insertar el dato siguiente de la posta: – Ubicación.
5. Introduce el dato y selecciona la opción “Aceptar”.	

	6. Valida los datos.
	7. Guarda el dato de la posta.
	8. Muestra el mensaje de información "Proceso Satisfactorio."
	9. El caso de uso termina.
Curso Alterno de los eventos:	
*a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
6.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	6.a.3. Muestra el mensaje de error "Existen datos incorrectos".
	6.a.4. Regresa al paso 5 del Curso Normal de Eventos.
Sección 1: "Modificar los datos de una posta de la guardia estudiantil"	
Curso Normal de los eventos:	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Modificar posta.	
	2. Muestra un listado con todas las postas existentes en el sistema.
3. Selecciona la posta a la cual le modificara los datos y selecciona la opción modificar datos.	
	4. Consulta la base de datos y obtiene todos los datos de la posta.
	5. Muestra los datos de la posta y permite modificarlos.
6. Modifica los datos de su interés y selecciona la opción "Modificar".	
	7. Valida los datos.

	8. Actualiza los datos de la posta.
	9. Muestra el mensaje de información "Proceso Satisfactorio."
	10.El caso de uso termina.
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
7.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	7.a.1. Muestra el mensaje de error "Existen datos incorrectos."
	7.a.2. Regresa al paso 6 del Curso Normal de Eventos.
Sección 2: Eliminar datos de una posta de la guardia estudiantil	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema
1. Selecciona la opción Eliminar Posta.	
	2.Muestra un listado con todas las postas de la guardia estudiantil existentes en el sistema.
3. Selecciona la posta a eliminar y la opción "Eliminar".	
	4. Muestra el mensaje de información siguiente: "Se eliminarán los datos de la posta seleccionada".
5. Selecciona la opción Eliminar.	
	6. Elimina los datos de la posta.
	7. Muestra el mensaje de información "Proceso Satisfactorio".
	8.El caso de uso termina.
Curso Alterno de los eventos:	

*a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	*.a.3. Cancela los datos.
	*.a.4. El caso de uso termina.
5. a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	5.a.2. El caso de uso termina.

Tabla 2.10 Descripción del caso de uso “Gestionar Postas de la Guardia Estudiantil”

Descripción del CUS “Gestionar Brigada”

Caso de Uso	Gestionar Brigada
Actores	Vicedecano de Extensión y Residencia (inicia).
Propósito	Permite al Vicedecano de Extensión y Residencia gestionar una brigada dentro del sistema.
Resumen	El CUS se inicia cuando el Actor selecciona la opción de Gestionar Brigada, el sistema muestra las opciones Insertar Brigada, Modificar Datos de Brigada y Eliminar Datos de Brigada. Seguidamente el Actor selecciona la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Actor y termina el caso de uso.
Referencias	RF 5
Precondiciones	Para modificar los datos de una brigada, la misma debe estar seleccionada. Para eliminar una persona la misma debe estar seleccionada.
Poscondiciones	Se creo, modificó o eliminó los datos de una brigada.
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Gestionar Brigada.	
	2. El sistema muestra las opciones siguientes: – Adicionar Brigada (Ver Sección “Adiconar

	Datos de una Brigada”). – Modificar Brigada (Ver Sección “Modificar Datos de una Brigada”). – Eliminar Brigada (Ver Sección “Eliminar Datos de una Brigada”).
3. Selecciona la opción Adicionar Brigada.	
	4. Muestra un formulario para insertar los siguientes datos de la brigada: – ID.
5. Introduce los datos y selecciona la opción Aceptar.	
	6. Valida los datos.
	7. Guarda los datos de la Brigada.
	8. Muestra el mensaje de información “Proceso Satisfactorio.”
	9. El caso de uso termina.
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
6.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	6.a.1. Muestra el mensaje de error “Existen datos incorrectos”.
	6.a.2. Regresa al paso 5 del Curso Normal de Eventos.
Sección 1:”Modificar datos de una brigada”	
Curso Normal de los eventos:	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Modificar Brigada.	
	2. Muestra un listado con todas las

	brigadas existentes en el sistema.
3. Selecciona la brigada a modificar.	
	4. Consulta la base de datos y obtiene el dato de la brigada. – ID.
	5. Muestra el dato de la brigada y permite modificarlo.
6. Modifica el dato de su interés y selecciona la opción “Modificar”.	
	7. Valida el dato.
	8. Actualiza el dato de la brigada.
	9. Muestra el mensaje de información “Proceso Satisfactorio.”
	10.El caso de uso termina.
Curso Alterno de los eventos:	
*a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
7.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	7.a.1. Muestra el mensaje de error “Existen datos incorrectos.”
	7.a.2. Regresa al paso 6 del Curso Normal de Eventos.
Sección 2: Eliminar datos de una brigada	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema
1. Selecciona la opción Eliminar Brigada.	
	2. Muestra un listado con todas las brigadas existentes en el sistema.
3. Selecciona la brigada la cual desea	

eliminar.	
	4. Muestra el mensaje de información siguiente: “Se eliminará la brigada”.
5. Selecciona la opción Eliminar.	
	6. Elimina la brigada.
	7. Muestra el mensaje de información “Proceso Satisfactorio.”
	8. El caso de uso termina.
Curso Alterno de los eventos:	
*a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
5. a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	5.a.1. El caso de uso termina.

Tabla 2.11 Descripción del caso de uso “Gestionar Brigada”

Descripción del CUS “Gestionar Rol”

Caso de Uso	Gestionar Rol
Actores	Vicedecano de Extensión y Residencia (inicia).
Propósito	
Resumen	El CUS se inicia cuando el Actor selecciona la opción de Gestionar Rol, el sistema muestra las opciones Crear Rol, Eliminar Rol, Modificar Rol, Listar Roles. Seguidamente el actor selecciona la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Actor y termina el caso de uso.
Referencias	RF 10
Precondiciones	Para modificar los datos de un rol, el mismo tiene que estar seleccionado. Para eliminar, el rol correspondiente debe estar seleccionado.
Poscondiciones	Se crearon, modificaron, eliminaron o listaron los datos de un rol determinado.

Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Gestionar Rol.	
	2. El sistema muestra las opciones siguientes: – Adicionar Rol (Ver Sección “Adicionar Rol”). – Modificar Rol (Ver Sección “Modificar Datos de un Rol”). – Eliminar Rol (Ver Sección “Eliminar Datos de Rol”). – Listar Rol (Ver Sección “Listar Roles”).
3. Selecciona la opción Adicionar Rol.	
	4. Muestra un formulario para insertar los datos siguientes de un rol: – Nombre. – Listado de Permisos.
5. Introduce los datos de su interés y selecciona la opción “Adicionar”.	
	6. Valida los datos.
	7. Guarda los datos del Rol.
	8. Muestra el mensaje de información “Proceso Satisfactorio.”
	9. El caso de uso termina.
Curso Alternativo de los eventos:	
*a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
6.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	6.a.1. Muestra el mensaje de error “Existen datos incorrectos.”

	6.a.2. Regresa al paso 5 del Curso Normal de Eventos.
Sección 1: "Modificar los datos de un rol"	
Curso Normal de los eventos:	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Modificar Rol.	
	2. Muestra un listado con todas los roles existentes en el sistema.
3. Selecciona el rol a el cual le modificara los datos y selecciona la opción "Modificar".	
	4. Consulta la base de datos y obtiene todos los datos del rol.
	5. Muestra los datos del rol y permite modificarlos.
6. Modifica los datos de su interés y selecciona la opción actualizar datos.	
	7. Valida los datos.
	8. Actualiza los datos del rol.
	9. Muestra el mensaje de información "Proceso Satisfactorio."
	10.El caso de uso termina.
Curso Alterno de los eventos:	
*a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
7.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	7.a.1. Muestra el mensaje de error "Existen datos incorrectos."

	7.a.2. Regresa al paso 6 del Curso Normal de Eventos.
Sección 2: Eliminar datos de un rol	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema
1. Selecciona la opción Eliminar Rol.	
	2. Muestra un listado con todos los roles existentes en el sistema.
3. Selecciona el rol a eliminar y la opción "Eliminar".	
	4. Muestra el mensaje de información siguiente: "Se eliminarán los datos del rol seleccionada".
5. Selecciona la opción Eliminar.	
	6. Elimina los datos del rol.
	7. Muestra el mensaje de información "Proceso Satisfactorio."
	8. El caso de uso termina.
Curso Alterno de los eventos:	
*a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
5. a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	5.a.1. El caso de uso termina.
Sección 2: Listar roles	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema
1. Selecciona la opción listar roles.	
	2. Muestra un listado con todos los roles existentes en el sistema especificando de ellos:

	– Nombre
3. Selecciona un rol de la lista y la opción ver detalles.	
	4. Muestra del rol seleccionado: – Nombre – Lista de privilegios
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.

Tabla 2.10 Descripción del caso de uso “Gestionar Rol”

Descripción del CUS “Gestionar Módulo”

Caso de Uso	Gestionar Módulo
Actores	Vicedecano de Extensión y Residencia (inicia).
Propósito	Permite al Vicedecano de Extensión y Residencia gestionar un nuevo módulo del sistema, permitiendo que el mismo sea implementado y añadido de una forma simple al sistema.
Resumen	El CUS se inicia cuando el Actor selecciona la opción de Gestionar Módulo, el sistema muestra las acciones Adicionar Módulo, Modificar Módulo y Eliminar Módulo. Seguidamente el Actor selecciona la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Actor y termina el caso de uso.
Referencias	RF 11
Precondiciones	Para modificar un Modulo, el mismo debe estar seleccionado. Para eliminar, el modulo correspondiente debe estar seleccionado.
Poscondiciones	Se creó, modificó o eliminó los datos de un usuario.
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Gestionar Módulo.	

