

Universidad de las Ciencias Informáticas

Facultad # 6



Titulo: “Sistema para la gestión de la información de profesores y estudiantes de la facultad: Desarrollo del Módulo Residencia.”

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores:

Yeneisi Jaime Yglesias.

Yarina Marlones Hidalgo.

Tutores:

Ing. Yadira Barroso Rodríguez. UCI

Ing. Yoendris Lacoste Ricardo. UCI

Ciudad de La Habana, Cuba, 2009

“Año del 50 Aniversario del Triunfo de la Revolución”

“SI UNA PERSONA ES PERSEVERANTE, AUNQUE SEA DURA DE ENTENDIMIENTO,
SE HARÁ INTELIGENTE; Y AUNQUE SEA DÉBIL SE TRANSFORMARÁ EN FUERTE.”

LEONARDO DA VINCI

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yeneisi Jaime Yglesias (Autor)

Yarina Marlones Hidalgo (Autor)

Ing. Yadira Barroso Rodríguez (Tutor)

Ing. Yoendris Lacoste Ricardo (Tutor)

DATOS DE CONTACTO

Yadira Barroso Rodríguez:

Ingeniera en Ciencias Informática. Profesora en adiestramiento, con 1 año de graduada.
ybarroso@uci.cu

Yoendris Lacoste Ricardo:

Ingeniero en Ciencias Informática. Profesor en adiestramiento, con 1 año de graduado.
ylacoste@uci.cu

AGRADECIMIENTOS

A nuestros padres porque todo lo que somos es el fruto de su sacrificio, amor infinito y apoyo incondicional.

A Persy por las horas que nos dedicó a dar los primeros pasos en Symphony.

A nuestros compañeros de proyecto por la ayuda que nos brindaron cuando lo necesitamos, gracias a todos.

A nuestros tutores Yadira y Yoendris por la ayuda brindada.

A la Revolución por habernos dado la oportunidad de estudiar en la Universidad de las Ciencias Informáticas y formarnos como profesionales.

A todos los que de una forma u otra han contribuido a nuestra formación como profesionales.

DEDICATORIA

De: Yarina Marlones Hidalgo

A mis queridísimos padres por ser quienes me han permitido llegar hasta aquí, gracias por todo el sacrificio para darme una buena educación, Gran parte de mi triunfo se los debo a ustedes:

A mi madre y hermana por ser las amigas que siempre han estado presentes en los buenos y malos momentos, por ser mis guías en este laberinto que es la vida.

A mi padre por estar siempre que lo necesite, y eternamente estar dispuesto a complacerme.

A mi familia por todo lo que han hecho por mí, en especial a mi tío Manuel Hidalgo por ser la fuerza que me hace levantar la cabeza y continuar adelante, aun en tiempos de debilidad.

A mi novio por toda la paciencia que ha tenido conmigo en estos tiempos difíciles, por su amor, comprensión y apoyo incondicional.

A Odalis y Santa por estar ahí, siempre que las necesite, brindándome su cariño y ayuda.

A todas aquellas personas que hicieron posible que este momento llegara. Sé que hoy cumplo el anhelo de todas esas personas que me quieren. Y trataré de no defraudarlos. Gracias por que sin su ayuda no habría podido llegar hasta aquí.

De: Yeneisi Jaime Yglesias

A quienes se merecen todo mi respeto, admiración y sobre todo un cariño infinito que brota de mi corazón: a mis padres que incondicionalmente me han apoyado en cada etapa de mi vida siendo ante todo mis mejores amigos, por entregarme todo el amor y la confianza que necesito para alcanzar un sueño y por darme fuerzas en momentos difíciles haciéndome saber que confían en mí, los quiero mucho.

A mi personita favorita, que cada día me enorgullece con cada paso que da, Adi, eres mi angelito, te quiero mucho.

A mi novio Adnier por ser tan especial para mí, por la paciencia con que ha sabido tratarme en cada momento que todo parecía desmoronarse, por el cariño de cada día, porque ante todas las cosas ha sabido ser mi amigo, te quiero.

A Rasmila y Víctor por su apoyo y ayuda incondicionales y a Danielita por alegrarme los días con cada ocurrencia.

A mis amigas: Yuri, Kíl, Mirelis, Yas, May, K, Lisset, Isis, sin ustedes no hubiesen sido estos 5 años de la universidad unos de los mejores de mi vida, gracias por todo su tiempo, sus consejos, su confianza, por las fiestas, por las no fiestas, por estar ahí cuando más lo necesité, porque cada una de ustedes me ha enseñado día a día cosas que nunca olvidaré. Las voy a extrañar muchísimo.

A mi amiga Yeny, mi camagüeyana favorita.

A todos aquellos que siempre confiaron en mí y que me brindaron su apoyo constante en todo momento.

RESUMEN

En el presente trabajo se realiza el análisis, diseño e implementación de las funcionalidades del Módulo Residencia del Sistema para la gestión de la Información de profesores y estudiantes de la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI), con el objetivo de proporcionarle al Vicedecanato de Residencia de la facultad la posibilidad de gestionar los datos de las cuarterías, trabajos socialmente útiles, guardias obreras estudiantiles, paradas de beca, las afectaciones de los edificios y las evaluaciones de los estudiantes referente a los procesos que tienen lugar en la residencia estudiantil de la facultad 6. El desarrollo de este módulo constituye una manera más rápida y eficiente de llevar el control y seguimiento de las actividades realizadas en la residencia estudiantil, de tal forma que la búsqueda de una información determinada sobre ellas sea más rápida y eficiente. Primero se analiza como es el funcionamiento de las actividades que se desarrollan en el área de residencia de la universidad, se explican las tecnologías, lenguajes y herramientas seleccionadas para el desarrollo de la aplicación y luego se exponen las características del sistema que se propone. Una vez desarrollado el mismo, este contribuirá a mejorar la gestión de la información de la facultad 6.

PALABRAS CLAVES

Cuarterías, trabajos socialmente útiles, guardias obreras estudiantiles, paradas de beca, área de residencia, gestión de la información.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
INTRODUCCIÓN.....	4
1.1-GESTIÓN DE LA INFORMACIÓN.....	4
1.2-SISTEMAS DE GESTIÓN DE LA INFORMACIÓN.....	4
1.3-SISTEMAS DE GESTIÓN DE LA INFORMACIÓN SIMILARES.....	5
1.4- METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	7
1.5-ROLES Y ARTEFACTOS.....	10
1.6-TECNOLOGÍAS, LENGUAJES Y HERRAMIENTAS PARA EL DESARROLLO.....	14
1.7- PATRONES.....	21
CONCLUSIONES.....	25
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	26
INTRODUCCIÓN.....	26
2.1-CARACTERIZACIÓN DEL PROBLEMA.....	26
2.2-OBJETO DE INFORMATIZACIÓN.....	26
2.3-PROPUESTA DEL SISTEMA.....	27
2.4-MODELAMIENTO DEL NEGOCIO.....	27
2.5-LEVANTAMIENTO DE REQUISITOS.....	48
2.6-MODELAMIENTO DEL SISTEMA.....	51
2.7-PAUTAS DEL DISEÑO.....	59
2.8-MATRIZ DE TRAZABILIDAD.....	60
CONCLUSIONES.....	60
CAPÍTULO 3: DISEÑO DEL SISTEMA.....	62
INTRODUCCIÓN.....	62
3.1-APLICACIÓN DE PATRONES.....	62
3.2-MODELO DE DISEÑO.....	67
3.3-DIAGRAMAS DE INTERACCIÓN.....	77

3.4-DIAGRAMA DESPLIEGUE	86
3.5-DIAGRAMA DE CLASES PERSISTENTES	88
3.6-MODELO DE DATOS.....	90
3.7-MAPA DE NAVEGACIÓN.	91
CONCLUSIONES.....	92
CAPITULO 4: IMPLEMENTACIÓN DEL SISTEMA.....	93
INTRODUCCIÓN.....	93
4.1-DIAGRAMAS DE COMPONENTES	93
4.2-EJEMPLOS DE MÉTODOS Y SU DESCRIPCIÓN.....	96
4.3-VALIDACIÓN DE LA APLICACIÓN.....	99
CONCLUSIONES.....	101
CONCLUSIONES GENERALES	102
RECOMENDACIONES.....	103
REFERENCIAS BIBLIOGRÁFICAS.....	104
BIBLIOGRAFÍA.....	106
ANEXO.1-DESCRIPCIONES EXTENDIDAS DE CUS	107
ANEXO.2-DIAGRAMA DE CLASES DEL DISEÑO.....	161
ANEXO.3-DIAGRAMAS DE SECUENCIAS.....	168
ANEXO.4-DIAGRAMA DE COMPONENTES	207
GLOSARIO DE TÉRMINOS	214

ÍNDICE DE FIGURAS

FIGURA 1 RUP EN DOS DIMENSIONES	8
FIGURA 2 LOS CASOS DE USO INTEGRAN EL TRABAJO.....	9
FIGURA 3 PATRÓN MÚLTIPLES ACTORES. ROLES COMUNES.....	22
FIGURA 4 EL PATRÓN MVC.....	25
FIGURA 5 DIAGRAMA DE CASOS DE USO DEL NEGOCIO	30
FIGURA 6 DIAGRAMA DE ACTIVIDADES CUN “REALIZAR CUARTELARÍAS”	39
FIGURA 7 DIAGRAMA DE ACTIVIDADES CUN “SOLICITAR EVALUACIÓN DE LOS ESTUDIANTES EN LA RESIDENCIA”	40
FIGURA 8 DIAGRAMA DE ACTIVIDADES CUN “REALIZAR TRABAJO SOCIALMENTE ÚTIL”	41
FIGURA 9 DIAGRAMA DE ACTIVIDADES CUN “REALIZAR GUARDIA OBRERA ESTUDIANTIL”	42
FIGURA 10 DIAGRAMA DE ACTIVIDADES CUN “SOLICITAR AFECTACIONES DE LOS APARTAMENTOS”	43
FIGURA 11 DIAGRAMA DE ACTIVIDADES CUN “SOLICITAR UBICACIÓN DE LOS ESTUDIANTES”	44
FIGURA 12 DIAGRAMA DE ACTIVIDADES CUN “REALIZAR PARADAS DE BECA”	45
FIGURA 13 DIAGRAMA DE OBJETOS	46
FIGURA 14 DIAGRAMA DE CUS	53
FIGURA 15 MATRIZ DE TRAZABILIDAD	60
FIGURA 16 PLANTILLA DECORADA CON UN LAYOUT.....	65
FIGURA 17 DIAGRAMA DE CLASES DEL DISEÑO DEL CU:”GESTIONAR EDIFICIO”	75
FIGURA 18 DIAGRAMA DE CLASES DEL DISEÑO DEL CU:”GESTIONAR TSU”	76
FIGURA 19 DIAGRAMA DE CLASES DEL DISEÑO: “GESTIONAR UBICACIÓN DEL ESTUDIANTE EN LA RESIDENCIA”	77
FIGURA 20 ESCENARIO: INSERTAR DATOS DEL EDIFICIO	78
FIGURA 21 ESCENARIO: GENERAR REPORTE DE LOS EDIFICIOS	79
FIGURA 22 ESCENARIO: MODIFICAR PLANIFICACIÓN DEL TRABAJO SOCIALMENTE ÚTIL	80
FIGURA 23 ESCENARIO: ELIMINAR PLANIFICACIÓN DEL TRABAJO SOCIALMENTE ÚTIL.....	81
FIGURA 24 ESCENARIO” INSERTAR DATOS DE LA UBICACIÓN DEL ESTUDIANTE”	82
FIGURA 25 ESCENARIO” MODIFICAR DATOS DE LA UBICACIÓN DEL ESTUDIANTE”	83
FIGURA 26 ESCENARIO” ELIMINAR DATOS DE LA UBICACIÓN DEL ESTUDIANTE”	84
FIGURA 27 ESCENARIO” VISUALIZAR DATOS DE LA UBICACIÓN DEL ESTUDIANTE”	85

FIGURA 28 ESCENARIO" BUSCAR DATOS DE LA UBICACIÓN DEL ESTUDIANTE"	86
FIGURA 29 DIAGRAMA DE CLASES PERSISTENTES	89
FIGURA 30 MODELO DE DATOS.....	90
FIGURA 31 MAPA DE NAVEGACIÓN	91
FIGURA 32 DIAGRAMA DE COMPONENTE CU: GESTIONAR EDIFICIOS.....	94
FIGURA 33 DIAGRAMA DE COMPONENTE CU: GESTIONAR TRABAJO SOCIALMENTE ÚTIL.....	95
FIGURA 34 DIAGRAMAS DE COMPONENTES CU:" GESTIONAR DATOS DE LA UBICACIÓN DEL ESTUDIANTE UBICACIÓN"	96
FIGURA 36 MÉTODO: EXECUTEOBTENERESTUDIANTEGRUPO ().....	97
FIGURA 37 MÉTODO: EXECUTEINSERTARUBICACIÓNESTUDIANTE ().....	98
FIGURA 39 MÉTODO: EXECUTEMOSTRARAUSENTES()	99

ÍNDICE DE TABLAS

TABLA 1 ACTORES DEL NEGOCIO	28
TABLA 2 TRABAJADORES DEL NEGOCIO	29
TABLA 3 DESCRIPCIÓN DEL CUN “REALIZAR CUARTELERÍA”	32
TABLA 4 DESCRIPCIÓN DEL CUN “SOLICITAR EVALUACIÓN DE LOS ESTUDIANTES EN LA RESIDENCIA”	33
TABLA 5 DESCRIPCIÓN DEL CASO DE USO “REALIZAR TRABAJO SOCIALMENTE ÚTIL”	34
TABLA 6 DESCRIPCIÓN DEL CUN “REALIZAR GUARDIA OBRERA ESTUDIANTIL”	35
TABLA 7 DESCRIPCIÓN DEL CUN “SOLICITAR AFECTACIONES DE LOS APARTAMENTOS”	37
TABLA 8 DESCRIPCIÓN DEL CUN “SOLICITAR UBICACIÓN DE LOS ESTUDIANTES”	37
TABLA 9 DESCRIPCIÓN DEL CUN” REALIZAR PARADAS DE BECA”	38
TABLA 10 DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA	52
TABLA 11 DESCRIPCIÓN DEL CUS” GESTIONAR EDIFICIOS”	54
TABLA 12 DESCRIPCIÓN DEL CUS” GESTIONAR APARTAMENTOS”	55
TABLA 13 DESCRIPCIÓN DEL CUS” GESTIONAR DATOS DE UBICACIÓN DEL ESTUDIANTE”	55
TABLA 14 DESCRIPCIÓN DEL CUS” GESTIONAR DATOS DE LA INSTRUCTORA”	55
TABLA 15 DESCRIPCIÓN DEL CUS” GESTIONAR CUARTELERÍA”	56
TABLA 16 DESCRIPCIÓN DEL CUS” GESTIONAR GUARDIA OBRERA ESTUDIANTIL”	56
TABLA 17 DESCRIPCIÓN DEL CUS” GESTIONAR TRABAJO SOCIALMENTE ÚTIL”	57
TABLA 18 DESCRIPCIÓN DEL CUS” GESTIONAR AFECTACIONES DE LOS APARTAMENTOS”	57
TABLA 19 DESCRIPCIÓN DEL CUS” GESTIONAR EVALUACIÓN DEL ESTUDIANTE”	58
TABLA 20 DESCRIPCIÓN DEL CUS” GESTIONAR PARADA DE BECA”	58
TABLA 21 DESCRIPCIÓN DEL CU” GESTIONAR EDIFICIO”	68
TABLA 22 DESCRIPCIÓN DEL CU” GESTIONAR APARTAMENTO”	68
TABLA 23 DESCRIPCIÓN DEL CU” GESTIONAR UBICACIÓN DEL ESTUDIANTE”	69
TABLA 24 DESCRIPCIÓN DEL CU” GESTIONAR DATOS DE LA INSTRUCTORA”	69
TABLA 25 DESCRIPCIÓN DEL CU” GESTIONAR CUARTELERÍA”	70
TABLA 26 DESCRIPCIÓN DEL CU” GESTIONAR GUARDIA OBRERA ESTUDIANTIL”	71
TABLA 27 DESCRIPCIÓN DEL CU” GESTIONAR TRABAJO SOCIALMENTE ÚTIL”	72
TABLA 28 DESCRIPCIÓN DEL CU” GESTIONAR EVALUACIÓN DEL ESTUDIANTE”	73
TABLA 29 DESCRIPCIÓN DEL CU” GESTIONAR AFECTACIONES DE LOS APARTAMENTOS”	73
TABLA 30 DESCRIPCIÓN DEL CU” GESTIONAR PARADA DE BECA”	75

INTRODUCCIÓN

La Informatización de la Sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones en la vida cotidiana, para satisfacer las necesidades de todas las esferas de la sociedad, en su esfuerzo por lograr cada vez, más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos. [1]

Cuba ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social, las Tecnologías de la Información y las Comunicaciones por lo que actualmente, ha estado enmarcada en la estrategia de lograr la informatización de las instituciones cubanas, para lograr más eficiencia en el desarrollo de sectores productivos, la salud, la actividad pública y la educación, en este sentido se han creado los Politécnicos de Informática, Los Joven Club de Computación y la Universidad de las Ciencias Informáticas(UCI).La cual juega un papel importante en la materialización de los proyectos asociados al programa cubano de informatización planteado con anterioridad, siendo partícipe de muchos eventos relacionados con la informatización e investigación, y en algunos de los casos la protagonista.

