

Universidad de las Ciencias Informáticas

Facultad 7



**Título: Aplicación Web para el control de la Guardia
Estudiantil y la Cuartelería de la Facultad 7**

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autores: Karel Bejerano González

Félix Miguel Montes de Oca Barrios

Tutores: Ing. Yarielys Hernández Fonticiella

Ing. Daichel Bárzaga Rosales

Ciudad de La Habana, Junio del 2009

“Año del 50 aniversario del triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

<Nombre del estudiante 1>

<Nombre del estudiante 2>

<Nombre del tutor 1>

<Nombre del tutor 2>

DATOS DE CONTACTO:

TUTORES:

Ing. Yarielys Hernández Fonticiella: Graduado de Ingeniero en Ciencias Informáticas en el año 2008. Actualmente labora en la Universidad de Ciencias Informáticas (UCI), desempeñándose como profesora de la Facultad No. 7. Está vinculada a la producción y labora en el Área Temática: Sistemas Especializados en Medicina.

Ing. Daichel Rosales Bárzaga: Graduado de Ingeniero en Ciencias Informáticas en el año 2008. Actualmente labora en la Universidad de Ciencias Informáticas (UCI), desempeñándose como profesor de la Facultad No. 7 vinculado a la producción en la misma.

AGRADECIMIENTOS:

A nuestros tutores, a los cuales agradecemos las tantas horas de esfuerzo y sacrificio.

De Karel

Agradecer a mis amigos por su amistad y ayuda incondicional en todo momento.

A Duniesky Sarmiento, José Rolando, Jorge del Toro Rodríguez, Bolmey, Ketty Fraga, a Seguí, Yanisleidys Álvarez Chamosa, Irina Carralero, Yalinet Padilla, Elaine Cuza y Dannier Flores por su aporte y ayuda en la realización de esta Tesis.

A los amigos de la vieja Guardia, de actividades, de todos los años juntos en la carrera, a esos que corrieron conmigo cuando me accidenté, a los que me conocen de verdad mi más profundo y sincero agradecimiento por estar ahí.

A Julitín, a jessye, al Miki, al Yosme, Albertico, Yasser, Félix, Lesther, Alfredo, Juan Pablo, San Juan, Aramís, Luis José, Yoelkis, Rubén, Yovanoty, Segura, Daniel (el mango), Causse, a todos ustedes gracias por existir.

A mis amigas Lissandra Estupiñán, Je-lo, Lilianna, Clenda Pérez Bisset, Anavel, Lianys, Elis, Yainely, Yudita la bella, Lizandra Morgado, Yisel, Adita, Issis, Arianna Medrano.

A Claritza por estar siempre allí cuando más lo necesité, por su sencillez, por su paciencia, por todo lo que me brindó sin pedir nada a cambio.

DEDICATORIA:

Dedico esta tesis a mis padres y a mi amigo Manuel Alejandro Buch Meneces que aunque no esté entre nosotros, nunca me olvidaré de él.

RESUMEN

En el presente trabajo se realiza el diseño e implementación de nuevas funcionalidades para el sitio de la Facultad 7 de la Universidad de las Ciencias Informáticas, PortalFacultad7. Las mismas persiguen el objetivo de proporcionarle a la dirección de la facultad la posibilidad de tener registrada toda la información necesaria referente a los procesos de la Guardia estudiantil y Cuartelería de la Facultad 7. Lo que posibilitará realizar de forma más eficiente el manejo y gestión de la información de estas actividades.

Para el desarrollo de la aplicación se utilizaron la metodología RUP. Como lenguaje de modelado se utilizará: UML y como herramienta de modelado visual: Visual Paradim, así como lenguajes de Programación: PHP como lenguaje de diseño se utilizará Dreamweaver y como framework el Symfony.

También el sistema proporcionará que se pueda llevar un control personalizado de la evaluación de los estudiantes en la residencia estudiantil, generar un conjunto de Reportes, en formato digital, acerca de estas dos actividades, que pueden ser impresos, también va a facilitar la planificación y el control de la Guardia Estudiantil y la Cuartelería a los responsables de estas tareas.

INTRODUCCIÓN.....	1
1. CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	5
1.1. Guardia Estudiantil y Servicio de Cuartelaría a nivel nacional	5
1.2. Guardia Estudiantil y Servicio de Cuartelaría en la UCI.....	5
1.3. Sistemas de Gestión	6
1.4. Tendencias y Tecnologías actuales.....	7
1.5. Arquitectura	7
1.6. El patrón Modelo Vista Controlador (MVC)	7
1.7. Metodologías de desarrollo de software	8
1.8. Rational Unified Process (RUP)	8
1.9. UML: Lenguaje Unificado de Modelado	10
1.10. Lenguajes de Programación	12
1.11. Sistemas Gestores de Bases de Datos.....	14
1.12. PostgreSQL.....	14
1.13. Herramientas utilizadas.....	16
1.14. Visual Paradigm.....	17
1.15. Adobe Dreamweaver.....	18
1.16. Symfony Framework.....	19
1.17. Editor de Base de Datos. SQL Manager.....	21
1.18. WebServices.....	21
1.19. WSDL.....	21
1.20. Conclusiones	22
2. CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA.....	23
2.1. Objetivos de la organización	23
2.2. Flujo actual de los Procesos	23
2.3. Descripción de los procesos de negocio.....	24
2.4. Reglas de Negocio.....	25
2.5. Objeto de informatización.....	25
2.6. Modelo de Negocio.....	25
2.7. Propuesta del sistema	35
2.8. Especificación de los requerimientos de software.....	35
2.9. Actores del Sistema.....	40
2.10. Listado de Casos de Uso del Sistema.....	40
2.11. Diagrama de Casos de Uso del Sistema.....	43
2.12. Casos de Uso por Ciclos	44
2.13. Casos de Uso Expandidos	45
2.14. Conclusiones	60

3.	CAPÍTULO III: DISEÑO DEL SISTEMA.....	61
3.1.	Diseño.....	61
3.2.	Diagramas de interacción.....	71
3.3.	Conclusiones.....	78
4.	CAPÍTULO IV: IMPLEMENTACIÓN.....	79
4.1.	Diagrama de Despliegue.....	79
4.2.	Modelo de Implementación.....	80
4.3.	Conclusiones.....	82
	CONCLUSIONES.....	83
	RECOMENDACIONES.....	84
	BIBLIOGRAFÍA.....	86
	GLOSARIO DE TÉRMINOS.....	87
	ANEXOS.....	88

INTRODUCCIÓN

En la actualidad se enfrentan grandes desafíos para lograr un mayor desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC). Es importante destacar que el uso de las TIC favorece el trabajo individual y el trabajo en grupo. La experiencia demuestra que los medios informáticos permiten intercambiar ideas, discutir y decidir en común situaciones complejas que en nuestro país están dirigidas a todas las esferas de la sociedad.

Para lograr un desarrollo tecnológico avanzado en todas estas esferas, las estrategias de informatización deben centrarse en los problemas que afectan a la sociedad cubana, logrando elevar la calidad de vida de los ciudadanos. Desde el inicio de la Revolución en 1959, los medios de producción pertenecen al pueblo y por ello es un deber de cada ciudadano velar por su cuidado y protección y así, en conjunto, ayudar más a la economía del país

En la Universidad de las Ciencias Informáticas (UCI), la Guardia Estudiantil (GE) y la Cuartelería tienen una vital importancia desde sus inicios. Los estudiantes, profesores y trabajadores de la UCI son los dueños de los medios de producción con los que se trabaja, razón por la cual se deben proteger. Dichas tareas se realizan en todas las facultades de la universidad y en ellas intervienen estudiantes y profesores. La Facultad 7 de la UCI y en especial su Vicedecano de Residencia, tienen la misión de elaborar la distribución de los grupos y los estudiantes para estas tareas: horario de las mismas, edificios donde ellas se realizarán, dar las evaluaciones, así como la publicación de la información.

Actualmente en la facultad 7, el proceso de gestión de la información y planificación de la guardia y la Cuartelería se realiza de forma manual. Se introducen los datos respectivos a una plantilla en Excel que no presenta en muchas ocasiones, las funcionalidades necesarias, como mostrar reportes del mes anterior.

El proceso donde se brinda la información diaria sobre estas actividades en la facultad 7 y la planificación de ellas se está realizando de forma engorrosa y difícil para la persona que lo planifica, incluso se corre el riesgo de cometer errores. Además, los estudiantes no cuentan con un sistema de avisos a través del cual puedan informarse sobre la cantidad y regularidad de guardias y Cuartelería realizadas. Además los profesores guías y jefes de grupo tienen como visualizar las evaluaciones de los estudiantes en el

desempeño de ellas, para posteriores análisis en las asambleas de brigada u otros momentos en que se ameriten. Los involucrados en estos procesos en la facultad ya sea la secretaria, las instructoras de los edificios o el Vicedecano de residencia, no cuentan con una herramienta capaz de viabilizar el proceso de gestión de información y planificación de la Guardia OE y la Cuartelería.

Para solucionar estas dificultades, es necesario desarrollar una aplicación capaz de automatizar todas estas actividades y facilitar el trabajo hasta el punto donde se pueda contar con todo el control necesario, sobre estos procesos, permitiendo así un trabajo más eficiente sobre estas actividades

Por tanto el **problema científico** a resolver queda formulado de la siguiente forma:

¿Cómo viabilizar la gestión de información y planificación de la Guardia Estudiantil y la Cuartelería de la Facultad 7?

El **objeto de estudio** El Proceso de gestión de información y planificación de la Guardia Estudiantil y Cuartelería en la UCI.

El **campo de acción** sería: El Proceso de gestión de información y planificación de la Guardia Estudiantil y Cuartelería en la Facultad 7.

El **objetivo general** de esta investigación es: Desarrollar una aplicación web que viabilice la gestión de información y planificación de la Guardia Estudiantil y la Cuartelería de la Facultad 7.

Para el desarrollo del trabajo se proponen las siguientes **tareas investigativas**:

- Analizar el estado del arte de los sistemas informáticos existentes relacionados con la planificación de actividades.
- Realizar un análisis crítico de la selección de la tecnología, metodología de desarrollo de software, lenguaje de programación, gestor de base de datos y herramientas a utilizar.
- Realizar entrevistas a los especialistas encargados de la gestión de información y planificación de la Guardia Estudiantil y la Cuartelería de la Facultad 7.

- Modelar las condiciones actuales que rigen el proceso de gestión de información y planificación de la Guardia Estudiantil y la Cuartelería de la Facultad 7.
- Analizar las necesidades de funcionamiento del sistema informático, describiendo la Especificación de Requisitos del Software.
- Analizar la propuesta de arquitectura definida en el Documento de Arquitectura de la Facultad 7.
- Obtener el Modelo de Sistema de la aplicación web.
- Seleccionar los patrones de arquitectura a utilizar.
- Obtener el Modelo de Diseño de la aplicación web.
- Obtener el Modelo de Implementación y los artefactos necesarios que describan la Base de Datos de la aplicación.
- Realizar la implementación de la aplicación web.

El trabajo se estructura en cuatro capítulos, que se describen a continuación:

Capítulo 1. Fundamentación Teórica: En este capítulo se analizarán conceptos referentes a: sistemas de gestión, tecnologías actuales, metodologías de desarrollo de software a usar, entre otros. Además se analizará el proceso de gestión de información relacionado con la GE y la Cuartelería en las Universidades Cubanas y se seleccionan: metodología, herramienta y lenguaje de modelado a utilizar para el desarrollo del software.

Capítulo 2. Características del sistema: Se describen los procesos de negocio y casos de uso del negocio. Se realiza un análisis de los requisitos funcionales y no funcionales del sistema. Se tratan las causas que originan la situación problemática y los procesos involucrados en el campo de acción. Se realiza una estrategia, con el objetivo de implantar la arquitectura candidata en el proceso de gestión de información relacionado con la GE y la Cuartelería.

Capítulo 3. Análisis y diseño del sistema: Se realizan las descripciones de clases y métodos más importantes, se propone un diseño para la implementación así como la descripción de las tablas de la Base de Datos y el Diagrama de Entidad Relación correspondiente a dichas clases.

Capítulo 4. Implementación: Se explica la organización del sistema y se presenta el diseño de la Base de datos y el diagrama de despliegue que va a visualizar los nodos físicos necesarios para el correcto funcionamiento de la aplicación.

1. CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se aborda la necesidad de implementar un sistema para gestionar la GE y Cuartelería en la UCI. Se hace un breve análisis de las distintas técnicas de programación existentes. También se mencionan las tendencias y tecnologías actuales en el desarrollo de aplicaciones y finalmente se fundamentan los objetivos propuestos.

1.1. Guardia Estudiantil y Servicio de Cuartelería a nivel nacional

Actividad que debe de realizar todo trabajador y estudiante ya que cada uno es el responsable de los medios de producción que disponen para su trabajo por lo que tienen la tarea de protegerlos. Su evaluación influye en la caracterización e integralidad del trabajador. Estas tareas no deben ser violadas bajo ninguna razón.

En un estudio que fue realizado se llegó a la conclusión de que en Cuba no existe ninguna aplicación que controle estas actividades. Éste, se realizó visitando los sitios de otras universidades, tales como la de Ciego de Ávila (UNICA), Granma, entre otras. Así como entrevistas a trabajadores de varias empresas.

1.2. Guardia Estudiantil y Servicio de Cuartelería en la UCI

La Universidad de las Ciencias Informáticas (UCI) por su característica, donde la producción juega un papel decisivo para el desarrollo de del país, posee una gran cantidad de medios de producción para el desarrollo de sus actividades de importancia para la economía del país, esa es una de las razones más importantes para la realización de la Cuartelería y la GE.

En la UCI se cuenta con un sistema para el control de estas actividades en la Facultad (FAC) 8 el cual actualmente no está en funcionamiento. La UCI cuenta con 10 facultades, cada una de ellas realiza su servicio de Cuartelería y guardia según sus características. Existen diferencias en cuanto: la distribución de los estudiantes en sus apartamentos y la forma de impartir la docencia. En algunas facultades donde se imparten clases en bloque la docencia es inviolable y la Cuartelería ocupa un segundo plano. Sin embargo en las restantes facultades el día de la Cuartelería es primordial y el estudiante está justificado a

faltar al aula. La forma de planificar la Guardia Estudiantil en la UCI es semejante en todas las facultades, pero en la facultad 7 se tienen en cuenta los días que hay pruebas parciales para reajustarla. Estos estudiantes deberán realizar esta actividad en otro momento. Toda esta información fue recopilada con las entrevistas realizadas a los Vice-Decanos de Residencia de las facultades.

Por todo lo antes planteado, la aplicación existente en la facultad 8 no se ajusta a las necesidades propias de la facultad 7, así como a las particularidades que esta tiene para el cumplimiento de las actividades en cuestión. En la mayoría de los casos, en la UCI, se realiza de forma exitosa la planificación y posterior evaluación de la Cuartelería y la Guardia Estudiantil. Pero en otros, no se realiza de la mejor forma, algunas veces por falta de orientación o la ausencia de una herramienta que gestione los procesos.

1.3. Sistemas de Gestión

Un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización. (1)

Gestionar es alcanzar los propósitos a través de la acción coordinada de personas, mejorar la gestión significará aumentar progresivamente la efectividad, la equidad y la eficiencia de los procesos. La gestión se hace evidente mediante indicadores que reflejan el avance hacia las metas establecidas, incluye además la constante voluntad y apropiación de las personas involucradas en cuanto a buscar nuevas formas de organizar el trabajo en busca de mejoras para alcanzar propósitos con la menor cantidad de recursos, incluyendo el tiempo.

La implementación de un sistema de gestión eficaz puede ayudar a:

- Gestionar los riesgos sociales, medioambientales y financieros
- Mejorar la efectividad operativa
- Reducir costos
- Aumentar la satisfacción de clientes y partes interesadas
- Proteger la marca y la reputación
- Lograr mejoras continuas
- Potenciar la innovación
- Eliminar las barreras al comercio

- Aportar claridad al mercado (2)

1.4. Tendencias y Tecnologías actuales

En la actualidad la tecnología Web ha alcanzado un gran auge, por lo que las aplicaciones Web como parte de las Tics están cada vez más presentes en Internet, evolucionando, haciéndose más dinámicas y ajustables a las necesidades de sus usuarios, de allí que desarrollar una aplicación Web, es de gran provecho para cualquier institución que lo amerite.

1.5. Arquitectura

La Arquitectura de Software no es más que 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. La misma establece los fundamentos para que analistas, diseñadores, programadores, trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades.

1.6. El patrón Modelo Vista Controlador (MVC)

Este patrón está basado en un patrón clásico del diseño Web conocido como arquitectura MVC, que está formado por tres niveles:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página Web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica del negocio (el modelo) y la presentación (la vista), lográndose un mantenimiento sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se

encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

1.7. Metodologías de desarrollo de software

Las metodologías de desarrollo de software pueden estar dadas por diversas y particulares características, por ejemplo:

- Para que estén enfocadas al individuo y a las interacciones del equipo de desarrollo sobre el proceso y las herramientas
- Para que tengan cambios durante el proyecto y el cliente sea parte del equipo de desarrollo
- Para que los grupos de trabajo estén diseñados con poco personal, generen pocos artefactos, posean pocos roles y no hagan énfasis en la arquitectura del software o puedan estar basadas en normas provenientes de estándares, seguidas por el entorno de desarrollo, con cierta resistencia a los cambios.
- También pueden estar dadas porque el cliente interactúe con el equipo de desarrollo mediante reuniones, los grupos de trabajos sean grandes y distribuidos, el proceso sea mucho más controlado, con numerosas políticas o normas, generen gran cantidad de artefactos y roles y la arquitectura del software sea esencial.

1.8. Rational Unified Process (RUP)

El Proceso Racional Unificado (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es un refinamiento realizado por Rational Software del más genérico proceso unificado.

Fue creado por Ivar Jacobson, Grady Booch y James Rumbaugh en 1998, su objetivo es producir software de alta calidad, que cumpla con los requerimientos de los usuarios dentro de la planificación y presupuesto establecidos. RUP es un proceso de desarrollo de software y junto a UML constituye la metodología estándar más utilizada para el desarrollo de un producto de software. El ciclo de vida de RUP se caracteriza por:

Dirigido por casos de uso: el proceso de desarrollo sigue un hilo, avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso son un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante.

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.

Iterativo e incremental: en donde el trabajo se divide en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto.

RUP define cuatro elementos: *los roles*, que responden a la pregunta ¿Quién?, *las actividades* que responden a la pregunta ¿Cómo?, *los artefactos*, que responden a la pregunta ¿Qué? y *los flujos de trabajo* que responde a la pregunta ¿Cuándo?

Un rol define las responsabilidades de un individuo, o de un grupo de individuos de un equipo de trabajo. Una persona puede desempeñar diversos roles y un rol puede ser representado por varias personas. Los roles definidos por RUP son: analistas, desarrolladores, administradores, soporte y producción, especialistas de pruebas y roles adicionales.

Una actividad es una unidad de trabajo que una persona que desempeñe un rol puede realizar. Las actividades tienen un objetivo concreto, crear o actualizar algún producto.

Un artefacto es un fragmento de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los artefactos son los resultados tangibles del proyecto que se crean y se usan hasta obtener el producto final.

Un flujo de trabajo es una relación de actividades que producen resultados observables dado por una secuencia de actividades realizadas por los diferentes roles.

Además de esto esta metodología divide el proceso de desarrollo en ciclos, teniendo un producto al culminar cada ciclo, estos se dividen en fases que finalizan con un hito, donde se debe tomar una decisión importante, las fases son:

Inicio: Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.

Elaboración: Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen.

Construcción: Se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.

Transición: Se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

1.9. UML: Lenguaje Unificado de Modelado

UML es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. Este lenguaje capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. A su vez, pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

No es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos mediante ingeniería inversa a partir de programas existentes. Es un lenguaje de propósito general para el modelado orientado a objetos. Hace algunos años, existían diversos métodos y técnicas Orientadas a Objetos, con muchos aspectos en común pero utilizando distintas notaciones, se presentaban inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, etc., además de pugnas entre enfoques, lo que generó la creación del UML como estándar para el modelado de sistemas de software principalmente, pero con posibilidades de ser aplicado a todo tipo de proyectos.

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

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- Relaciones: relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones”. (3)

Tiene como objetivos fundamentales:

- Ser un lenguaje de modelado de propósito general que puede ser usado por todos los modeladores.
- No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- No pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. Incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son el encapsulamiento y componentes.
- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.

1.10. Lenguajes de Programación

Para la realización de este epígrafe se ha utilizado como referencia bibliográfica: (4)

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

Un lenguaje de programación permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Una característica relevante de los lenguajes de programación es precisamente que más de un programador puedan tener un conjunto común de instrucciones que puedan ser comprendidas entre ellos para realizar la construcción del programa de forma colaborativa.

PHP (Hypertext Preprocessor) (5)

El Hypertext Preprocessor, es un lenguaje de programación interpretado de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Una de sus características más potentes, es su soporte para gran cantidad de bases de datos, entre las que se pueden mencionar: InterBase, MSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. También ofrece la integración con varias bibliotecas externas, que brindan al desarrollador la posibilidad de realizar cualquier tarea, desde generar documentos en pdf (Portable Document Format) hasta analizar código XML (extensible Markup Language) y también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la librería GTK+. Es software libre, lo que implica menos costes y servidores más baratos que otras alternativas.

Es muy rápido y su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado. Es multiplataforma, funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server) de forma que el código que se haya creado para una de ellas, no tiene porqué modificarse al pasar a la otra. Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP (Active Server Pages).

Tiene una de las comunidades más grandes en Internet, con lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y más recursos. Ofrece una solución simple y universal para las paginaciones dinámicas de la Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día, a diferencia con el código de otros lenguajes.

Debido a su amplia popularidad está perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar sus capacidades. Su funcionamiento se puede describir a través de los pasos siguientes:

- Escribir en las páginas HTML pero con el código PHP dentro.
- Guardar la página en el servidor Web.
- Un navegador solicita una página al servidor.
- El servidor interpreta el código PHP.

En ningún caso se envía código PHP al navegador, por lo que todas las operaciones realizadas son transparentes al usuario, el código PHP es ejecutado en el servidor y el resultado es enviado al navegador. El resultado es normalmente una página HTML. Por lo que al usuario le parecerá que está visitando una página HTML que cualquier navegador puede interpretar.

Al ser un lenguaje que se ejecuta en el servidor, no es necesario que el navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. Además se encuentra libre en el mercado y se puede acceder a él por medio de Internet. Constituye una petición del cliente, el desarrollo del producto empleado el lenguaje de programación PHP, además, luego de analizar las características y ventajas que posee, expuestas anteriormente, se considera que es el más apropiado para la implementación del software que se pretende realizar.

1.11. Sistemas Gestores de Bases de Datos.

Una base de datos o banco de datos (en inglés: database) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Existen unos programas denominados Sistemas Gestores de bases de Datos, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

1.12. PostgreSQL.

Para la realización de este epígrafe se ha utilizado como referencia bibliográfica: (6)

PostgreSQL es un potente Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) de código abierto. Tiene más de 15 años de activo proceso de desarrollo a nivel mundial y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad, integridad de los datos, y la corrección. Es multiplataforma y funciona en los principales sistemas operativos, como Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. También apoya el almacenamiento de grandes objetos binarios, imágenes, sonidos o vídeo.