	<p>2. El sistema muestra las opciones siguientes:</p> <ul style="list-style-type: none"> – Adicionar Módulo. (Ver Sección “Adicionar Datos de Módulo”) – Modificar Módulo (Ver Sección “Modificar Datos de Módulo”). – Eliminar Módulo (Ver Sección “Eliminar Datos de Módulo”).
3. Selecciona la opción Adicionar Módulo.	
	<p>4. Muestra un formulario para insertar los datos siguientes del Módulo:</p> <ul style="list-style-type: none"> – Nombre.
5. Introduce el nombre del módulo y selecciona la opción “Crear”.	
	6. Valida el dato.
	7. Guarda el dato del módulo.
	<p>8. Solicita las funcionalidades del módulo y para ellas:</p> <ul style="list-style-type: none"> – Nombre. – Vinculo.
9. Introduce los datos solicitados por el sistema y selecciona la opción “Adicionar” para cada funcionalidad.	
	10. Valida los Datos.
	11. Adiciona a la Base de Datos la funcionalidad especificada.
	12. Regresa al paso 9 del curso normal de eventos.
Curso Alternativo de los eventos:	
*a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.

6.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	6.a.1. Muestra el mensaje de error “Alguno de los datos son incorrectos.”
	6.a.2. Regresa al paso 5 del Curso Normal de Eventos.
10.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	10.a.1.Muestra el mensaje de error “Alguno de los datos son incorrectos.”
	10.a.2.Regresa al paso 5 del Curso Normal de Eventos.
Sección 1:”Modificar datos de Módulo”	
Curso Normal de los eventos:	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Modificar Módulo.	
	2. Muestra un listado con todos los módulos del sistema.
3. Selecciona el módulo al cual le modificara los datos.	
	4. Consulta la base de datos y obtiene todos los datos del módulo.
	5. Muestra los datos del módulo y permite modificarlos.
6. Modifica los datos de su interés y selecciona la opción “Modificar”.	
	7. Valida los datos.
	8. Actualiza los datos del módulo.
	9. Muestra el mensaje de información “Proceso Satisfactorio.”
	10.El caso de uso termina.
Curso Alterno de los eventos:	

*.a Selecciona la opción Cancelar	
Acción del actor	Respuesta del sistema
	*.a.3. Cancela los datos.
	*.a.4. El caso de uso termina.
7.a Existen datos incorrectos	
Acción del actor	Respuesta del sistema
	7.a.3. Muestra el mensaje de error “Alguno de los datos son incorrectos.”
	7.a.4. Regresa al paso 6 del Curso Normal de Eventos.
Sección 2: Eliminar datos de Módulo	
Curso Normal de los eventos:	
Acción del actor	Respuesta del sistema
1. Selecciona la opción Eliminar Módulo.	
	2. Muestra un listado con todos los módulos del sistema.
3. Selecciona el módulo a eliminar y la opción “Eliminar Módulo”.	
	4. Muestra el mensaje de información siguiente: “Se eliminarán los datos del módulo”.
5. Selecciona la opción Eliminar.	
	6. Elimina los datos del módulo.
	7. Muestra el mensaje de información “Proceso Satisfactorio.”
	8. El caso de uso termina.
Curso Alterno de los eventos:	
*.a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema
	*.a.1. Cancela los datos.
	*.a.2. El caso de uso termina.
5. a Selecciona la opción Cancelar.	
Acción del actor	Respuesta del sistema

	5.a.1. El caso de uso termina.
--	--------------------------------

Tabla 2.11 Descripción del caso de uso "Gestionar Módulo"

Anexo 3: Diagrama de clases de análisis del resto de los caso de uso.

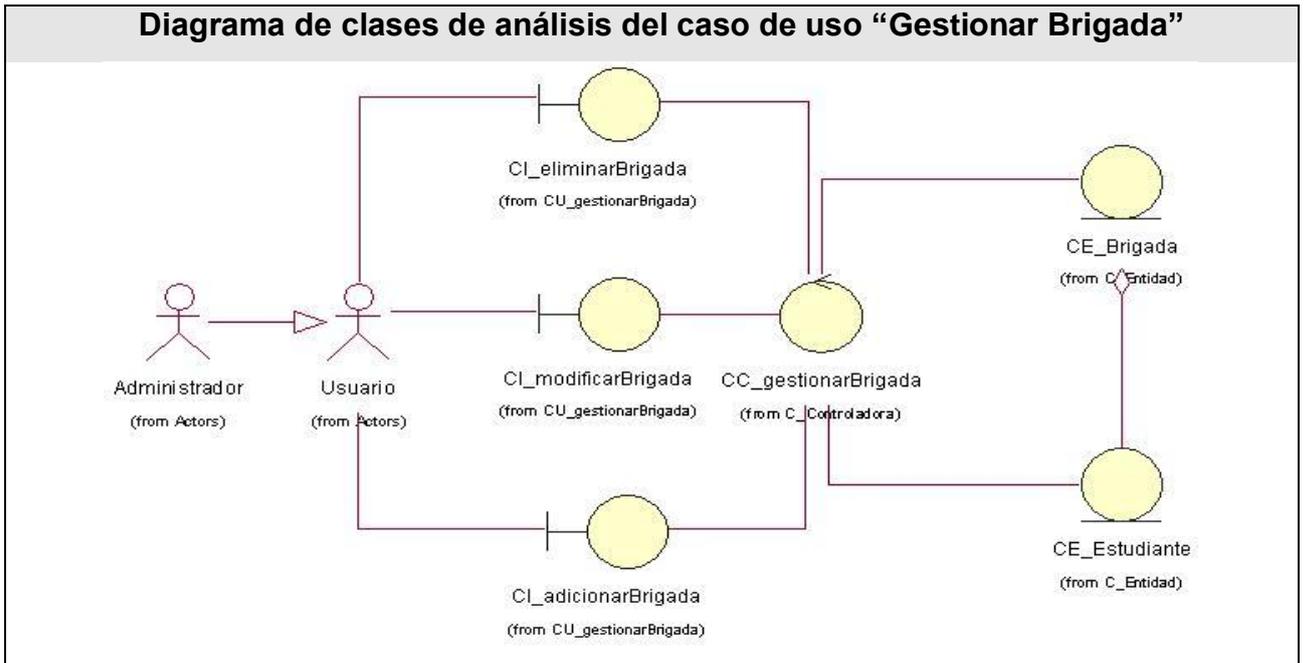


Diagrama de clases de análisis del caso de uso "Gestionar Modulo"

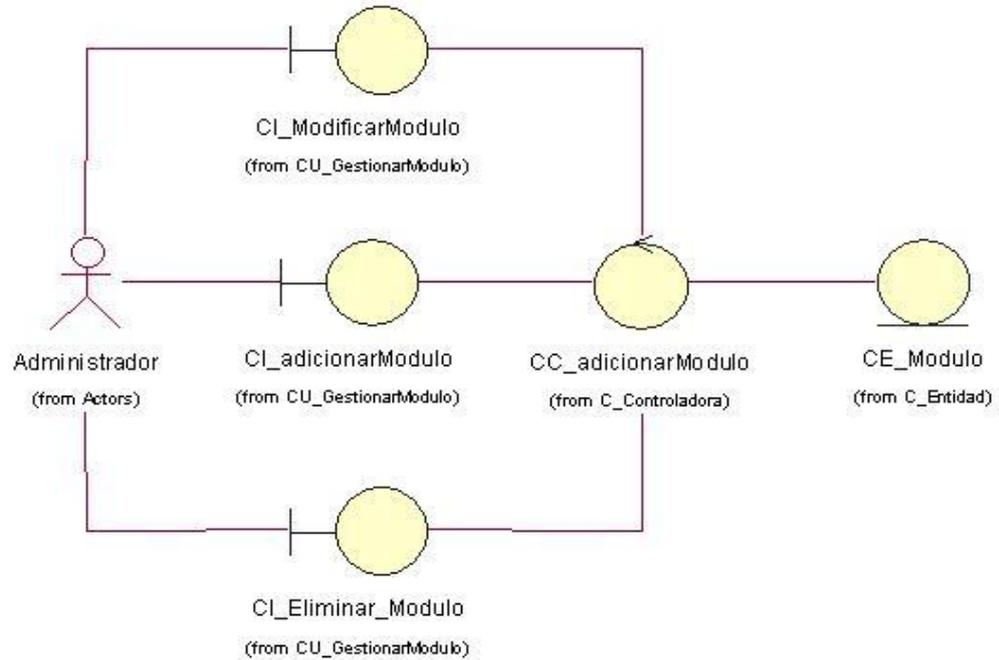


Diagrama de clases de análisis del caso de uso "Gestionar Rol"