De esta manera se crea una cultura digital en la población como paso fundamental de la sociedad hacia el objetivo de lograr un desarrollo sostenible.

La Universidad de las Ciencias Informáticas (UCI) emerge como una de las instituciones que mayor expectativa asegura en el orden económico, político y social de nuestro país. Este proyecto, concebido como la primera universidad surgida al calor de la Batalla de Ideas, tiene características que la distinguen como única de su tipo en el Sistema educacional cubano. Caracterizada por su dinamismo, hace de la comunidad universitaria un ente intranquilo y enérgico; por su flexibilidad es totalmente adaptable al cambio; por su diversidad se convierte en un termómetro de disímiles indicadores; y por su extensión y concurrencia de procesos es una fiel muestra de una ciudad del futuro.

Hacer de la Universidad de las Ciencias Informáticas una ciudad donde todos y cada uno de los procesos estén sustentados sobre una infraestructura tecnológica capaz de responder a las necesidades de los que aquí conviven, es una tarea en la que directivos, trabajadores y estudiantes están inmersos desde sus inicios. Es por este motivo que la vicerrectoría de residencia de la UCI en coordinación con las facultades docentes y las organizaciones estudiantiles se encarga de organizar las actividades que se llevan a cabo en la residencia estudiantil con el objetivo de mantener en buen estado esta área de la universidad.

La facultad 6 de la UCI no está exenta de ello, ya que posee una matrícula de alrededor de 853 estudiantes que conviven a diario, de ahí la necesidad de tener un control sobre cada una de las actividades que tienen lugar en esta área. Sin embargo la facultad no cuenta con un Sistema que permita controlar los procesos actuales de la residencia, dígase la cuartelería, la guardia obrera estudiantil, el trabajo socialmente útil, las paradas de beca, la ubicación de los estudiantes.

Los procesos planteados con anterioridad se llevan manuales y/o a través de la herramienta Microsoft Office Excel y Microsoft Word, lo que trae como consecuencia:

- El volumen de información es cada vez mayor, lo cual provoca que la búsqueda de la misma en determinado momento se torne un poco lenta.
- No existe la posibilidad de que varios usuarios puedan acceder a la información almacenada simultáneamente, lo que provoca en muchas ocasiones un atraso a la hora de contar con la información necesaria en tiempo.
- La información no se encuentra totalmente centralizada, es decir, se envían informaciones por correo electrónico o se hacen llegar por medio de algún dispositivo de almacenamiento, lo cual puede provocar que dicha información sea consultada por personas no autorizadas.

Analizando lo anteriormente planteado se define como **problema científico**: ¿Cómo mejorar la gestión de los procesos de la residencia estudiantil de la facultad 6?

Como **objeto de estudio** se define: Gestión de información de la residencia estudiantil, y como **campo de acción**: Proceso de desarrollo de software para la gestión de la información de la residencia estudiantil en la facultad 6.

El **objetivo general de la investigación** es: Desarrollar el módulo Residencia del Sistema para la gestión de la información de profesores y estudiantes de la facultad 6.

Como **objetivos específicos**:

- Identificar las funcionalidades que debe cumplir el sistema.
- Analizar el módulo Residencia.
- Diseñar el módulo Residencia.
- Implementar los componentes del módulo Residencia.

Para dar cumplimiento a estos objetivos se realizarán las siguientes **tareas**:

- Estudio de la residencia estudiantil de la facultad 6.
- Estudio de sistemas similares a la gestión de la información de la residencia.
- Asimilación de las tecnologías seleccionadas por el grupo de arquitectura.
- Realización de entrevistas con el cliente para capturar los requisitos de la aplicación.
- Análisis y Diseño del módulo Residencia.
- Implementación de los componentes del módulo Residencia.
- Validación de la aplicación a nivel de desarrollador.

El documento está estructurado de la siguiente forma:

Capítulo 1: Fundamentación Teórica. El capítulo incluye un análisis de los sistemas de gestión de la información existentes, se mencionan las tecnologías, metodologías y herramientas que serán usadas para dar solución al problema planteado, se exponen los roles que se desempeñan en el proceso de desarrollo y los artefactos que estos realizan, así como una breve descripción de los patrones de diseño y arquitectura que se utilizan.

Capítulo 2: Características del Sistema. Contiene los requerimientos funcionales y no funcionales que darán solución al problema planteado. Además se identifican los casos de uso y los actores del sistema y se brinda una descripción textual de estos casos de uso.

Capítulo 3: Diseño del Sistema. Se definen las clases del diseño. Se exponen los diagramas de clases del diseño y los diagramas de interacción realizados en el diseño. Se comentan los patrones de diseño y de arquitectura aplicados. Se muestra a través del diagrama de despliegue la distribución de los componentes físicos necesarios para la implantación del sistema.

Capítulo 4: Implementación del Sistema. Contiene la representación de los diagramas de componentes, se muestran fragmentos de código de los principales métodos implementados y por último se muestran los resultados de las pruebas aplicadas y un análisis de los resultados obtenidos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

El capítulo incluye un estado del arte del tema tratado, de las tendencias, técnicas, tecnologías, metodologías y herramientas usadas en la actualidad o de las que sirven de apoyo para la solución del problema que se enfrenta.

1.1-Gestión de la Información

La gestión de la información es un amplio campo de estudio, pero esencialmente es el proceso de gestión de datos de una organización o empresa, es el proceso de desarrollo de prácticas y procedimientos relativos a los datos [2]. La correcta gestión de la información conoce, incorpora y vincula todos los tipos de datos, de todas las áreas de la organización y se relaciona con todos los procesos, desde la generación de datos internos, la selección y adquisición de documentos hasta la organización de su uso. Por lo tanto la gestión de la información implica:

- Determinar la información que se precisa.
- Recoger y analizar la información.
- Registrar y recuperar la información cuando sea necesaria.
- Utilizar la información.
- Divulgar la información.[3]

1.2-Sistemas de gestión de la Información

Un Sistema de Gestión de Información (SGI) es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una organización. Los sistemas de gestión de información para la gestión del conocimiento constituyen hoy una alternativa de imprescindible presencia en cada organización. Al permitir operar casi todos los activos tangibles e intangibles de la institución y llegar a convertirse en la herramienta integral de gerencia más cotizada y necesaria para alcanzar con éxito los resultados propuestos por la organización.

Un Sistema de Gestión de Información realiza cuatro actividades básicas:

- **Entrada de Información:** es el proceso mediante el cual el sistema de gestión de información toma los datos que requiere para procesar la información.

- **Almacenamiento de información:** es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.
- **Procesamiento de Información:** es la capacidad del sistema de gestión de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados.
- **Salida de Información:** es la capacidad de un sistema de gestión de información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, graficadores, cintas magnéticas, disquetes.

1.3-Sistemas de Gestión de la Información similares

En la actualidad las instituciones cubanas se encuentran inmersas en un proceso de crecimiento y extensión, lo que trae como consecuencia un aumento considerable en el volumen de los datos que manejan. El procedimiento que se emplea para transformar estos datos en información en muchos casos es manual, lo que deriva que en muchas ocasiones el resultado obtenido no posea la calidad requerida, afectando el valor del conocimiento derivado de esta información, lo que repercute negativamente en la toma de decisiones y en la eficacia del proceso de gestión.

Para mejorar el proceso de gestión de la información anteriormente explicado, la educación superior cubana se ha planteado como una estrategia para su desarrollo la informatización; siendo ésta uno de sus retos más importantes en los últimos años, tomando en cuenta las facilidades que ofrecen las Tecnologías de la Información y las Comunicaciones (TIC) para procesar grandes volúmenes de información de forma ágil y confiable. Por lo que se han realizado aplicaciones en distintas universidades del país que permita un rápido y adecuado manejo de datos para mejorar el procesamiento de la información. Como ejemplo de ello está:

Sistema automatizado informativo para la comunidad de residencia estudiantil de la Universidad de Pinar del Rio “Hermanos Saiz Montes de Oca”. (SAICRE).

Este sistema se ha diseñado y elaborado con el objetivo de gestionar la información de los becados de la Comunidad de Residencia Estudiantil en la Universidad de Pinar del Rio (UPR).

El software fue desarrollado con uso del gestor de base de datos Microsoft Access y de su lenguaje

residente “Visual Basic para Aplicaciones”, utilizándose para su análisis y diseño la notación UML y la herramienta CASE Rational Rose Enterprise Edition. Cuenta con un sistema de ayuda creado en RoboHELP Office 9.0 y funcionalidades para garantizar la seguridad de la información.

SAICRE permite:

- Controlar el registro de becados.
- Controlar el registro de indicadores de los becados:
 - ✓ Evaluaciones.
 - ✓ Sanciones.
 - ✓ Reconocimientos.
 - ✓ Actividades.
 - ✓ Tratamientos.

Esto facilita la gestión de la información de los becados, permitiendo controlar los datos de cada uno como: evaluaciones semestrales y una anual en el curso, cuarterías, guardias, inspecciones a los cuartos donde el orden, la limpieza e higiene, el cuidado de la propiedad social y la estética son los principales parámetros evaluados. [4]

Este sistema utiliza como gestor de Base de Datos y herramienta de modelado softwares propietarios. Aunque tiene funcionalidades en común con la aplicación a desarrollar, estas están encaminadas a resolver las necesidades de la universidad de Pinar del Rio por lo que no es viable su utilización en el sistema a implementar, además carece de otras funcionalidades necesarias para la gestión de la información de la residencia en la facultad 6.

En la Universidad de las Ciencias Informáticas el desarrollo de aplicaciones de este tipo ha sido una necesidad desde su formación, es por eso que varias de sus facultades se han dado a la tarea de desarrollar sistemas para el manejo de la información, se tiene como ejemplo de sistema de gestión de la información:

Sistema de Gestión de información de la Facultad 8 (SGIF 8). Es un sistema para gestionar toda la información referente a la facultad 8, posee 6 módulos, de estos uno es para gestionar gran parte de la información no docente que se lleva a cabo en la facultad y vinculadas a la residencia estudiantil como son:

- Ubicar a los estudiantes en los apartamentos de los edificios con los que dispone la facultad.

- Conocer en qué edificio, paso de escalera y apartamento de la residencia se encuentra ubicado un estudiante de la facultad o un grupo de estudiantes.
- Emitir partes de guardia.
- Conocer cómo se ha comportado la guardia estudiantil de una brigada específica hasta el momento.
- Consultar los partes de la guardia estudiantil de cualquier día.

El sistema ha sido desarrollado utilizando una arquitectura cliente-servidor, además puede ser utilizado en múltiples plataformas. El lenguaje de programación Web utilizado es PHP (Personal Home Page), como herramienta case se definió Rational Rose, RUP como metodología, UML como lenguaje de modelado y se seleccionó CodeIgniter como frameworks para la aplicación. [5]

Este sistema carece de funcionalidades necesarias para la gestión de la información de la residencia en la facultad 6, no gestiona las afectaciones de los apartamentos, las evaluaciones de los estudiantes en la residencia y las paradas de beca.

1.4- Metodología de desarrollo de software

Al enfrentarse al desarrollo de un producto de software, se hace indispensable tener básicamente una metodología para su desarrollo. Una metodología es un conjunto ordenado de pasos a seguir para cumplir un objetivo. Dentro de la Ingeniería de Software, el objetivo es el desarrollo de software de alta calidad que cumpla con las necesidades del usuario. El "Proceso Unificado" es el resultado final de tres décadas de desarrollo y uso práctico. Esta es una de las causas que conlleva a que sea la metodología que mejor se ajusta a las necesidades que existen actualmente en el desarrollo de software, pues propone un modelo iterativo e incremental, muy acorde con la naturaleza cambiante de los requisitos en muchos proyectos.

RUP elimina los errores cometidos en las iteraciones previas, logra así que al final del proceso se obtenga como resultado un producto de calidad. Para eso se agrupan las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales divididos en 4 fases. Los 6 primeros flujos son conocidos como flujos de ingeniería y los tres últimos como de apoyo [6], donde todos y cada uno de ellos cobran vital importancia en el proceso de desarrollo de software. En la siguiente figura se puede encontrar el gráfico conocido por RUP en dos dimensiones, donde se representan los flujos mencionados y la fase en que cada uno de ellos cobra su mayor desempeño.

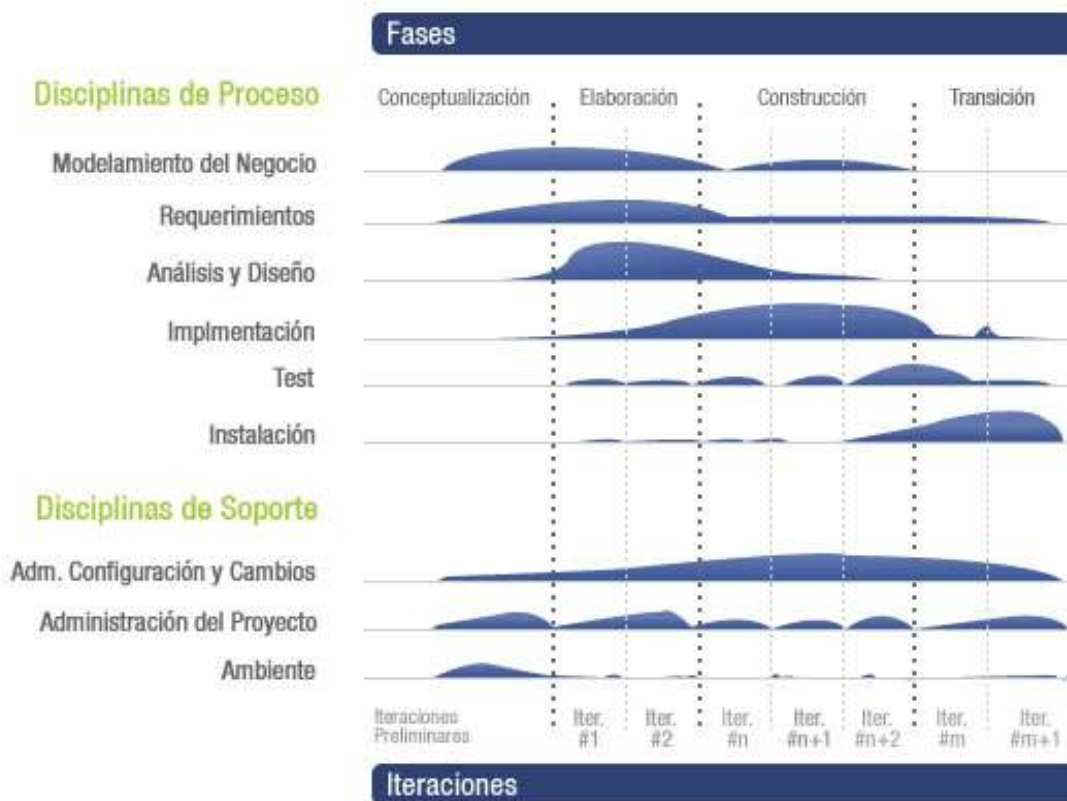


Figura 1 RUP en dos dimensiones

El ciclo de vida de RUP tiene tres características fundamentales:

➤ **Proceso dirigido por casos de uso:**

Los casos de uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un caso de uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los casos de uso representan los requisitos funcionales del sistema. En RUP los casos de uso no solo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permiten establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo. En la figura siguiente se hace una representación de esta afirmación.

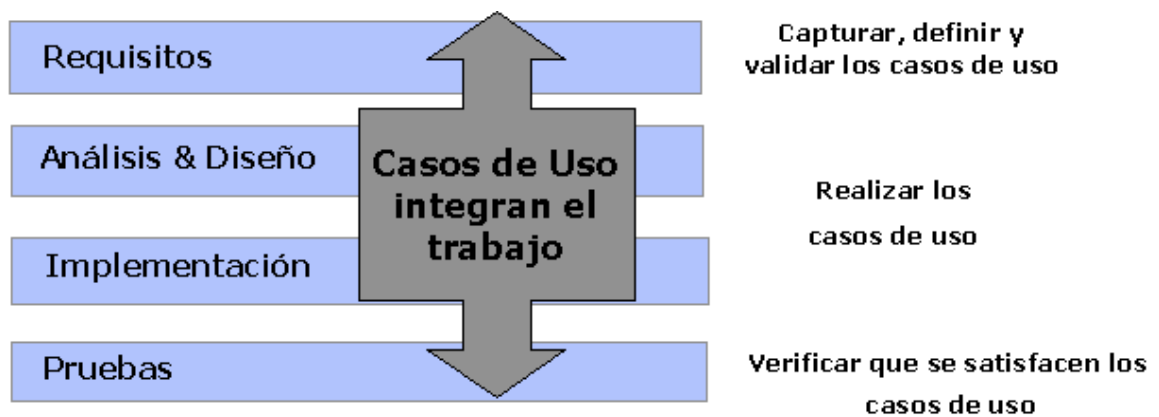


Figura 2 Los Casos de uso integran el trabajo.

➤ **Proceso iterativo e incremental:**

El trabajo se divide en partes más pequeñas o mini proyectos. El equilibrio entre casos de uso y arquitectura se logra durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto. Y en cada iteración se aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura, para obtener un resultado final sin problemas de no haber corregido errores anteriores.

➤ **Proceso centrado en la arquitectura:**

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema. [7]

1.5-Roles y artefactos

A partir del desarrollo del presente trabajo de diploma de acuerdo a las necesidades actuales del sistema se exponen los roles que se desempeñan y los artefactos que se generan según la metodología escogida.