Características de PostgreSQL:

- Objeto-relacionales: PostgreSQL en cada tabla define una clase que permite implementar la herencia entre tablas o clases, funciones y operadores polimórficos.
- Implementación del estándar SQL92/SQL99.
- Código abierto: Debido a la licencia liberal, PostgreSQL puede ser usado, modificado y distribuido por todo el mundo de forma gratuita para cualquier fin, ya sea de datos, comerciales o académicas.
- Múltiples lenguajes de procedimientos: Los disparadores y otros procedimientos pueden ser escritos en varios lenguajes de procedimientos. Código del lado del servidor es comúnmente escrito en PL / PostgreSQL, un lenguaje de procedimiento similar al de Oracle PL / SQL. También se puede desarrollar código del lado del servidor en Tcl, Perl, incluso bash (el de código abierto Linux / Unix Shell).

- Múltiples-cliente API: PostgreSQL soporta el desarrollo de aplicaciones cliente en varios lenguajes, interfaz para PostgreSQL desde C, C ++, ODBC, Perl, PHP, Tcl / Tk, y Python.
- Tipos de datos: integer, string, numeric, boolean, char, varchar, date, interval, y timestamp, tipos geométrica, tipo de datos booleanos y tipos de datos diseñados específicamente para hacer frente a las direcciones de red.
- La extensibilidad: Es una de las características más importantes de Postgre SQL, éste, puede ser ampliado, se pueden añadir nuevos tipos de datos, nuevas funciones y operadores, e incluso nuevos lenguajes de procedimiento y de cliente.
- Tiene un soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados.
- MVCC, o Control de Concurrencia Multi-Versión (Multi-Versión Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios, es decir, mientras un proceso escribe no se bloquean las tablas, ni siquiera las tuplas. Mediante el uso de MVCC, PostgreSQL evita el problema de que procesos lectores esperen a que se termine de escribir por completo. MVCC está considerado mejor que el bloqueo a nivel de tupla porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.
- Integridad Referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

Todo ello unido a las ventajas con que cuenta el Gestor como son: Instalación ilimitada, mejor soporte que los proveedores comerciales, ahorros considerables en costos de operación Estabilidad y confiabilidad legendarias, extensible multiplataforma, diseñado para ambientes de alto volumen, herramientas gráficas de diseño y administración de bases de datos, hacen de él uno de los mejores, más completo y óptimo gestores de BD con el que se pueda desarrollar un software. Por lo que se decidió usarlo para el desarrollo del sistema en propuesta.

1.13. Herramientas utilizadas.

Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software.

CASE es también definido como el Conjunto de métodos, utilidades y técnicas que facilitan el mejoramiento del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. (7)

Para llevar a cabo el sistema se hace indispensable tener en cuenta la utilización de algunas herramientas necesarias en el diseño de interfaz, el trabajo con las imágenes y el modelado del proyecto.

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son las aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas contribuyen de manera directa en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. Cada una de estas herramientas persigue nueve Objetivos principales:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar, desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.

1.14. Visual Paradigm.

La herramienta CASE seleccionada en el proyecto para realizar el modelado fue Visual Paradigm for UML 6.0 Enterprise Edition que utiliza el Lenguaje Unificado de Modelado (UML) como lenguaje de modelado. Está diseñada para distintos usuarios entre los que se incluyen ingenieros de software, analistas de sistemas, analistas de negocios, arquitectos y desarrolladores. Está orientada a la creación de diseños y se usa el paradigma de programación orientada a objetos. Visual Paradigm (VP) incluye una herramienta llamada Visual Architect que permite la generación de código para el manejo de la base de datos. Con esta herramienta se puede generar código para los lenguajes PHP, JAVA y C# y para los gestores de base de datos DB2, Informix, SQL Server, MySQL, Oracle y PostgreSQL.

Beneficios de **Visual Paradigm** para UML:

➤ **Persistencia de forma fácil.**

Los desarrolladores emplean mucho esfuerzo en salvar y cargar objetos entre la memoria y la base de datos lo que hace que el programa sea complicado y difícil de mantener. VP simplifica estas tareas mediante la generación de una capa de persistencia entre objeto y modelos de datos.

➤ **Generador de mapeo objeto-relacional sofisticado.**

La capa de mapeo objeto-relacional que se genera incorpora características como soporte de transacciones, capaz de conectar en caché, agrupación de conexiones y personalización de sentencias SQL.

➤ **Amplia cobertura para bases de datos.**

Soporta una amplia gama de base de datos, incluidos Oracle, DB2, Cloudscape / Derby, Sybase Adaptive Server Enterprise, Sybase SQL Anywhere, Microsoft SQL Server, PostgreSQL, MySQL y otros.

➤ **Base de datos de ingeniería inversa.**

Permite la ingeniería inversa existente en una base de datos a través JDBC en el modelo entidad-relación. Los desarrolladores pueden transformar el modelo entidad-relación al modelo de objetos y rediseñar la base de datos para un mayor desarrollo.

➤ Integración con IDE.

VP-UML no sólo es una aplicación independiente, se puede integrar a los principales Integrated Development Environments (IDEs): Eclipse/WebSphere®, Borland JBuilder®, NetBeans/Sun™ ONE, IntelliJ IDEA™ Oracle JDeveloper, BEA WebLogic Workshop™ y Visual Studio .NET.

Es una herramienta diseñada para desarrollar software que utiliza la Programación Orientada a Objetos. Busca reducir la duración de un ciclo de desarrollo brindando ayuda a arquitectos, analistas, diseñadores y desarrolladores.

Presenta varios beneficios entre los que se destacan:

- Navegación intuitiva entre el modelo visual y el código.
- Poderosa herramienta de generación de PDF/HTML a partir de diagramas UML.
- Sincronización entre el código fuente y el modelo en tiempo real.
- Soporte para toda la notación UML.
- Importa Rational Rose Project.
- Soporta UML 2.0.
- Presenta un diseño basado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Ofrece capacidades de ingeniería directa e inversa.
- Se integra al IDE de Eclipse

Por las anteriores características mencionadas, adicionadas a las ventajas que presenta en el diseño de sistemas y a la agilidad que proporciona en la modelación, se determinó emplear Visual Paradigm en la posterior propuesta de solución.

1.15. Adobe Dreamweaver.

Es una aplicación en forma de estudio, basada en la forma de estudio de Adobe Flash pero con más parecido a un taller destinado para la construcción y edición de sitios y aplicaciones Web basados en

estándares. Creado inicialmente por Macromedia (actualmente producido por Adobe Systems). Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas. (8)

La gran base de este editor sobre otros es su gran poder de ampliación y personalización del mismo, puesto que en este programa, sus rutinas (como la de insertar un hipervínculo, una imagen o añadir un comportamiento) están hechas en Javascript-C, lo que le ofrece una gran flexibilidad en estas materias. Esto hace que los archivos del programa no sean instrucciones de C++ sino, rutinas de Javascript que hace que sea un programa muy fluido, que todo ello hace, que programadores y editores web hagan extensiones para su programa y lo ponga a su gusto. (9)

Dreamweaver permite al usuario utilizar la mayoría de los navegadores Web instalados en su ordenador para pre visualizar las páginas web. También dispone de herramientas de administración de sitios dirigidas a principiantes como, por ejemplo, la habilidad de encontrar y reemplazar líneas de texto y código por cualquier tipo de parámetro especificado, hasta el sitio web completo.

Un aspecto de alta consideración de Dreamweaver es su arquitectura extensible, él, permite el uso de "Extensiones" y ha tenido un gran éxito desde finales de los 90 y actualmente mantiene el 90% del mercado de editores HTML por lo que ha permitido que Dreamweaver gane el apoyo de una gran comunidad de desarrolladores de extensiones que hacen posible la disponibilidad de extensiones gratuitas y de pago para la mayoría de las tareas de desarrollo Web.

1.16. **Symfony Framework.**

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener.

Es un completo framework diseñado para optimizar, de acuerdo a sus características, el desarrollo de las aplicaciones Web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la

aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel, es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. (10)

“Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por Terceros.” (11)

Por todo lo anteriormente planteado y además de ser un requisito en el documento de Arquitectura propuesto por la facultad es que se decidió usar dicho Framework para la implementación del sistema en cuestión.

1.17. Editor de Base de Datos. SQL Manager

En la actualidad, las bases de datos creadas bajo lenguaje SQL o afines, se han convertido en parte de toda creación de utilidades y sitios de Internet. Además, existen numerosas herramientas que se relacionan directamente con servidores de bases de datos.

EMS SQL Manager for PostgreSQL Lite es un claro ejemplo, ya que esta aplicación de orden avanzado, ofrece la oportunidad de administrar completamente los servidores basados en lenguaje Postgre SQL.

Este software cuenta con un funcionamiento verdaderamente sólido y comprensible si el usuario en cuestión, conoce al pie de la letra el funcionamiento y dinámica de trabajo del lenguaje nombrado anteriormente. EMS SQL Manager for PostgreSQL Lite posee una interfaz gráfica realmente intuitiva y bastante similar al clásico explorador de Windows, solo que ciertamente, cuenta con muchas más prestaciones.

1.18. WebServices.

Los servicios Web XML permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientemente de cómo se hayan creado las aplicaciones, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles los dispositivos utilizados para obtener acceso a ellas. Aunque los servicios Web XML son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada

1.19. WSDL

WSDL es el acrónimo de Web Services Description Language (lenguaje de descripción de servicios Web). La notación que utiliza un archivo WSDL para describir formatos de mensaje se basa en el estándar XML. Esto significa que es un idioma de programación neutral y basado en estándares, lo cual es perfecto para describir interfaces de servicios XML Web Services que pueden abrirse desde una gran variedad de plataformas y lenguajes de programación. Además de describir el contenido de un mensaje, WSDL define el lugar en el que está disponible el servicio y qué protocolo de comunicaciones se utiliza para hablar al servicio. Esto significa que el archivo WSDL define todos los elementos necesarios para escribir un

programa que pueda funcionar con un servicio XML Web. Existen varias herramientas disponibles para leer un archivo WSDL y generar el código necesario para establecer comunicación con un servicio XML Web

1.20. Conclusiones

Con desarrollo del capítulo se llegó a la conclusión de que no existe una manera de trabajo que estandarice la forma en que se debe proceder con la planificación de la GE y el servicio de Cuarterería, ni como dar un reporte de las evaluaciones de los grupos y los estudiantes en la UCI. Por lo que es preciso entonces realizar una aplicación que viabilice esta gestión y cumpla con las necesidades planteadas. Para ello se utiliza la metodología RUP. Como lenguaje de modelado se utiliza: UML y como herramienta de modelado visual: Visual Paradim. Como lenguaje de Programación: PHP, como framework el Symfony y como lenguaje de diseño se utiliza Dreamweaver.

2. CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se enuncian y describen los procesos de negocio, se exponen las reglas que estos deben cumplir, se dan a conocer cuáles son los procesos a automatizar, además de describirse los actores y trabajadores que intervienen en el mismo; así como la realización de los diagramas de casos de uso del negocio, diagramas de actividades y el modelo de objeto.

También se identifican cuáles son los requisitos funcionales y no funcionales con los que debe cumplir la aplicación. Se muestra además una descripción detallada de cada uno de los casos de uso del sistema que proporcionará que se pueda llevar un control personalizado de la evaluación de los estudiantes en la residencia estudiantil, generar un conjunto de Reportes, en formato digital, acerca de estas dos actividades, que pueden ser impresos, también va a facilitar la planificación y el control de la Guardia Estudiantil y la Cuartelería a los responsables de estas tareas.

2.1. Objetivos de la organización

La Guardia Estudiantil y servicio de Cuartelería de la Facultad 7 se enmarca dentro de las doctrinas de la Guardia Estudiantil y Servicio de Cuartelería de la UCI y tiene como misión, velar por el cuidado y protección de todos los medios de producción, así como de aquellos medios con que cuenta la FAC para el desenvolvimiento y desarrollo pleno de la vida universitaria de sus estudiantes y trabajadores.

2.2. Flujo actual de los Procesos

En la Universidad de la Ciencias informáticas diariamente se realiza la Guardia Estudiantil y la Cuartelería, garantizando así, tanto la protección de todos los medios básicos de la universidad, como el buen funcionamiento y desarrollo de la vida universitaria en general. Dentro de los procesos fundamentales que han sido identificados y que se enmarca dentro del problema se encuentran:

- Planificar Guardia.
- Planificar Cuartelería
- Evaluar estudiante en la Cuartelería
- Obtener evaluación de la Cuartelería

A continuación se detallan cada uno de los procesos expuestos anteriormente.

Con el objetivo de **Planificar la Guardia** que se realiza en la residencia Estudiantil, el Vicedecano orienta a su Secretaria que confeccione la Distribución de la GE de la Facultad. Utilizando, para realizar la actividad el Listado de Grupos de la Facultad. Una vez acabada dicha distribución se le informa al Vicedecano de Residencia

Para **Planificar Cuartelerías** el Vicedecano de Residencia orienta a la Instructora que confeccione la Distribución de la Cuartelería de su(s) edificio(s). Utilizando para realizar la actividad el Listado de Estudiantes por Edificio. Una vez acabada dicha distribución se le informa al estudiante.

En cuanto a **Evaluar estudiante en la Cuartelería** es un proceso que puede ser comenzado tanto por el estudiante como por la Instructora de edificio. Si es comenzado por el estudiante, éste, le pide su evaluación de Cuartelería y la instructora se la proporciona y le pide que firme su evaluación. Si el caso de uso es comenzado por la instructora, esta da la evaluación al estudiante y le pide a este que firme dicha evaluación.

Con el propósito de **Obtener evaluación en la Cuartelería** el usuario pide la evaluación a la instructora y esta le da la evaluación del estudiante.

2.3. Descripción de los procesos de negocio

A continuación se muestra una descripción de los procesos de negocio por separado, que contribuirá a un mejor entendimiento de los mismos.

2.3.1 Cuartelería.

La Cuartelería es una actividad que es realizada por los estudiantes, de lunes a sábado. Se encuentra en cada paso de escalera de su respectivo edificio un estudiante de cuartelero. Ellos tienen la responsabilidad de tener un control de todo el personal ajeno que accede al edificio, además deben mantener la limpieza del paso de escalera y las áreas de los alrededores del edificio. Diariamente se emite una evaluación (B,

R, M) de cada una de las cuartelarias realizadas, contribuyendo así a la evaluación integral de cada estudiante.

2.3.2 Guardia estudiantil.

En la facultad existe una planificación para la guardia estudiantil, donde se refleja el grupo de estudiantes (brigada) y el profesor responsable, además del día en que le corresponde realizar la guardia. Una vez realizada la guardia se firma una hoja con los datos pertenecientes a la misma (hora de entrada y salida, posta a realizar la guardia, estudiantes que asistieron, profesor encargado de controlar dicha guardia), así como otros datos de interés.

2.4. Reglas de Negocio.

- La guardia estudiantil es realizada por una brigada solamente en un día.
- Un estudiante realiza la guardia en una sola posta ese día.
- Un solo estudiante está de cuartelero en un paso de escalera.

2.5. Objeto de informatización.

Al analizar todo el trabajo se ha determinado el objeto a informatizar, que será el diseño de una aplicación web que viabilice el proceso de gestión de información y planificación de la Guardia Estudiantil y la Cuartelería de la Facultad 7.

Se pretende que el sistema gestione la información y planificación, o sea, que brinde la posibilidad de Insertar, Listar y Modificar listados de brigadas para realizar la guardia. También permitirá gestionar los listados de los estudiantes por edificio para la confección de la Cuartelería, permitiendo: Insertar, Buscar, Modificar y Visualizar los mismos.

2.6. Modelo de Negocio.

El primer flujo de trabajo que propone RUP para el desarrollo de un sistema informático es el Modelado del negocio. El cual tiene como objetivos fundamentales:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización.

Para lograr esos objetivos el proceso de Modelado posibilita obtener una visión de la organización, que permite definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos.

2.6.1 Actores del Negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Actores del Negocio	Descripciones
Vicedecano de Residencia	Es uno de los directivos de la Facultad que solicita informaciones y orienta actividades relativas a los procesos que se realizan en la residencia estudiantil.
Estudiante	El estudiante es el encargado de realizar la GE y la Cuartelería, así como de solicitar y firmar su evaluación.
Profesor	El profesor puede requerir la evaluación de sus estudiantes para otras actividades

Tabla2.1.Actores del Negocio

2.6.2 Trabajadores del negocio.

Un trabajador del negocio representa un rol desempeñado en las realizaciones de CUN, éste colabora con otros trabajadores, es notificado de los eventos del negocio y manipula las entidades del negocio para realizar sus responsabilidades.

Trabajadores del Negocio	Descripciones
Instructora	Es la encargada planificar con el Listado de Estudiantes por Edificio (Listado Est_Edif) la Distribución de la Cuartelería y dar las evaluaciones a los estudiantes en sus respectivos edificios
Secretaria	Es la encargada de planificar la Distribución de la Guardia Estudiantil con el Listado de Grupos por Facultad (Listado Grupo_Fac)).

Tabla 2.2. Trabajadores del Negocio

2.6.3 Diagrama de Casos de Uso del Negocio.

Un diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio.

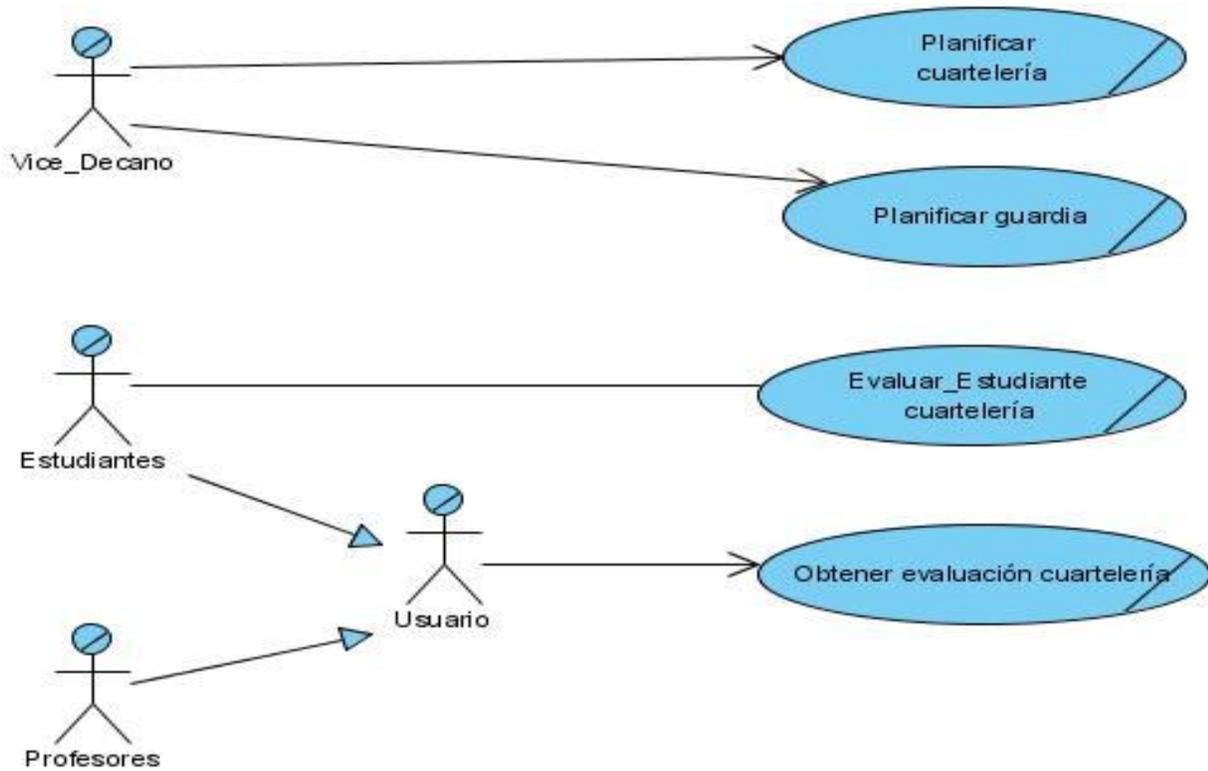


Figura 2.1. Diagrama de casos de uso del negocio.

2.6.4 Especificación de los Casos de Uso.

Caso de Uso “Planificación de Cuartelería”

Caso de uso:	“ Planificación de Cuartelería”
Actores:	Vicedecano de Residencia
Trabajadores:	Instructora de Edificio
Resumen:	El caso de uso se inicia cuando el Vicedecano de Residencia orienta a la Instructora que confeccione la Distribución de la Cuartelería de su(s) edificio(s). Utilizando para realizar la actividad el Listado de Estudiantes por Edificio. Una vez acabada dicha distribución se le informa al estudiante.

Tabla 2.3. Caso de Uso “Planificación de Cuartelería”.

Caso de Uso “Planificación de la Guardia Estudiantil (GE)”

Caso de uso:	“ Planificación de la Guardia Estudiantil (GE) ”
Actores:	Vicedecano de Residencia
Trabajadores:	Secretaria del Vicedecano
Resumen:	El caso de uso se inicia cuando el Vicedecano de Residencia orienta a su Secretaria que confeccione la Distribución de la GE de la Facultad. Utilizando para realizar la actividad el Listado de Grupos de la Facultad. Una vez acabada dicha distribución se informa al Vicedecano de Residencia y este a su vez a los estudiantes.

Tabla 2.4. Caso de Uso “Planificación de la Guardia Estudiantil (GE)”.

Caso de Uso “Evaluar_Estud_Cuarteleria”

Caso de uso:	“Evaluar Estud_Cuarteleria”
Actores:	Estudiante
Trabajadores:	Instructora de Edificio
Resumen:	Si es comenzado por el estudiante este le pide su evaluación de Cuartería y la instructora se la facilita y le pide que firme su evaluación. Si el caso de uso es comenzado por la instructora esta da la evaluación al estudiante y le pide a este que firme dicha evaluación.

Tabla 2.5. Caso de Uso “Evaluar_Estud_Cuarteleria”.

Caso de Uso “Obtener Evaluación de Cuartería”

Caso de uso:	“Obtener Evaluación de Cuartería”
Actores:	Usuarios (Estudiantes y Profesores)
Trabajadores:	Instructora de Edificio
Resumen:	El caso de uso comienza cuando el usuario pide la evaluación a la instructora y esta le da la evaluación del estudiante.

Tabla 2.6. Caso de Uso “Obtener Evaluación de Cuartería”.

2.6.5 Diagrama de Actividades

Los casos de uso del negocio consisten en la descripción de la secuencias de actividades que, en conjunto, producen algo observable para el actor del negocio. El proceso consiste en un flujo básico de una o más alternativas. La estructura del mismo se describe gráficamente con la ayuda de un diagrama de actividad.