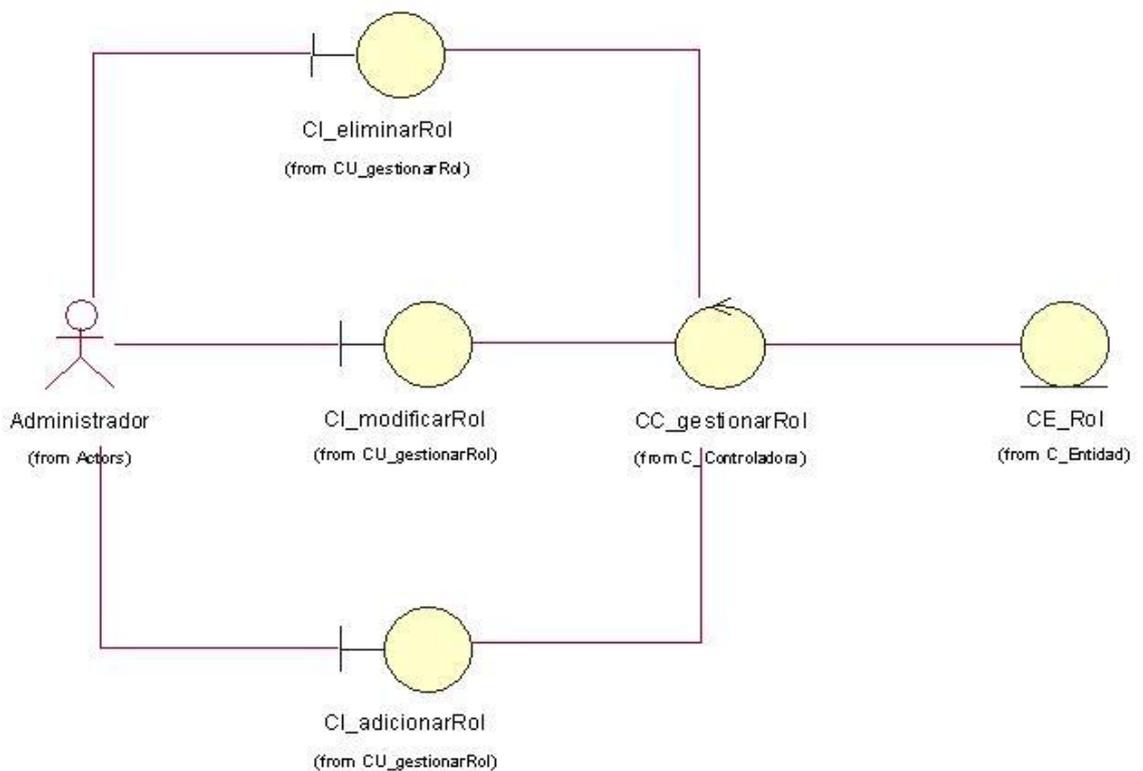


Diagrama de clases de análisis del caso de uso "Gestionar Postas"

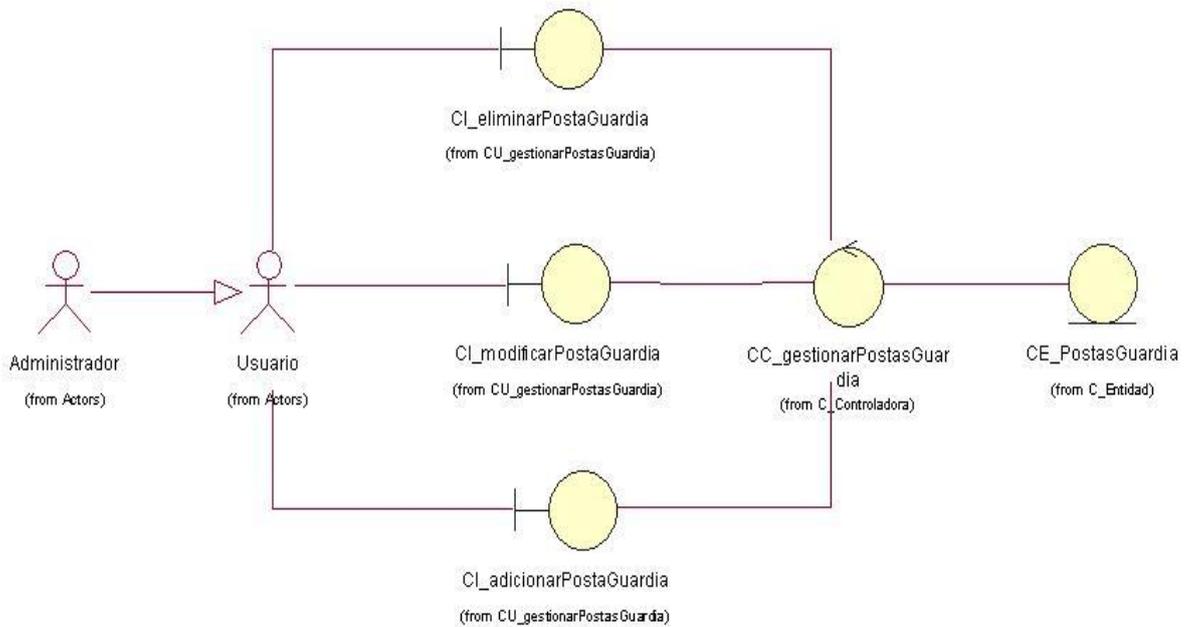
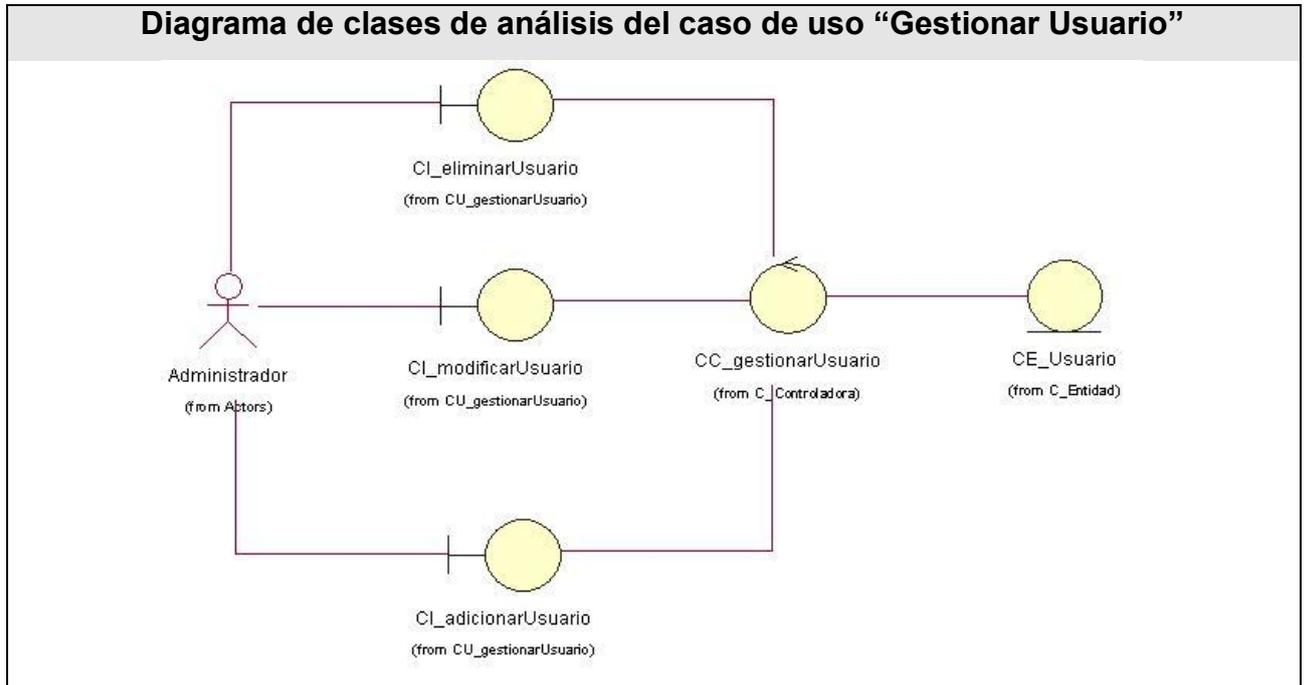


Diagrama de clases de análisis del caso de uso "Buscar Estudiante"





Anexo 4: Diagrama de clases Web del resto de los casos de uso.

Diagrama de clases Web para el escenario "Modificar Cuartelería"

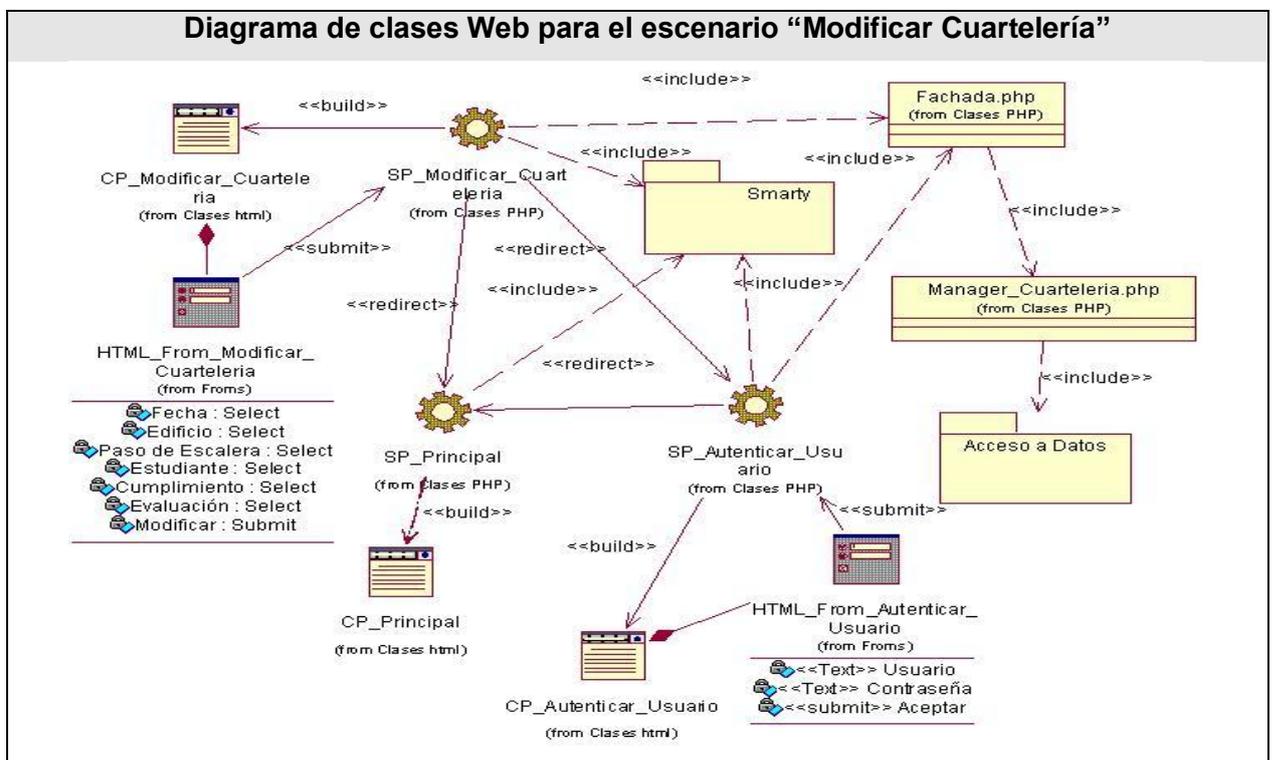


Diagrama de clases Web para el escenario "Listar Cuartelería"