1.5.1-Rol Analista

En la metodología RUP el analista del sistema es el responsable de conducir y coordinar los requerimientos y los casos de uso modelando y delimitando la funcionalidad del sistema y delimitando el sistema; por ejemplo, estableciendo qué actores y casos de uso existen y cómo interactúan.

Analista: Agrupa los roles que están involucrados fundamentalmente en la extracción e investigación de los requisitos del sistema. Este grupo está formado por los siguientes roles:

- **Analista de procesos del negocio:** Responsable de definir la arquitectura del negocio; los casos de uso del negocio y actores, así como sus relaciones. Los artefactos que realizan son: glosario del negocio, modelo de casos de uso del negocio, modelo de análisis del negocio, metas del negocio, objetivos de la organización, visión del negocio, documento de arquitectura del negocio, especificación suplementaria del negocio y reglas del negocio.
- **Diseñador de negocio:** Encargado de detallar la especificación de la organización o parte de ella. Los artefactos que realizan son: caso de uso del negocio, actor del negocio, realización de caso de uso del negocio, sistema del negocio, entidad del negocio, trabajador del negocio y eventos del negocio.
- **Analista del sistema:** Dirige y coordina el proceso de extracción de requisitos y desarrollo del modelo de casos de uso, definiendo la funcionalidad y límites del sistema. Los artefactos que realizan son: plan de gestión de requerimientos, glosario, atributos de requerimientos, especificación suplementaria, solicitudes de los stakeholders, documento visión, modelo de casos de uso, actor y Storyboard.
- **Especificador de requerimientos:** Se encarga de especificar los detalles de una o varias partes de la funcionalidad del sistema, describiendo uno o varios aspectos de los requisitos. Los artefactos que realizan son: casos de uso, especificación de requerimiento del software, paquete de casos de uso y requerimientos del software.

Los artefactos realizados por el analista del sistema que se obtendrán en la presente investigación son:

- **Glosario del negocio:** Lista de conceptos asociados al negocio que son comúnmente usados y que deben ser del dominio del equipo de desarrollo para poder modelar el negocio y dar una solución a la problemática encontrada.
- **Modelo de casos de uso del negocio:** Describe los procesos del negocio de una empresa en términos de casos de uso y actores del negocio, que se corresponden con los procesos del negocio y los clientes, respectivamente.
- **Modelo de análisis del negocio:** Describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso del negocio.
- **Caso de uso del negocio:** Representa un proceso del negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio.
- **Actor del negocio:** Lo que se modela como actor es el rol que se desempeña cuando se interactúa con el negocio para beneficiarse de sus resultados.
- **Realización de caso de uso del negocio:** Muestra cómo colaboran los trabajadores y entidades de negocio para ejecutar el proceso.
- **Entidad del negocio:** Representa a los objetos que los trabajadores del negocio toman, inspeccionan, manipulan, producen o utilizan durante la realización de los casos de uso de negocio. Comúnmente representan un documento o una parte esencial de un producto.
- **Trabajador del negocio:** Es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio.
- **Glosario:** Términos comunes que se utilizan para describir el sistema.
- **Modelo de casos de uso:** Es un modelo del sistema que contiene actores, casos de uso y sus relaciones.
- **Actor:** Rol que se desempeña al interactuar con el sistema, generalmente son los trabajadores del negocio y en algunos casos los actores del negocio que van a interactuar con el Sistema.
- **Casos de uso:** Fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.
- **Especificación de los requerimientos del software:** Es la captura de requerimientos de software para el sistema o una parte de este.

- **Paquete de casos de uso:** Es una colección de casos de uso, actores, relaciones, diagramas y otros paquetes; que se usan para estructurar el modelo de casos de uso en partes más pequeñas.
- **Descripción de la arquitectura** (vista del modelo de casos de uso): contiene una vista de arquitectura del modelo de casos de uso, que representa los casos de uso significativos para la arquitectura. [7]

1.5.2-Rol Diseñador

En la metodología RUP el diseñador es el responsable de diseñar una parte del sistema cumpliendo con las restricciones de los requerimientos, arquitectura y proceso de desarrollo del proyecto, identifica y define las responsabilidades, operaciones, atributos y relaciones de los elementos de diseño. Debe asegurarse que el diseño es consistente con la arquitectura del software y que está detallado al punto que se puede proceder con la implementación.

Diseñador: Este rol dirige el diseño de una parte del sistema, dentro de las restricciones de los requisitos, arquitectura y proceso de desarrollo para el proyecto. Este grupo está formado por los siguientes roles:

- **Diseñador de Base de Datos:** Este rol dirige el diseño de la estructura de almacenamiento de datos persistentes que se utilizará en el sistema.
- **Diseñador de Cápsulas:** Este rol diseña cápsulas, asegurando que el sistema puede responder a los sucesos a tiempo, de acuerdo con los requisitos de concurrencia.
- **Diseñador de interfaz de usuario:** Este rol coordina el diseño de la interfaz de usuario. Esto incluye recopilar los requisitos de utilización y los diseños de interfaz de usuario candidata a la creación de prototipos para cumplir estos requisitos.

Los artefactos realizados por el diseñador que se obtendrán en la presente investigación son:

- **Realización de casos de uso del diseño:** Es una colaboración en el modelo de diseño que describe como se realiza un caso de uso específico, y como se ejecuta en términos de casos de uso del diseño. Una realización de caso de uso del diseño proporciona una traza directa a una realización de caso de uso del análisis en el modelo de análisis.

- **Modelo del diseño:** Describe la realización de guiones de uso, y sirve como una abstracción del modelo de implementación y el código fuente. El modelo de diseño se utiliza como entrada esencial para actividades en implementación y prueba.
- **Clases del diseño:** Una clase es una descripción de un conjunto de objetos que comparten las mismas responsabilidades, relaciones, operaciones, atributos, y la semántica.
- **Realización de Caso de uso del diseño:** Es una colaboración en el modelo de diseño que describe como se realiza un caso de uso específico, y como se ejecuta en términos de casos de uso del diseño. Una realización de caso de uso del diseño proporciona una traza directa a una realización de caso de uso del análisis en el modelo de análisis.
- **Paquetes de diseño:** Son una colección de clases, relaciones, realizaciones de casos de uso, diagramas y otros paquetes que están de alguna forma relacionados. Es usado para estructurar el modelo de diseño dividiéndolo en partes más pequeñas.
- **Subsistema de diseño:** Es una parte del sistema que encapsula el comportamiento, incluye interfaces y paquetes.
- **Prototipo de interfaz de usuario:** Ayudan a comprender y especificar las interacciones entre actores humanos y el Sistema durante la captura de requisitos. No solo ayuda a desarrollar una interfaz gráfica mejor, sino también a comprender los casos de uso. [7]

1.5.3-Rol Implementador

El rol Implementador es responsable de desarrollar y de probar componentes de acuerdo con los estándares adoptados del proyecto para la integración en subSistemas más grandes. El artefacto que realiza el implementador que se obtendrá en la presente investigación es:

- **Subsistema de Implementación:** Este artefacto consta de un conjunto de elementos de implementación. Estructura el modelo de implementación dividiéndolo en componentes más pequeños que se pueden integrar y probar separadamente.
- **Prueba del desarrollador:** Este artefacto abarca el trabajo tradicionalmente pensado bajo las categorías siguientes: Pruebas de unidad, parte de las Pruebas de integración, y algunos aspectos de lo que se denomina Pruebas del sistema.
- **Elementos de implementación:** Los elementos de implementación son la parte física de la implementación, incluyen los archivos y directorios. Incluyen ficheros de código (fuentes, binarios o ejecutables), ficheros de datos y de documentación como ficheros de ayuda online. [7]

1.6-Tecnologías, lenguajes y herramientas para el desarrollo

1.6.1-Tecnología del lado del cliente

Los lenguajes del lado del cliente son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento. Entre ellos encontramos Javascript y HTML.

JavaScript

JavaScript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. [8] Está diseñado para controlar la apariencia y manipular los eventos dentro de la ventana del navegador Web y es soportado por la gran mayoría de los navegadores, lo que lo posiciona en el lugar del lenguaje de programación web del lado del cliente más utilizado y le confiere la característica de ser multiplataforma.

JavaScript cuenta con efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo y permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas con programas como: calculadoras, agendas, tablas de cálculo, calendarios [9] y validar informaciones entradas por el usuario para verificar su veracidad o su forma de escribirse.

HTML

HTML es la abreviatura de "HyperText Mark-up Lenguaje" (Lenguaje de Marcas Hipertextuales), diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. Gracias a los navegadores como: Internet Explorer, Opera, Firefox, Netscape y Safari, el HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para la Web. Este lenguaje permite la introducción de referencias a otras páginas por medio de los enlaces hipertexto. [10]

1.6.2-Tecnología del lado del servidor

PHP5

Los lenguajes Del lado del servidor son aquellos que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Entre los lenguajes

que trabajan del lado del servidor se pueden citar algunos, que se destacan por ser los más utilizados por los programadores como son PERL, ASP, PHP, Java, JSP, entre otros. Estos lenguajes desarrollan la lógica de negocio dentro del servidor, además se encargan de los accesos a los distintos Sistemas Gestores de Bases de Datos. Para el desarrollo de la presente aplicación será utilizado PHP como lenguaje de programación.

PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. [11]

El lenguaje de programación que soportara la aplicación será PHP 5. Una esperada evolución del, tal vez, más popular de los lenguajes de programación de páginas y aplicaciones web del lado del servidor.

El principal objetivo de PHP5 ha sido mejorar los mecanismos de POO (Programación Orientada a Objeto) para solucionar las carencias de las anteriores versiones. Un paso necesario para conseguir que PHP sea un lenguaje apto para todo tipo de aplicaciones y entornos, incluso los más exigentes.

1.6.3-Lenguaje de modelado

UML

El Lenguaje Unificado de Modelado permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. UML no es un lenguaje de programación, sino un lenguaje de propósito general para el modelado orientado a objetos. También puede considerarse como un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes. Intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos; se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo. En estos momentos se convierte en un standard, ya que es el resultado de las tres opciones existentes en el mercado. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otra rama [6]. Permite la modificación de todos sus miembros mediante estereotipos y restricciones. Un estereotipo nos permite indicar especificaciones del lenguaje al que se refiere el diagrama de UML. Una restricción identifica un comportamiento forzado de una clase o relación, es decir, mediante la restricción estamos forzando el comportamiento que debe tener el objeto al que se le aplica.

Dentro de las ventajas que proporciona este lenguaje de modelado podemos destacar las siguientes:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyendo así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo.
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- UML es independiente del proceso, aunque e para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

1.6.4-Herramienta CASE (Computer Aided Software Engineering)

Visual Paradigm 6.1

Es una potente herramienta CASE empleada para visualizar y diseñar elementos de software, para ello utiliza el lenguaje UML, proporciona a los desarrolladores una plataforma que les permite diseñar un producto con calidad de forma rápida. Facilita la interoperabilidad con otras herramientas CASE como Rational Rose. Se integra con diversos IDE´s como: NetBeans (de Sun), Eclipse (de IBM), JDeveloper (de Oracle), JBuilder (de Borland). Está disponible en varias ediciones: Enterprise, Professional, Community, Standard, Modeler y Personal. Genera código y realiza ingeniería inversa para diferentes lenguajes de programación como: Java, C++, CORBA IDL, PHP, XML Schema y ADA. En adición se genera código para C#, Visual Basic.net, Object Definition Lenguaje (ODL), Flash Action Script, Delphi, Perl y Python. Se integra con el Visio para importar imágenes del mismo para realizar los diagramas de despliegue. Además exporta e importa los diagramas en el estándar XML.

Visual Paradigm es la herramienta CASE que se empleará en la modelación de este proyecto por su característica de ser multiplataforma, por las facilidades que brinda y por ser una herramienta que se puede utilizar legalmente.

- Ofrece entorno de creación de diagramas para UML 2.0.
- Disponibilidad en múltiples plataformas.
- Disponibilidad de integrarse en los principales IDEs.

- Soporta una gama de lenguajes en la Generación de Código e Ingeniería Inversa en Java, C++, CORBA IDL, PHP, Esquema de XML, Ada y Python.

La Generación de Código soporta C #, VB .NET, Lenguaje de Definición de Objeto (ODL), Flash Action Script, Delphi, Perl, Objetivo-C, y Ruby.

1.6.5-Framework para el desarrollo

Symfony 6.1.2

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

En el desarrollo del presente trabajo de diploma se escogió a **Symfony** por sus características y funcionalidades, después de un estudio que realizó el grupo de arquitectura del proyecto. Symfony es un completo framework diseñado para optimizar, gracias 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. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

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. Symfony 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.

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 las librerías de otros fabricantes [12].

1.6.6-Ambiente de Desarrollo Integrado (IDE)

Eclipse 3.4

Eclipse es el entorno de desarrollo integrado IDE (Integrated Development Environment) seleccionado para implementar este sistema. El Eclipse es un IDE para todo tipo de aplicaciones, inicialmente desarrollado por IBM, y actualmente gestionado por la Fundación Eclipse. La característica clave de Eclipse es la extensibilidad. Eclipse es una gran estructura formada por un núcleo y muchos plugins que van conformando la funcionalidad final. La forma en que los plugins interactúan es mediante interfaces o puntos de extensión; así, los nuevos aportes se integran sin dificultad ni conflictos. [13] Se seleccionó como IDE de desarrollo ECLIPSE 3.4, el cual brinda nuevas facilidades como la disposición de un entorno mucho más flexible y profesional para controlar todo el ciclo de vida de un desarrollo. Además, brinda capacidades de refactorización del código fuente permitiendo adecuar el comportamiento externo de una función o clase sin cambiar el funcionamiento interno.

Ventajas:

- Es una herramienta de código abierto.
- Soporta herramientas que manipulan diferentes tipos de lenguajes, como por ejemplo Java, C, C++.
- Se ejecuta en varios sistemas operativos incluyendo Windows y Linux.
- Provee a los desarrolladores, herramientas (ej.- PDE) que facilitan la creación de plugins.

1.6.7-Sistema Gestor de Base de Datos (SGBD)

PostgreSQL 8.2

Un sistema Gestor de Base de Datos (SGBD) puede definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

El Sistema Gestor de bases de datos que almacena los datos de la aplicación es PostgreSQL v8.2.

PostgreSQL es un sistema de base de datos relacional perteneciente al ámbito del software libre que destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL. Cuenta con versiones para una amplia gama de sistemas operativos, entre ellos: Linux, Windows, Mac SX, Solaris, BSD, Tru64 y otros más.

PostgreSQL soporta ACID, o lo que es lo mismo, la realización de transacciones seguras; también, vistas, uniones, claves extranjeras, procedimientos almacenados, triggers, entre otros. Incluye la mayor parte de los tipos de datos especificados en los estándares SQL92 y SQL99, como: entero, numérico, booleano, char, varchar, fecha, interval o timestamp.

Otras características interesantes de PostgreSQL son las siguientes:

- Alta concurrencia, que evita tener que bloquear una tabla cuando se está escribiendo en ella.
- Copias de seguridad en línea.
- Replicación asíncrona.
- Transacciones anidadas.
- Optimizador de consultas.

Si de cifras se trata, es importante saber que en PostgreSQL el tamaño máximo de la base de datos es ilimitado; el de una tabla asciende a 32 TB, el de una fila a 1.6 TB y el de un campo de datos a 1 GB; el número de filas en una tabla es ilimitado, pero no el de columnas, que oscila entre 250 y 1600 columnas por tabla.

1.6.8-Servidor Web. Apache 2.2

Apache es un servidor Web, flexible, rápido y eficiente, de código fuente abierto, continuamente actualizado y adaptado a los nuevos protocolos. Está disponible para diferentes plataformas como: FreeBSD, NetBSD, OpenBSD, GNU/Linux, Mac OS y Mac OS X Server, Netware, Solaris, Windows, entre otras. Con los diferentes módulos de apoyo que proporciona y con la API de programación de módulos, puede ser adaptado a diferentes entornos y necesidades. Gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP.

1.6.9-Sistema de control de versiones: Subversion 1.4.5

Subversion 1.4 es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se lo conoce también como svn por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion 1.4 es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

Ventajas:

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Puede ser servido mediante Apache. Esto permite que clientes, utilicen Subversion en forma transparente.
- Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.
- Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos (SQL, LDAP, PAM).

1.7- Patrones

Un patrón es una unidad de información nombrada, instructiva e intuitiva que captura la esencia de una familia exitosa de soluciones probadas a un problema recurrente dentro de un cierto contexto. El objetivo de los patrones es crear un lenguaje común a una comunidad de desarrolladores para comunicar experiencias sobre los problemas y sus soluciones. Pueden referirse a distintos niveles de abstracción, desde un proceso de desarrollo hasta la utilización eficiente de un lenguaje de programación. [14]

1.7.1-Patrón de casos de uso

Los patrones de Casos de uso a utilizar son los siguientes:

- Patrón CRUD
- Patrón Múltiples actores

-CRUD (Creating, Reading, Updating, Deleting)

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual, consta de un caso de uso, llamado Información CRUD o Gestionar información, modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples.

-Múltiples actores

Roles comunes

Puede suceder que los dos actores jueguen el mismo rol sobre el caso de uso. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso.