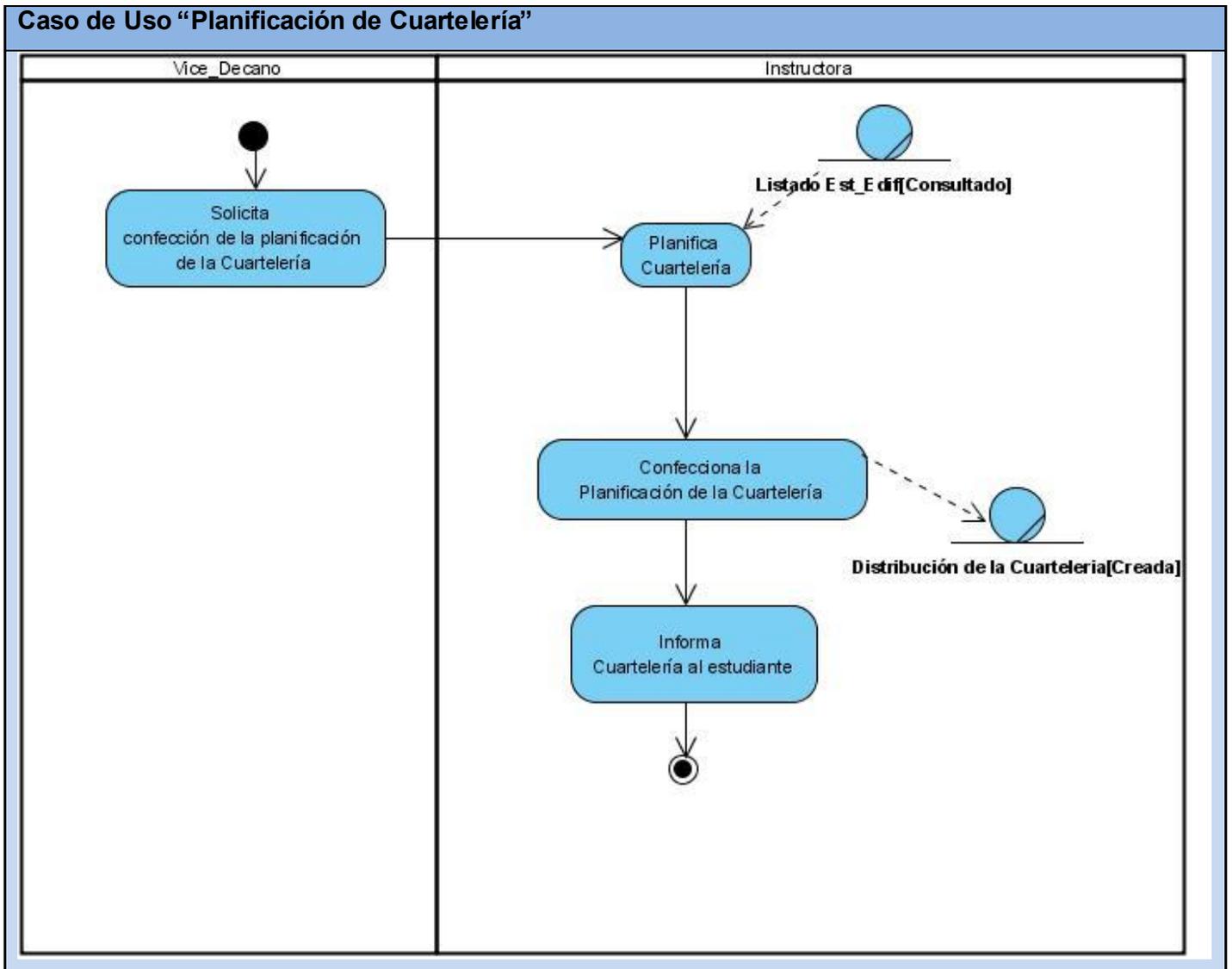


Figura 2.2. Caso de Uso "Planificación de Cuartelería".

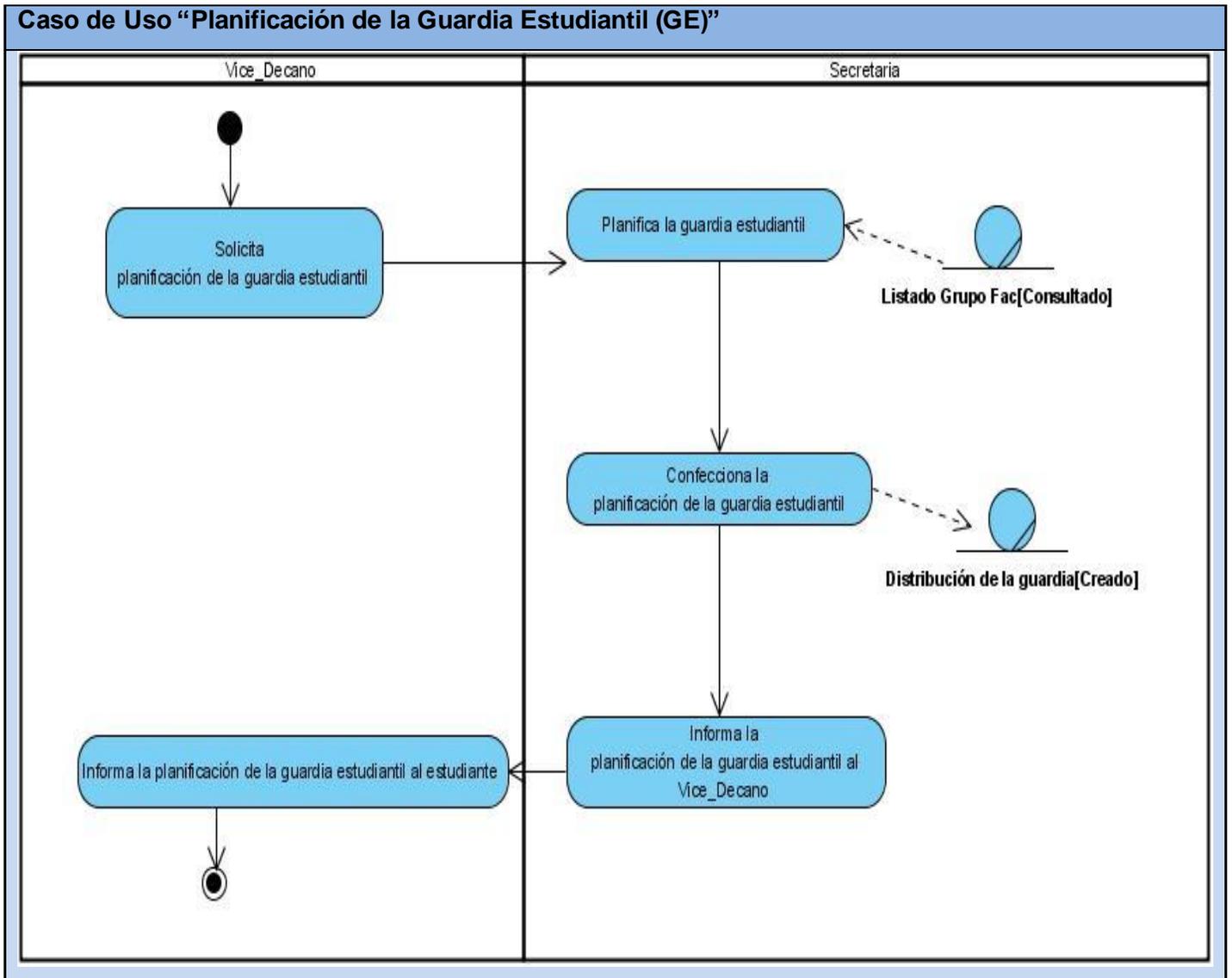


Figura 2.3. Caso de Uso “Planificación de la Guardia Estudiantil (GE)”.

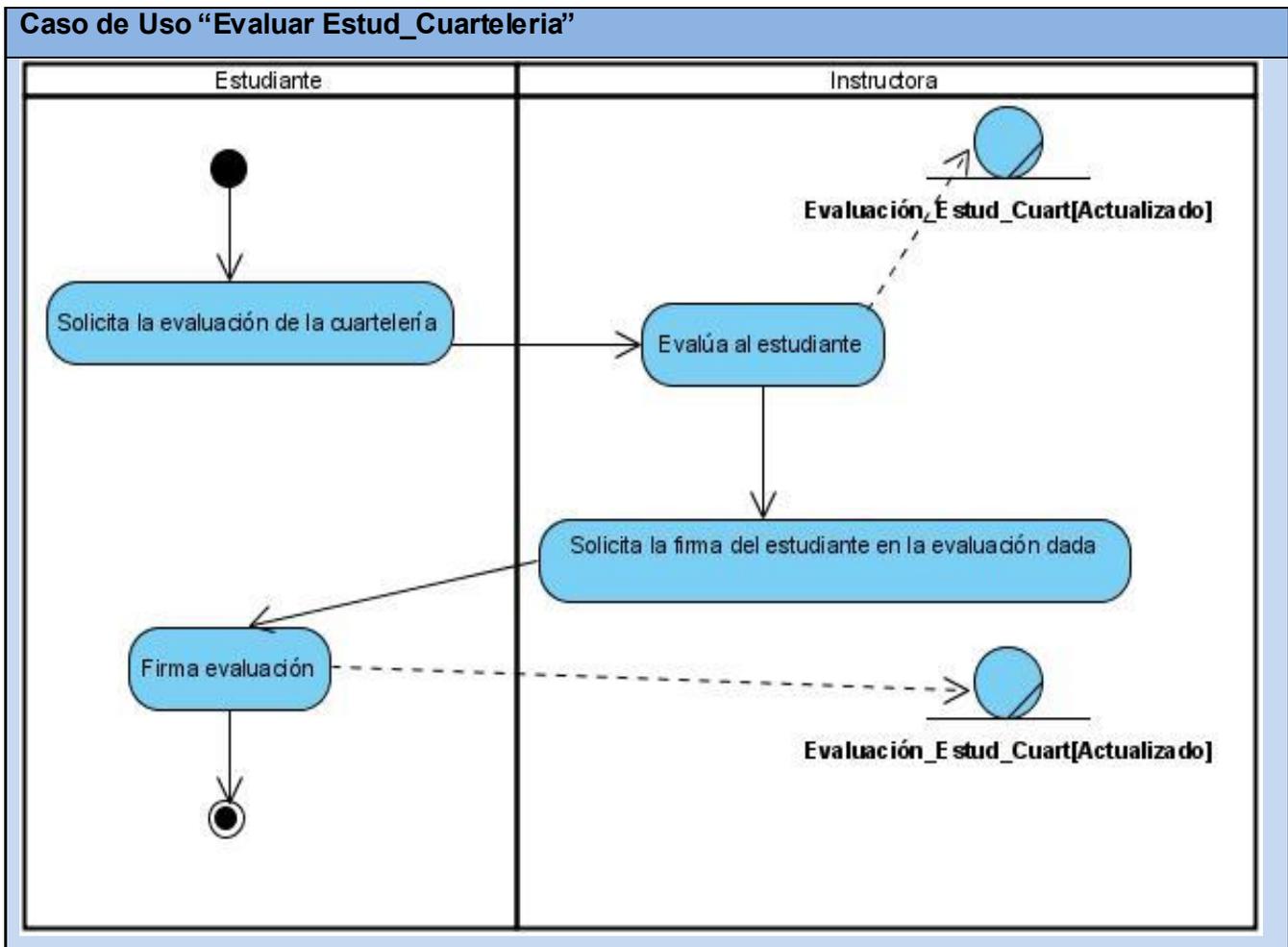


Figura 2.4. Caso de Uso "Evaluar Estud_Cuarteleria".

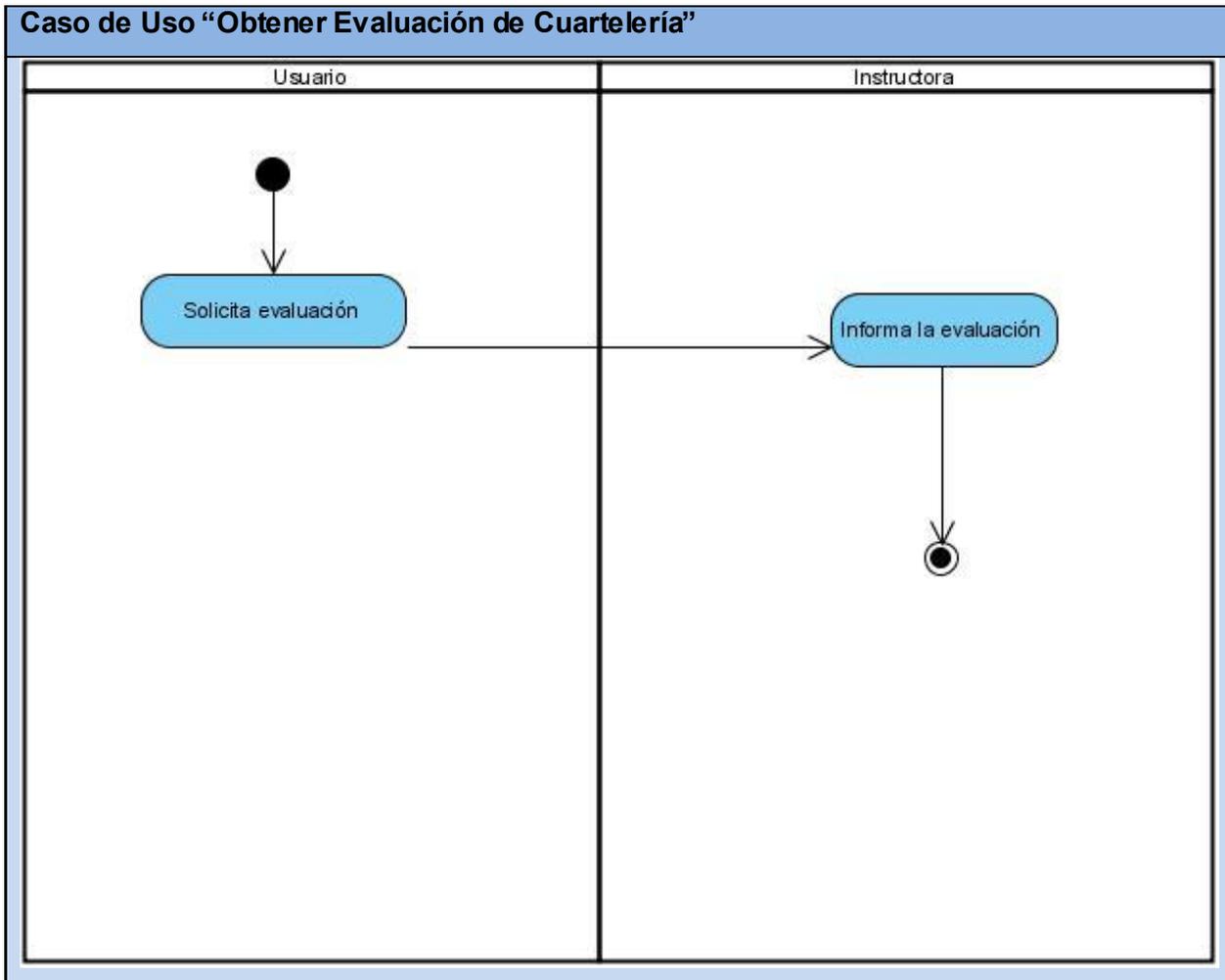


Figura 2.5. Caso de Uso "Obtener Evaluación de Cuartelería"

2.7. Propuesta del sistema

A continuación se hará una descripción breve de las principales características del sistema que se va a desarrollar, lo cual podrá ser constatado a través del listado de los requisitos del software.

2.8. Especificación de los requerimientos de software

La Especificación de Requisitos de Software (ERS) es una descripción completa del comportamiento del sistema que se va a desarrollar, es decir son especificaciones de las características, cualidades, capacidades y comportamiento de un sistema, descritos de manera entendible para el equipo de desarrollo, clientes y usuarios y en función de la satisfacción de sus necesidades. Los casos de uso también son conocidos como requisitos funcionales. Además de los casos de uso, la ERS también contiene requisitos no funcionales (o complementarios).

2.8.1 Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. A partir del estudio y las investigaciones realizadas se obtuvieron los siguientes requerimientos funcionales que ha de cumplir el sistema, con vista a la satisfacción final del cliente:

➤ **RF 1 Autenticar Usuario.**

- 1.1. Autenticar usuario.
- 1.2. Cerrar sesión.

➤ **RF 2 Gestionar Estudiantes.**

- 2.1 Insertar estudiantes.
- 2.2 Modificar estudiante.
- 2.3 Eliminar Estudiante.

2.4 Buscar Estudiante.

➤ **RF 3 Gestionar Cuartelerías.**

3.1 Insertar Cuartelería.

3.2 Modificar Cuartelería.

➤ **RF 4 Gestionar Guardia Estudiantil.**

4.1 Insertar Guardia.

4.2 Modificar Guardia.

4.3 Buscar Guardia.

➤ **RF 5 Gestionar la asignación de roles.**

5.1 Asignar roles

5.2 Eliminar asignación de roles

➤ **RF 6 Mostrar Reportes**

6.1 Planificación Mensual de la Guardia.

6.2 Cuartelería Semanal por Edificio

6.3 Evaluación de los Estudiantes en la Cuartelería por Grupo.

2.8.2 Requisitos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son esas características que posibilitan que el producto sea atractivo, usable, rápido, confiable, etc. Normalmente

están vinculados a requerimientos funcionales, es decir una vez que se conozca lo que el sistema debe hacer, se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

➤ **Requerimientos de Software:**

. **RNF 1:** Navegador Internet Explorer v6.0 o superior, Mozilla Firefox v2.x.

. **RNF 2:** Servidor Web Apache v2.x o superior con módulo PHP 5.2.3 disponible, este debe estar configurado con las extensiones, “soap”, “ldap”.

. **RNF 3:** Como Sistema Gestor de Base de Datos: PostgreSQL

➤ **Requerimientos de Hardware:**

RNF 4: Para los servidores tanto Web como SGBD: PENTIUM IV o superior con 256 MB de RAM o más.

RNF 5: Capacidad de disco duro, preferiblemente mayor a los 10 GB.

➤ **Restricciones para el diseño e implementación:**

RNF 6: Utilizar como lenguaje del lado del servidor al PHP v5.2.3 o superior y del lado del cliente: HTML y Java Script.

RNF 7: Para la programación en PHP se recomienda el IDE: Zend Studio, con la ayuda del Framework Symfony.

RNF 8: El código deberá ser reutilizable.

➤ **Requerimientos de Apariencia o interfaz externa:**

RNF 9: Estará diseñado para resolución de 1024x768, aunque deberá verse en 800x600 o cualquier resolución superior a esta.

RNF 10: La interfaz debe ser agradable para el usuario, que combine correctamente los colores, tipo de letra y tamaño, que los iconos estén en correspondencia con lo que representan no debe contener muchas imágenes que demoren las respuestas al usuario.

➤ **Requerimientos de Seguridad:**

RNF 11: Se podrá acceder al sistema solamente después de autenticarse.

RNF 12: Autorización (Atribución a los usuarios respecto a sus funciones de trabajo.)

RNF 13: Contar con un sistema de permisos y usuarios para el acceso a la información. .

RNF 14: El sistema debe estar disponible las 24 horas del día (disponibilidad).

➤ **Requerimientos de Usabilidad:**

RNF 15: El sistema debe permitir el acceso concurrente de diferentes usuarios, en dependencia del nivel de usabilidad que este definida para cada uno.

➤ **Requerimientos de Rendimiento:**

RNF 16: El sistema deberá ser capaz de gestionar toda la información y dar respuesta a las solicitudes lo más rápido posible.

➤ **Requerimientos Políticos Culturales:**

RNF 18: El sistema solo podrá ser utilizado en territorio cubano.

RNF 19: El sistema debe tener una interfaz que esté acorde con el lugar donde se implantará, es decir, que refleje los ideales de la organización y que sea amena, cumpliendo con las pautas de diseño.

RNF 20: El producto no debe contener palabras en otros idiomas.

RNF 21: El producto debe respetar los términos empleados normalmente por los especialistas en el tema de la esfera que se automatiza.

➤ **Requerimientos Legales:**

RNF 22: La mayoría de las herramientas de desarrollo son libres y sus licencias están avaladas.

RNF 23: Reconocido y autorizado por instancias superiores tales como la dirección de producción de la UCI (facultad 7).

RNF 24: Documentación legal de uso como Declaración de Autoría.

➤ **Requerimientos de Confiabilidad:**

RNF 25: Los reportes que se obtendrán deben ser 100% precisos y reales.

RNF 26: Deben establecerse los mecanismos necesarios para el restablecimiento del sistema ante fallos de comunicación u otros.

RNF 27: Garantiza un control estricto sobre el tráfico de información.

RNF 28: Chequeo constante de la integridad y consistencia en los datos.

Ya identificados y definidos los requisitos tanto funcionales como no funcionales que debe cumplir el sistema y haciendo uso de ellos se comienza la etapa del modelado del sistema.

2.9. Actores del Sistema

Actores	Descripción
Usuarios	Persona que va a tener algún rol en el sistema, es decir, los que van a poder trabajar con la aplicación.
Instructora de edificio	Es la encargada de cuando ya los estudiantes estén vinculados a los respectivos edificios donde van a vivir, realizar la distribución de la Cuartelería. Además de dar a los estudiantes la evaluación de la misma
Secretaria del Vicedecano	Es la encargada de crear la distribución de la Guardia Estudiantil (GE).
Administrador	Es el encargado de crear los edificios de la residencia de la facultad. Una vez creados estos edificios, ubicar a los estudiantes por áreas temáticas en su respectivo edificio. Además tiene la responsabilidad de crear nuevas cuentas de usuarios.

Tabla 2.6. Definición de Actores.

2.10. Listado de Casos de Uso del Sistema

CU 1	Autenticar Usuario
Actor	Usuarios
Descripción	Permite que solamente puedan acceder a la aplicación todos aquellos que trabajarán directamente con la misma, dándole acceso a operar en el sistema, según los vínculos a los que tienen permiso.
Referencia	RF 1

Tabla 2.7. Autenticar Usuario.

CU 2	Gestionar Edificio
Actor	Administrador
Descripción	Se gestionará toda la información de los edificios, se crearán los mismos, se modificarán y se eliminarán si ya no son necesarios.
Referencia	RF 2

Tabla 2.8. Gestionar Edificio.

CU 3	Gestionar Estudiante
Actor	Administrador
Descripción	Se les dará ubicación a todos los estudiantes en sus respectivos edificios, distribuidos por áreas temáticas.
Referencia	RF 3

Tabla 2.9. Gestionar Estudiante.

CU 4	Gestionar Cuarterías
Actor	Instructora de edificio
Descripción	A cada estudiante se le dará un día para que realicen la Cuartería. Si el año que cursa el estudiante es mayor o igual que tercero, entonces se tendrá en cuenta los días que tiene clases, para que no les coincida con el día de la cuartería. Si el estudiante realizó su servicio de cuartería se le tiene que dar una evaluación.
Referencia	RF 4

Tabla 2.10. Gestionar Cuarterías.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

CU 5	Gestionar Guardia Estudiantil (GE)
Actor	Secretaria del Vicedecano
Descripción	A cada grupo de la Facultad se le asignara un día para realizar la Guardia Estudiantil, así como a los profesores.
Referencia	RF 5

Tabla 2.11. Gestionar Guardia Estudiantil (GE).

CU 6	Gestionar la asignación de roles
Actor	Administrador
Descripción	Se le asignará un rol a un usuario determinado, para que éste tenga los permisos y privilegios adecuados para que interactúe con el sistema y trabaje en el mismo según su autorización.
Referencia	RF 6

Tabla 2.12. Gestionar asignación de roles.