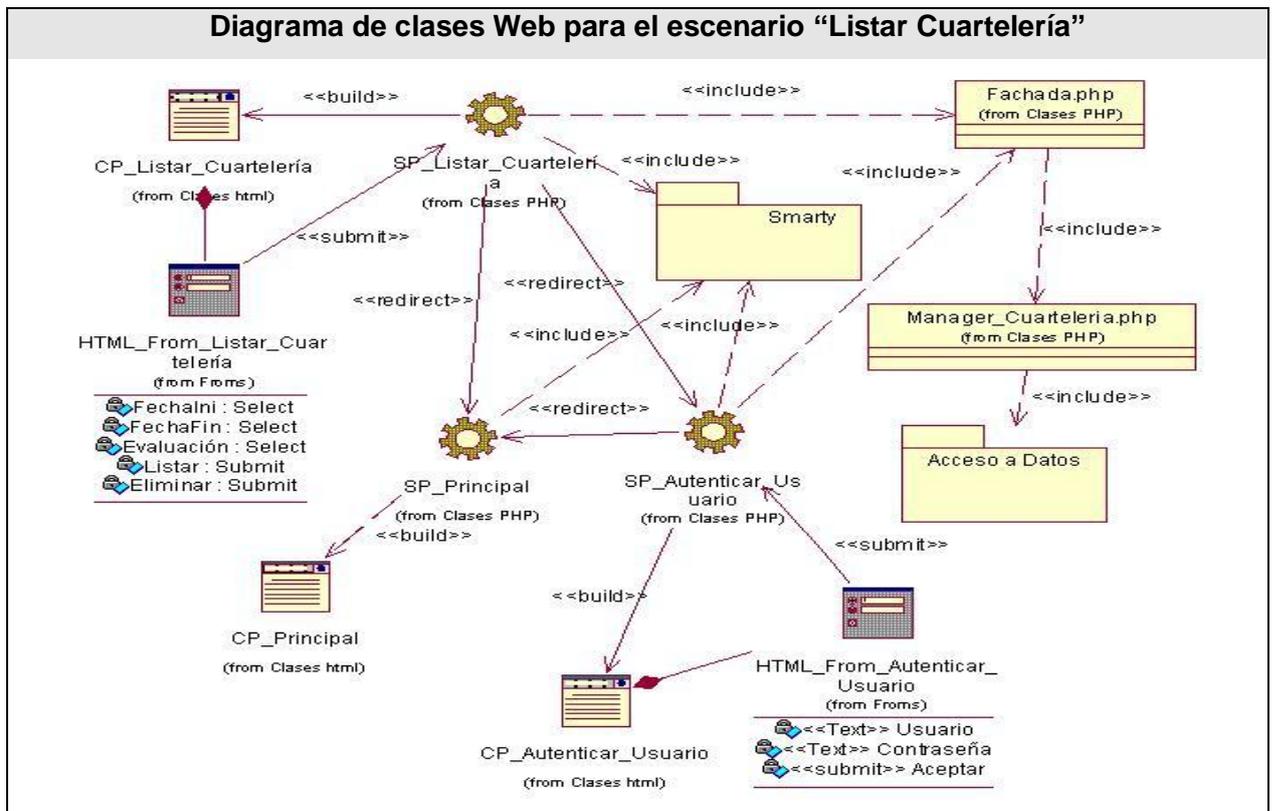


Diagrama de clases Web para el escenario "Adicionar TSU"

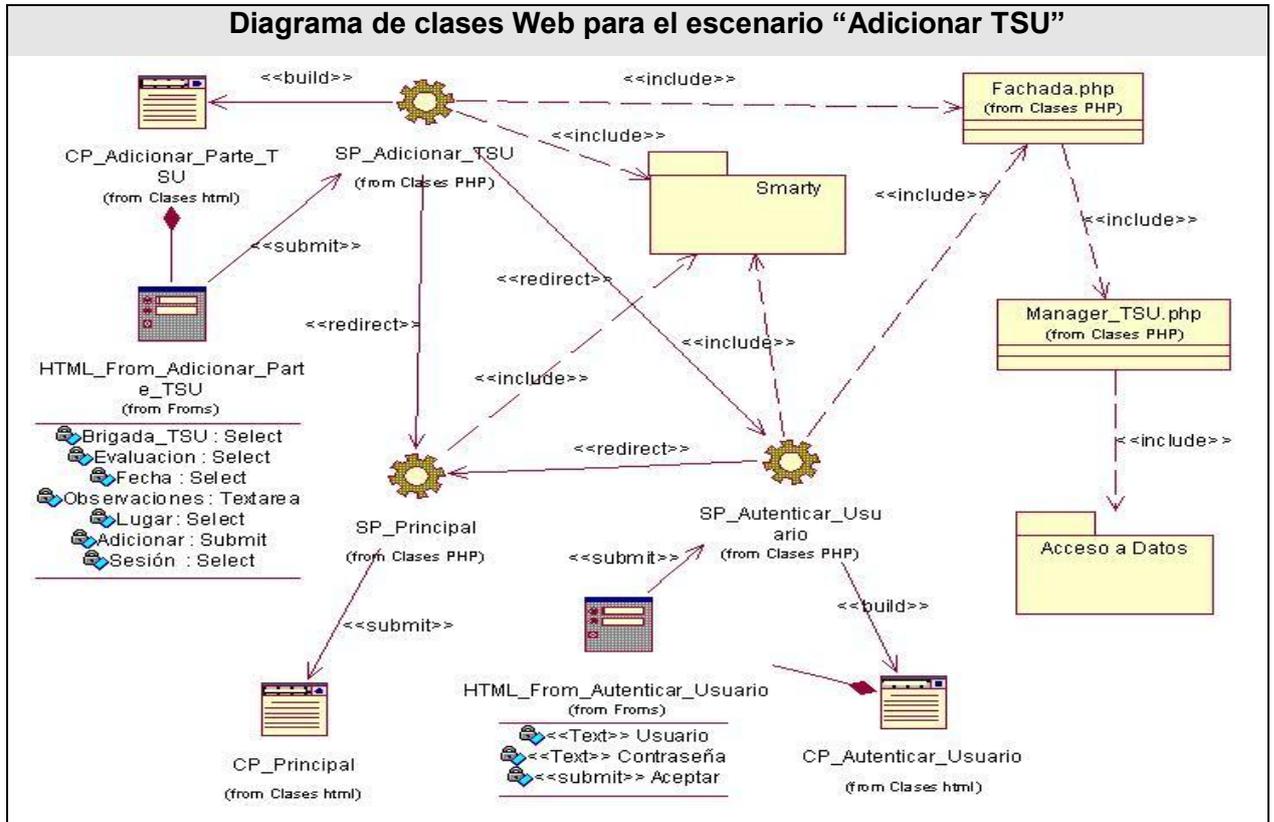


Diagrama de clases Web para el escenario "Modificar TSU"

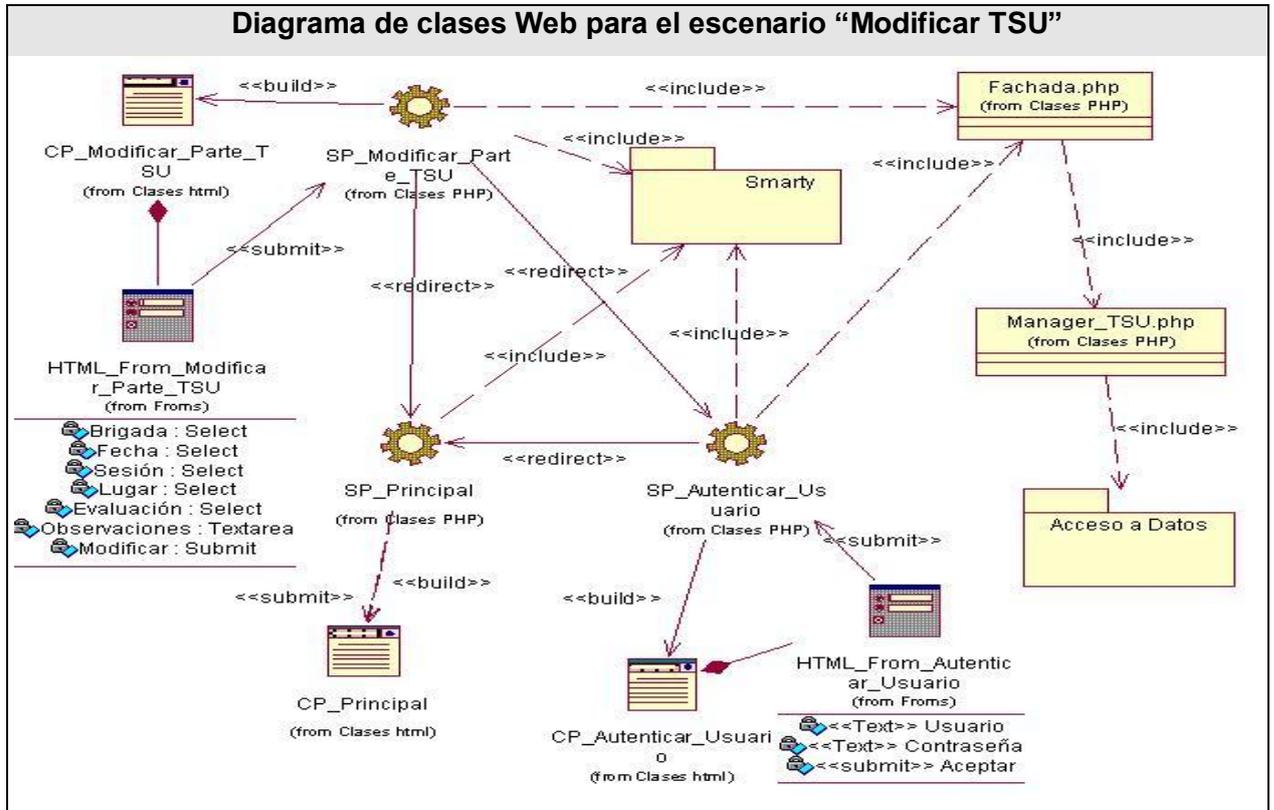


Diagrama de clases Web para el escenario "Adicionar Persona"