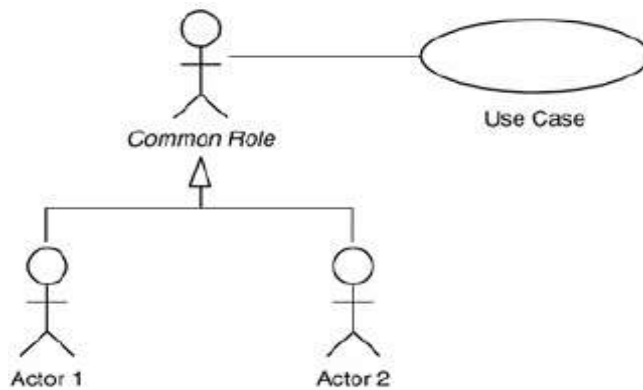


Figura 3 Patrón Múltiples actores. Roles comunes

1.7.2-Patrones de arquitectura y 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. Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. [15]

-Patrones GRASP (General Responsibility Assignment Software Patterns): describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades.

Creador

Este patrón se encarga de asignarle a la clase B la responsabilidad de crear una instancia de clase A. B es un creador de los objetos A.

Experto

Este se encarga de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Alta Cohesión

Este patrón se encarga de asignar una responsabilidad de modo que la cohesión siga siendo alta. Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo por hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande.

Controlador

Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase.

Bajo Acoplamiento

Este patrón se encarga de asignar una responsabilidad para mantener bajo acoplamiento. Las clases deben comunicarse con un número pequeño de clases tanto como sea posible.

-Patrones GOF (Gang of Four)

En la categoría creacionales: Muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente decidiendo que clases instanciar o sobre que objetos otro delegará responsabilidades.

- Singleton (Instancia única): Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella.
- Abstract Factory (Fábrica abstracta): Proporciona una interfaz para crear familias de objetos o que dependen entre sí, sin especificar sus clases concretas.

En la categoría estructurales: Describen la forma en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros.

- Decorator (Envoltorio): Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad.
- Composite (Objeto compuesto): Combina objetos en estructuras de árbol para representar jerarquías de parte-todo. Permite que los clientes traten de manera uniforme a los objetos individuales y a los compuestos.

Categoría Comportamiento: Se utilizan para organizar, manejar y combinar comportamientos.

- Command (Acción): Encapsula una petición en un objeto, permitiendo así parametrizar a los clientes con distintas peticiones, encolar o llevar un registro de las peticiones y poder deshacer las operaciones. [15]

Los patrones de arquitectura ayudan a especificar la estructura fundamental de una aplicación, expresan el esquema fundamental de organización para sistemas de software, proveen un conjunto de

subsistemas predefinidos, especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Para el desarrollo de la aplicación informática que acompaña el presente trabajo de diploma se utilizará el framework Symfony, el cual está basado en el patrón MVC.

Modelo de Vista Controlador (MVC): Symfony 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 de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como en 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. [16]

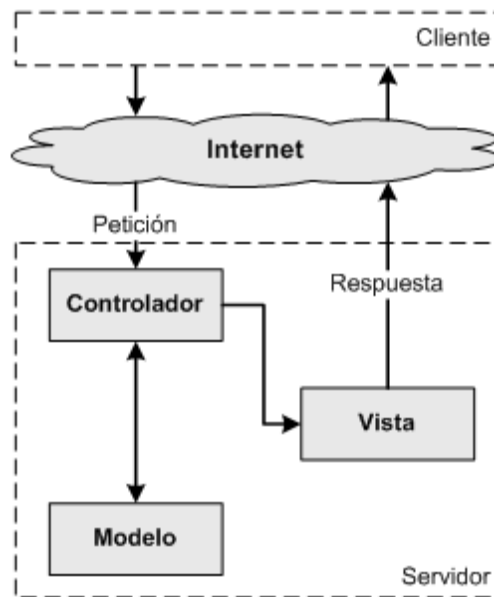


Figura 4 El patrón MVC

Conclusiones

En este capítulo se realizó un estudio del estado del arte del problema a resolver, analizando sistemas similares existentes. A partir del estudio detallado de estos sistemas y el análisis de los procesos que posee implementados, se llega a la conclusión de que no cuentan con funcionalidades necesarias para la correcta gestión de las actividades que se llevan a cabo en la residencia estudiantil, por lo que surge la necesidad de crear el módulo residencia que pertenece al sistema de gestión de la información de profesores y estudiantes de la facultad 6. Se definen para el diseño e implementación del sistema tecnologías, técnicas y metodologías que han sido objeto de estudio de este capítulo, para obtener una aplicación que cumpla con todos los requerimientos exigidos y que permita a los clientes finales un entorno de trabajo amigable y flexible.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Introducción

En el presente capítulo se enuncian y describen los procesos de negocio, se exponen las reglas que debe cumplir el negocio en cuestión, se dan a conocer cuáles serán 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 objetos. También se identifican cuáles son los requisitos funcionales y no funcionales que debe cumplir la aplicación, así como la descripción detallada de cada uno de los casos de usos del sistema.

2.1- Caracterización del problema

La Universidad de las Ciencias Informáticas surge como única de su tipo en el sistema educacional cubano en el calor de la batalla de ideas, la cual está integrada por miles de estudiantes de todo el país, los cuales se encuentran distribuidos en las diferentes facultades que integran dicha universidad. Cada facultad consta de un área de residencia en la cual conviven los estudiantes.

Para lograr el buen funcionamiento de los procesos que se llevan a cabo en estas áreas las facultades cuentan con un vicedecanato de residencia, donde los directivos están inmersos en la labor de lograr que los estudiantes cumplan con lo establecido por la dirección de la universidad, dígase la realización de la Guardia Obrera Estudiantil, el Trabajo Socialmente Útil, la Cuarterería y las parada de becas convocadas por la FEU. El vicedecano además es el encargado de llevar el control de las ubicaciones de los estudiantes, así como las evaluaciones y asistencia alcanzadas en la realización de las actividades de la residencia. Para un mejor manejo de dicha información se hace necesario conocer el funcionamiento de dichos procesos que se realizan actualmente en la facultad.

2.2- Objeto de informatización

Los procesos actuales se llevan a cabo manualmente y en ocasiones a través de la herramienta Microsoft Office Excel y Microsoft Word, manejándose un gran volumen de información siendo cada vez mayor, esto provoca que la búsqueda de la misma en determinado momento se torne un poco lenta.

La información no se encuentra totalmente centralizada, es decir, se envían informaciones por correo electrónico o se hacen llegar por medio de algún dispositivo de almacenamiento, lo cual puede provocar que dicha información sea consultada por personas no autorizadas. El aumento de

información provoca en muchas ocasiones un atraso a la hora de contar con la información necesaria en tiempo.

2.3-Propuesta del Sistema

Se propone implementar un sistema que facilite el manejo y control de la información sobre la residencia estudiantil de la facultad 6, que sea de fácil acceso pero con una fuerte seguridad e integridad. El sistema debe ser explícito en la información que se muestre, por esto es la importancia de una buena interfaz de usuario con una clara descripción de lo que ahí se muestra, debe centrarse en las funcionalidades que se requieren y que estas se ejecuten con la mayor rapidez posible. El sistema debe permitir insertar, modificar y actualizar, así como buscar y guardar toda la información que se maneje.

2.4-Modelamiento del Negocio

El flujo de trabajo de Modelamiento del negocio da una visión de qué es necesario hacer para dar respuesta a las solicitudes del usuario, lo cual se logra definiendo los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos. El Modelamiento del negocio tiene como objetivos:

- 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.
- Derivar los requerimientos del sistema que va a soportar la organización.
- Lograr una comunicación efectiva entre los usuarios y el equipo de proyecto con el objetivo de llegar a un entendimiento de lo que hay que hacer, es la clave del éxito en la producción de un software.

El negocio actual se basa en llevar un control total de los procesos que se llevan a cabo en la residencia estudiantil de la facultad. Se desea registrar la ubicación de los estudiantes por apartamentos, la planificación del trabajo socialmente útil, la cuartería, la guardia obrera estudiantil y la evaluación de los estudiantes en la beca. Además se quiere controlar la relación de edificios y apartamentos con que cuenta la residencia de la facultad, así como las afectaciones que presentan los mismos.

2.4.1-Identificación de los actores y trabajadores del negocio

Actores del negocio: grupo de individuos, organización, máquina o sistema de información externo al negocio y que interactúa con él.

Actores del Negocio

Decana	Directivo de la facultad que solicita una información determinada referente a la residencia estudiantil y posee privilegios y permisos para consultar dicha información.
Estudiante	Persona que debe cumplir con la realización de la guardia obrera estudiantil, el trabajo socialmente útil y la cuartería, de la cual se emite una evaluación.
Profesor Guía	Persona responsable de un grupo de estudiantes de la facultad y que controla las evaluaciones de los mismos.

Tabla 1 Actores del Negocio

Trabajadores del negocio: grupo de individuos, máquina o sistema automatizado que participan directamente en los procesos que se llevan a cabo en el negocio, ya que son los que realizan las actividades y son propietarios de elementos pero no obtienen ningún beneficio con los resultados del proceso.

Trabajadores del negocio

Trabajador	Descripción
Vicedecano de residencia y extensión (VDRE)	Es la persona que lleva el control de las actividades que se realizan en la residencia estudiantil, así como la planificación de las mismas, encargándose de dar solución a las solicitudes hechas por los actores del negocio.

Instructora	Es la persona encargada de organizar la cuartería en el edificio que le corresponde atender, así como controlar su cumplimiento, la limpieza de los apartamentos y el estado del edificio en general. No se beneficia en ningún momento de las acciones ejecutadas en los procesos de negocio, sino que se limita a ejecutarlas.
Secretaria del Vicedecano de residencia y extensión (Sec VDRE)	Es la persona que asiste al Vicedecano de Residencia y según los permisos dados accede a determinada información.
Jefe de brigada	Estudiante al frente de la brigada, responsable de tomar la asistencia en las guardias obreras estudiantiles.

Tabla 2 Trabajadores del negocio

2.4.2-Diagrama de casos de uso del negocio

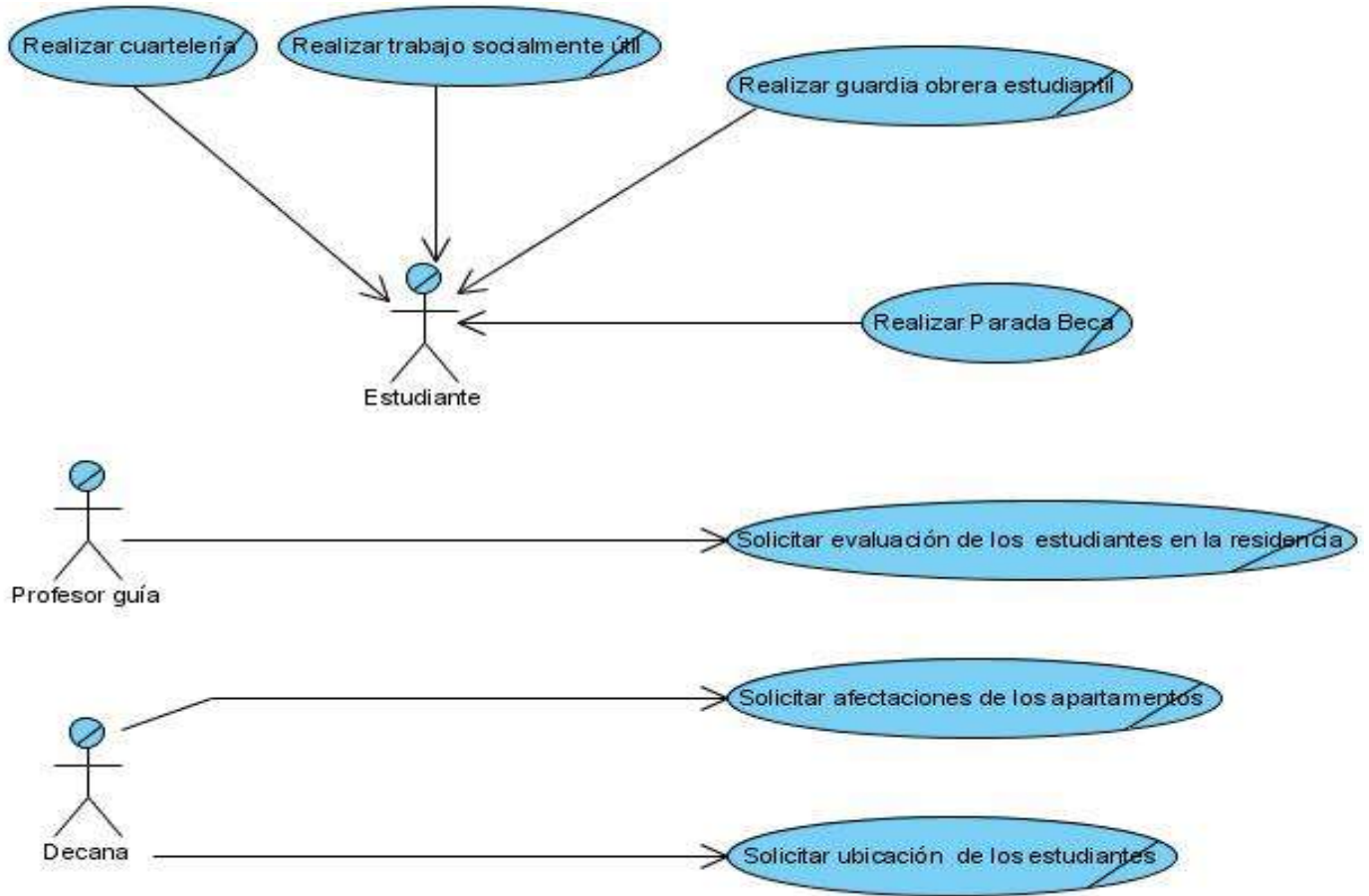


Figura 5 Diagrama de casos de uso del negocio

2.4.3-Realización de los casos de uso del negocio

El detallar los casos de uso del negocio, tiene como objetivo fundamental describir su flujo de sucesos, brindando una mayor comprensión de los procesos del negocio a clientes y desarrolladores. Se describe de forma textual como se llevan a cabo las actividades dentro del negocio y quienes las realizan. Por medio de los diagramas de actividades se muestra el flujo de los procesos de manera gráfica, a través de las calles, se especifican las responsabilidades de los actores y trabajadores del negocio y a través del flujo de objetos cómo se utilizan las entidades del negocio.

2.4.3.1-Descripción de los casos de uso del negocio

Descripción del CUN “Realizar cuartelería”

Caso de Uso:	Realizar cuartería
Actores:	Estudiante
Trabajadores:	Instructora (inicia)
Resumen:	El caso de uso se inicia cuando la instructora planifica la distribución de la cuartería, le informa al estudiante el día que le corresponde, este la realiza, la instructora le da una evaluación, el estudiante firma la planilla de cumplimiento y culmina el caso de uso.
Precondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
<p>3. El estudiante se presenta y realiza la cuartería.</p> <p>6. El estudiante recibe la evaluación y confirma que está de acuerdo.</p>	<ol style="list-style-type: none"> 1. La instructora planifica la distribución de la cuartería. 2. La instructora informa el día que debe realizar la cuartería el estudiante. 4. La instructora registra los datos del estudiante para controlar la asistencia en una planilla. 5. La instructora informa de la evaluación obtenida al finalizar la cuartería al estudiante según su comportamiento durante la misma.

7. El estudiante firma la planilla de cumplimiento y culmina el caso de uso.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
2.1. El estudiante no se presenta a hacer la cuartelería.	4.1. La instructora da evaluación de mal al estudiante.
Poscondiciones	Se recoge la asistencia a la cuartelería.
Posibles Mejoras	Informatizar el control de asistencia a la cuartelería.

Tabla 3 Descripción del CUN “Realizar cuartelería”

Descripción del CUN “Solicitar evaluación de los estudiantes en la residencia”

Caso de Uso:	Solicitar evaluación de los estudiantes en la residencia
Actores:	Profesor guía (inicia)
Trabajadores:	Vicedecano de residencia y extensión (VDRE)
Resumen:	Inicia cuando el profesor guía del grupo solicita al VDRE las evaluaciones de un estudiante o varios. El VDRE busca el listado en el que se encuentra registrada esa información, reúne las evaluaciones solicitadas y culmina el CU.
Precondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio (Sec. V.E.R)
1. El profesor guía solicita evaluaciones de la residencia de un determinado estudiante o varios.	2. El VDRE busca las evaluaciones en el registro correspondiente.

5. El profesor guía recibe el listado con las evaluaciones solicitadas y culmina el caso de uso.	<p>3. El VDRE confecciona un listado con las evaluaciones solicitadas del estudiante(s).</p> <p>4. El VDRE entrega el listado confeccionado.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Poscondiciones	
Posibles Mejoras	Informatizar la búsqueda de evaluaciones de estudiantes en la residencia.

Tabla 4 Descripción del CUN “Solicitar evaluación de los estudiantes en la residencia”

Descripción del caso de uso “Realizar trabajo socialmente útil”

Caso de Uso:	Realizar trabajo socialmente útil(TSU)
Actores:	Estudiante
Trabajadores:	Vicedecano de residencia y extensión (VDRE) (inicia), Secretaria del Vicedecano de residencia y extensión (Sec VDRE)
Resumen:	Inicia cuando el VDRE realiza la planificación del trabajo socialmente útil y le informa a los estudiantes, estos lo realizan según el día planificado, la Sec VDRE recoge la asistencia y culmina el CU.
Precondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio

<p>3. Los estudiantes reciben la planificación del trabajo socialmente útil.</p> <p>4. Los estudiantes realizan el trabajo el día señalado en la sesión definida.</p>	<p>1. El VDRE planifica la distribución del trabajo socialmente útil.</p> <p>2. El VDRE informa el día y sesión correspondientes a cada brigada.</p> <p>5. La Sec VDRE recoge la asistencia de los estudiantes al trabajo socialmente útil en un documento.</p> <p>4. La Sec VDRE envía el documento al VDRE.</p> <p>5. El VDRE recibe el documento de la asistencia al trabajo socialmente útil y termina el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
2.1. Los estudiantes no realizan el trabajo el día señalado en la sesión definida.	4.1 La Sec VDRE le envía información al VDRE de la situación.
Poscondiciones	Se recoge la asistencia de los estudiantes al trabajo socialmente útil
Posibles Mejoras	Informatizar el control de asistencia de los estudiantes al trabajo socialmente útil.