CU 7	Gestionar Reportes
Actor	Usuarios
Descripción	Se gestionará toda la información acerca de la realización de las guardias y las Cuarterías así como la posibilidad de brindar reportes acerca de la planificación, evaluación, cantidad y regularidad con que se realizan dichas tareas.
Referencia	RF 7

Tabla 2.13. Gestionar Reportes

2.11. Diagrama de Casos de Uso del Sistema.

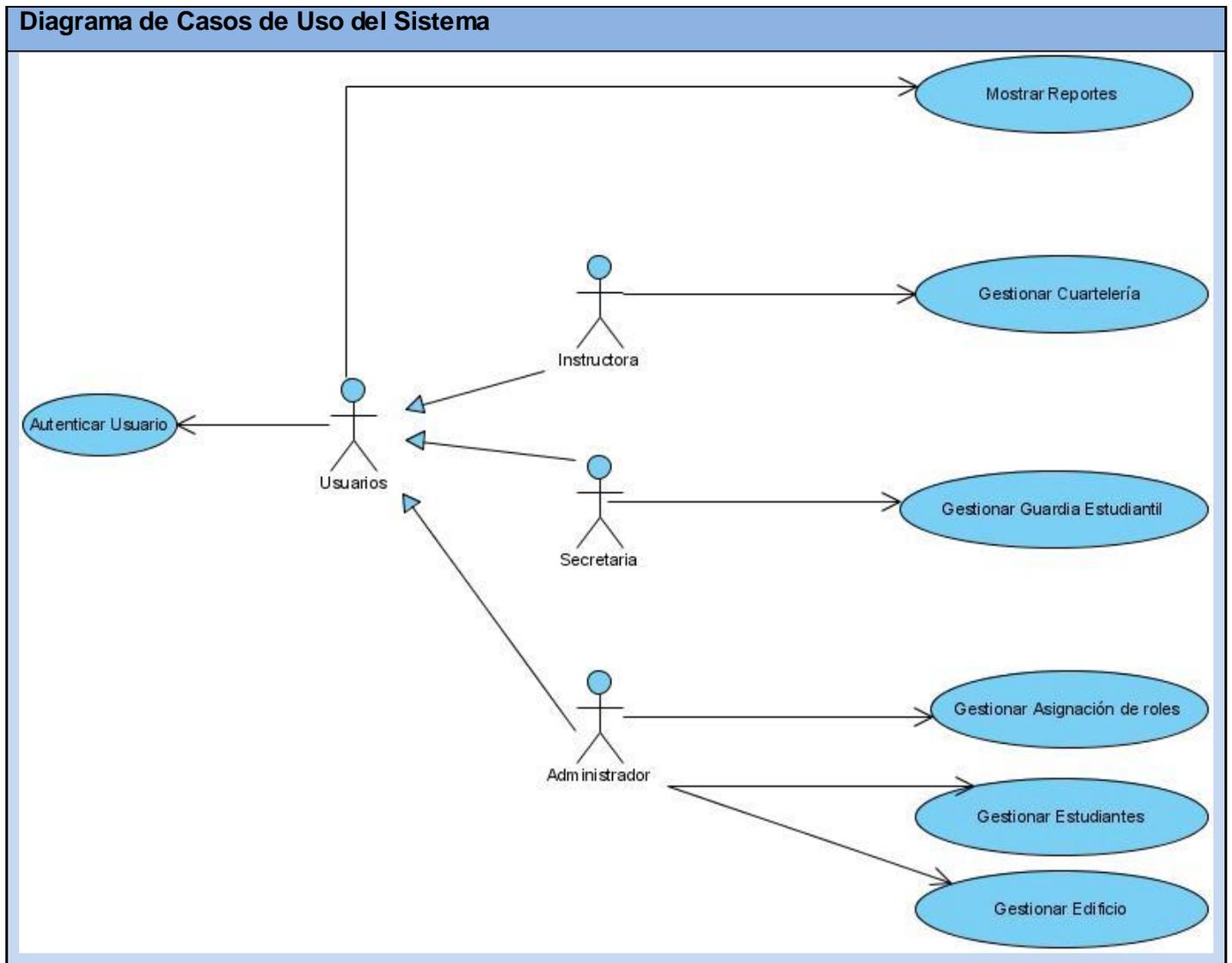


Figura 2.6. Diagrama de Casos de Uso del Sistema.

2.12. Casos de Uso por Ciclos

Código	Caso de Uso	Ciclo	Descripción
CU 1	Autenticar Usuario	1	Se decide desarrollar en el primer ciclo debido a la importancia que tiene el manejo de la seguridad del sistema con el objetivo de evitar la pérdida o alteración de información por personal no autorizado al mismo.
CU 2	Gestionar Edificio	1	Se decide desarrollar en el primer ciclo pues constituye una prioridad inmediata en la aplicación.
CU 3	Gestionar Estudiante	1	Se decide desarrollar en el primer ciclo pues es de vital importancia para la conformación de la distribución de la cuartería
CU 4	Gestionar Cuartería	1	Se decide desarrollar en el primer ciclo pues es una de las tareas fundamentales de la aplicación.
CU 5	Gestionar Guardia Estudiantil (GE)	1	Se decide desarrollar en el primer ciclo pues al igual que el Caso de Uso anterior es una de las tareas fundamentales de la aplicación.

CU 6	Gestionar asignación de roles	1	Se decide desarrollar en el primer ciclo pues es necesario que estén definidos los roles determinados para que un usuario específico pueda planificar la guardia estudiantil y la cuartelería.
CU 7	Gestionar Reportes	1	Se decide desarrollar en éste ciclo porque es una prioridad poder conocer reportes determinados sobre la realización de la guardia y la cuartelería

Tabla 2.13. Casos de Uso por Ciclos

2.13. Casos de Uso Expandidos

Autenticar Usuario

Descripción del Caso de Uso

Caso de Uso	Autenticar Usuario
Actores	Usuarios
Resumen	Permite que solamente puedan acceder a la aplicación todos aquellos que trabajarán directamente con la misma, dándole acceso a operar en el sistema, según los vínculos a los que tienen permiso.
Precondiciones	La aplicación debe estar disponible

Referencias	RF 1
Prioridad	Crítica
Flujo Normal de Eventos	
“Sección Autenticar”	
Acción del actor	Respuesta del Sistema
1. El actor entra usuario y contraseña.	2. El sistema comprueba si los datos entrados son o no correctos. Si son correctos entra a la aplicación con todas las funcionalidades a las que tiene acceso.
<i>Prototipo de Interfaz</i>	
Flujo Alternativo de Eventos	
Sección “Autenticar”	
Acción del Actor	Respuesta del Sistema
	2.1 Si los datos son incorrectos el sistema muestra un mensaje de error “Usuario o contraseña incorrecta” .
Pos condiciones	El usuario queda autenticado en el sistema.

Tabla 2.14. Autenticar Usuario.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

Gestionar Edificio

Descripción del Caso de Uso

Caso de Uso	Gestionar Edificio
Actores	Administrador
Resumen	El caso de Uso se inicia cuando se manipula toda la información acerca de los edificios: se crearán los mismos, se modificarán y se eliminarán si ya no son necesarios, terminando así el caso de uso.
Precondiciones	La información de los servicios con los que se trabajarán, deben estar disponibles y actualizados.
Referencias	RF 1
Prioridad	Crítica.
Flujo Normal de Eventos	
“Sección Crear Edificio”	
Acción del actor	Respuesta del Sistema
1. El administrador accede a la opción “Crear Edificio”	2. El sistema muestra un formulario con los campos correspondientes para crear un nuevo edificio.
3. Introduce los datos de forma correcta y presiona el botón aceptar.	4. Valida los datos
	5. Adiciona Edificio a la Base de Datos
Flujos alternos	
3. a. Introduce datos de forma incorrecta	4.1 Muestra un mensaje de advertencia y retorna al paso 3.
	4.2 En caso de coincidir algún dato

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

	muestra un mensaje de advertencia y retorna al paso 3
Flujo Normal de Eventos	
“Sección Modificar Edificio”	
Acción del actor	Respuesta del Sistema
6. Selecciona la opción modificar	7. Muestra los datos listos para ser modificados
8. Modifica los datos necesarios	9. Valida los nuevos datos
	10. Actualiza los datos de los edificios en la Base de Datos
Flujos alternos	
8. a. Realiza los cambios introduciendo datos incorrectos	9.1 En caso de error muestra un mensaje de advertencia y brinda la posibilidad de corregir el/los dato/s escritos incorrectamente y regresa el paso 8.
	9.2 En caso de coincidir algún dato que deba ser único, se muestra un mensaje de advertencia y regresa al paso 8.
Flujo Normal de Eventos	
“Sección Eliminar Edificio”	
Acción del actor	Respuesta del Sistema
11. Selecciona la opción eliminar	12. El sistema muestra un formulario con los edificios para ser eliminados
13. Escoge el edificio a eliminar.	14. El sistema muestra un mensaje para saber si es el edificio que se desea eliminar.
15. Confirma el mensaje	16. Elimina el edificio de la base de datos

Flujos alternos	
15. a. No desea eliminar el edificio	16.1 No se realiza ninguna operación
Prototipo de Interfaz	
Pos condiciones	Edificios creados en la BD. Edificios modificados en la BD Edificios eliminados en la BD.

Tabla 2.15. Gestionar Edificio.

Gestionar Estudiante

Descripción del Caso de Uso

Caso de Uso	Gestionar Estudiante
Actores	Administrador
Resumen	El Caso de Uso se inicia cuando se da la ubicación a todos los estudiantes en sus respectivos edificios, distribuidos por áreas temáticas. Si es necesario se modificarán o si ya no están en la escuela se eliminarán, terminando así el caso de uso.
Precondiciones	La información con la cual se trabajará debe estar disponible y actualizada
Referencias	RF 1
Prioridad	Crítica.
Flujo Normal de Eventos	
“Sección Crear Estudiante”	
Acción del actor	Respuesta del Sistema
1. El administrador accede a la	2. El sistema muestra un formulario para

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

opción “Adicionar Estudiante”	escoger un edificio y apartamento, además de mostrar los estudiantes para asignarlo a ese edificio.
3. Escoge el edificio y el apartamento, además del grupo, el sexo y el estudiante a adicionar y presiona la opción aceptar	4. Valida los datos insertados.
	5. Adiciona estudiante a la Base de Datos.
Flujos alternos	
3. a. Introduce datos de forma incorrecta	4.1 Muestra un mensaje de advertencia y retorna al paso 3.
	4.2 En caso de coincidir algún dato muestra un mensaje de advertencia y retorna al paso 3
Flujo Normal de Eventos	
“Sección Modificar Estudiante”	
Acción del actor	Respuesta del Sistema
6. Selecciona la opción modificar	7. Muestra todos los edificios
8. Selecciona el edificio donde está el estudiante a modificar.	9. Busca el estudiante en la base de datos y muestra sus datos
10. Modifica datos del estudiante	11. Valida los datos modificados.
	12. Actualiza los datos del estudiante en la Base de Datos.
Flujos alternos	
10. a. Realiza los cambios introduciendo datos incorrectos	11.1 En caso de error muestra un mensaje de advertencia y brinda la posibilidad de corregir el/los dato/s escritos

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

	incorrectamente y se regresa el paso 10.
	11.2 En caso de coincidir algún dato que deba ser único, se muestra un mensaje de advertencia y regresa al paso 10.
Flujo Normal de Eventos	
“Sección Eliminar Estudiante”	
Acción del actor	Respuesta del Sistema
13. Selecciona la opción eliminar	14. El sistema muestra un formulario con los edificios
15. Escoge el edificio donde se encuentra el estudiante.	16. El sistema muestra todos los estudiantes de este edificio
17. Selecciona el estudiante	18. Muestra un mensaje si se desea eliminar este estudiante.
19. Confirma la eliminación.	20. El sistema elimina el estudiante de la Base de Datos.
Flujos alternos	
19. a. No desea eliminar estudiante	20.1 No realiza ninguna acción.
Prototipo de Interfaz	
Pos condiciones	Estudiantes creados en la BD Estudiantes modificados en la BD Estudiantes eliminados en la BD.

Tabla 2.16. Gestionar Estudiante

Gestionar Cuartelería

Descripción del Caso de Uso

Caso de Uso	Gestionar Cuartelería
Actores	Instructora de Edificio
Resumen	El caso de Uso se inicia cuando al estudiante se le planifica un día para que realice la cuartelería. Si el año que cursa es mayor o igual que tercero, entonces se tendrá en cuenta los días de clases, para que no les coincida con el día de la cuartelería. Si el estudiante realizó dicho servicio se le tiene que dar una evaluación, terminando así el caso de uso.
Precondiciones	La Instructora de Residencia debe estar autenticada en el sistema y debe tener los permisos necesarios asignados
Referencias	RF 4
Prioridad	Crítica.
Flujo Normal de Eventos	
“Sección Crear Cuartelería”	
Acción del actor	Respuesta del Sistema
1. La instructora de edificio selecciona la opción crear Cuartelería	2. El sistema muestra un formulario con los campos correspondientes para crear la Cuartelería.
3. Escoge el edificio, estudiante y la fecha de la realización de la Cuartelería.	4. Valida los datos
	5. Adiciona Cuartelería a la Base de Datos

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

Flujos alternos	
3. a. Introduce datos de forma incorrecta	4.1 Muestra un mensaje de advertencia y retorna al paso 3.
	4.2 En caso de coincidir algún dato muestra un mensaje de advertencia y retorna al paso 3
Flujo Normal de Eventos	
“Sección Modificar Cuartelería”	
Acción del actor	Respuesta del Sistema
6. Selecciona la opción modificar	7. Muestra los edificios donde quiere ser modificada la Cuartelería.
8. Selecciona el edificio y la Cuartelería a modificar.	9. Muestra los datos de la Cuartelería a modificar
10. Modifica los datos.	11. Valida los datos modificados
	12. Adiciona a la Base de Datos la nueva Cuartelería
Flujos alternos	
10. a. Realiza los cambios introduciendo datos incorrectos	11.1 En caso de error muestra un mensaje de advertencia y brinda la posibilidad de corregir el/los dato/s escritos incorrectamente y se regresa el paso 10.
	11.2 En caso de coincidir algún dato que deba ser único, se muestra un mensaje de advertencia y regresa al paso 10.
Flujo Normal de Eventos	
“Sección Eliminar Cuartelería”	
Acción del actor	Respuesta del Sistema

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

13. Selecciona la opción eliminar	14. Sistema muestra un formulario con los edificios donde se va a eliminar la Cuartelería.
15. Escoge el edificio y la Cuartelería que se va a eliminar.	16. El sistema muestra un mensaje para saber si es la Cuartelería que se desea eliminar.
17. Confirma el mensaje	18. Elimina el edificio de la base de datos
Flujos alternos	
17. a. No desea eliminar el edificio	18.1 No se realiza ninguna operación
<i>Prototipo de Interfaz</i>	
Pos condiciones	Cuartelarías creadas, en la BD Cuartelarías modificadas en la BD Cuartelarías eliminadas en la BD.

Tabla 2.17. Gestionar Cuartelería.

Gestionar Guardia Estudiantil (GE)

Descripción del Caso de Uso

Caso de Uso	Gestionar Guardia Estudiantil (GE)
Actores	Secretaria del Vicedecano
Resumen	El caso de Uso se inicia cuando a cada grupo de la Facultad se le asigna un día para realizar la Guardia Estudiantil, terminando así el caso de uso
Precondiciones	La Secretaría del Vicedecano de residencia debe estar autenticada en el sistema y tener los permisos necesarios asignados

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

Referencias	RF 1
Prioridad	Crítica.
Flujo Normal de Eventos	
“Sección Crear Guardia Estudiantil”	
Acción del actor	Respuesta del Sistema
1. La Secretaria del Vicedecano accede a la opción “Crear Guardia Estudiantil”	2. El sistema muestra un formulario con todos los campos correspondientes.
3. Escoge el profesor, la fecha y el grupo de la guardia.	4. Valida los datos escogidos.
	5. Adiciona la guardia a la Base de Datos.
Flujos alternos	
3. a. Escoge datos de forma incorrecta	4.1 En caso de faltar algún campo por llenar o datos incorrectos muestra un mensaje de advertencia y retorna al paso 3.
	4.2 En caso de coincidir algún dato muestra un mensaje de advertencia y retorna al paso 3
Flujo Normal de Eventos	
“Sección Modificar Guardia Estudiantil”	
Acción del actor	Respuesta del Sistema
6. Selecciona la opción modificar	7. Muestra todos las guardias
8. Selecciona la guardia a modificar.	9. Muestra los datos de la guardia a modificar
10. Modifica datos de la guardia.	11. Valida los datos modificados.
	12. Actualiza los datos del estudiante en

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

	la Base de Datos
Flujos alternos	
10. a. Realiza los cambios introduciendo datos incorrectos	11.1 En caso de error muestra un mensaje de advertencia y brinda la posibilidad de corregir el/los dato/s escritos incorrectamente y se regresa el paso 10.
	11.2 En caso de coincidir algún dato que deba ser único, se muestra un mensaje de advertencia y regresa al paso 10.
Flujo Normal de Eventos	
“Sección Eliminar Guardia Estudiantil”	
Acción del actor	Respuesta del Sistema
13. Selecciona la opción eliminar	14. El sistema muestra un formulario con todas las guardias
15. Escoge la guardia que desea eliminar	16. Muestra un mensaje si se desea eliminar este estudiante
17. Confirma la eliminación.	18. El sistema elimina la guardia de la Base de Datos.
Flujos alternos	
17. a. No desea eliminar guardia.	18.1 No realiza ninguna acción.
Prototipo de Interfaz	
Pos condiciones	Cuartelerías creadas en la BD Cuartelerías modificadas en la BD

	Cuartelerías y eliminadas en la BD.
--	-------------------------------------

Tabla 2.18. Gestionar Guardia Estudiantil (GE).

Gestionar asignación de roles

Descripción del Caso de Uso

Caso de Uso	Gestionar asignación de roles
Actores	Administrador
Resumen	El caso de Uso se inicia cuando se gestiona la información acerca de asignación de roles. Se asignarán roles y se eliminarán si ya no son necesarios, terminando así el caso de uso.
Precondiciones	El Vicedecano de residencia debe estar autenticado en el sistema y tener los permisos necesarios asignados
Referencias	RF 1
Prioridad	Crítica
Flujo Normal de Eventos	
“Sección asignar roles”	
Acción del actor	Respuesta del Sistema
1. El administrador accede a la opción “Editar roles”	2. El sistema muestra un formulario con los campos correspondientes para asignar un rol.
3. Introduce los datos y presiona el botón aceptar.	4. Valida los datos
	5. Adiciona el usuario a la Base de Datos

Flujos alternos	
3. a. Introduce datos de forma incorrecta	4.1 Muestra un mensaje de advertencia y retorna al paso 3.
	4.2 En caso de coincidir algún dato muestra un mensaje de advertencia y retorna al paso 3
Flujo Normal de Eventos	
“Sección Eliminar rol”	
Acción del actor	Respuesta del Sistema
1. Selecciona la opción eliminar	2. El sistema marca el usuario que debe ser eliminado
	3. Elimina el rol de la base de datos
Flujos alternos	
5. a. No desea eliminar el usuario	6.1 No se realiza ninguna operación
Prototipo de Interfaz	
Pos condiciones	Roles asignados en la BD Roles eliminados en la BD.

Tabla 2.19. Gestionar Asignación de roles.

Mostar Reportes

Descripción del CU

Caso de Uso	Mostar Reportes
Actores	Usuarios
Resumen	El caso de Uso se inicia cuando se gestiona toda la información acerca de la planificación y evaluación de los estudiantes en el desempeño de estas actividades así como la cantidad y regularidad con que las hacen, terminando así el caso de Uso
Precondiciones	Debe estar autenticado
Referencias	RF 1
Prioridad	Opcional
Flujo Normal de Eventos	
Acción del actor	Respuesta del Sistema
1. El Usuario selecciona la opción "Gestionar Reportes"	2. El sistema mostrará una interfaz que brinda la posibilidad de seleccionar el tipo de reportes que se desea ver.
3. El Usuario selecciona el reporte deseado	4. El sistema muestra una interfaz que contiene los datos estadísticos de dicho reporte y una opción que le permite mandarlo a imprimir si lo desea
5. El Usuario manda a imprimir el reporte si lo desea.	6. EL sistema imprime el reporte
Prototipo de Interfaz	
Pos condiciones	Se muestra el reporte deseado.

Tabla 2.20. Mostar Reportes.

2.14. Conclusiones

En este capítulo se describieron los procesos de negocio, para un mejor entendimiento fueron expuestos los requisitos funcionales y no funcionales que debe cumplir la aplicación. Se desarrollaron además los respectivos artefactos correspondientes a los Flujos de trabajo de Modelado del Negocio y requerimiento. Así como la descripción textual de cada uno, para tener un mayor conocimiento y claridad a la hora de realizar la aplicación web.

3. CAPÍTULO III: DISEÑO DEL SISTEMA

En este capítulo se describen los casos de uso donde se encuentran involucrados los procesos fundamentales que el sistema debe cumplir. Se muestran mediante los diagramas de clases las interacciones de cada uno de los casos de uso, brindándose al mismo tiempo una breve descripción de las clases fundamentales para el desarrollo, así como una breve descripción de la base de datos y las tablas que la componen.

3.1. Diseño

En los diagramas de clases del diseño se representan las relaciones físicas entre los objetos del sistema, especificándose de cada clase los atributos con su visibilidad y métodos que poseen correspondencia directa con los de la implementación, la navegabilidad y dependencias que existen entre ellas. Las representaciones de dichas clases son en correspondencia con el tipo de sistema que se esté contrayendo y el lenguaje de programación que se esté usando. En este caso se utilizan los estereotipos Web encajando todas las clases perfectamente en cuatro estereotipos:

Páginas clientes: Son las páginas encargadas de permitir a los usuarios interactuar con el sistema tanto para hacer solicitudes como para que las respuestas a las mismas sean mostradas.

Clases servidoras: Son las encargadas de la construcción de forma dinámica de las páginas clientes y sirven de enlace entre estas y el resto de las clases.

Clases controladoras: Tienen la responsabilidad de realizar las operaciones que responden a los procesos de negocio y dar respuestas a las solicitudes hechas por el usuario.

Clases entidad: Son las responsables de la persistencia de los datos físicamente.

3.1.1 Patrones de Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

3.1.2 Patrón Modelo-Vista-Controlador (MVC)

Para la realización de este epígrafe se ha utilizado como referencia bibliográfica. (12)

El patrón MVC es un patrón de diseño de software en el cual todo el proceso está dividido en tres capas, típicamente estas capas son: el Modelo, la Vista y el Controlador.

En la Vista cuenta con dos elementos fundamentales, la capa externa que es el archivo que contiene el código HTML de las interfaces de usuario que no varía para todas o parte de las páginas web y las Plantillas que contienen los formularios y datos específicos de cada página.

En el Controlador, se encuentra el controlador frontal que constituye el único punto de entrada a la aplicación y cuya principal función es re-direccionar las peticiones realizadas por la Vista, hacia las acciones que constituyen los controladores de los diferentes módulos que conforman la aplicación.

En el Modelo se encuentran la capa de acceso a datos y la capa de abstracción de la base de datos. La primera contiene todas las funcionalidades que responden a la lógica del negocio y la segunda provee una abstracción de la base de datos que posibilitará que la aplicación no dependa de un sistema gestor de BD determinado.

De esta forma cuando el usuario interactúa con la aplicación lo hace mediante la Vista, que envía las peticiones al controlador frontal, que las re-direcciona hacia la acción correspondiente, encargándose esta de llamar a la función implementada en el Modelo, donde se obtiene la respuesta, que es enviada a través del Controlador hacia la Vista.

Algunos de los principales beneficios que brinda aplicar el patrón MVC son:

- Menor acoplamiento.
 - Desacopla las vistas de los modelos.
 - Desacopla los modelos de la forma en que se muestran e ingresan los datos
- Mayor cohesión.
 - Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo de negocio).
- Las vistas proveen mayor flexibilidad y agilidad.
 - Se pueden crear múltiples vistas de un modelo.
 - Las vistas pueden anidarse.
 - Se puede cambiar el modo en que una vista responde al usuario sin cambiar su representación visual.
 - Se pueden sincronizar las vistas.
 - Las vistas pueden concentrarse en diferentes aspectos del modelo.
- Más claridad de diseño.
- Facilita el mantenimiento.
- Mayor escalabilidad.

3.1.2.1 Diagrama de clases del diseño.

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces, en una aplicación. Normalmente contiene la siguiente información:

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

- Navegabilidad.
- Dependencias.