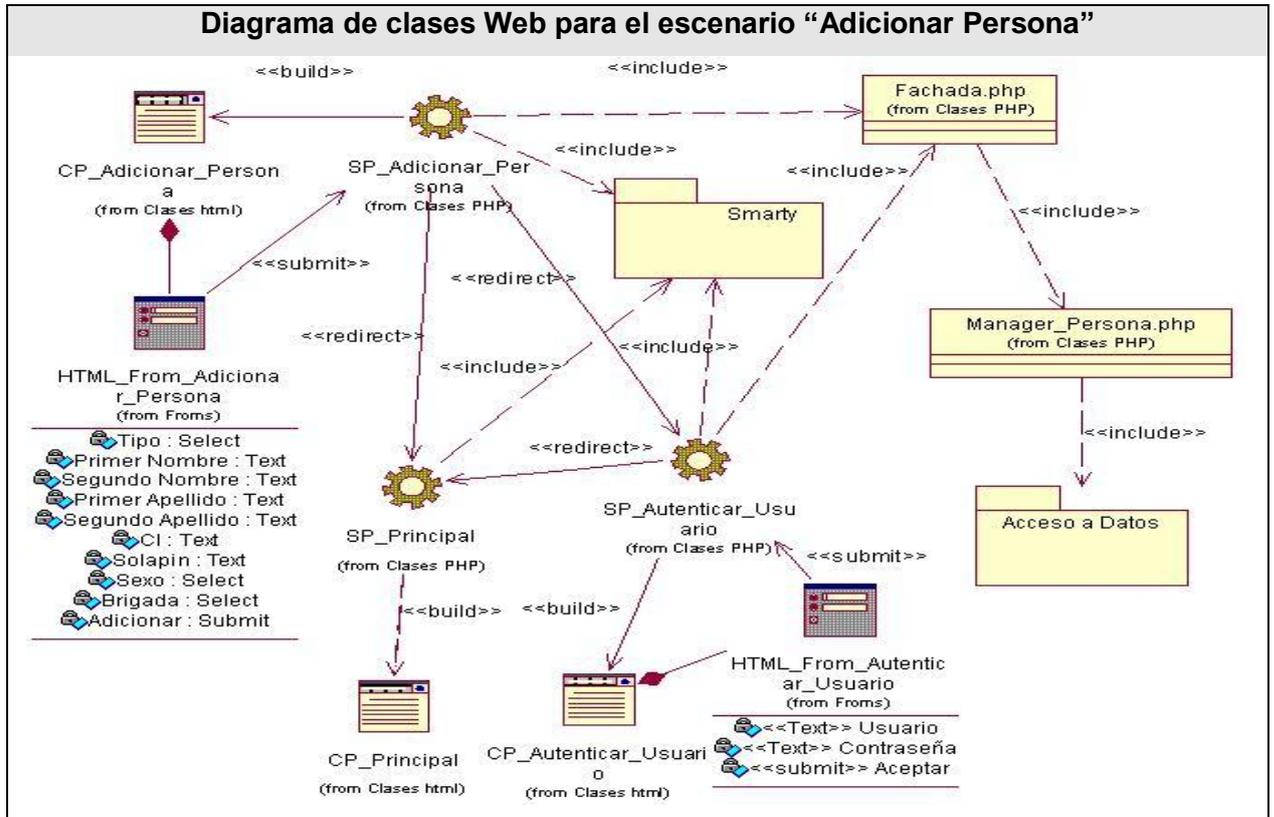


Diagrama de clases Web para el escenario “Modificar Persona”

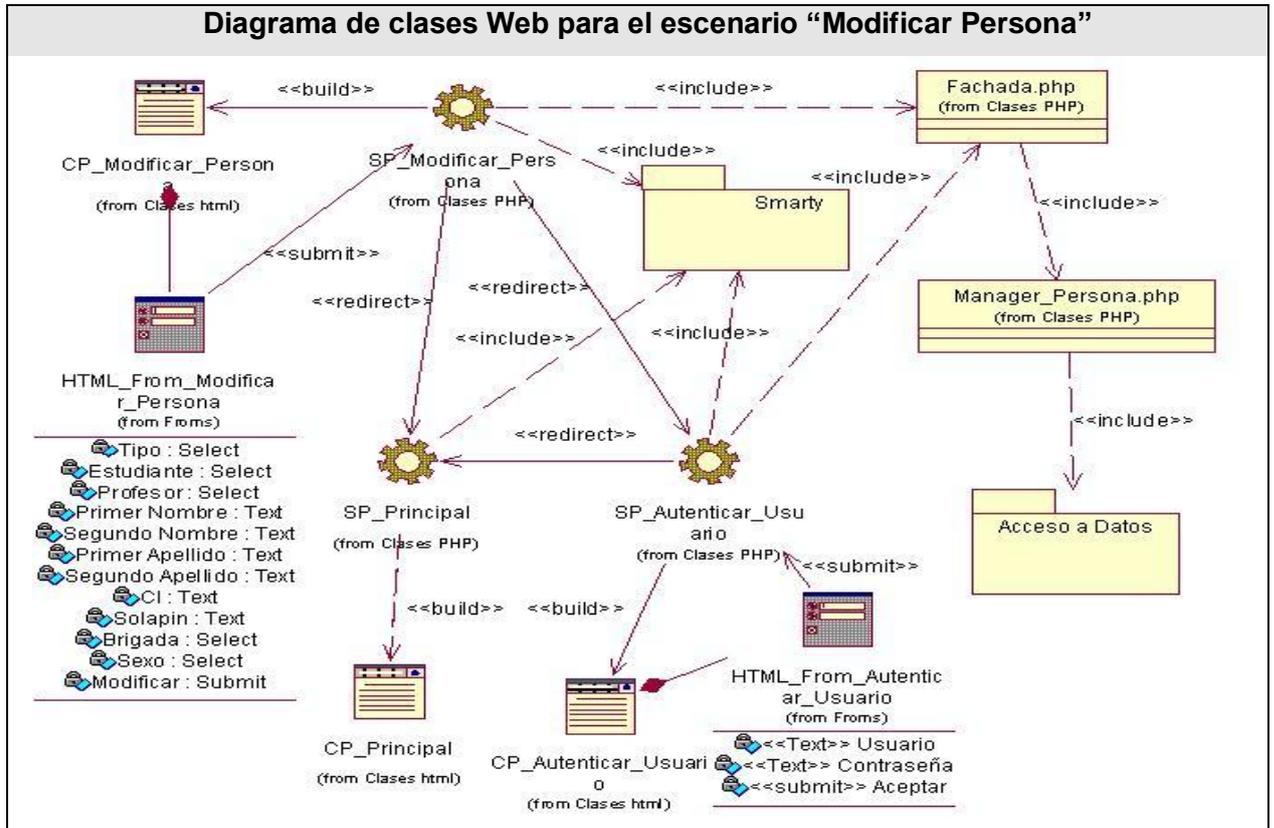


Diagrama de clases Web para el escenario "Adicionar Modulo"