Tabla 5 Descripción del caso de uso “Realizar trabajo socialmente útil”

Descripción del CUN “Realizar guardia obrera estudiantil”

Caso de Uso:	Realizar guardia obrera estudiantil(GOE)
Actores:	Estudiante
Trabajadores:	Vicedecano de residencia y extensión (VDRE) (inicia), Jefe de brigada
Resumen:	Inicia cuando el VDRE realiza la planificación de la guardia obrera

	estudiantil, le informa a los estudiantes, estos la realizan según el día planificado, el jefe de brigada recoge la asistencia correspondiente y culmina el caso de uso.
Precondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
<p>3. Los estudiantes reciben la planificación de la guardia obrera estudiantil.</p> <p>4. Los estudiantes realizan la guardia el día señalado.</p>	<p>1. El VDRE planifica la distribución de la guardia obrera estudiantil.</p> <p>2. El VDRE informa el día correspondiente a cada brigada.</p> <p>5. El jefe de brigada recoge la asistencia de los estudiantes a la guardia obrera estudiantil en un documento.</p> <p>6. El jefe de brigada envía el documento al VDRE.</p> <p>7. El VDRE recibe el documento con la asistencia a la guardia obrera estudiantil.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
2.1. Los estudiantes no realizan la guardia el día señalado.	5.1. El jefe de brigada le envía información al VDRE de la situación.
Poscondiciones	Se recoge la asistencia de los estudiantes a la guardia.
Posible Mejoras	Informatizar el control de asistencia de los estudiantes a la guardia obrera estudiantil.

Tabla 6 Descripción del CUN “Realizar guardia obrera estudiantil”

Descripción del CUN “Solicitar afectaciones de los apartamentos”

Caso de Uso:	Solicitar afectaciones de los apartamentos
---------------------	--

Tabla 7 Descripción del CUN “Solicitar afectaciones de los apartamentos”

Descripción del CUN “Solicitar ubicación de los estudiantes”

Caso de Uso:	Solicitar ubicación de los estudiantes	
Actores:	Decana (inicia)	
Trabajadores:	Vicedecano de residencia y extensión (VDRE)	
Resumen:	Inicia cuando la decana solicita la ubicación de estudiantes en la facultad. El VDRE busca la información solicitada, se la entrega a la decana y culmina el caso de uso.	
Precondiciones:		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. La decana solicita la ubicación de estudiantes en la residencia.	2. El VDRE busca la información solicitada en los registros de ubicación.	
	3. El VDRE conforma un documento en el que se recoge la ubicación solicitada.	
	4. El VDRE le envía el documento a la decana.	
4. La decana recibe el documento con la ubicación solicitada y culmina el caso de uso.		
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
Poscondiciones	Se confecciona el listado de ubicación.	
Posibles mejoras	Informatizar la búsqueda de ubicación de los estudiantes en la residencia.	

Tabla 8 Descripción del CUN “Solicitar ubicación de los estudiantes”

Descripción del CUN” Realizar paradas de beca”

Caso de Uso:	Realizar paradas de beca	
Actores:	Estudiante	
Trabajadores:	Vicedecano de residencia y extensión (VDRE) (inicia)	
Resumen:	Inicia cuando el VDRE planifica el día en que se va efectuar la parada de beca y le informa a los estudiantes, estos limpian el apartamento que les corresponde, el vicedecano pasa la inspección, asigna la puntuación y culmina el caso de uso.	
Precondiciones:		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
3. Los estudiantes limpian el apartamento que les corresponde.	1. El VDRE planifica el día en que se va a efectuar la parada de beca. 2. El VDRE informa a los estudiantes la planificación. 4. El VDRE pasa la inspección por los apartamentos. 5. El VDRE le asigna a cada apartamento una puntuación según el estado de cada uno de ellos. 6. El VDRE registra las puntuaciones asignadas en un documento y culmina el caso de uso.	
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
Poscondiciones	Se registran las puntuaciones de las paradas de beca.	
Posibles Mejoras	Informatizar los resultados de las paradas de beca.	

Tabla 9 Descripción del CUN” Realizar paradas de beca”

2.4.3.2-Diagramas de actividades

Diagrama de actividades CUN “Realizar cuartería”

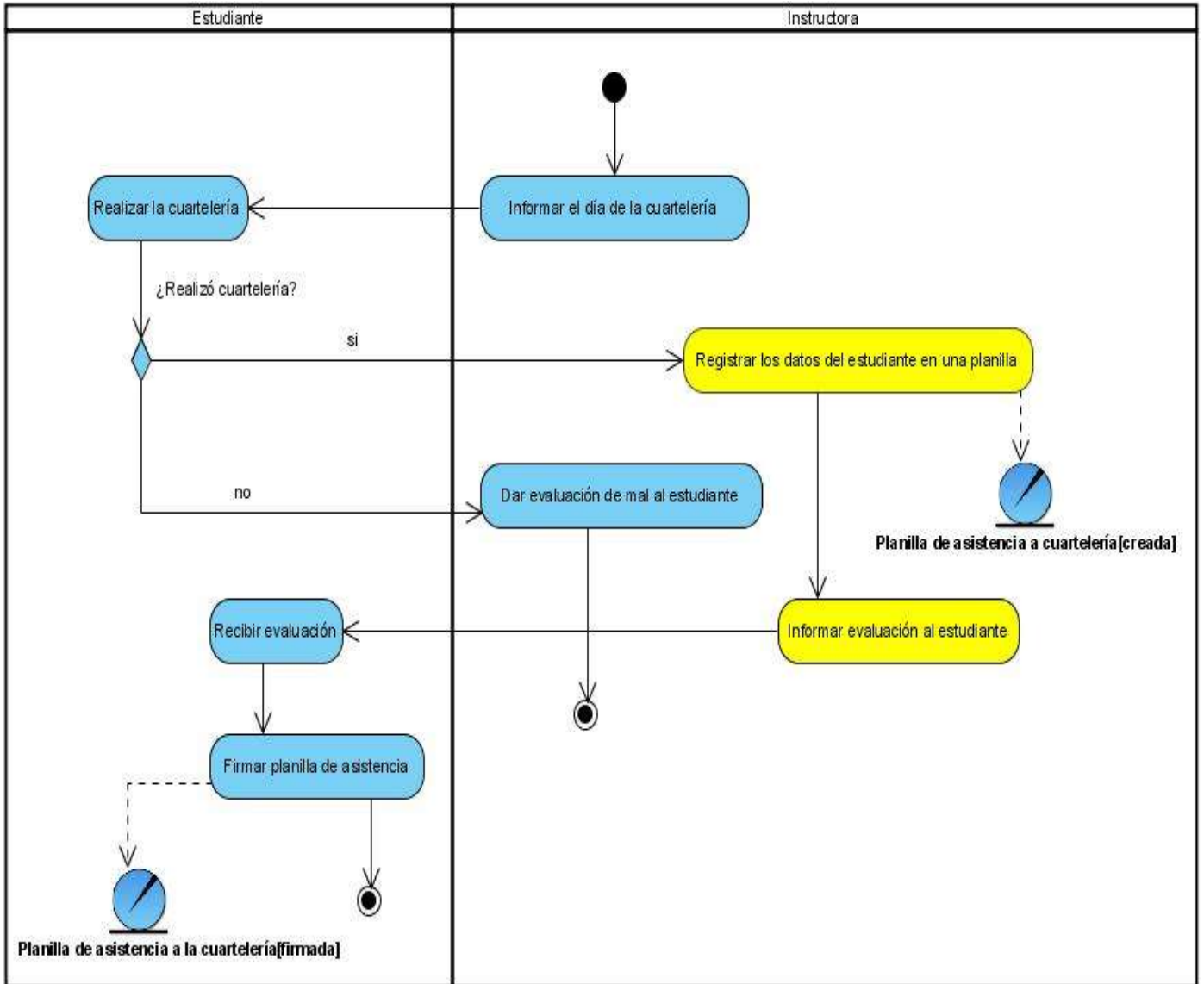


Figura 6 Diagrama de actividades CUN “Realizar cuarterías”

Diagrama de actividades CUN “Solicitar evaluación de los estudiantes en la residencia”

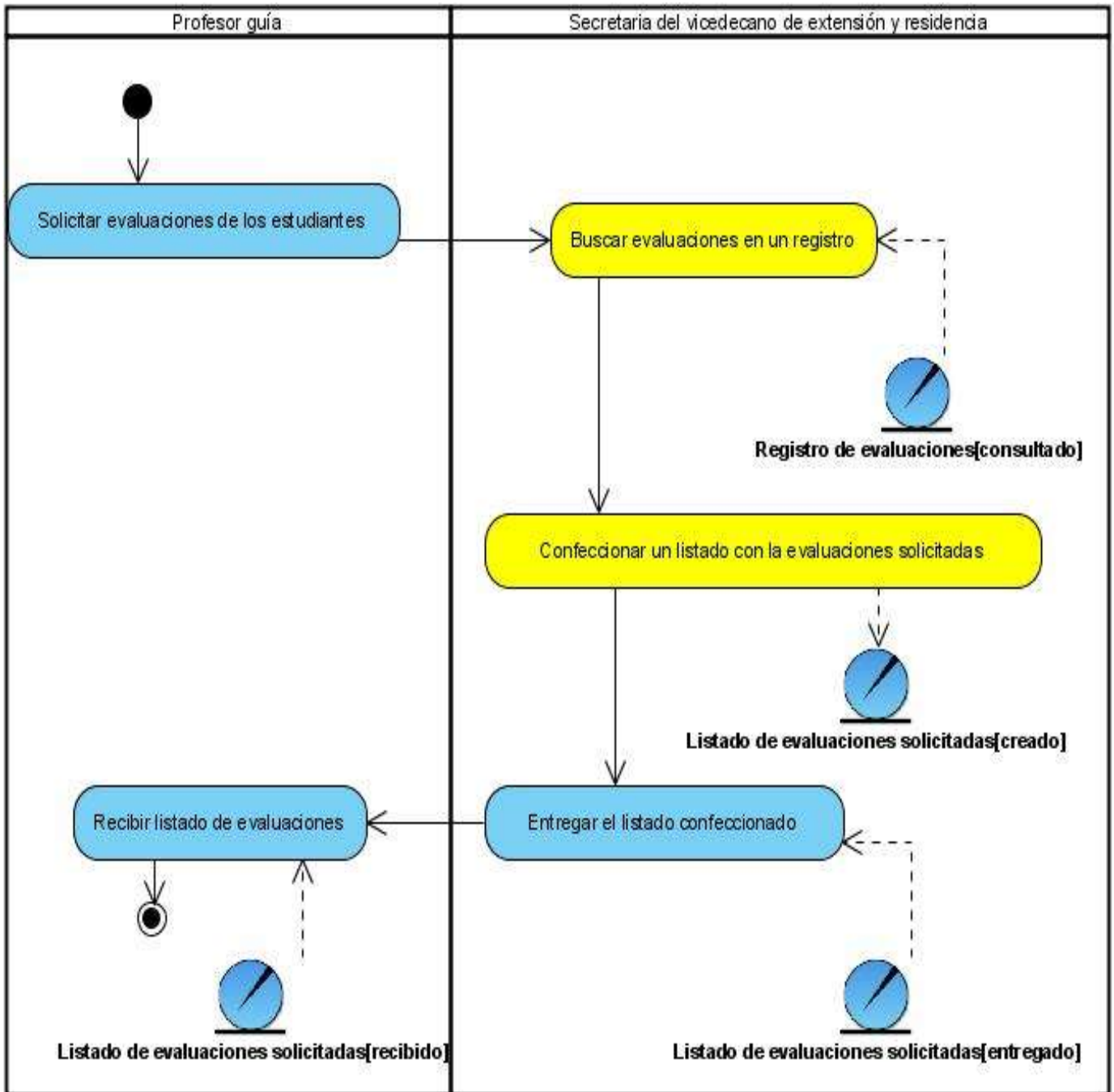


Figura 7 Diagrama de actividades CUN “Solicitar evaluación de los estudiantes en la residencia”

Diagrama de actividades CUN “Realizar trabajo socialmente útil”

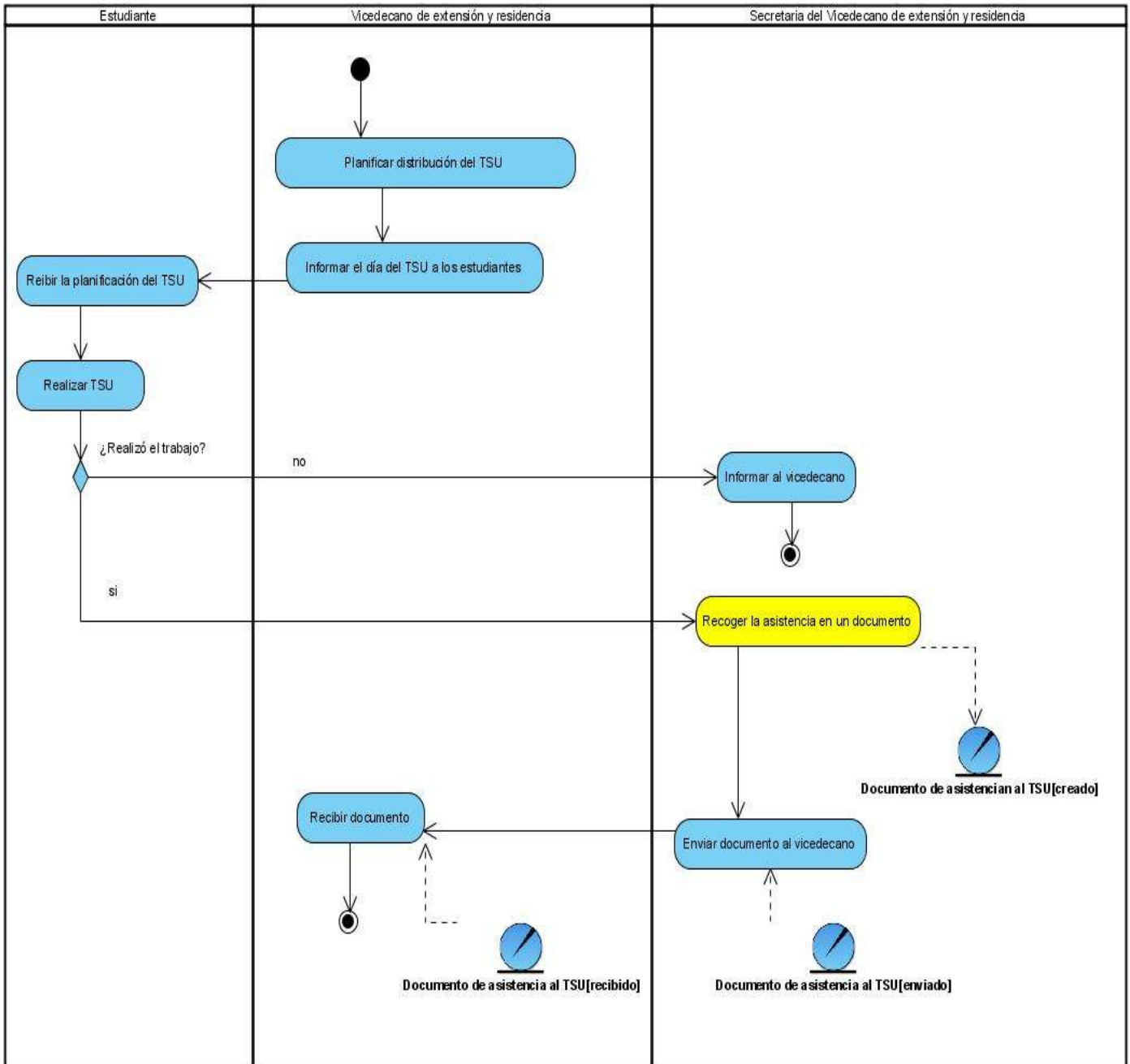


Figura 8 Diagrama de actividades CUN “Realizar trabajo socialmente útil”

Diagrama de actividades CUN “Realizar guardia obrera estudiantil”