A continuación se muestran los diagramas de clases del diseño del software:

DCD_Autenticar Usuario.

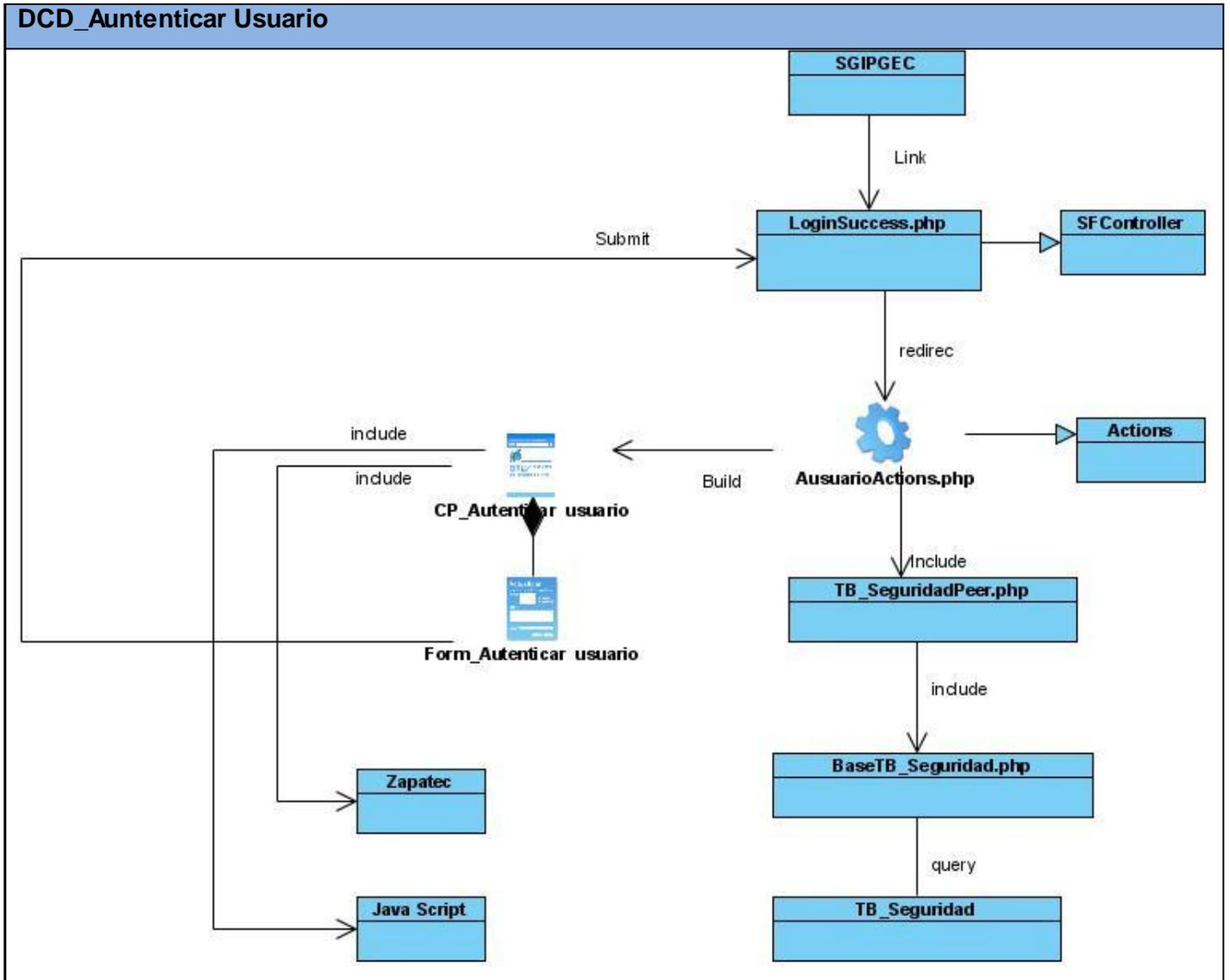


Fig.3.1 Diagrama de Clases del Diseño Autenticar Usuario.

DCD_Gestionar asignación de roles.

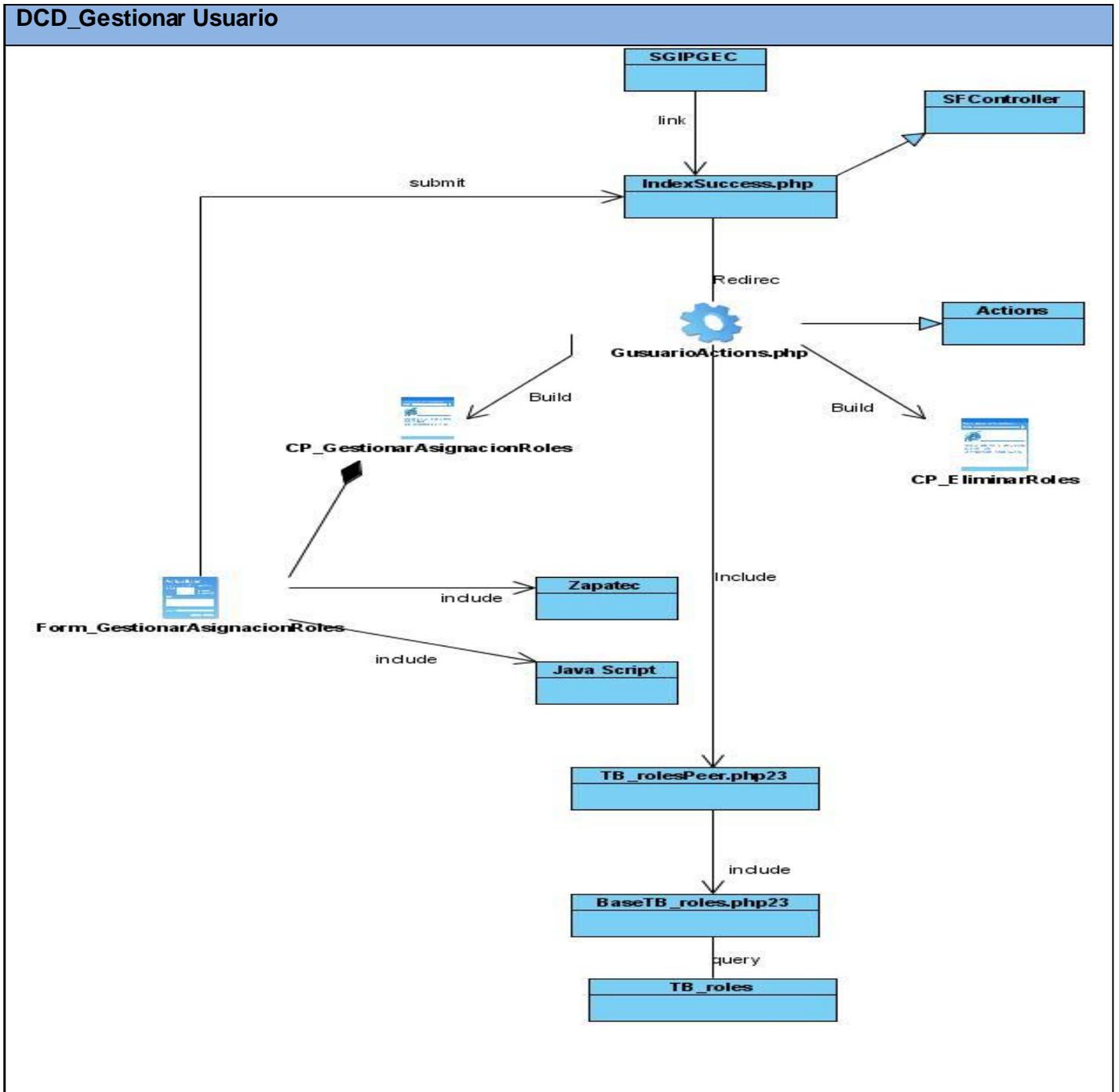


Fig.3.2 Diagrama de Clases del Diseño Gestionar asignación de roles.

DCD_Gestionar Estudiante.

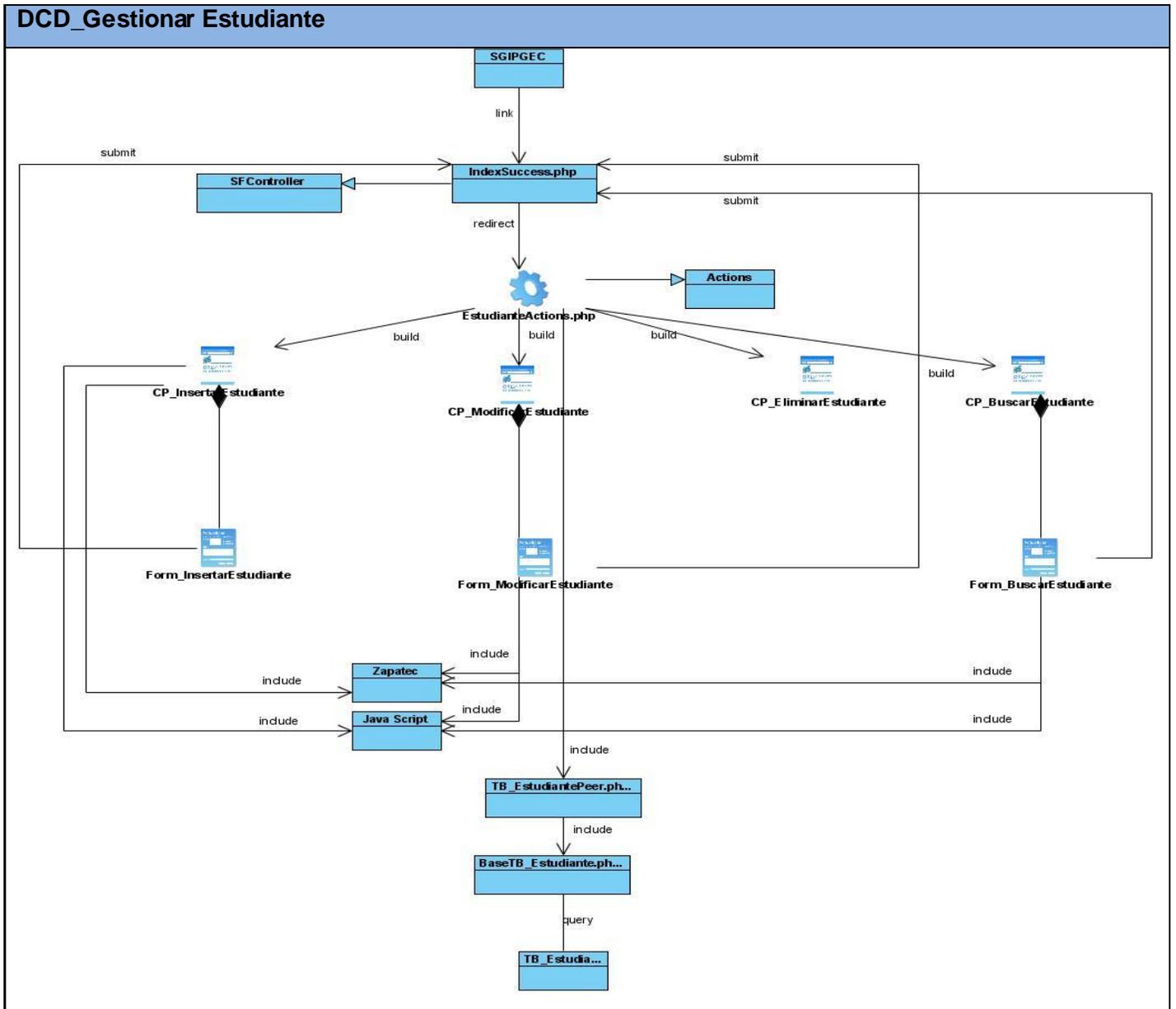


Fig.3.3 Diagrama de Clases del Diseño Gestionar Estudiante.

DCD_Gestionar Edificio.

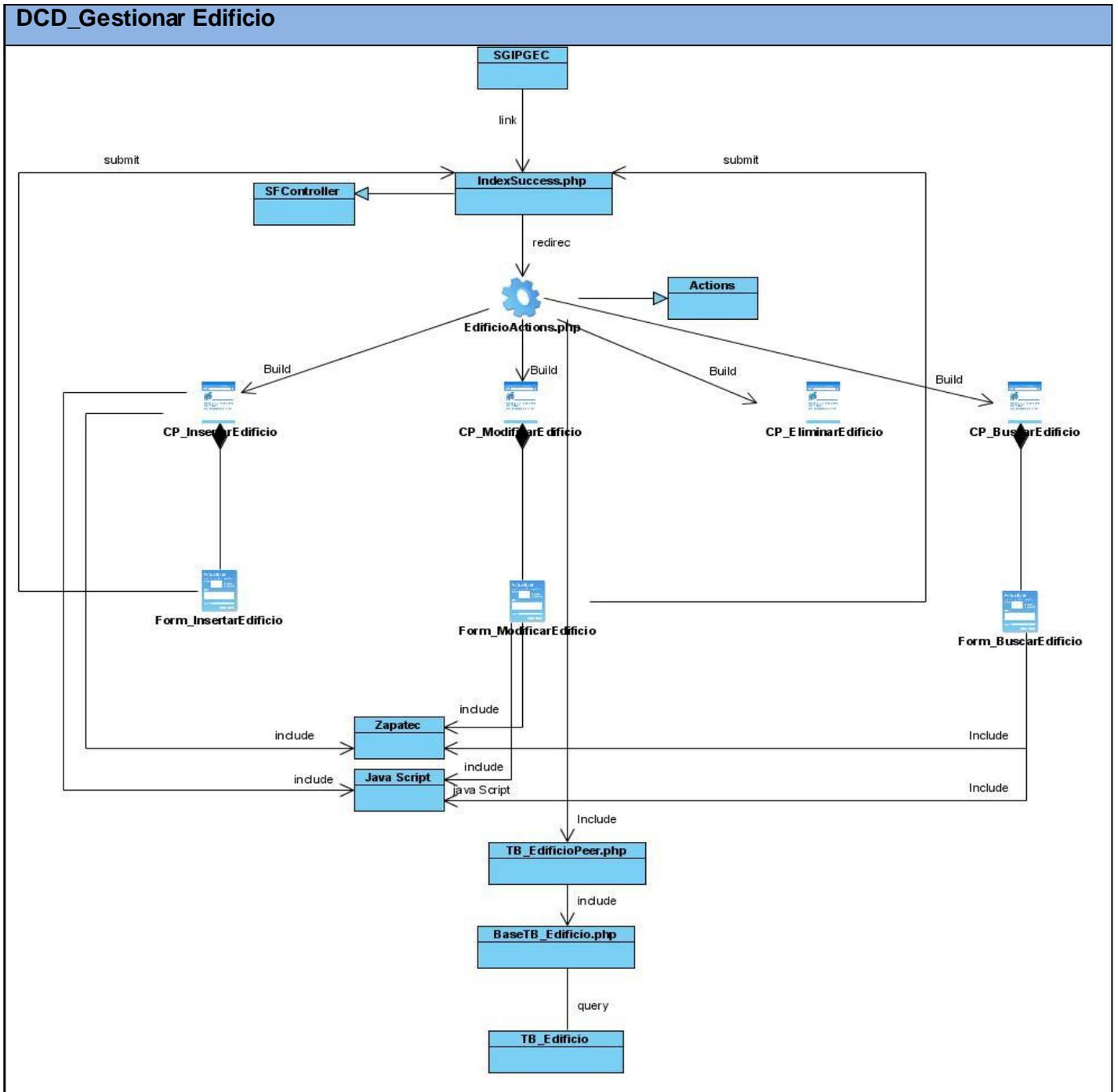


Fig.3.4 Diagrama de Clases del Diseño Gestionar Edificio.

DCD_Gestionar Guardia.

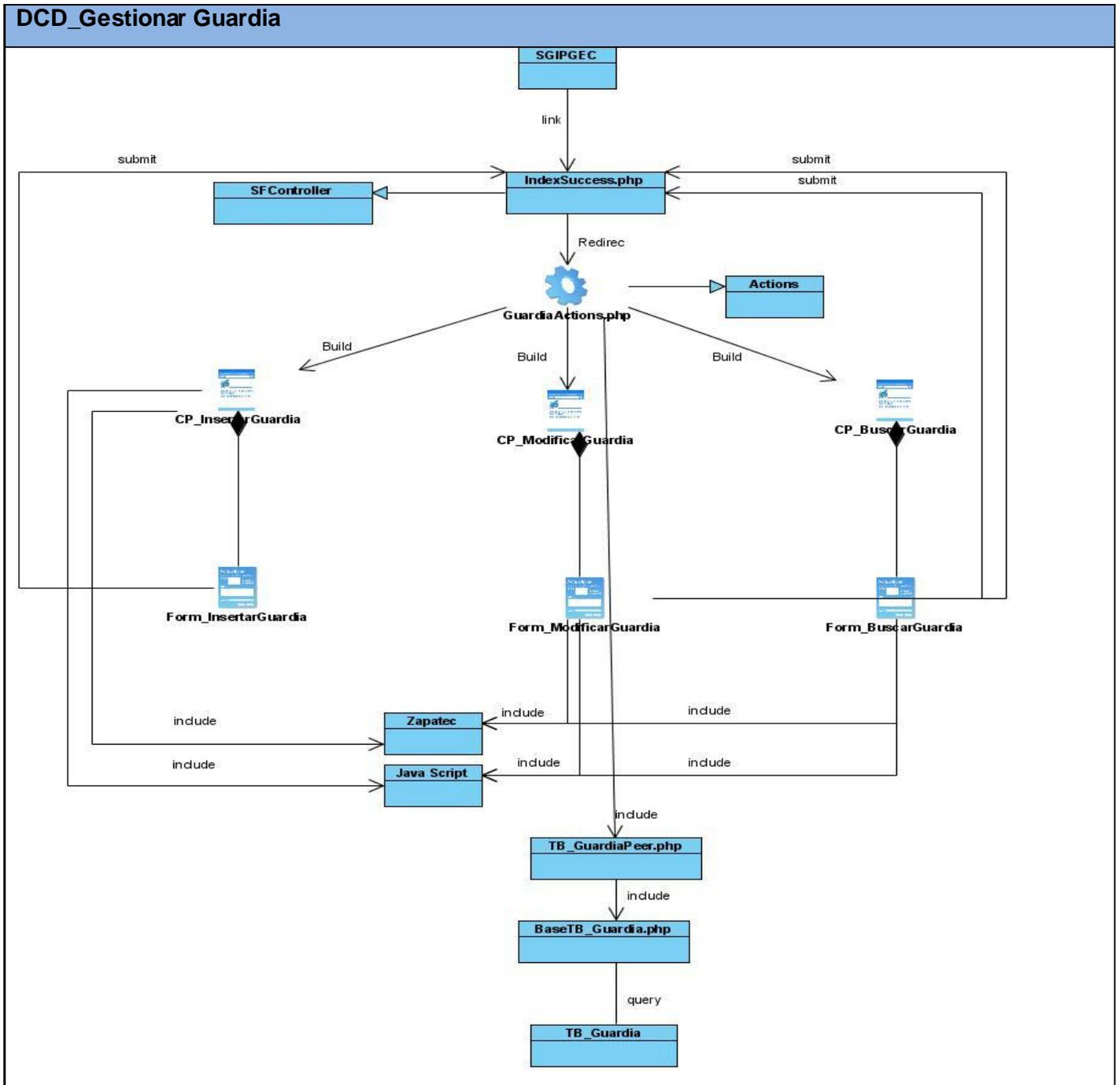


Fig.3.5 Diagrama de Clases del Diseño Gestionar Guardia.

DCD_Gestionar Cuartelería.

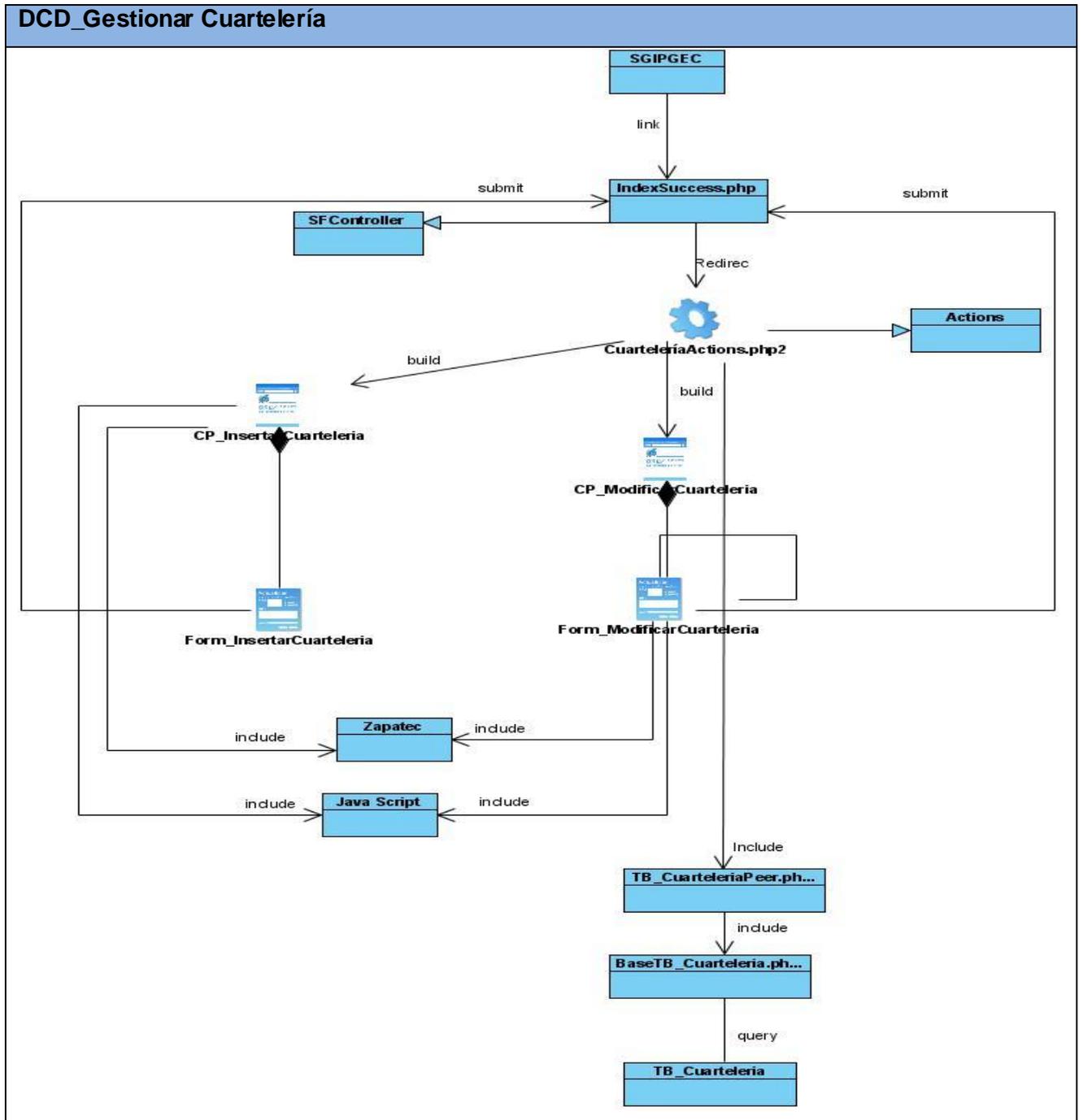


Fig.3.6 Diagrama de Clases del Diseño Gestionar Cuartelería.

DCD_Mostrar Reportes.