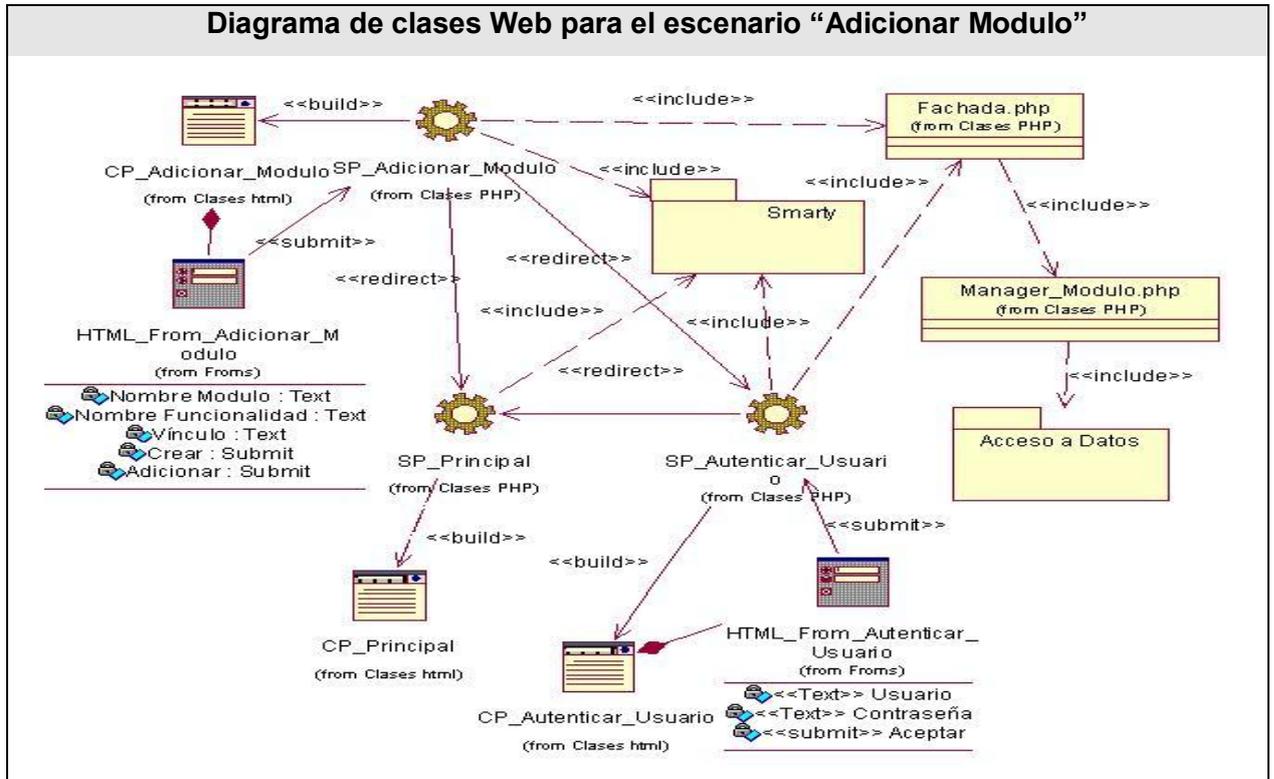


Diagrama de clases Web para el escenario “Editar Modulo”

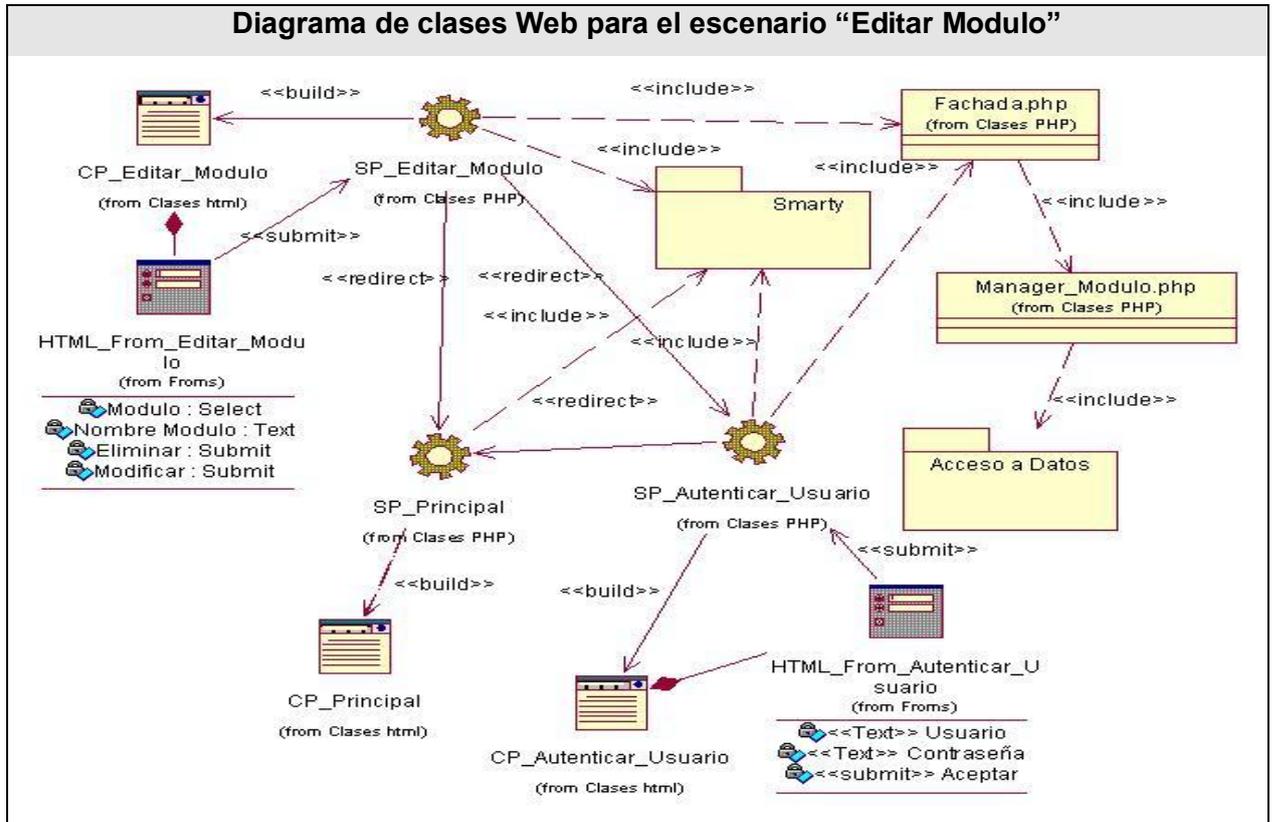


Diagrama de clases Web para el escenario "Listar Rol"

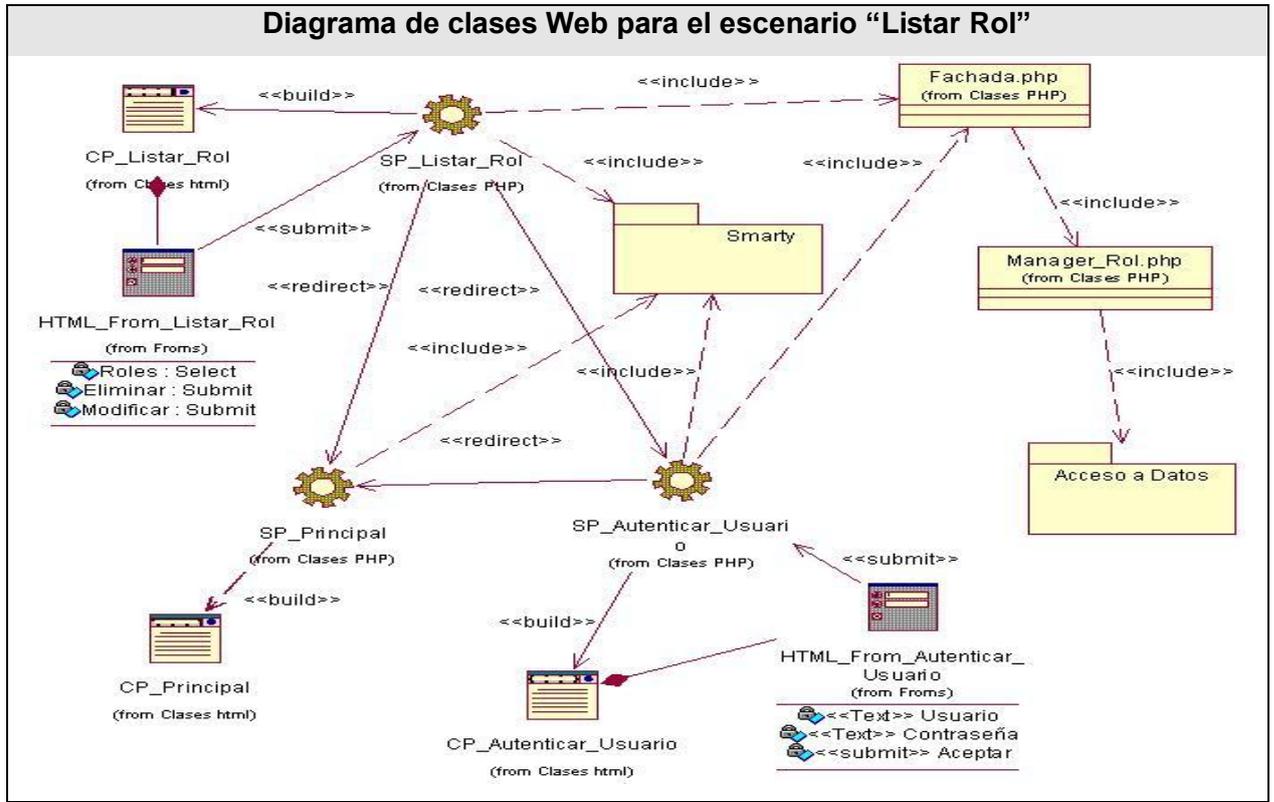


Diagrama de clases Web para el escenario "Modificar Rol"