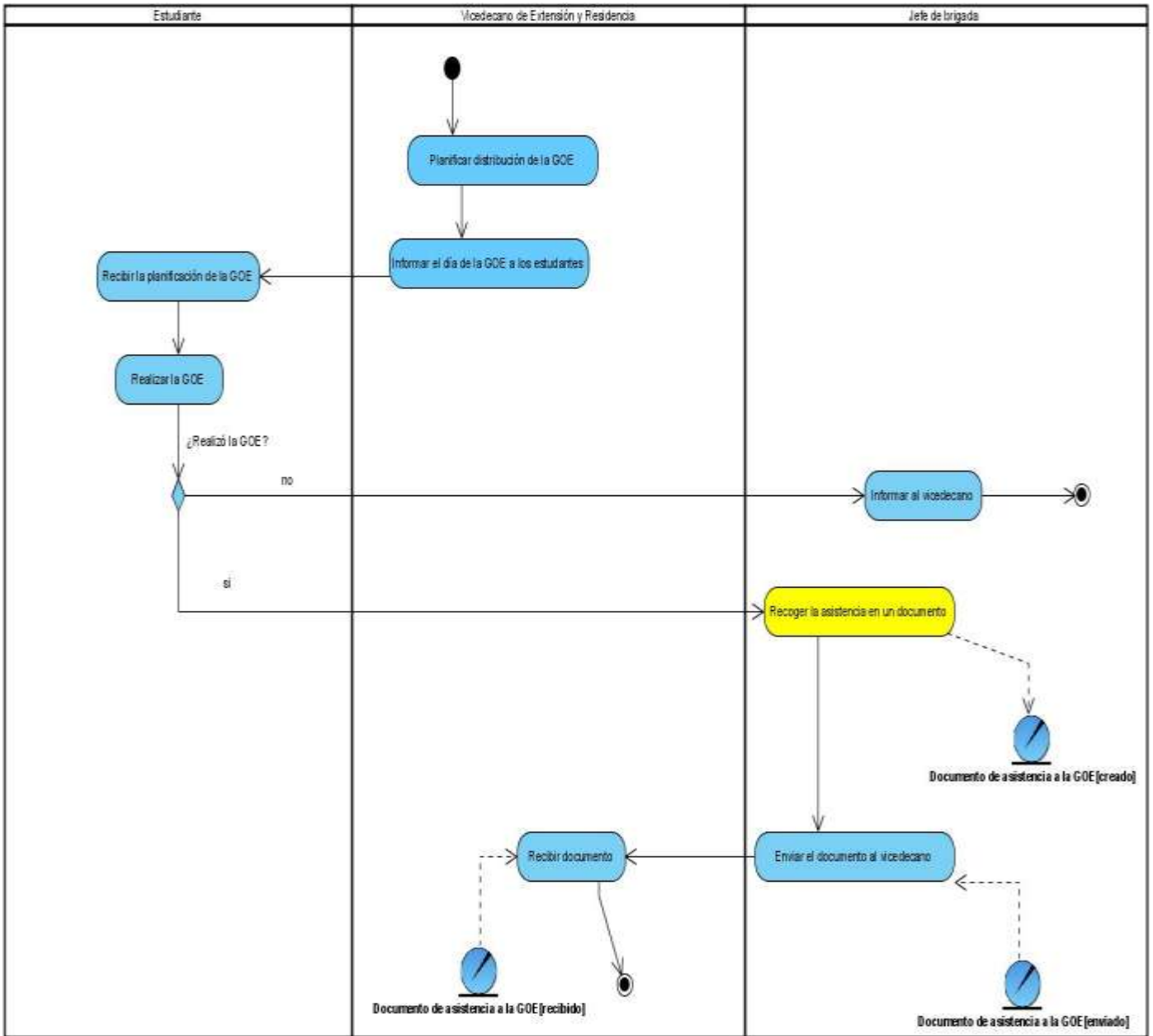


Figura 9 Diagrama de actividades CUN “Realizar guardia obrera estudiantil”

Diagrama de actividades CUN “Solicitar afectaciones de los apartamentos”

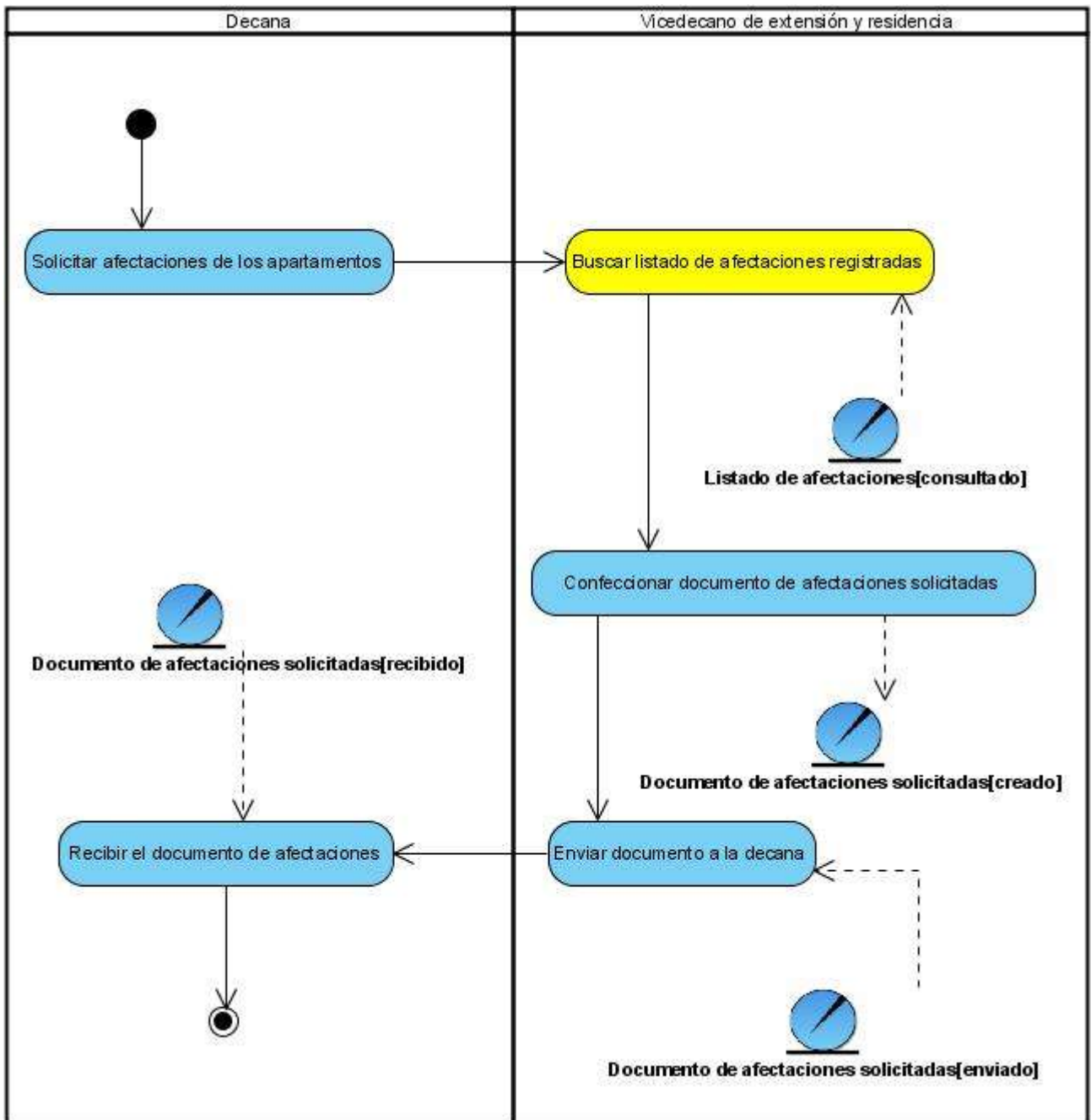


Figura 10 Diagrama de actividades CUN “Solicitar afectaciones de los apartamentos”

Diagrama de actividades CUN “Solicitar ubicación de los estudiantes”

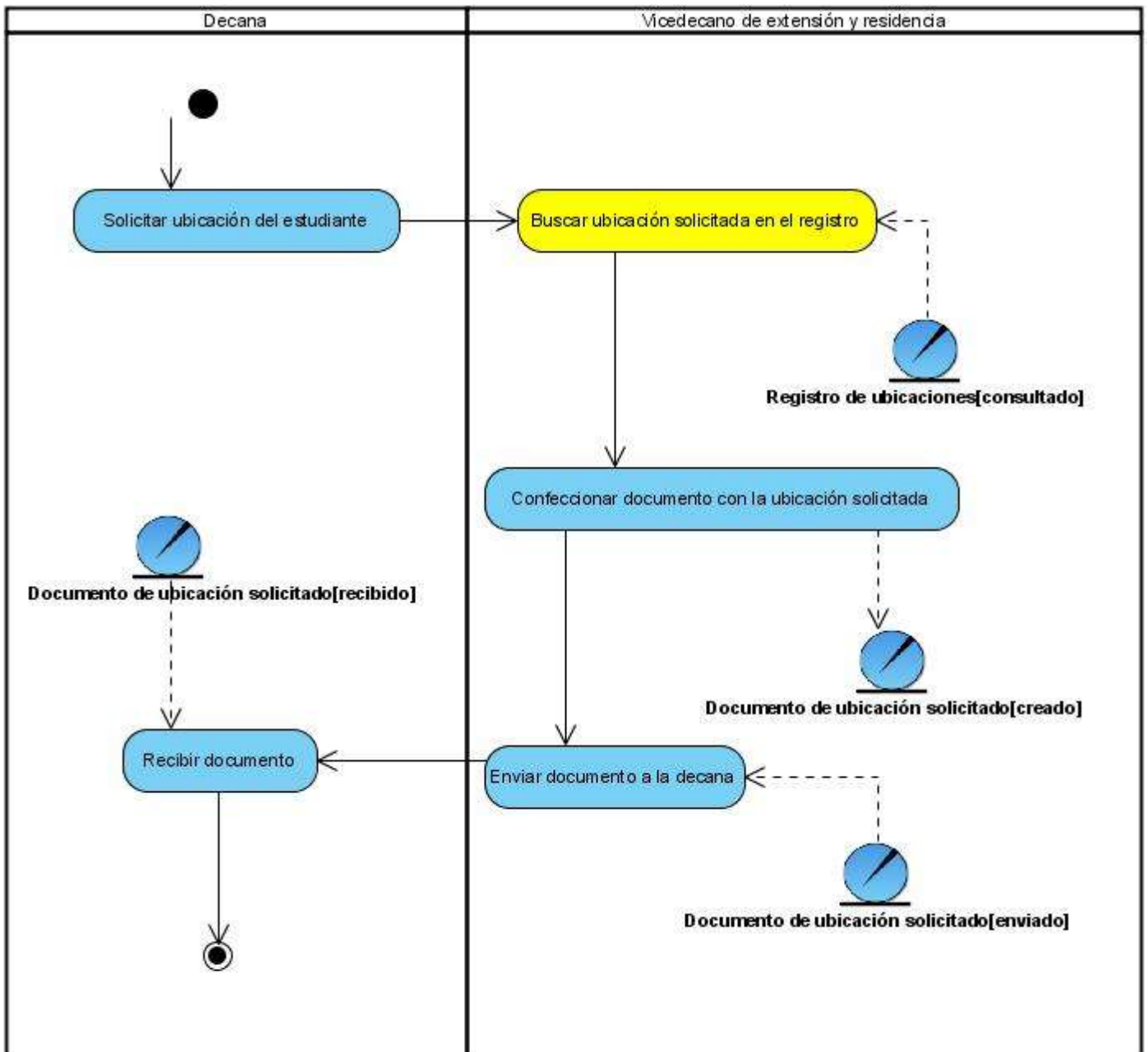


Figura 11 Diagrama de actividades CUN “Solicitar ubicación de los estudiantes”

Diagrama de actividades CUN “Realizar Paradas de beca”

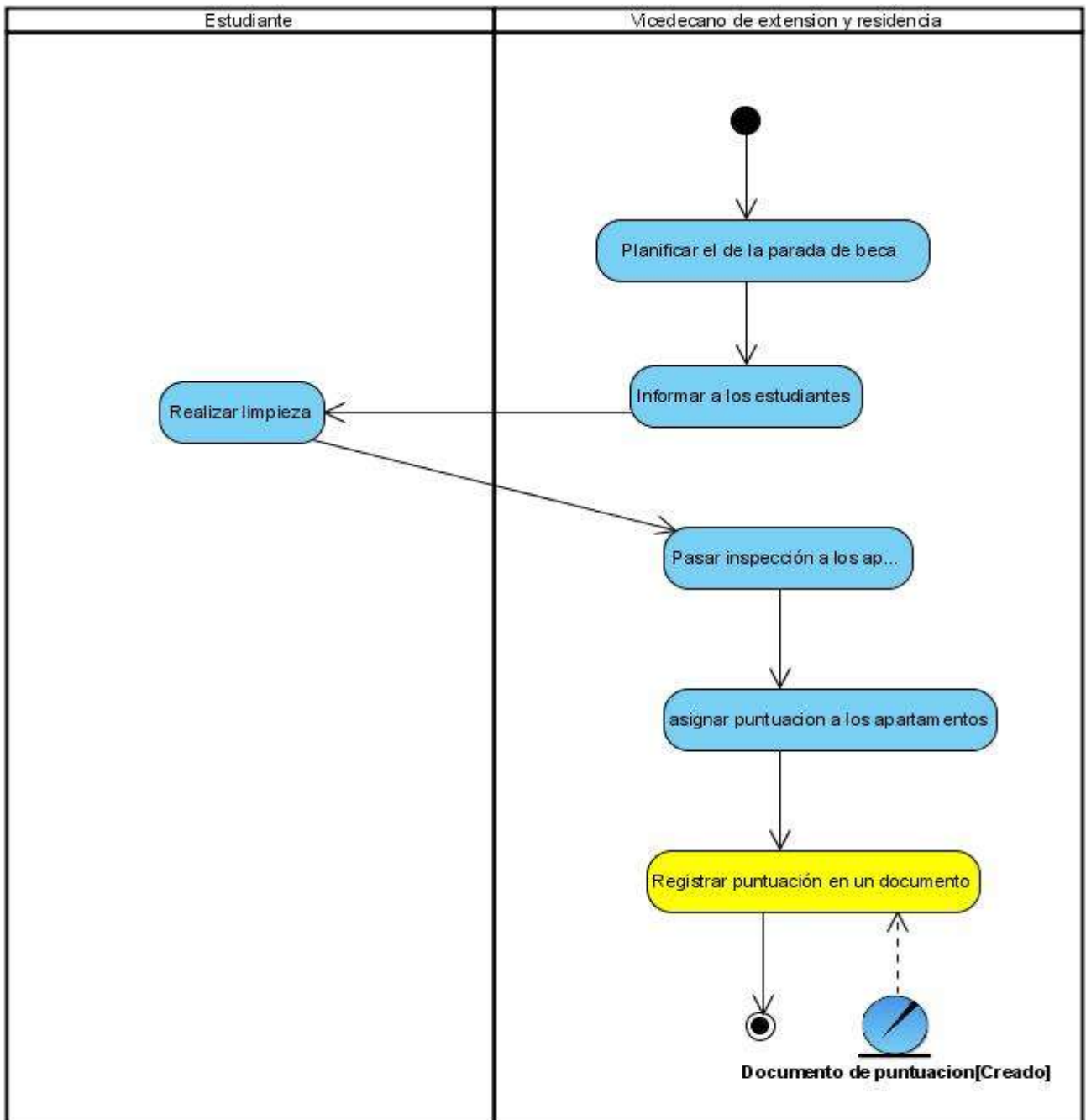


Figura 12 Diagrama de actividades CUN “Realizar Paradas de beca”

2.4.3.3-Modelo de objetos

El modelo de objetos del negocio, es un modelo interno a un negocio, muestra la relación entre los trabajadores y entidades del negocio dentro del flujo de trabajo de Modelamiento del negocio.

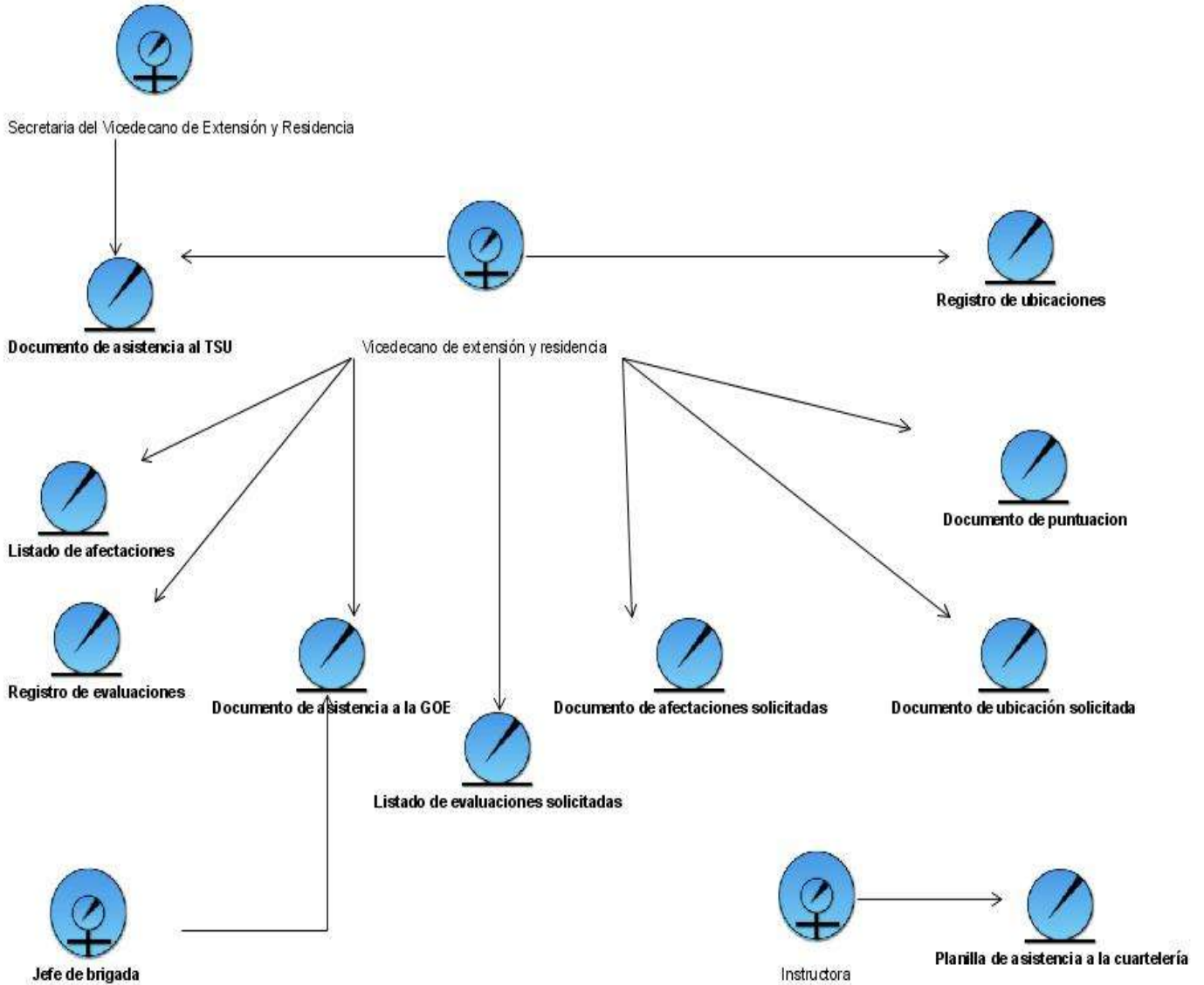


Figura 13 Diagrama de objetos

2.4.3.4-Reglas del negocio

Las reglas del negocio regulan aspectos del negocio, ya que describen políticas que deben cumplirse o condiciones que deben ser satisfechas.