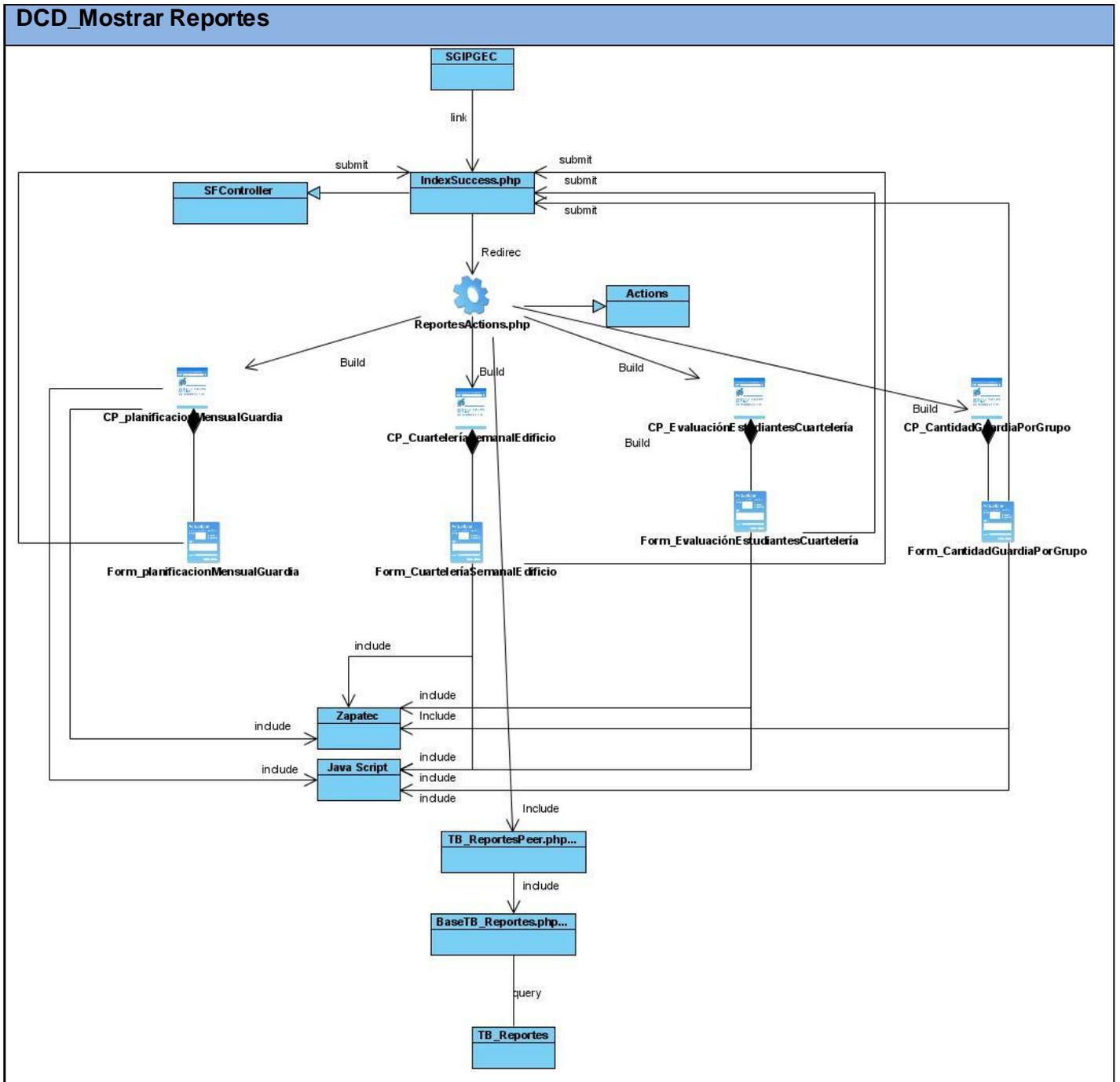


Fig.3.7 Diagrama de Clases del Diseño Gestionar Reportes.

3.2. Diagramas de interacción.

Un diagrama de interacción explica gráficamente las interacciones existentes entre las instancias (y las clases) del modelo de estas. El punto de partida de las interacciones es el cumplimiento de las condiciones de los contratos de operación. El UML define dos tipos de estos diagramas; ambos sirven para expresar interacciones semejantes o idénticas de mensaje:

1. Diagramas de colaboración.
2. Diagramas de secuencia.

Las interacciones entre los objetos y las clases, de la solución propuesta como resultado del diseño del software, fueron modeladas a través de diagramas de secuencia. A continuación se muestran algunos ejemplos:

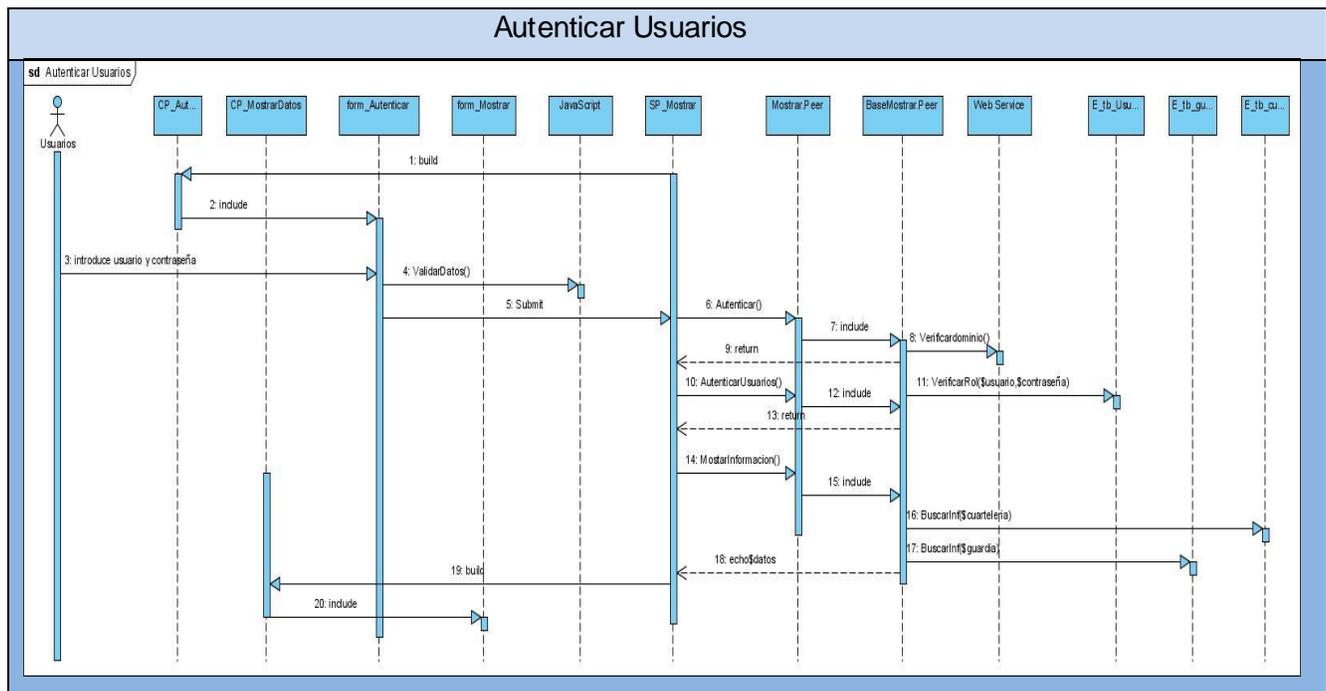


Figura 3.1. “Diagrama de Secuencia “Autenticar Usuarios.”

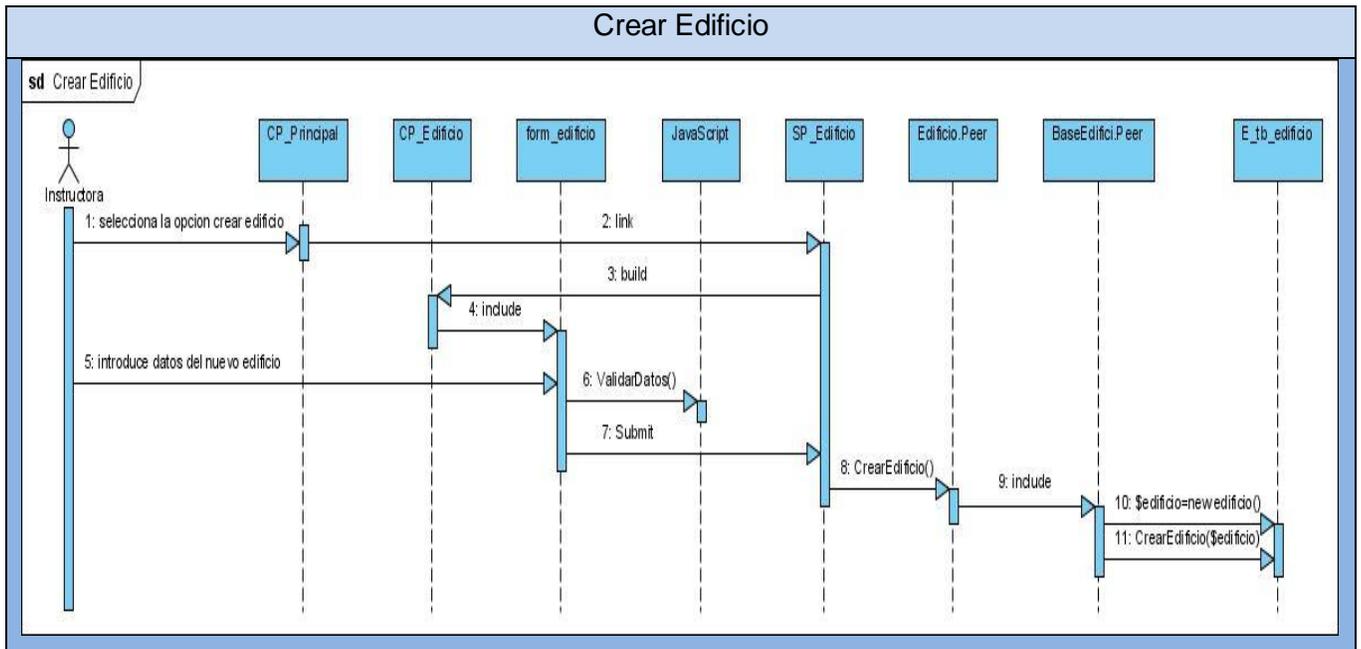


Figura 3.2. "Diagrama de Secuencia "Crear Edificio"

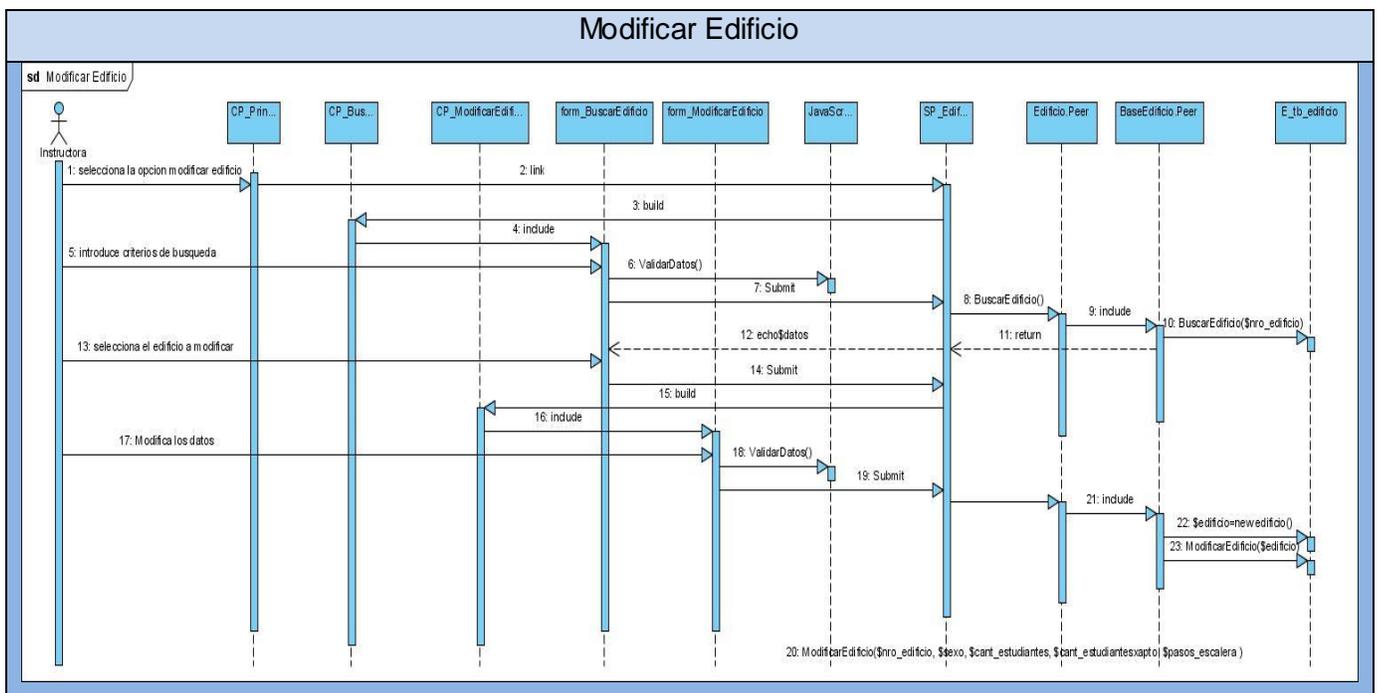


Figura 3.3. "Diagrama de Secuencia "Modificar Edificio".

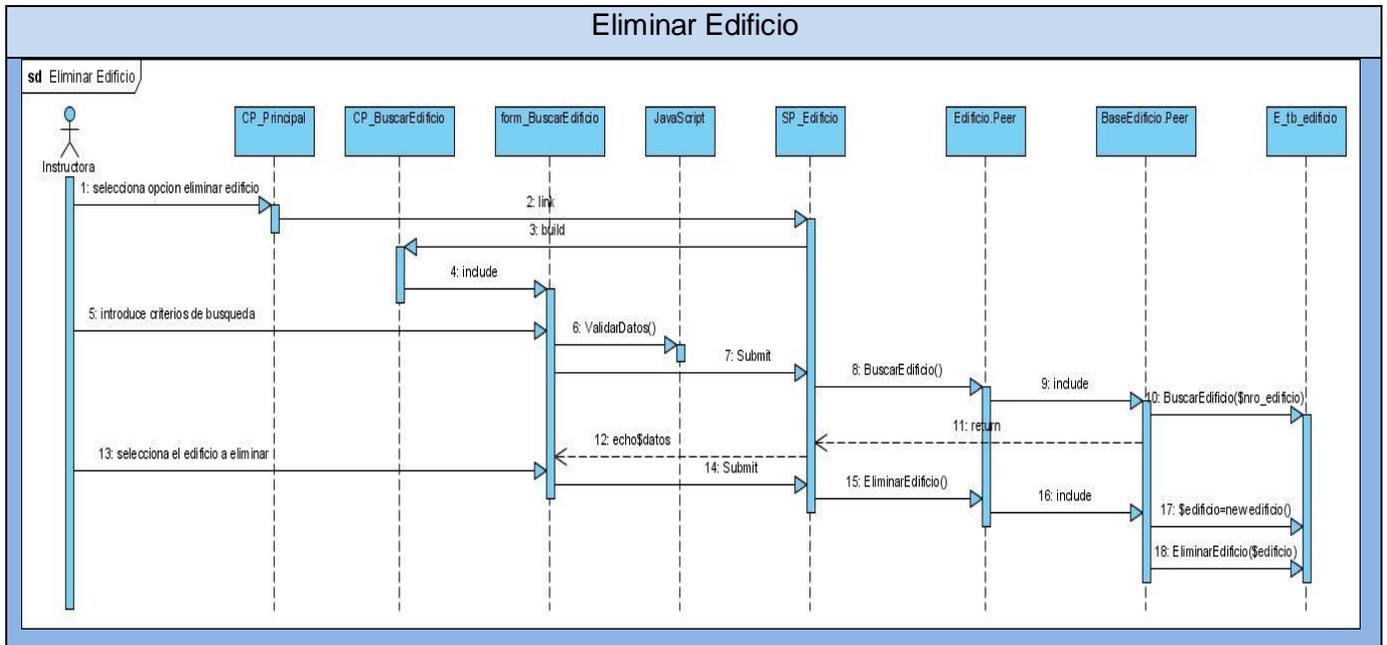


Figura 3.4. "Diagrama de Secuencia "Eliminar Edificio".

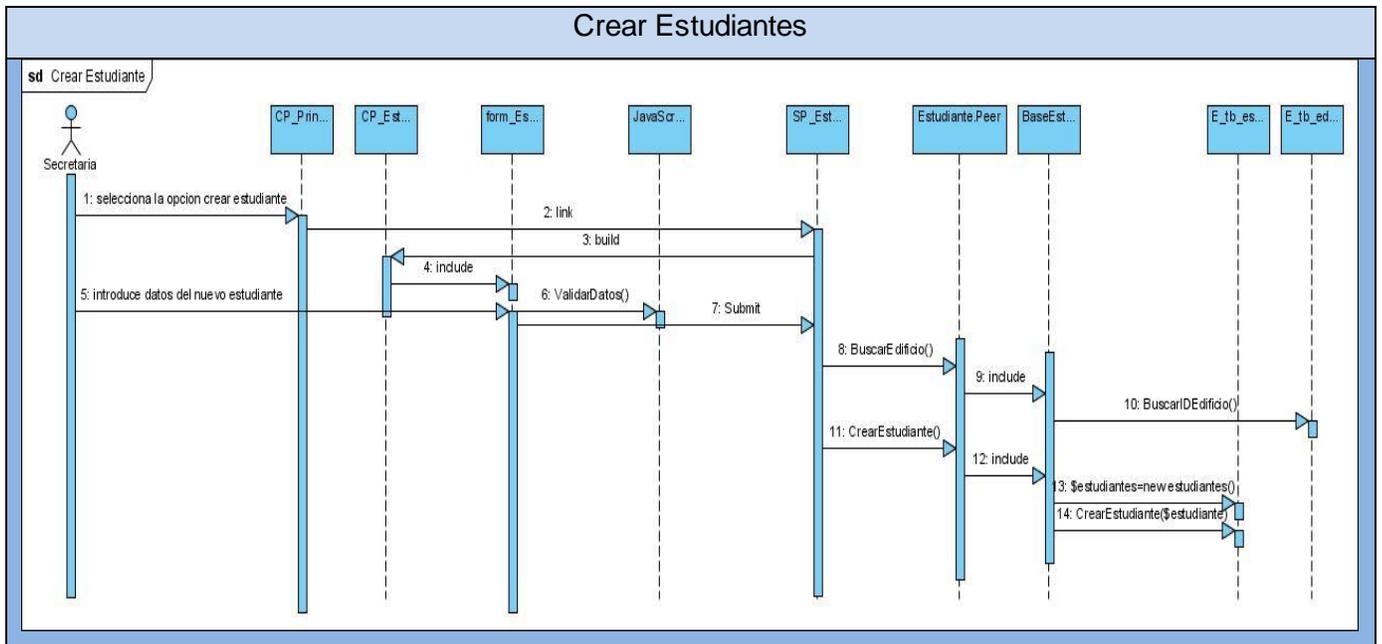


Figura 3.5. "Diagrama de Secuencia "Crear Estudiantes".

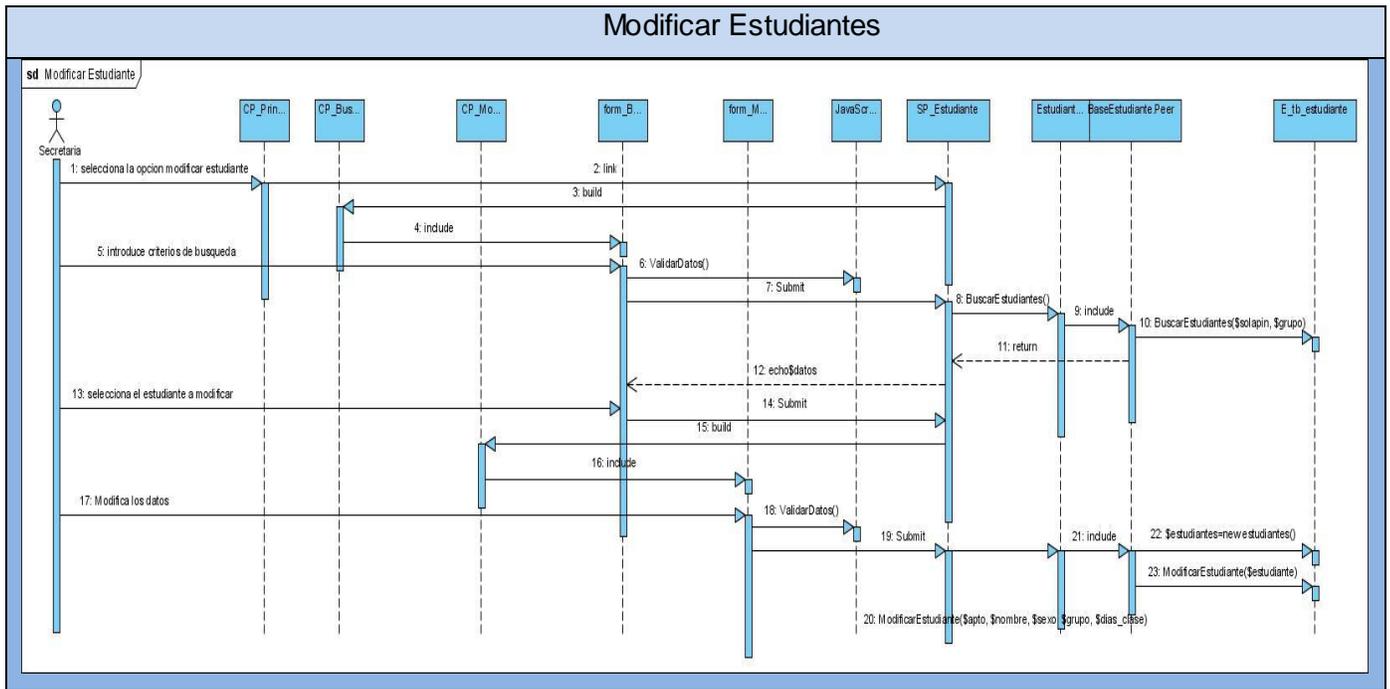


Figura 3.6. "Diagrama de Secuencia "Modificar Estudiantes"

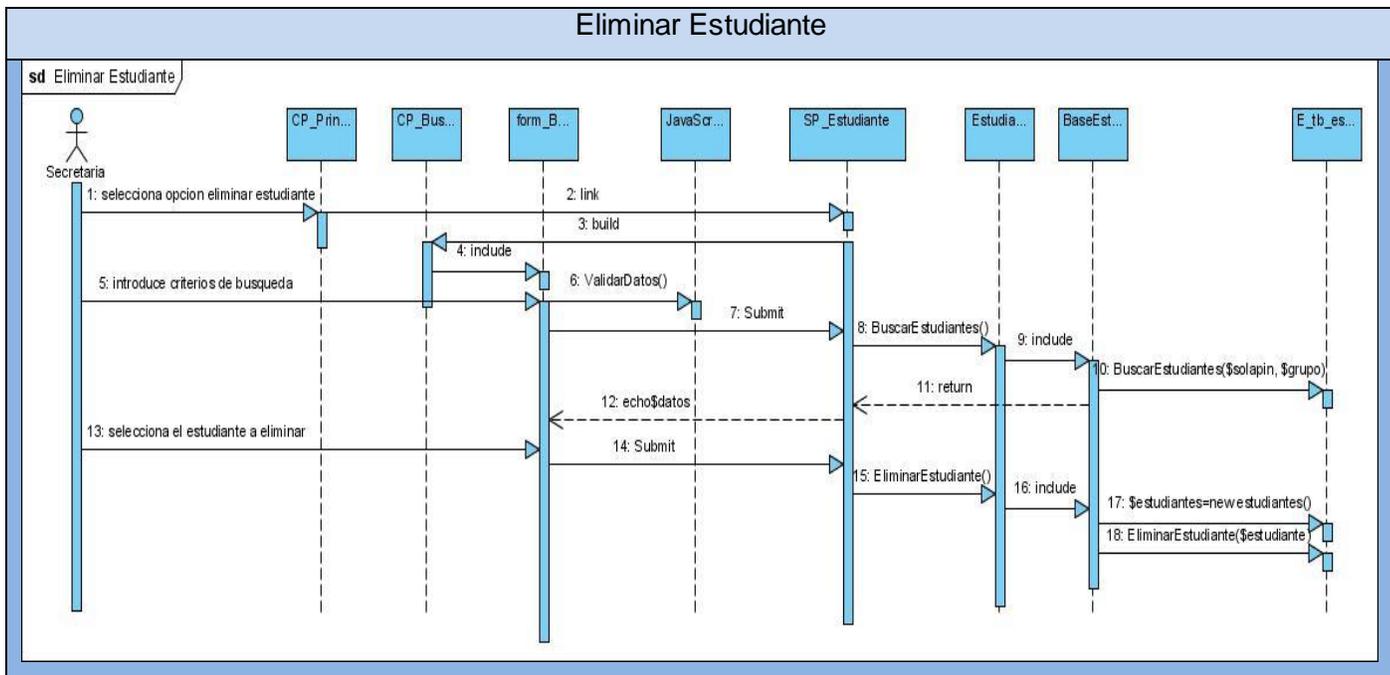


Figura 3.7 "Diagrama de Secuencia "Eliminar Estudiante".