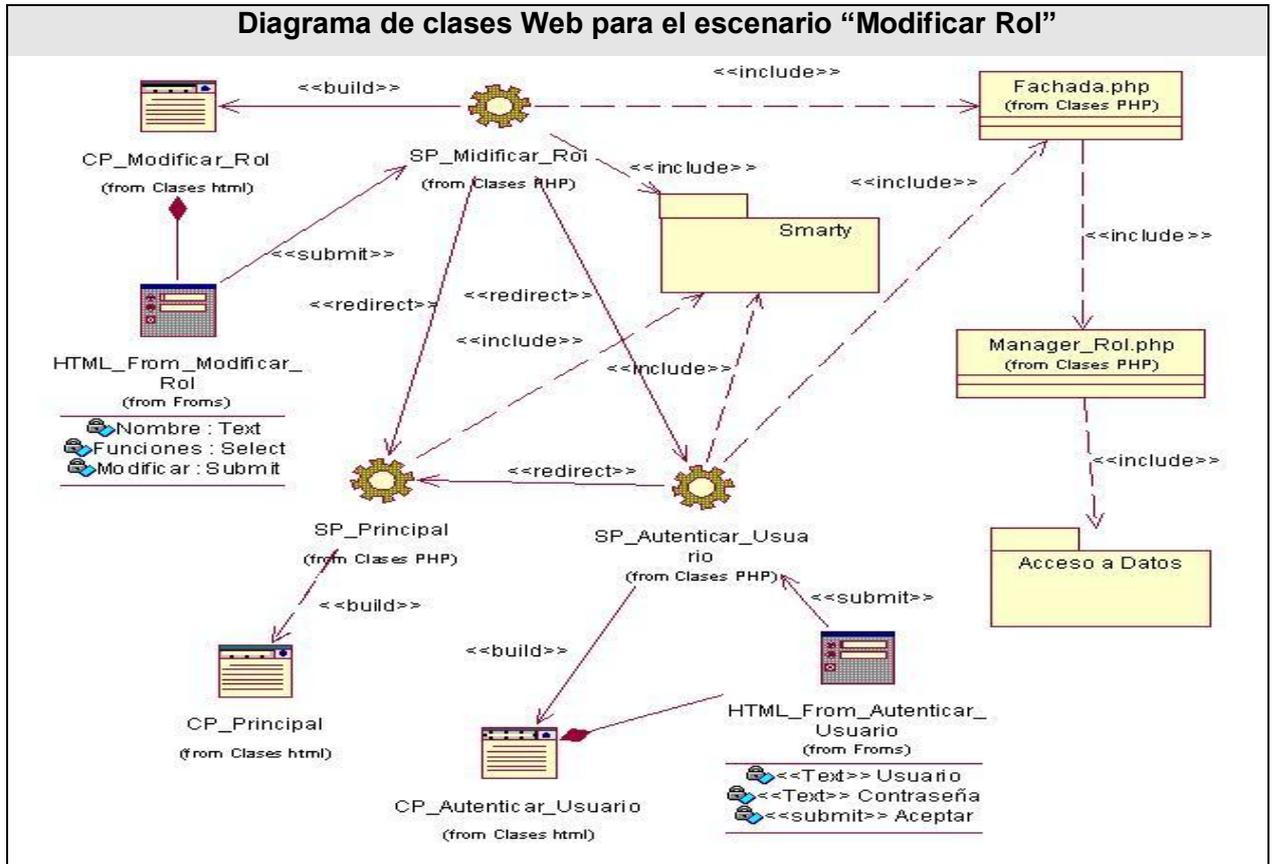


Diagrama de clases Web para el escenario "Adicionar Usuario"

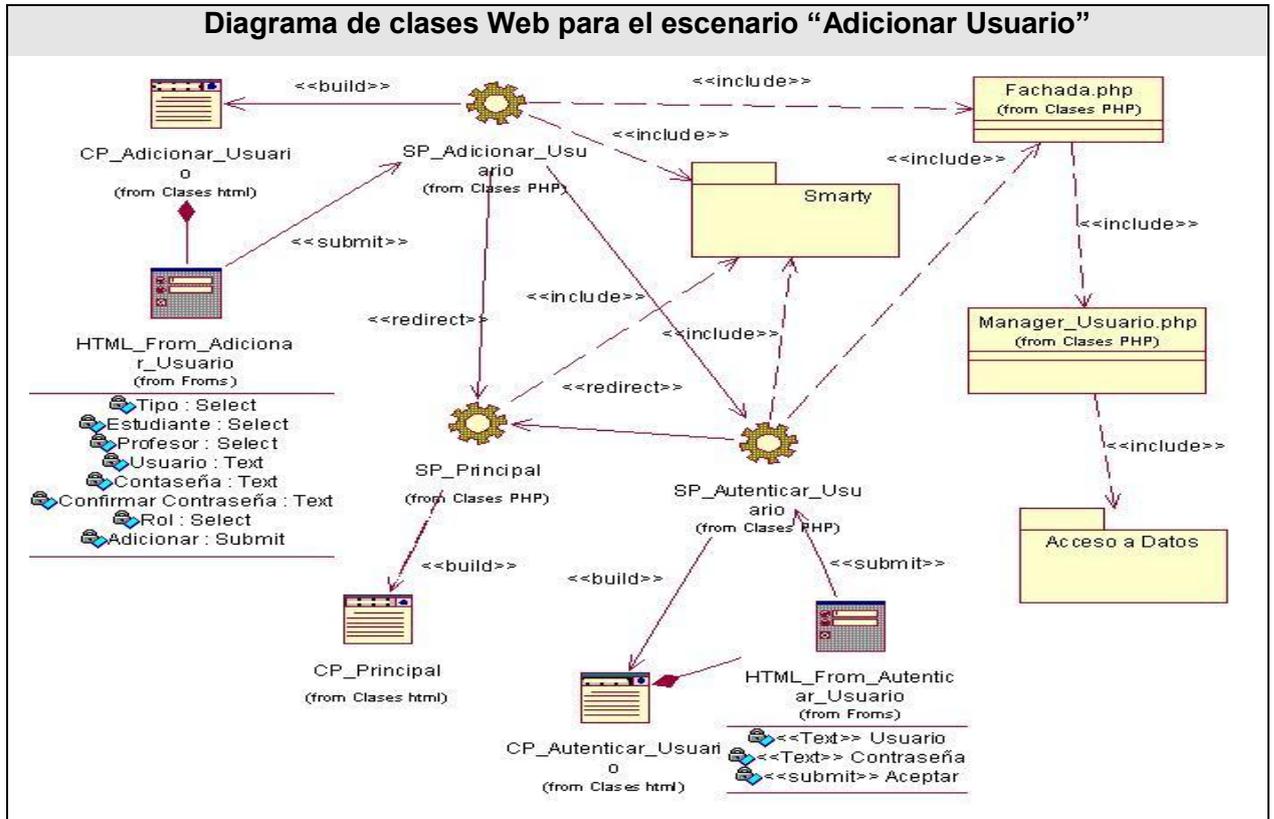
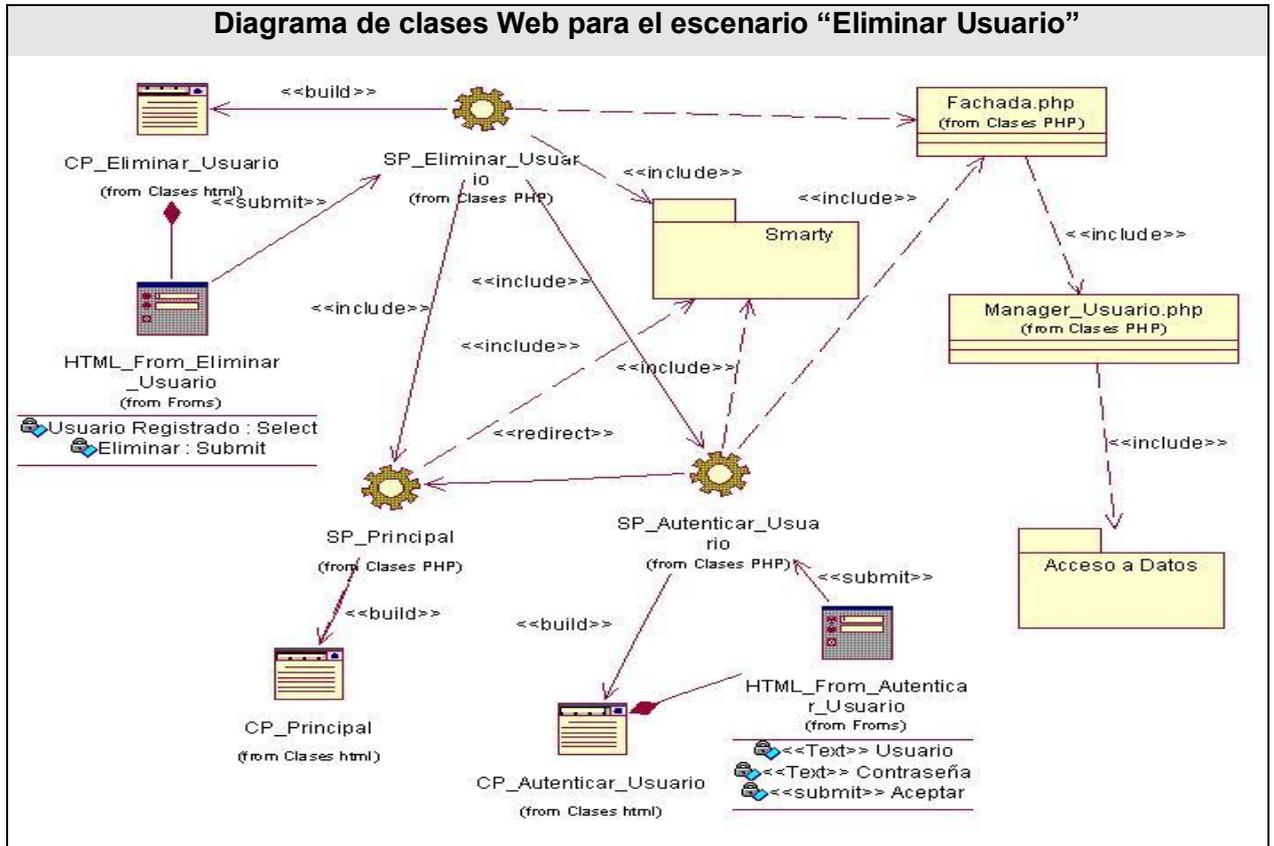


Diagrama de clases Web para el escenario "Eliminar Usuario"



Anexo 5: Diagrama de componentes del resto de los caso de uso.