1. Cuando comienza el curso el estudiante es ubicado en un apartamento según el sexo.
2. En un apartamento no pueden vivir a la vez varones y hembras.
3. El número que identifica a cada apartamento se conforma con el número del edificio, seguido del número de paso de escalera y seguido además del número consecutivo que posee cada apartamento en ese paso de escalera.
4. Todos los meses se le da al estudiante una evaluación según su comportamiento en la residencia.
5. Una brigada realiza el trabajo socialmente útil una vez al mes.
6. En un día realizan el trabajo socialmente útil una brigada en la sesión de la mañana y la otra en la sesión de la tarde o una brigada realiza las dos sesiones.
7. El tiempo que demora una brigada en realizar la guardia obrera estudiantil está en dependencia de la cantidad de grupos que haya en la facultad.
8. Un estudiante está de cuartelero en un paso de escalera de su edificio.
9. Cuando un estudiante no se presenta a realizar la guardia, trabajo socialmente útil, cuartelería se convierte en ausente.
10. La mayor cantidad de pisos de un edificio es 5.
11. La mayor cantidad de pasos de escalera de un edificio es 4 y la menor 1.
12. La mayor cantidad de apartamentos de un edificio por pisos es de 2 ó 4.

2.5-Levantamiento de Requisitos

2.5.1-Requisitos funcionales

Los requisitos funcionales son aquellas capacidades o condiciones que el sistema debe cumplir. Ellos definen qué es lo que el sistema debe hacer y permiten identificar las funcionalidades requeridas.

RF1-Gestionar datos de los edificios

RF1.1-Insertar datos del edificio.

RF1.2-Modificar datos del edificio.

RF1.3-Visualizar datos del edificio.

RF1.4-Eliminar datos del edificio.

RF1.5-Generar reporte de las capacidades libres por edificios.

RF2-Gestionar datos de los apartamentos

RF2.1-Insertar datos de los apartamentos.

RF2.2-Modificar datos del apartamento.

RF2.3-Visualizar datos del apartamento.

RF2.4-Eliminar datos del apartamento.

RF3-Gestionar datos de ubicación del estudiante

RF3.1-Insertar datos de ubicación del estudiante en la residencia.

RF3.2-Modificar datos de ubicación del estudiante en la residencia.

RF3.3-Buscar ubicación del estudiante.

RF3.4-Visualizar los datos de ubicación del estudiante.

RF3.5-Eliminar datos de ubicación del estudiante.

RF4-Gestionar datos de las instructoras

RF4.1-Insertar datos de la instructora.

RF4.2-Modificar datos de la instructora.

RF4.3-Visualizar datos de la instructora.

RF4.4-Eliminar datos de la instructora.

RF6-Gestionar cuartería

RF5.1-Insertar planificación de la cuartería.

RF5.2-Modificar planificación cuartería.

RF5.3-Visualizar planificación de la cuartería.

RF5.4-Eliminar planificación de la cuartería.

RF5.5-Insertar asistencia a la cuartería.

RF5.6-Modificar asistencia a la cuartería.

RF5.7-Generar reporte de la cuartería.

RF6-Gestionar guardia obrera estudiantil

RF.1-Insertar planificación de la guardia obrera estudiantil.

RF6.2-Modificar planificación de la guardia obrera estudiantil.

RF6.3-Visualizar planificación de la guardia obrera estudiantil.

RF6.4-Eliminar planificación de la guardia obrera estudiantil.

RF6.5-Insertar asistencia de la guardia obrera estudiantil.

RF6.6-Modificar asistencia de la guardia obrera estudiantil.

FR6.7-Generar reporte de la guardia obrera estudiantil.

RF7- Gestionar Trabajo Socialmente Útil (TSU)

FR7.1-Insertar planificación del TSU.

RF7.2-Modificar planificación del TSU.

RF7.3-Visualizar planificación del TSU.

RF7.4-Eliminar planificación del TSU.

RF7.5-Insertar asistencia al TSU.

RF7.6-Modificar asistencia al TSU.

RF7.7-Generar reporte del TSU.

RF8-Gestionar afectaciones de los apartamentos

RF8.1-Insertar afectaciones de los apartamentos.

RF8.2-Modificar afectaciones de los apartamentos.

RF8.3-Eliminar afectaciones de los apartamentos.

RF8.4-Generar reporte de las afectaciones de los apartamentos.

RF9-Gestionar evaluación del estudiante

- RF9.1-Insertar evaluación del estudiante.
- RF9.2-Modificar evaluación del estudiante.
- RF9.4-Eliminar evaluación del estudiante.
- RF9.5-Generar reporte de la evaluación del estudiante.

RF10-Gestionar fecha de las paradas de beca

- RF10.1- Insertar fecha de las paradas de beca.
- RF10.2- Visualizar fecha de las paradas de beca.
- RF10.3- Eliminar fecha de las paradas de beca.
- RF10.4- Modificar fecha de las paradas de beca.
- RF10.5-Generar reporte de la parada de beca.

RF12- Gestionar parada de beca apartamento

- RF12.1-Insertar puntuación de los apartamentos de las paradas de beca.
- RF12.2-Modificar puntuación de los apartamentos de las paradas de beca.
- RF12.3-Eliminar puntuación de los apartamentos de las paradas de beca.
- RF12.4-Visualizar puntuación de los apartamentos de las paradas de beca.

RF11- Gestionar parada de beca edificio

- RF11.1-Insertar puntuación de los edificios de las paradas de beca.
- RF11.2-Modificar puntuación de los edificios de las paradas de beca.
- RF11.3-Eliminar puntuación de los edificios de las paradas de beca.
- RF11.4-Visualizar puntuación de los edificios de las paradas de beca.

2.2-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.

Requerimientos de Software

Para el desarrollo:

- Sistema operativo Windows 2000 o superior.
- Navegador web Internet Explorer 5.5 o superior o Mozilla Firefox 2.0 o superior.

- Servidor web Apache 2.2 o superior.
- Servidor de Base de Datos PostgreSQL 8.2.

Para la explotación:

Cliente:

- Sistema operativo Windows 2000 o superior.
- Navegador web Internet Explorer (5.5 o superior) o Mozilla Firefox 2.0 o superior.

Servidor:

- Sistema operativo Windows 2000 o superior.
- Servidor de Base de Datos PostgreSQL 8.2.
- Servidor web Apache 2.2 o Superior, PHP 5.2 o superior.

Requerimientos de Hardware

Para el desarrollo:

- Pentium III o superior.
- 256 MB de memoria RAM o superior.
- Disco duro con capacidad de 40 GB o superior.

Para la explotación:

Cliente:

- Pentium II o superior.
- 64 MB de memoria RAM o superior.
- Disco duro con capacidad de 20 GB o superior.

Servidor:

- Pentium IV o superior a 2.0 GHZ.
- 2 GB de memoria RAM o superior.
- Disco duro con capacidad de 80 GB o superior.

2.6-Modelamiento del Sistema

2.6.1-Actores del Sistema

Descripción de los actores del sistema

Actores	Justificación
Vicedecano de residencia y extensión	Gestiona toda la información relacionada con los estudiantes en la residencia. Registra, busca, completa o elimina los datos de los estudiantes, apartamentos, edificios y actividades en la residencia, además puede generar reportes según criterios especificados.
Secretaria del Vicedecano de residencia y extensión	Solo posee acceso a consultar e ingresar la información relacionada con el Trabajo Socialmente Útil (TSU) y la Guardia Obrera Estudiantil (GOE).
Instructora	Solo posee acceso a consultar e ingresar la información relacionada con la cuarterería y las afectaciones de los apartamentos.
Profesor guía Decana Vicedecano de Formación Jefes de año	Solo poseen acceso a consultar toda la información relacionada con los estudiantes en la residencia.

Tabla 10 Descripción de los actores del sistema

2.6.2-Diagrama de casos de uso del sistema

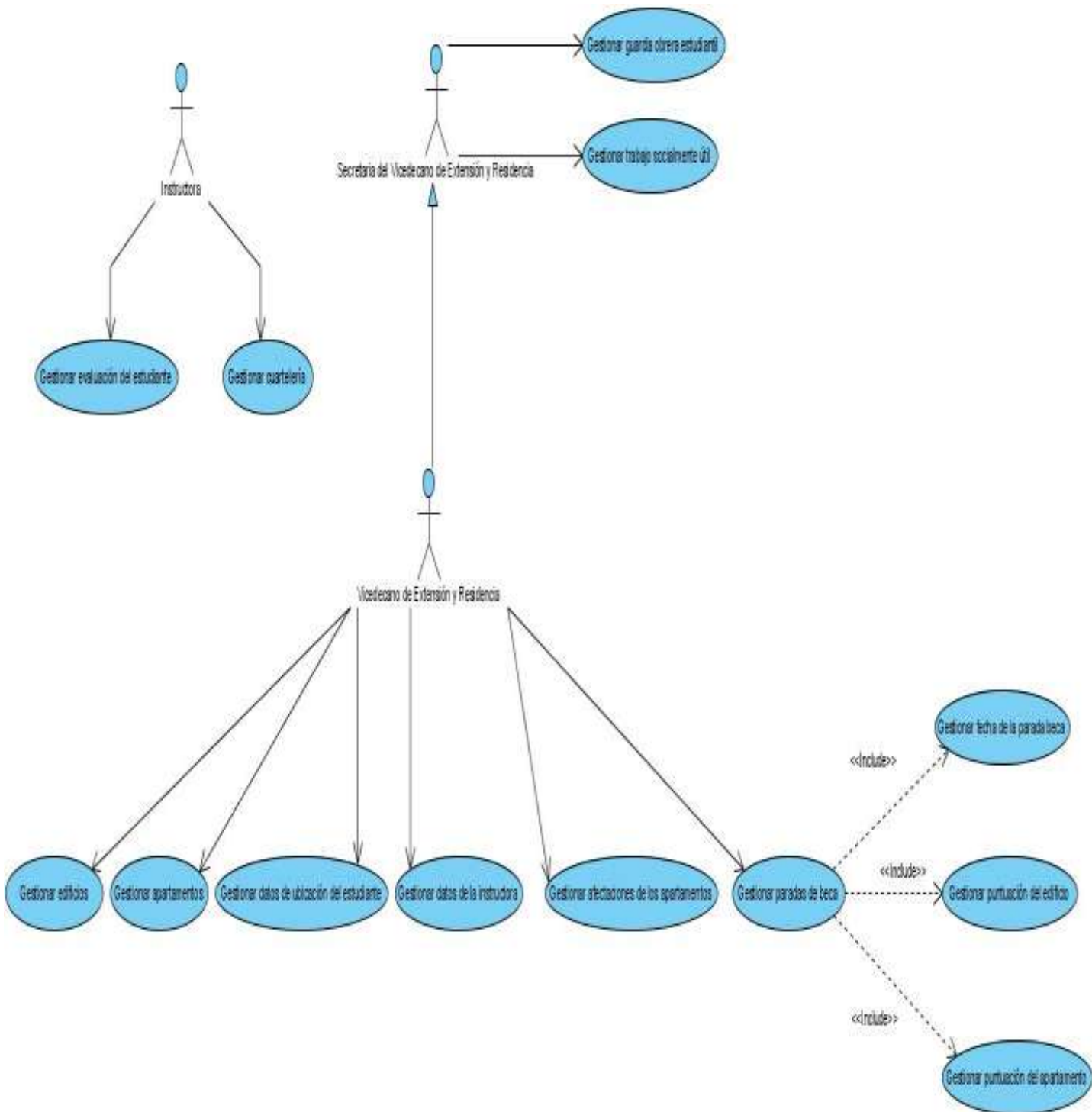


Figura 14 Diagrama de CUS

2.6.3-Descripción de los casos de uso del sistema

Descripción del CUS" Gestionar edificios"

CU_1	Gestionar edificios
Actor	Vicedecano de residencia y extensión (inicia)
Descripción	El caso de uso comienza cuando el Vicedecano de Extensión y Residencia decide Insertar datos del edificio, Modificar datos del edificio, Visualizar datos del edificio, Eliminar datos del edificio o Generar reporte de las capacidades libres por edificio. Luego de que el Vicedecano de Extensión y Residencia indique los datos del edificio a insertar, modificar o visualizar el sistema realiza la opción seleccionada y termina el caso de uso.
Referencia	RF1, RF1.1, RF1.2, RF1.3, RF1.4, RF1.5
Prioridad	Crítico

Tabla 11 Descripción del CUS" Gestionar edificios"

Descripción del CUS" Gestionar apartamentos"

CU_2	Gestionar apartamentos
Actor	Vicedecano de residencia y extensión
Descripción	El caso de uso comienza cuando el Vicedecano de Extensión y Residencia decide Insertar datos del apartamento, Modificar datos del apartamento, Visualizar datos del apartamento, Eliminar datos del apartamento o Generar reporte de las capacidades libres por apartamento. Luego que el Vicedecano de Extensión y Residencia indique los datos del apartamento a insertar, modificar, visualizar o generar reporte el sistema realiza la opción seleccionada y termina el caso de uso"
Referencia	RF2, RF2.1, RF2.2, RF2.3, RF2.4, RF2.5
Prioridad	Crítico

Tabla 12 Descripción del CUS” Gestionar apartamentos”

Descripción del CUS” Gestionar datos de ubicación del estudiante”

CU_3	Gestionar datos de ubicación del estudiante
Actor	Vicedecano de residencia y extensión
Descripción	El caso de uso comienza cuando el Vicedecano de Extensión y Residencia decide Insertar, Modificar, Visualizar, Buscar o Eliminar datos de ubicación del estudiante. Luego que el Vicedecano de Extensión y Residencia indique los datos de ubicación a insertar, modificar, visualizar o eliminar el sistema realiza la opción seleccionada y termina el caso de uso.
Referencia	RF3, RF3.1, RF3.2, RF3.3, RF3.4, RF3.5
Prioridad	Crítico

Tabla 13 Descripción del CUS” Gestionar datos de ubicación del estudiante”

Descripción del CUS” Gestionar datos de la instructora”

CU_4	Gestionar datos de la instructora
Actor	Vicedecano de residencia y extensión
Descripción	El caso de uso comienza cuando la instructora decide Insertar evaluación del estudiante, Modificar evaluación del estudiante o Generar reporte de la evaluación del estudiante. Luego de que la instructora indique la evaluación a insertar, modificar o generar reporte el sistema realiza la opción seleccionada y termina el caso de uso.
Referencia	RF4, RF4.1, RF4.2, RF4.3, RF4.4
Prioridad	Secundario

Tabla 14 Descripción del CUS” Gestionar datos de la instructora”

Descripción del CUS” Gestionar Cuartelería”

CU_5	Gestionar cuartería
Actor	Instructora
Descripción	El caso de uso comienza cuando la instructora decide Insertar planificación de la cuartería, Modificar planificación de la cuartería y Visualizar la planificación de la cuartería, así como Insertar y Modificar la asistencia de los estudiantes a la misma, y además permite generar reportes.
Referencia	RF5, RF5.1, RF5.2, RF5.3, RF5.4, RF5.5, RF5.6, RF5.7
Prioridad	Crítico

Tabla 15 Descripción del CUS” Gestionar Cuartería”

Descripción del CUS” Gestionar Guardia Obrera Estudiantil”

CU_6	Gestionar guardia obrera estudiantil
Actor	Vicedecano de residencia y extensión , Secretaria del Vicedecano de residencia y extensión
Descripción	El caso de uso comienza cuando el Vicedecano de Extensión y residencia decide Insertar planificación de la guardia obrera estudiantil, Modificar planificación de la guardia obrera estudiantil y Visualizar la planificación de la guardia obrera estudiantil, así como Insertar y Modificar la asistencia de los estudiantes a la misma, y además permite generar reportes.
Referencia	RF6, RF6.1, RF6.2, RF6.3, RF6.4, RF6.5, RF6.6, RF6.7
Prioridad	Crítico

Tabla 16 Descripción del CUS” Gestionar Guardia Obrera Estudiantil”