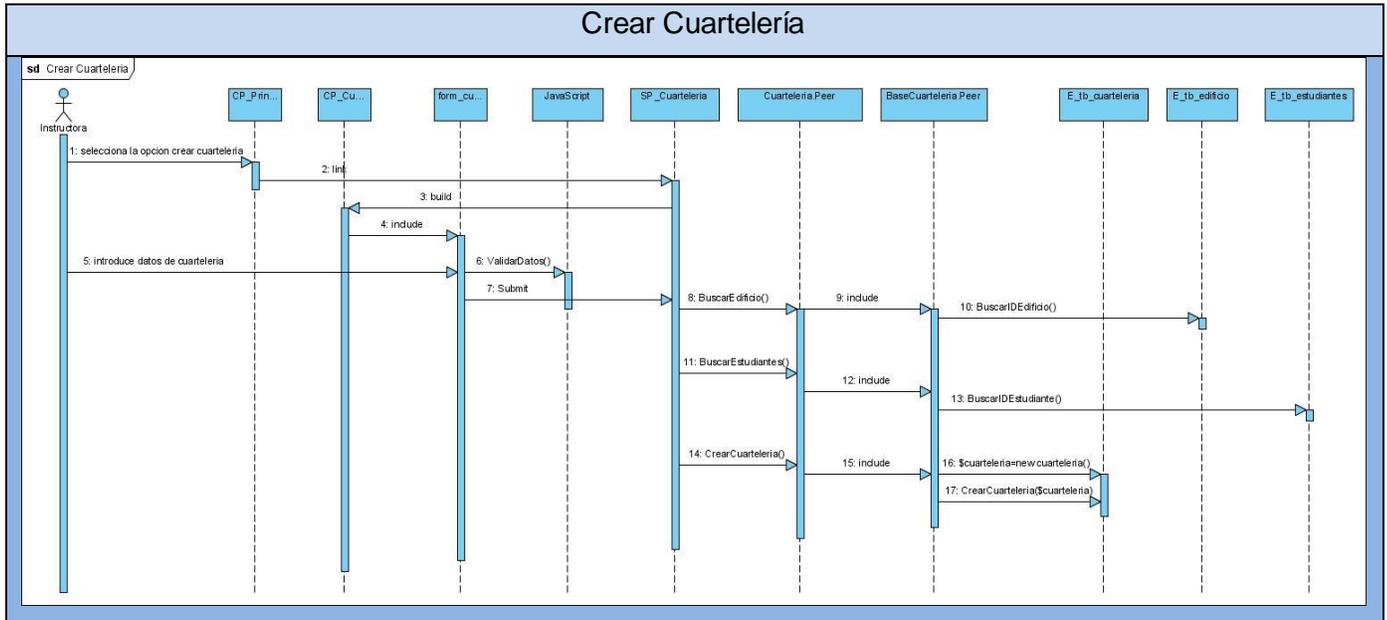


Figura 3.8. "Diagrama de Secuencia "Crear Cuartería".

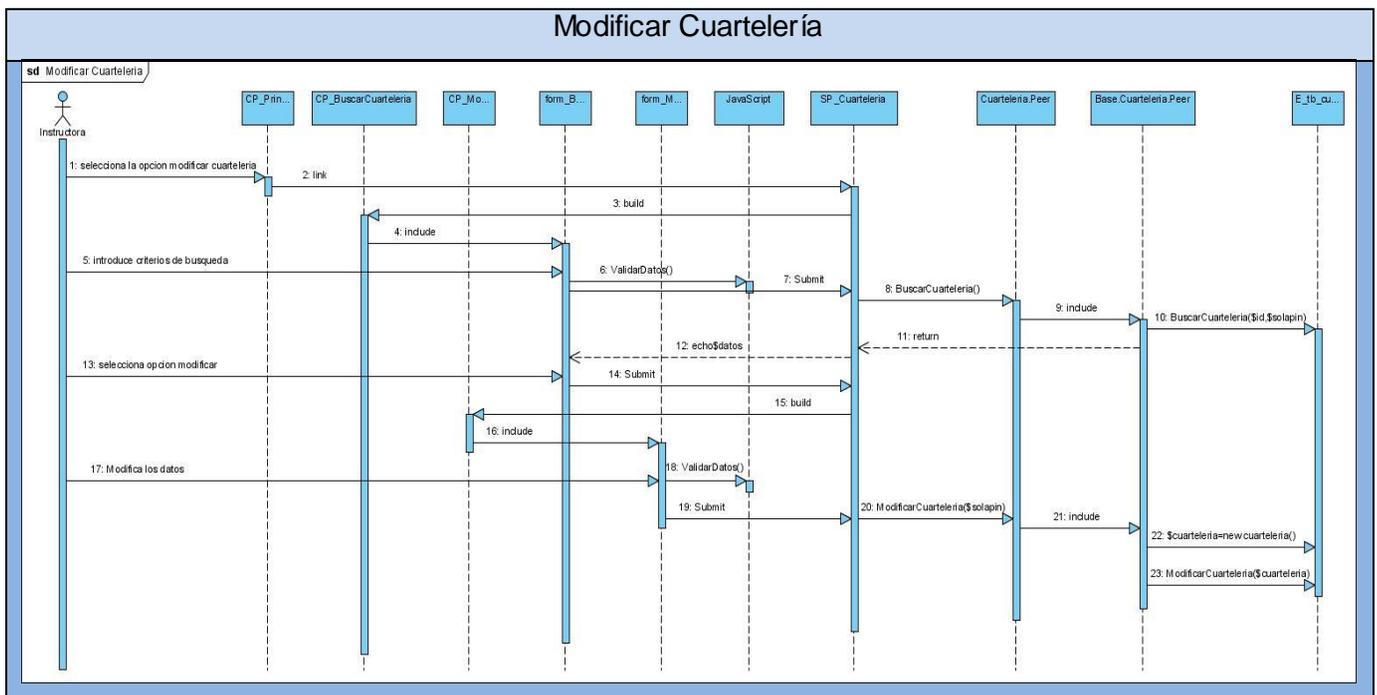


Figura 3.9. "Diagrama de Secuencia "Modificar Cuartería".

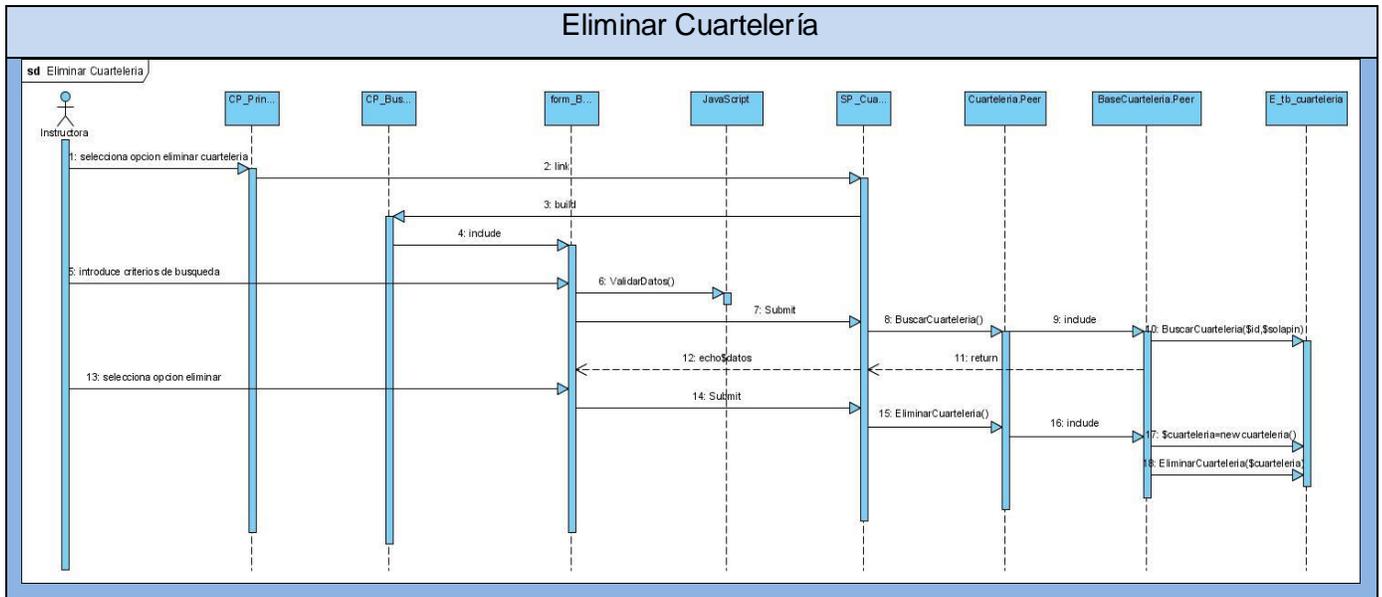


Figura 3.10. "Diagrama de Secuencia "Eliminar Cuartelería".

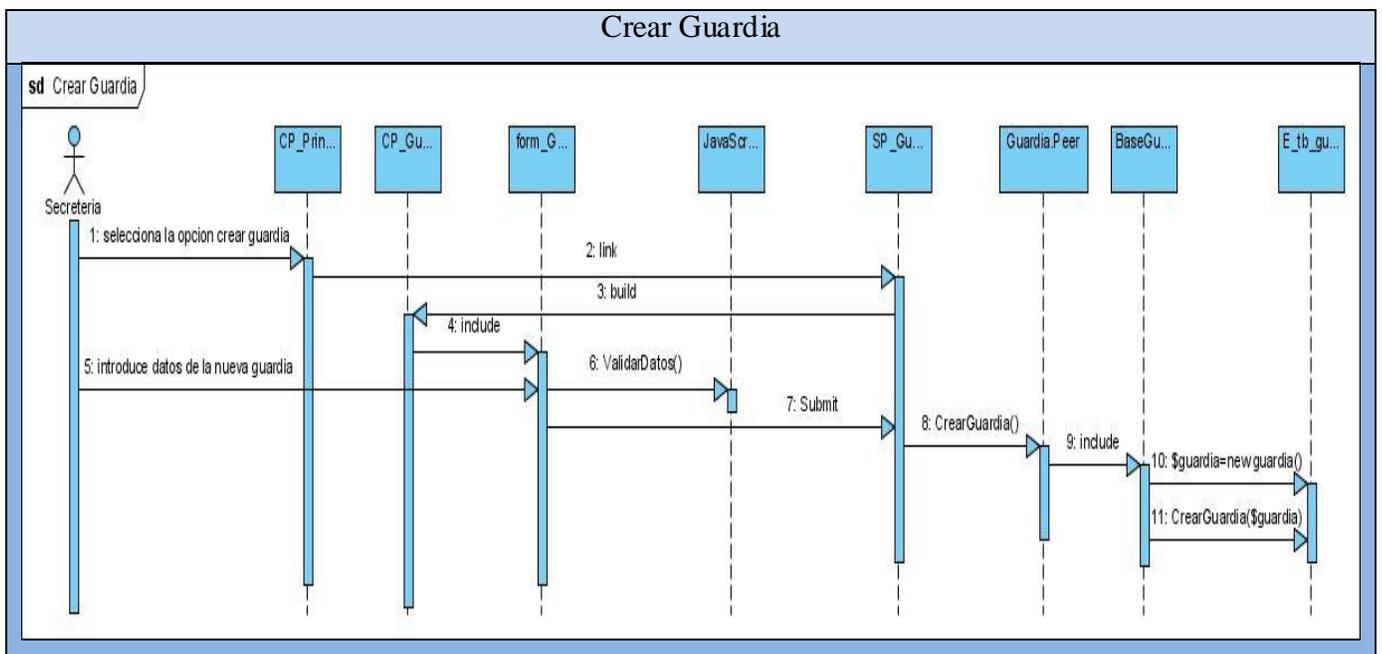


Figura 3.11. "Diagrama de Secuencia "Crear Guardia".

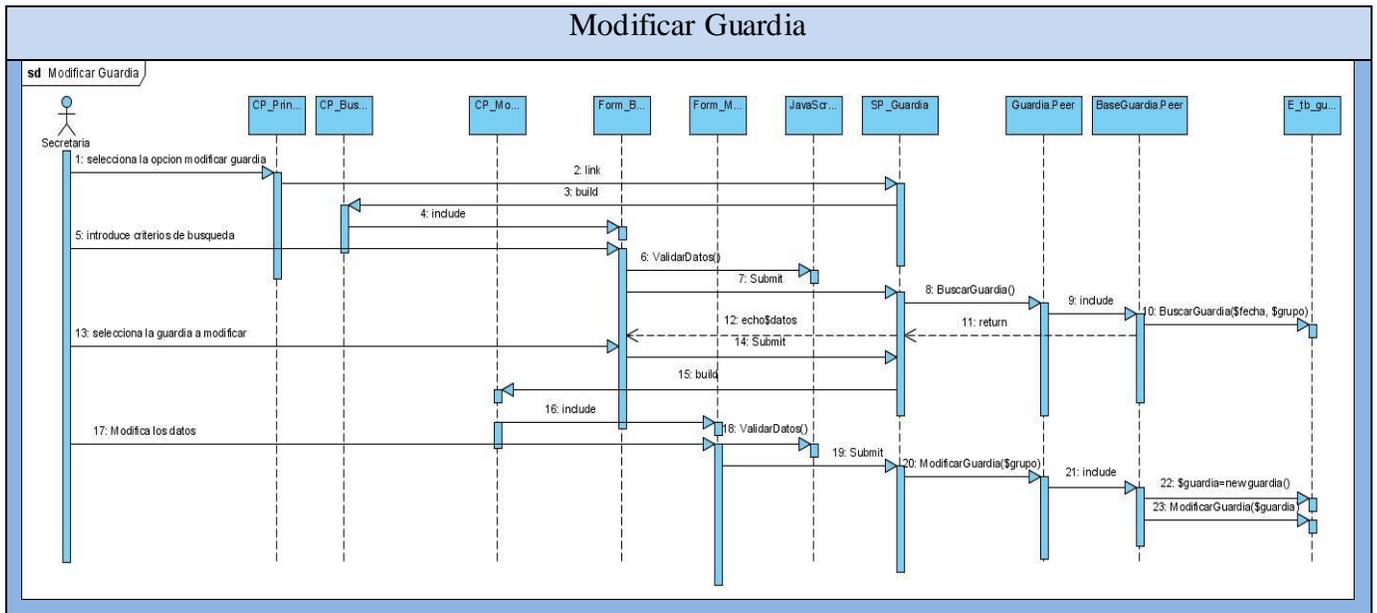


Figura 3.12. "Diagrama de Secuencia "Modificar Guardia".

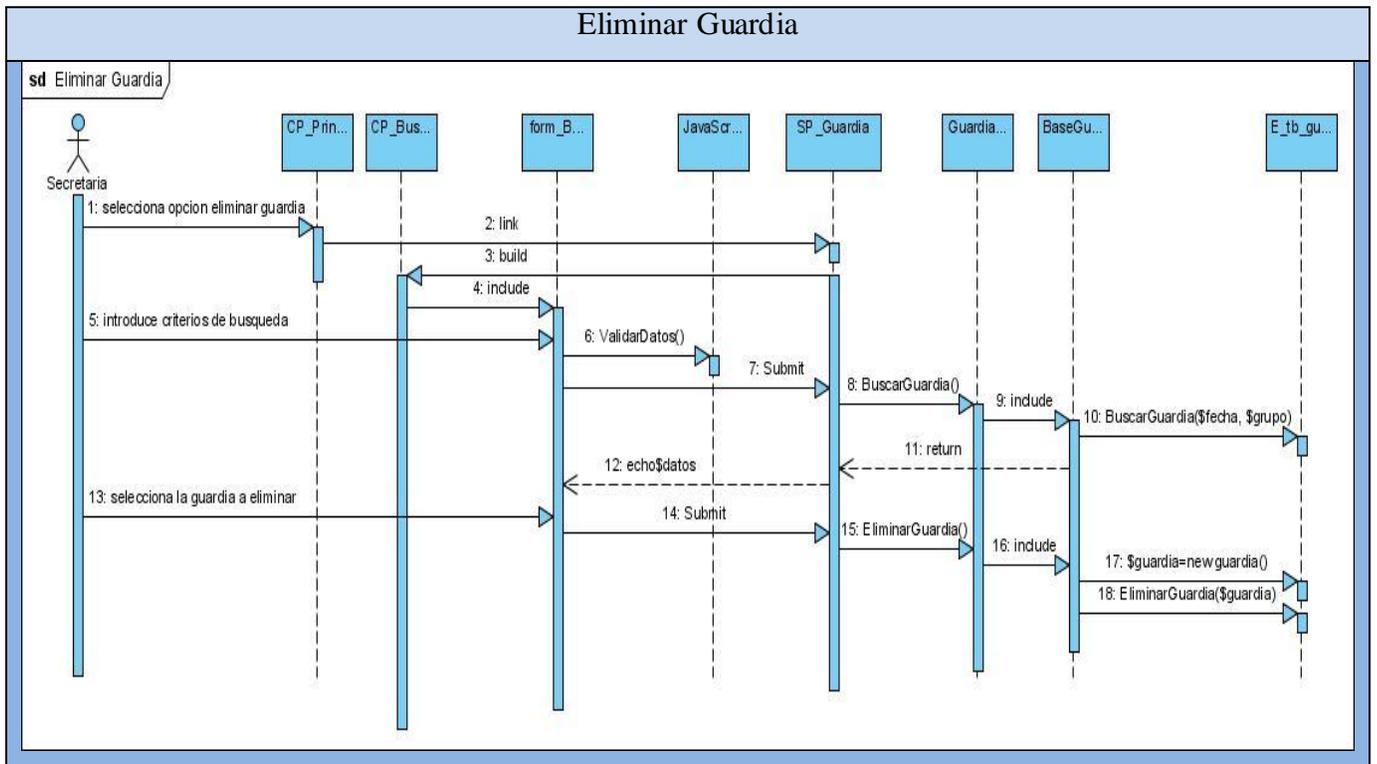


Figura 3.16. Diagrama de Secuencia "Eliminar Guardia"

3.3. Conclusiones

En este capítulo se seleccionaron un conjunto de patrones de diseño; los cuales, fueron aplicados al diseño del software que dará cumplimiento al objetivo específico planteado. Se obtuvo el Modelo de Diseño consecuente tanto con los requisitos funcionales, como con los no funcionales, con los cuales debía cumplir la aplicación web a modelar. Se obtuvieron artefactos tales como: diagramas de clases y de secuencia, los cuales se espera que sirvan como entrada principal para implementación del software, estos son de gran importancia para la implementación de la aplicación.

4. CAPÍTULO IV: IMPLEMENTACIÓN

El propósito fundamental de este capítulo es definir cómo desarrollar la arquitectura comenzando con el resultado de la etapa de Diseño e implementando las clases y subsistemas definidos en el capítulo anterior en términos de componentes. Se modelan los diagramas de componentes y de despliegue quedando conformado el modelo de implementación del sistema, dando una visión de cómo quedará construida y distribuida la aplicación

4.1. Diagrama de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final. Representa la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Se representa mediante un grafo de nodos unidos por conexiones de comunicación. En dicho diagrama se muestra una PC Cliente en donde el usuario accede a la aplicación, una PC Servidor Web que es donde estará situada la aplicación, una PC Servidor de Base de Datos en donde se encontrará la base de datos del sistema y una impresora para que el usuario tenga la posibilidad de imprimir los reportes efectuados por la aplicación.