Diagrama de componentes del caso de uso “Buscar Estudiante”

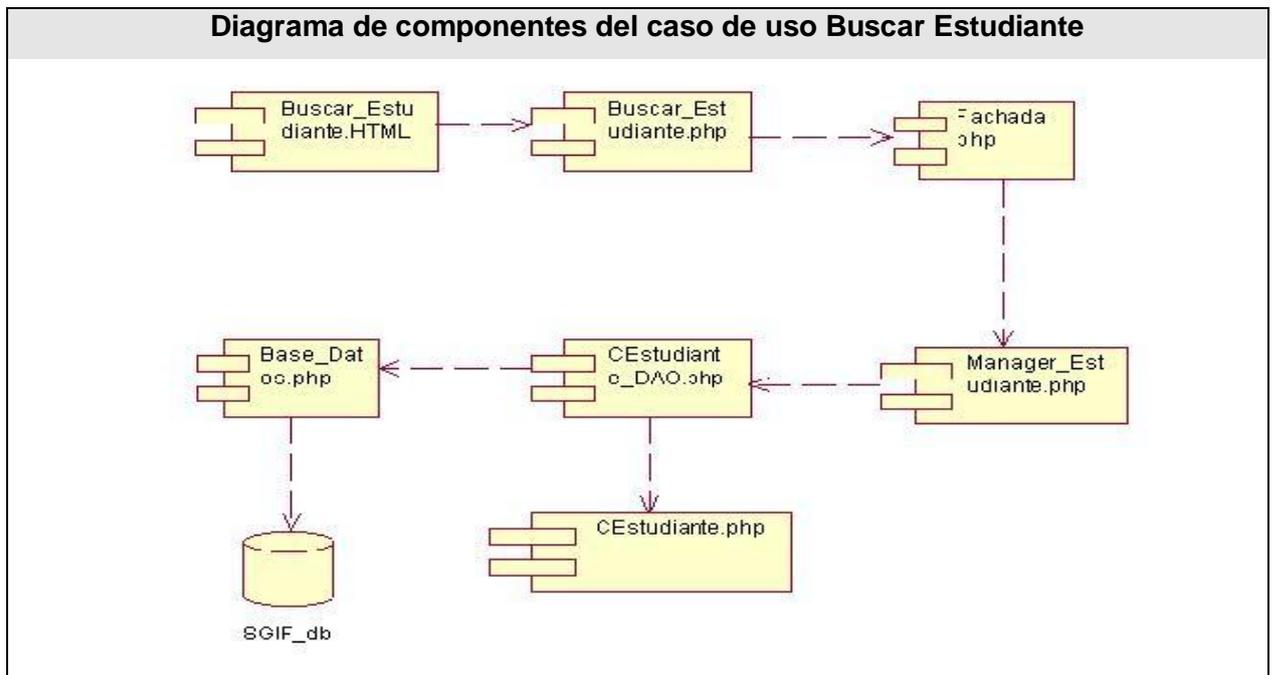


Diagrama de componentes del caso de uso “Gestionar Modulo”

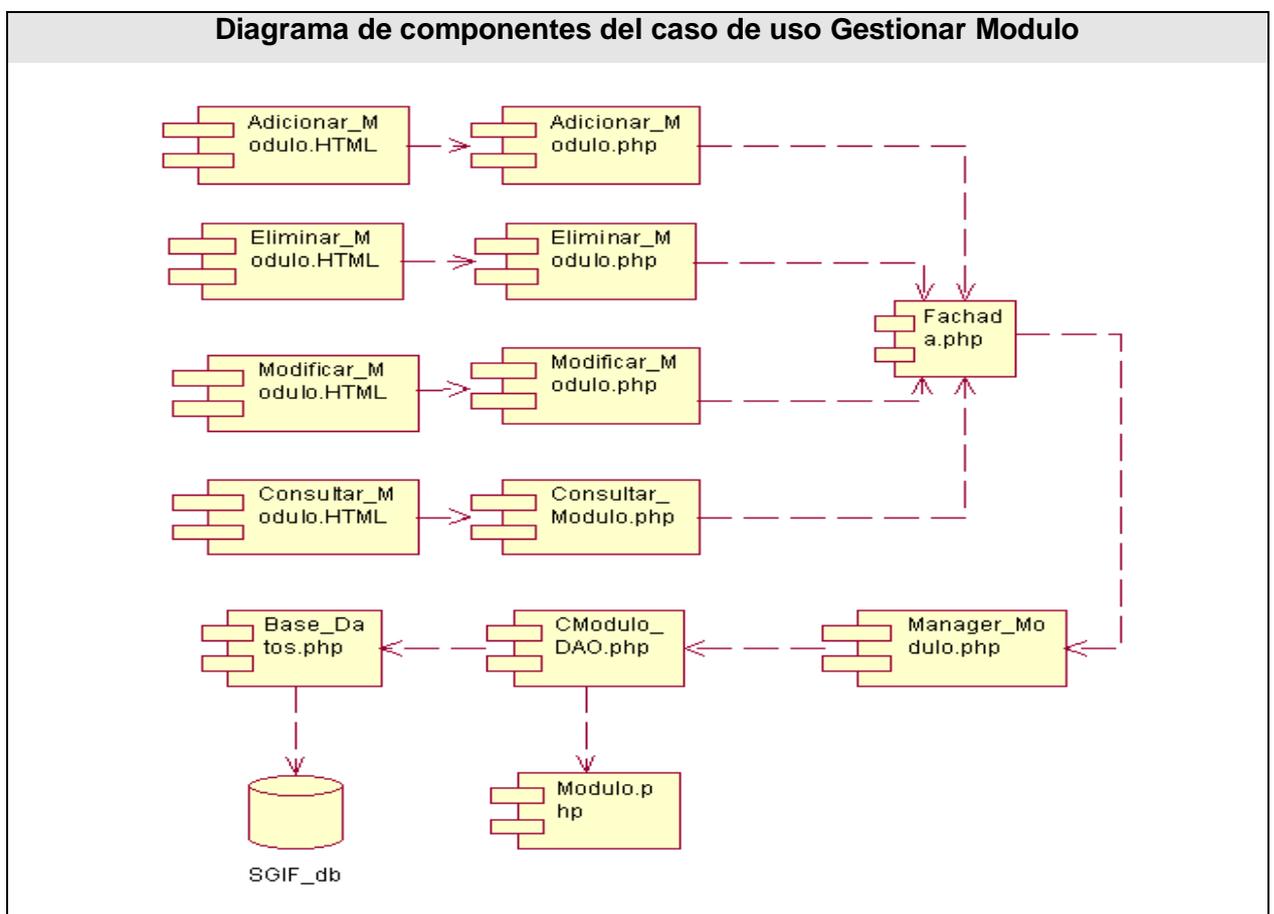


Diagrama de componentes del caso de uso “Gestionar Posta”

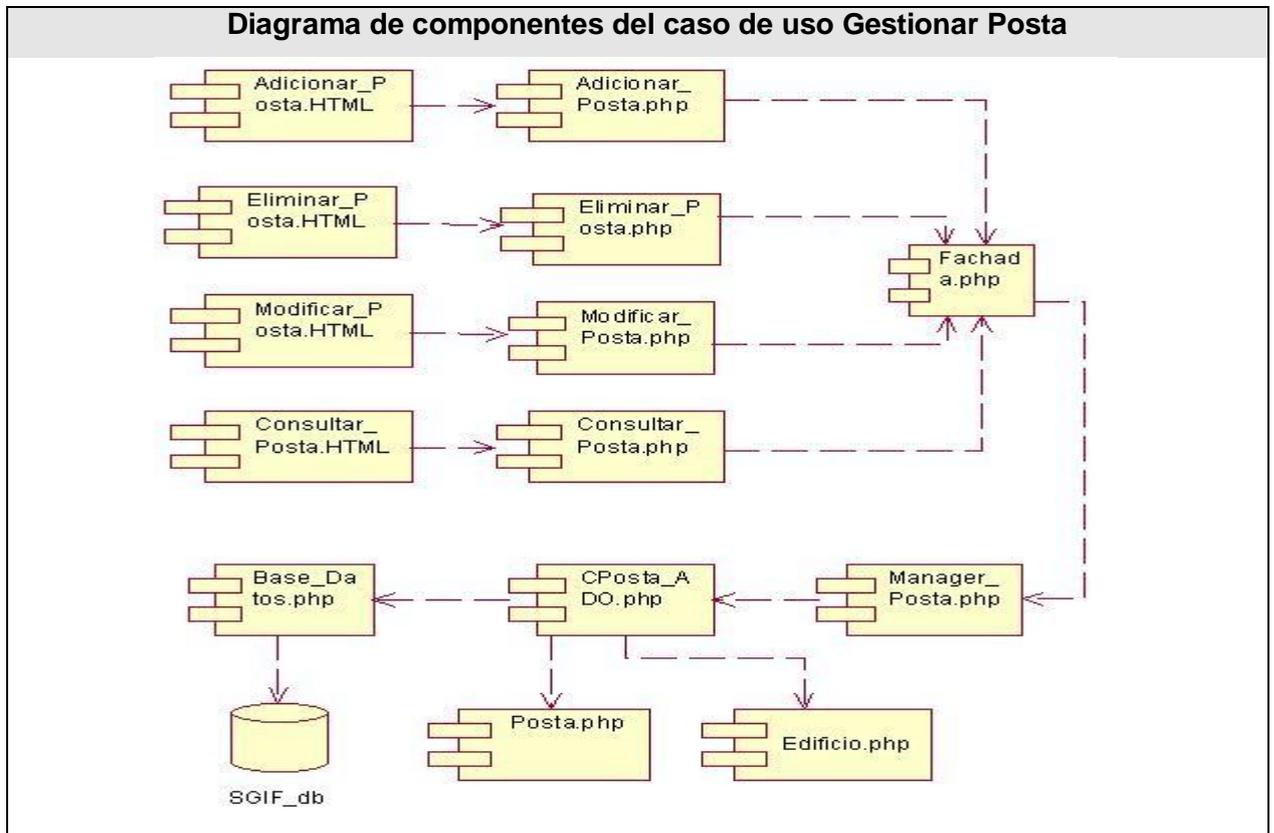


Diagrama de componentes del caso de uso “Gestionar Usuario”