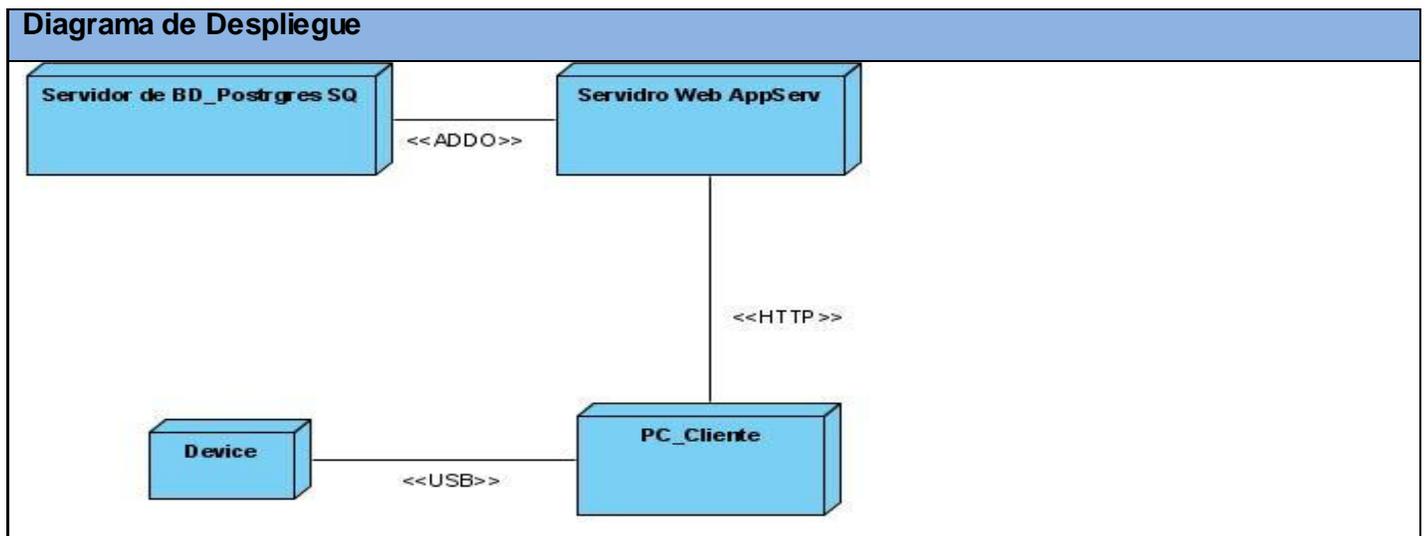


Figura 4.1. Diagrama de Despliegue

4.2. Modelo de Implementación

Un diagrama de implementación muestra las dependencias entre las partes de código del sistema (diagrama de componentes) o la estructura del sistema en ejecución (diagrama de despliegue): los diagramas de componentes se utilizan para modelar la vista de implementación estática de un sistema, mientras que los diagramas de despliegue se utilizan para modelar la vista de despliegue estática

4.3. Diagrama de componentes del Sistema

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

Los diagramas de componentes muestran los componentes software que constituyen una parte reusable, sus interfaces, y sus interrelaciones. Un paquete en un diagrama de componentes representa una división física del sistema. Los paquetes se organizan en una jerarquía de capas donde cada capa tiene una interfaz bien definida

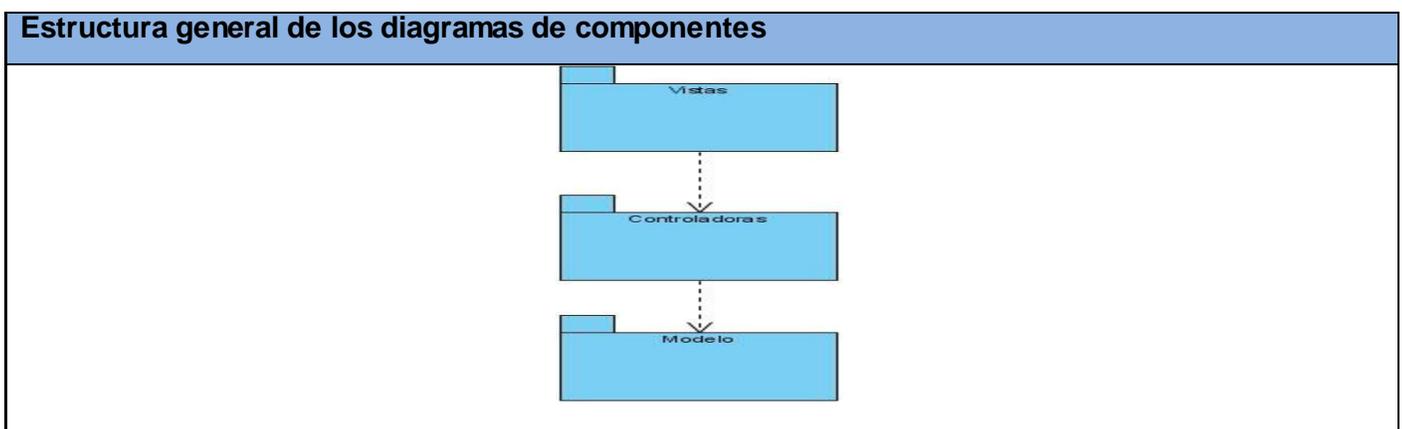


Figura 4.2. Estructura general de los diagramas de componentes

4.2.1 Diagramas de componentes por casos de uso.

Diagrama de componentes del caso de uso “Gestionar Cuartelería”.

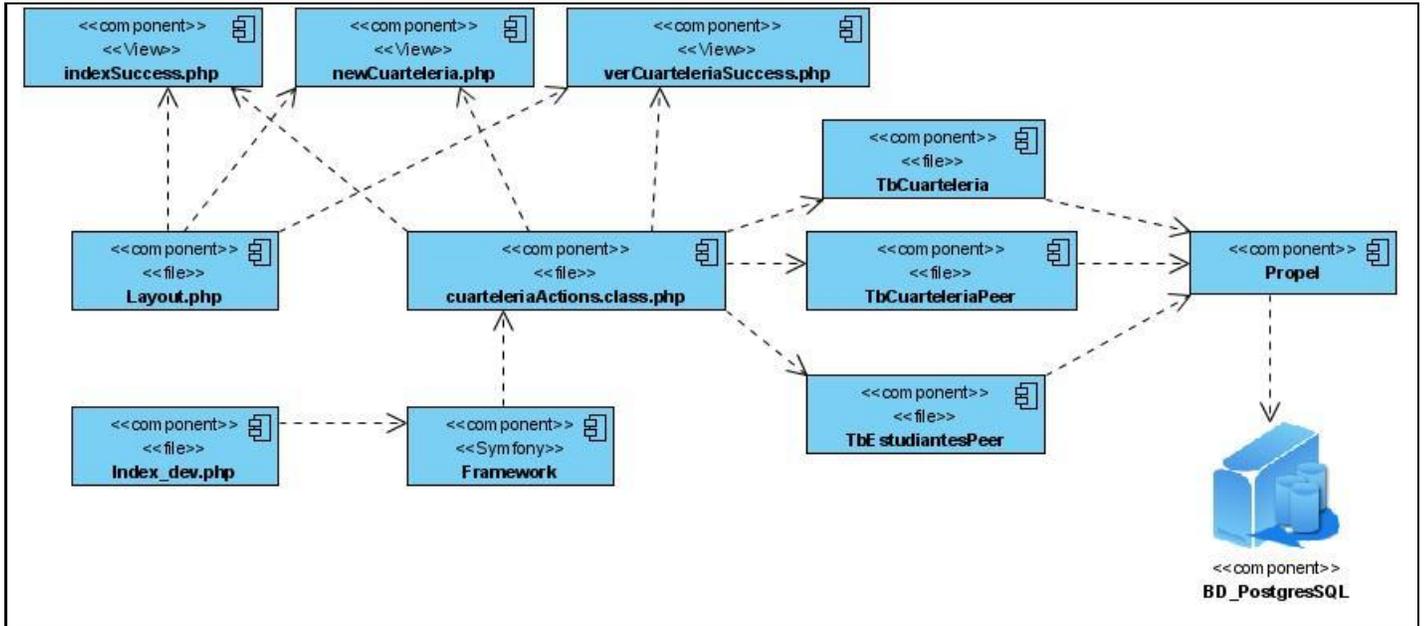


Diagrama de componentes del caso de uso “Gestionar Guardia”.

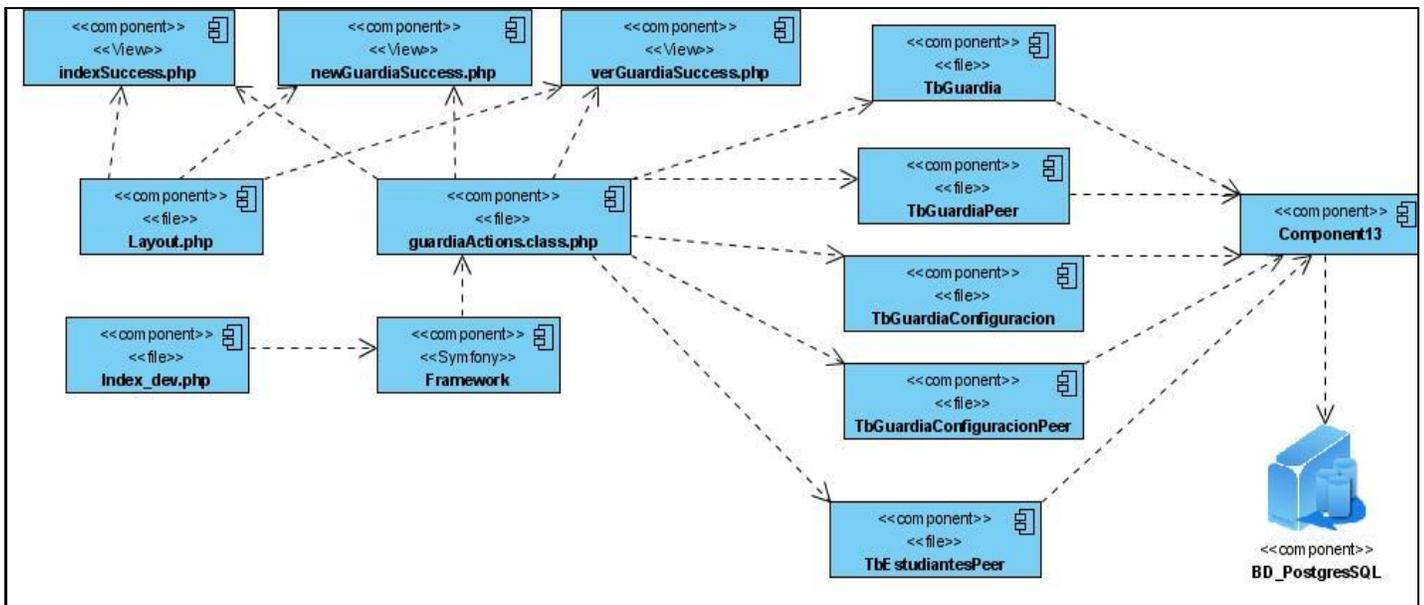
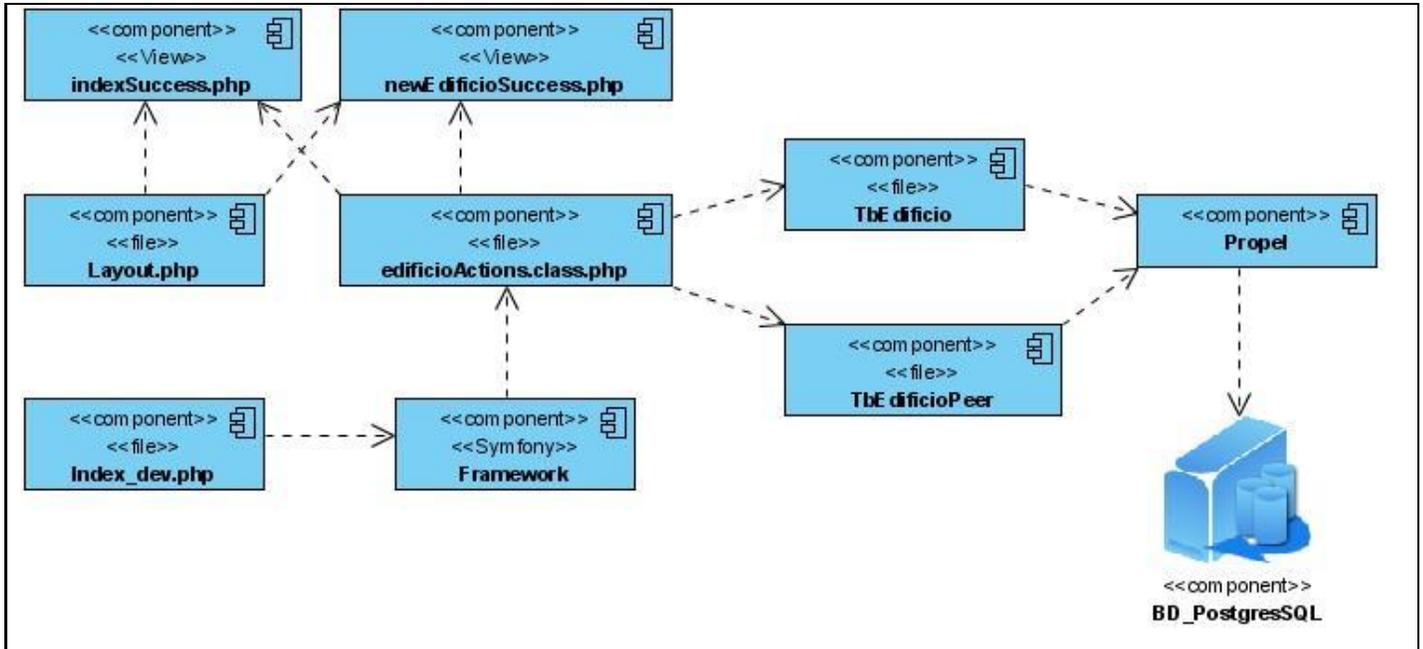


Diagrama de componentes del caso de uso “Gestionar Edificio”.



4.4. Conclusiones

En este capítulo se ha mostrado mediante el diagrama de despliegue los nodos necesarios para la utilización del sistema, garantizando que este funcione correctamente. Lo que se traduce en la utilización de una PC Cliente en donde el usuario accede a la aplicación, una PC Servidor Web que es donde estará situada la aplicación. Además de una PC Servidor de Base de Datos en donde se encontrará la base de datos del sistema y una impresora para que el usuario tenga la posibilidad de imprimir los reportes efectuados por la aplicación. También se mostró la relación existente entre todos los archivos que conforman el sistema, y su relación con la base de datos mediante el Diagrama de Componentes del Sistema.

CONCLUSIONES

Como conclusiones de la presente investigación se tienen las siguientes:

- Se obtuvieron todos los artefactos consistentes en el Modelo de negocio, Modelo de Casos de Uso del Sistema, Modelo de Diseño, Modelo de implementación, resultantes de realizar la ingeniería del software según los flujos de trabajo que propone RUP como metodología de desarrollo utilizada.
- Se obtuvo una aplicación que permite integrar un conjunto de funcionalidades, al Portal Facultad 7, logrando gestionar la información y planificación de la Guardia y la Cuartelería.

RECOMENDACIONES

Luego de haber cumplido con los objetivos propuestos se recomienda:

- Perfeccionar los criterios de búsqueda de la aplicación, haciéndolos más específicos.
- Agregarle nuevas funcionalidades al sistema, con el objetivo de que el usuario pueda obtener reportes personalizados, exportables en formato digital.

1. [En línea] [Citado el: 15 de Noviembre de 2008.]
<http://www.bsigroup.com.mx/es-mx/Auditoria-y-Cert>.
2. Idem a Referencia 1.
3. [En línea] [Citado el: 25 de Enero de 2009.]
<http://www.info-ab.uclm.es/asignaturas/42579/pdf/>.
4. [En línea] [Citado el: 11 de Abril de 2009.]
<http://www.lenguajes-de-programacion.com/lenguaje>.
5. [En línea] [Citado el: 23 de Febrero de 2009.]
<http://libertonia.escomposlinux.org/story/2004/7/15/115328/134..>
6. [En línea] [Citado el: 20 de Febrero de 2009.]
http://www.netpecos.org/docs/mysql_postgres/index.html.
7. [En línea] [Citado el: 19 de Marzo de 2009.]
<http://www.monografias.com/trabajos24/herramientas-case/herramientas-case.shtml>.
8. Idem a Referencia 1
9. Idem a Referencia 2
10. [En línea] [Citado el: 12 de Febrero de 2009.]
http://www.librosweb.es/symfony_1_2/capitulo1/symfony_en_pocas_palabras.html.
11. Idem a Referencia 10.
12. [En línea] [Citado el: 10 de Diciembre de 2008.]
<http://web2development.blogspot.com/2007/05/patron-mvc.html>.

BIBLIOGRAFÍA

1. Desarrolloweb.com. [En línea] [Citado el: 24 de Febrero de 2009.]
<http://www.desarrolloweb.com/articulos/1581.php>.
2. Dreamweaver. [En línea] [Citado el: 12 de Diciembre de 2008.]
<http://www.desarrolloweb.com/articulos/332.php>.
3. Ensemble3D. [En línea] [Citado el: 3 de Diciembre de 2008.]
<http://www.ensemble3d.com/jscript/index.html>.
4. Maestros del Web. [En línea] [Citado el: 12 de Febrero de 2009.]
<http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
5. Monografías. [En línea] [Citado el: 15 de Noviembre de 2008.]
<http://www.monografias.com/trabajos24/herramientas-case/herramientas-case.shtml>.
6. Sitios de Descargas. [En línea] [Citado el: 15 de Enero de 2009.]
[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/).
7. [En línea] [Citado el: 25 de Noviembre de 2008.] <http://biblioteca.uci.cu/sbd/biuci/index.html>.
8. [En línea] [Citado el: 20 de Noviembre de 2008.] <http://www.scribd.com/doc/297224/RUP>.
9. [En línea] [Citado el: 20 de Febrero de 2009.] <http://www.symfony.es/>.
10. [En línea] [Citado el: 16 de Noviembre de 2008.]
http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.
11. [En línea] [Citado el: 12 de Enero de 2009.]
http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html.
12. [En línea] [Citado el: 09 de Enero de 2009.]
https://developer.mozilla.org/es/Gu%C3%ADa_JavaScript_1.5/Concepto_de_JavaScript.
13. [aut. libro] Enrique Barreiro. Tema2 -ingeniería de requerimientos.

GLOSARIO DE TÉRMINOS

CASE: Computer Aided Software Engineering.

CUN: Sigla que se utiliza para referirse al término Caso de Uso del Negocio.

CUS: Sigla que se utiliza para referirse al término Caso de Uso del Sistema.

Hardware: Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.

HTML: Lenguaje usado para escribir documentos para servidores World Wide Web

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.

TCP/IP: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

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.

USB: Interfaz estándar que facilita la conexión de periféricos a un ordenador.

ANEXOS

INICIO LA FACULTAD FORMACIÓN PRODUCCIÓN **EXTENSIÓN UNIVERSITARIA** ORGANIZACIONES SERVICIOS

Movimiento de Artistas Aficionados | Deportes | Trabajo Socialmente Útil | Emulación | **Residencia y Guardia Estudiantil**

[Inicio](#) > [Extensión Universitaria](#) > [Residencia y Guardia Estudiantil](#)

Residencia y Guardia Estudiantil

Funcionalidades

- Gestionar Edificios
- Gestionar Ubicación
- Gestionar Cuartelería
- Gestionar Evaluación
- Gestionar Guardia
- Editar Roles
- Reportes**
- Planificación Mensual de la Guardia
- Cuartelería Semanal por Edificio
- Evaluación de los Estudiantes en la Cuartelería por

Numero edificio	Estudiante	Fecha	Evaluacion
54	Jorge Moreira Rodriguez	2009-06-11 00:00:00	Seleccione
79	Eduardo Merino Hernández	2009-06-10 00:00:00	Mal
79	Rafael Fernández Peñalver	2009-06-15 00:00:00	Bien
81	Claudia Verdés González	2009-06-19 00:00:00	Seleccione
81	Hilda María Rodriguez Gómez	2009-06-10 00:00:00	Bien
81	María de Jesús Cartelle Cruz	2009-06-11 00:00:00	Regular
81	YARIELYS HERNANDEZ FONTICIELLA	2009-06-10 00:00:00	Muy Bien

Página 1 de 1 (7 filas)

[Noticias](#) | [Fotoreportajes](#) | [Misión Ambiental](#) | [Defensa Civil](#) | [Eventos](#) | [Contactos](#) | [Temas Calientes](#) | [Nacionales](#) | [Internacionales](#)

Copyright © 2009 Open Source Matters. Todos los derechos reservados.
Diseñada por [Ing. William Sófiora Cruz](#).

Local intranet 100%

Fig. 1. "Evaluación de los estudiantes en la cuartelería"

INICIO LA FACULTAD FORMACIÓN PRODUCCIÓN **EXTENSIÓN UNIVERSITARIA** ORGANIZACIONES SERVICIOS

Movimiento de Artistas Aficionados | Deportes | Trabajo Socialmente Útil | Emulación | **Residencia y Guardia Estudiantil**

Inicio > Extensión Universitaria > Residencia y Guardia Estudiantil

Residencia y Guardia Estudiantil

Funcionalidades

- Gestionar Edificios
- Gestionar Ubicación
- Gestionar Cuartelería
- Gestionar Evaluación
- Gestionar Guardia
- Editar Roles
- Reportes**
- Planificación Mensual de la Guardia
- Cuartelería Semanal por Edificio
- Evaluación de los Estudiantes en la Cuartelería por Grupo

Nueva Reporte

Todos Por Grupos

Numero edificio	Estudiante	Grupo	Fecha	Evaluacion
54	Jorge Moreira Rodriguez	07304	2009-06-11 00:00:00	
79	Eduardo Merino Hernández	07302	2009-06-10 00:00:00	Mal
79	Rafael Fernández Peñalver	07502	2009-06-15 00:00:00	Bien
81	Claudia Verdés González	07208	2009-06-19 00:00:00	
81	Hilda María Rodríguez Gómez	07304	2009-06-10 00:00:00	Bien
81	Maria de Jesús Cartelle Cruz	07304	2009-06-11 00:00:00	Regular
81	YARIELYS HERNANDEZ FONTICIELLA	EG7308	2009-06-10 00:00:00	Muy Bien

Pagina 1 de 1 (7 filas)

Noticias | Fotoreportajes | Misión Ambiental | Defensa Civil | Eventos | Contactos | Temas Calientes | Nacionales | Internacionales

Copyright © 2009 Open Source Matters. Todos los derechos reservados.
Diseñada por Ing. William Sófiora Cruz.

Done Local intranet 100%

Fig.2 “Evaluación de los estudiantes en la cuartelería por grupo”

INICIO | LA FACULTAD | FORMACIÓN | PRODUCCIÓN | **EXTENSIÓN UNIVERSITARIA** | ORGANIZACIONES | SERVICIOS

Movimiento de Artistas Aficionados | Deportes | Trabajo Socialmente Útil | Emulación | **Residencia y Guardia Estudiantil**

Inicio > Extensión Universitaria > Residencia y Guardia Estudiantil

Residencia y Guardia Estudiantil

Funcionalidades

- Gestionar Edificios
- Gestionar Ubicación
- Gestionar Cuartelería
- Gestionar Evaluación
- Gestionar Guardia
- Editar Roles
- Reportes**
- Planificación Mensual de la Guardia
- Cuartelería Semanal por Edificio
- Evaluación de los Estudiantes en la Cuartelería por

Nuevas Ubicaciones

Edificios *

Apartamentos

Grupos *

Estudiantes

Nombre y Apellidos	Grupo	Apartamento	Acciones
Arlis Antonio Capdesuñer Lorient	07203	80101	
Claudia Verdés González	07208	81106	

[Noticias](#) | [Fotoreportajes](#) | [Misión Ambiental](#) | [Defensa Civil](#) | [Eventos](#) | [Contactos](#) | [Temas Calientes](#) | [Nacionales](#) | [Internacionales](#)
 Copyright © 2009 Open Source Matters. Todos los derechos reservados.
 Diseñada por [Ing. William Sófiora Cruz](#).

Local intranet 100%

Fig.3. "Gestionar ubicación de estudiantes"

INICIO LA FACULTAD FORMACIÓN PRODUCCIÓN **EXTENSIÓN UNIVERSITARIA** ORGANIZACIONES SERVICIOS

Movimiento de Artistas Aficionados | Deportes | Trabajo Socialmente Útil | Emulación | **Residencia y Guardia Estudiantil**

Inicio > Extensión Universitaria > Residencia y Guardia Estudiantil

Residencia y Guardia Estudiantil

Funcionalidades

- Gestionar Edificios
- Gestionar Ubicación
- Gestionar Cuartelería
- Gestionar Evaluación
- Gestionar Guardia
- Editar Roles
- Reportes**
- Planificación Mensual de la Guardia
- Cuartelería Semanal por Edificio
- Evaluación de los Estudiantes en la Cuartelería por

Nueva Guardia

Grupos *

Este campo es requerido!

Fecha * ...

Grupo	Fecha	Acciones
07203	2009-06-05	<input type="checkbox"/>
07203	2009-06-17	<input type="checkbox"/>
07507	2009-06-11	<input type="checkbox"/>

Pagina 1 de 1 (3 filas)

[Noticias](#) | [Fotoreportajes](#) | [Misión Ambiental](#) | [Defensa Civil](#) | [Eventos](#) | [Contactos](#) | [Temas Calientes](#) | [Nacionales](#) | [Internacionales](#)
 Copyright © 2009 Open Source Matters. Todos los derechos reservados.
 Diseñada por Ing. William Sófiora Cruz.

Done Local intranet 100%

Fig.4. "Gestionar guardia"