Descripción del CUS” Gestionar Trabajo Socialmente Útil”

CU_7	Gestionar Trabajo Socialmente Útil
-------------	---

Actor	Vicedecano de residencia y extensión, Secretaria del Vicedecano de residencia y extensión.
Descripción	El caso de uso comienza cuando el Vicedecano de Extensión y residencia decide Insertar planificación del trabajo socialmente útil, Modificar planificación del trabajo socialmente útil y Visualizar la planificación del trabajo socialmente útil, así como Insertar y Modificar la asistencia de los estudiantes al mismo, y además permite generar reportes.
Referencia	RF7, RF7.1, RF7.2, RF7.3, RF7.4, RF7.5, RF7.6, RF7.7
Prioridad	Crítico

Tabla 17 Descripción del CUS” Gestionar Trabajo Socialmente Útil”

Descripción del CUS” Gestionar afectaciones de los apartamentos”

CU_8	Gestionar afectaciones de los apartamentos
Actor	Vicedecano de residencia y extensión
Descripción	El caso de uso comienza cuando el Vicedecano de Extensión y Residencia decide Insertar afectaciones de los apartamentos, Modificar afectaciones de los apartamentos, Eliminar afectaciones de los apartamentos o Generar reporte de afectaciones de los apartamentos. Luego de que el Vicedecano de Extensión y Residencia indique los datos del edificio a insertar, modificar o visualizar el sistema realiza la opción seleccionada y termina el caso de uso.
Referencia	RF8, RF8.1, RF8.2, RF8.3, RF8.4
Prioridad	Crítico

Tabla 18 Descripción del CUS” Gestionar afectaciones de los apartamentos”

Descripción del CUS” Gestionar evaluación del estudiante”

CU_9	Gestionar evaluación del estudiante
-------------	--

Actor	Instructora
Descripción	El caso de uso comienza cuando la instructora decide Insertar evaluación del estudiante, Modificar evaluación del estudiante o Generar reporte de la evaluación del estudiante. Luego de que la instructora indique la evaluación a insertar, modificar o generar reporte el sistema realiza la opción seleccionada y termina el caso de uso.
Referencia	RF9, RF9.1, RF9.2, RF9.3, RF9.4
Prioridad	Secundario

Tabla 19 Descripción del CUS” Gestionar evaluación del estudiante”

Descripción del CUS” Gestionar parada de beca”

CU_10	Gestionar paradas de beca
Actor	Vicedecano de residencia y extensión
Descripción	El caso de uso comienza cuando el vicedecano de residencia y extensión decide Insertar, Modificar, Visualizar y Eliminar las fechas en que se efectuarán las paradas de beca así como las puntuaciones de los apartamentos y edificios en dichas paradas de beca, además permite generar reporte de la puntuación de las paradas de beca tanto para los edificios como para los apartamentos, luego de que el vicedecano de residencia y extensión indique la fecha o puntuación a insertar, modificar, visualizar, eliminar o generar reporte el sistema realiza la opción seleccionada y termina el caso de uso.
Referencia	RF10, RF10.1, RF10.2, RF10.3, RF10.4, RF10.5, RF10.6, RF10.7
Prioridad	Secundario

Tabla 20 Descripción del CUS” Gestionar parada de beca”

2.7-Pautas del Diseño.

Para lograr una agradable apariencia y facilitar el uso del software se definieron algunas pautas de diseño como son:

- Cuando se le solicita datos al usuario se hará a través de una tabla de dos columnas. Ésta tabla debe estar centrada en el formulario. En la columna izquierda se ubicará el nombre del dato solicitado al usuario alineado a la derecha y en la columna derecha el componente adecuado para recoger el dato solicitado y éste estará alineado a la izquierda.
- Los mensajes de error ocurridos durante la validación de los formularios se mostrarán mediante las respectivas ventanas emergentes en dependencia de si se notificará un error, una alerta o alguna otra información relacionada a la entrada de datos y se resaltará en color rojo el campo que contiene el error.
- Los componentes como campos de selección y campos de texto ubicados en la misma columna deben mantener la uniformidad con respecto al tamaño. Su tamaño debe estar en correspondencia con el texto que se va a recoger en ellos.
- Para la selección de datos de tablas nomencladores como Provincia, Municipio, se hará uso del componente Listbox, y el elemento 0 debe ser textualmente “<<Selecione>>”.
- Para representar campos cuyos datos de entrada son obligatorios, se colocará al lado derecho del componente en el cual el usuario entrará los datos un asterisco en color rojo.
- Para tablas donde se solicitan datos de localidad, estos deben estar organizados verticalmente siempre presentando primero en orden de arriba hacia abajo a los de menor extensión.

2.8-Matriz de trazabilidad.

-->	Gestionar edi...	Gestionar ap...	Gestionar Ubi...	Gestionar dat...	Gestionar cua...	Gestionar gu...	Gestionar tra...	Gestionar afe...	Gestionar ev...	Gest
Insertar datos del edificio	X									
Modificar datos del edificio	X									
Eliminar datos del edificio	X									
Visualizar datos del edificio	X									
Generar reporte de capacidades libres por edificio	X									
Insertar datos del apartamento		X								
Modificar datos de los apartamentos		X								
Eliminar datos de los apartamentos		X								
Visualizar datos del apartamento		X								
Insertar ubicacion										
Modificar ubicacion										
Eliminar ubicacion										
Generar reporte de la ubicacion										
Insertar datos de la instructora			X							
Modificar datos de la instructora			X							
Eliminar datos de la instructora			X							
Visualizar datos de la instructora			X							
Insertar planificacion de la cuarteleria					X					
Modificar planificacion de la cuarteleria					X					
Eliminar planificacion de la cuarteleria					X					
Visualizar planificacion de la cuarteleria					X					
Insertar asistencia a la cuarteleria					X					
Modificar asistencia a la cuarteleria					X					
Generar reporte de la cuarteleria					X					
Insertar planificacion de la guardia						X				
Modificar planificacion de la guardia						X				
Eliminar planificacion de la guardia						X				
Visualizar planificacion de la guardia						X				
Insertar asistencia a la guardia						X				
Modificar asistencia a la guardia						X				
Insertar planificacion del tsu							X			
Modificar planificacion del tsu							X			
Eliminar planificacion del tsu							X			
Visualizar planificacion del tsu							X			
Insertar asistencia al tsu							X			
Modificar asistencia al tsu							X			
Generar reporte del tsu							X			
Insertar afectaciones de los apartamentos								X		
Modificar afectaciones de los apartamentos								X		
Eliminar afectaciones de los apartamentos								X		
Visualizar afectaciones de los apartamentos								X		
Generar reporte de afectaciones de los apartam...								X		
Insertar evaluacion del estudiante									X	
Modificar evaluacion del estudiante									X	
Eliminar evaluacion del estudiante									X	
Visualizar evaluacion del estudiante									X	
Generar reporte de evaluacion del estudiante									X	
Insertar fecha parada beca										X
Modificar fecha parada beca										X
Eliminar fecha parada beca										X
Visualizar fecha parada beca										X
Insertar puntuacion edificio										
Modificar puntuacion edificio										
Eliminar puntuacion edificio										
Visualizar puntuacion edificio										
Insertar puntuacion apartamento										
Modificar puntuacion apartamento										
Eliminar puntuacion apartamento										
Visualizar puntuacion apartamento										
Generar reporte de la parada de beca										

Figura 15 Matriz de trazabilidad

Conclusiones

Partiendo de la comprensión de los procesos del negocio en este capítulo, se definieron las principales funcionalidades que debe tener el sistema. Se elaboró el diagrama de casos de uso del

negocio y sistema, se definieron 58 requisitos funcionales, agrupados en 10 casos de usos los cuales se describieron textualmente, se elaboraron los diagramas de objetos de negocio y los de actividades de cada caso de uso. Todo lo anterior, proporciona una visión general de lo que el sistema hace.

CAPÍTULO 3: DISEÑO DEL SISTEMA.

Introducción

En este capítulo se aborda el flujo de trabajo de RUP Análisis y Diseño del sistema; mediante los artefactos que se generan en este flujo se modelan los casos de usos seleccionados. Se profundiza en los casos de usos de manera que permitan expresar una vista interna del sistema. El propósito fundamental es lograr una mejor y más detallada vista de la aplicación en relación a la propuesta final del sistema así como obtener una implementación que satisfaga las necesidades de los usuarios. El diseño tiene en cuenta los requisitos no funcionales, en resumen, cómo cumple el sistema sus objetivos.

3.1-Aplicación de patrones

3.1.1-Aplicación de patrones GRASP en Symfony.

Son muchos los patrones que se aplican en la implementación con Symfony, a continuación se mencionan algunos ejemplos de los evidenciados, ubicándolos en las capas de Modelo y Control que plantea el patrón arquitectónico MVC.

Experto: Es uno de los patrones más utilizados al trabajar con Symfony, con la inclusión de Propel para mapear la BD. Propel genera las clases para la gestión de las entidades con las responsabilidades asignadas correctamente según el patrón Experto, las clases de abstracción de datos (Peer del Modelo) cuentan con un conjunto de funcionalidades relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan.

```
abstract class BaseEdificioPeer {  
  
    const DATABASE_NAME = 'propel';  
  
    const TABLE_NAME = 'edificio';  
  
    const CLASS_DEFAULT = 'lib.model.Edificio';  
  
    const NUM_COLUMNS = 5;  
  
    const NUM_LAZY_LOAD_COLUMNS = 0;  
  
    const ID_EDIFICIO = 'edificio.ID_EDIFICIO';  
  
    const NUMERO_EDIFICIO = 'edificio.NUMERO_EDIFICIO';  
  
    const CANT_APARTAMENTOS = 'edificio.CANT_APARTAMENTOS';  
  
    const CANT_PASOS_ESCALERA = 'edificio.CANT_PASOS_ESCALERA';  
  
    const SEXO = 'edificio.SEXO';  
  
}
```

```

abstract class BaseEdificio extends BaseObject implements Persistent {

    const PEER = 'EdificioPeer';

    protected static $peer;

    protected $id_edificio;

    protected $numero_edificio;

    protected $cant_apartamentos;

    protected $cant_pasos_escalera;

    protected $sexo;
    
```

```

public static function getEdificios()
{
    $edificios=EdificioPeer::doSelect(new Criteria());

    $arreglo_edificios=array('-1'=>'');
    if ($edificios)
    foreach ($edificios as $edif)
    {
        if ($edif)
        $arreglo_edificios[$edif->getIdEdificio()]= $edif->getNumeroEdificio();
    }

    return $arreglo_edificios;
}
    
```

Creador: En la clase edificiosActions se encuentran las acciones definidas para el módulo gestionar_edificios. En dichas acciones se crean los objetos de las clases que son los que instancias a las clases del modelo, lo que evidencia que la clase edificiosActions es “creador” de dichas entidades. Ejemplo del uso de este patrón es en la acción executeInsertar () mediante la creación de las instancias de las clases entidades que contienen los datos del edificio.

```

public function executeInsertar(sfWebRequest $request)
{
    $this->form = new EdificioForm();
}
    
```

Alta Cohesión: El trabajar con Symfony permite la organización del trabajo en cuanto a la estructura del proyecto, esto proporciona crear y trabajar con clases con una alta cohesión. Ejemplo de esto se evidencia en la clase Actions, la cual está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas, teniendo un sentido común y un propósito único, siendo las mismas las encargadas de controlar las acciones de las plantillas.

Bajo Acoplamiento: En el modelo también se encuentran las clases que implementan la lógica de negocio y de acceso a datos, estas clases no tienen asociaciones con las de la vista o el controlador por lo que la dependencia en este caso es baja, poniéndose de manifiesto este patrón.

Controlador: Este patrón se observa en las clases `sfFrontController`, `sfFrontWebController`, `sfContext`, los “actions”, el `index.php` del ambiente, etc. Symfony aplica el patrón “Front Controller” (Controlador frontal) y por tanto tiene una estructura bien organizada de controladores, que parte desde el “`index.php`” del ambiente y terminan en los “actions”. Hay controladores que se encargan de la seguridad del sistema trabajando con ficheros YML, otros sólo velan por identificar mediante unos datos las clases que deben realizar determinadas tareas.

3.1.2-Aplicación de patrones GoF en Symfony

Decorador: este método corresponde a la clase abstracta `sfView`, padre de todas las vistas, las que contienen un decorador para permitir añadir funcionalidades dinámicamente.

El archivo llamado `layout.php` es el que contiene el Layout de la página. Este archivo, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este comportamiento es una implementación del patrón de diseño llamado “Decorator”.

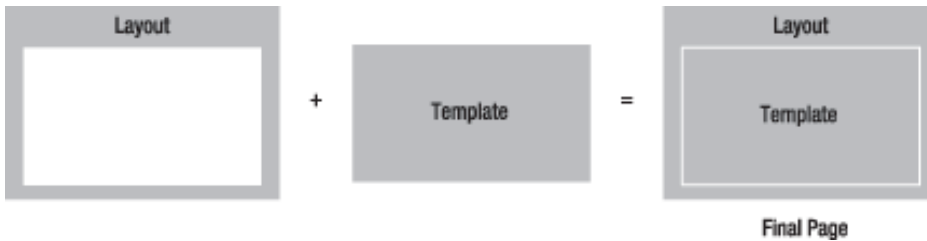


Figura 16 Plantilla decorada con un layout

Singleton: se aplica en el método getInstance de la clase sfRouting, el cual garantiza que esa clase sólo tenga una única instancia. La clase sfRouting es una de las que utiliza el controlador frontal (sfFrontWebController), es muy utilizada porque es la encargada de enrutar todas las peticiones que se hagan a la aplicación. La clase sfRouting define otros métodos muy útiles para la gestión manual de las rutas: ClearRoutes (), hasRoutes (), getRoutesByName.

Command: Este patrón se pone de manifiesto en la clase sfWebFrontController, en el método dispatch (), que es la encargada de determinar cual módulo y acción usar en dependencia de la petición de parámetros especificados por el usuario. Esta clase es la que está por defecto. La clase sfRouting también aplica este patrón, se encuentra desactivada por defecto y actúa en dependencia de las necesidades del administrador del sistema donde se aplique el framework, la cual se puede activar o desactivar. Pero donde más se evidencia este patrón es en la clase sfWebFrontController.

```

/**
 * Dispatches a request.
 *
 * This will determine which module and action to use by request parameters
 specified by the user.
 */
public function dispatch()
{
    try
    {
        if (sfConfig::get('sf_logging_enabled'))
        {
            $this->getContext()->getLogger()->info('{sfController} dispatch
request');
        }

        // reinitialize filters (needed for unit and functional tests)
        sfFilter::$filterCalled = array();

        // determine our module and action
        $request = $this->getContext()->getRequest();
        $moduleName = $request->getParameter('module');
        $actionName = $request->getParameter('action');

        // make the first request
        $this->forward($moduleName, $actionName);
    }
    ..
}

```

3.1.3-Patrón MVC que implementa Symfony

El uso de un framework que utiliza MVC obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por el framework. El código de la presentación se guarda en la vista, el código de manipulación de datos se guarda en el modelo y la lógica de procesamiento de las peticiones constituye el controlador. Aplicar el patrón MVC a una aplicación resulta bastante útil además de restrictivo.

La implementación que realiza Symfony de la arquitectura MVC incluye varias clases como son:

- `sfController` es la clase del controlador. Se encarga de decodificar la petición y transferirla a la acción correspondiente.
- `sfRequest` almacena todos los elementos que forman la petición (parámetros, cookies, cabeceras, etc.)
- `sfResponse` contiene las cabeceras de la respuesta y los contenidos. El contenido de este objeto se transforma en la respuesta HTML que se envía al usuario.
- El contexto (que se obtiene mediante `sfContext::getInstance()`) almacena una referencia a todos los objetos que forman el núcleo de Symfony y puede ser accedido desde cualquier punto de la aplicación.

La vista se encarga de producir las páginas que se muestran como resultado de las acciones, aquí se encuentra el layout, que es común para todas las páginas. La vista en Symfony está compuesta por diversas partes, estando cada una de ellas especialmente preparada para que pueda ser fácilmente modificable por la persona que normalmente trabaja con cada aspecto del diseño de las aplicaciones.

En el controlador se encuentran las acciones, siendo estas el corazón de la aplicación, ya que contienen toda la lógica de la aplicación. Las acciones utilizan el modelo y definen variables para la vista. Cuando se realiza una petición web en una aplicación Symfony, la URL define una acción y los parámetros de la petición.

En el Modelo se encuentran las clases, las cuales se generan de forma automática en dependencia de la estructura de la BD. En Symfony, el acceso a los datos, se realiza mediante objetos. Propel es el motor generador que se encarga de esta generación automática para construir sus clases, creando la estructura y generando el código de las mismas. Se crean 4 archivos por cada tabla de la BD. Los cuales se expondrán brevemente a continuación:

A medida que el desarrollo del proyecto va avanzando, puede ser necesario añadir métodos y propiedades personalizadas en los objetos del modelo y con esto se aumenten las tablas o columnas. Además, cada vez que se modifica se deben regenerar las clases del modelo de objetos. Si se añaden los métodos personalizados en las clases que se generan, se borrarían cada vez que se vuelvan a generar esas clases.

Las clases con nombre Base son las que se generan directamente a partir del esquema. No se deberían modificar esas clases, porque cada vez que se genera el modelo, se borran todas las clases. Por otra parte, las clases de objetos propias que heredan de las clases con nombre Base, no se modifican por lo que son las clases en las que se añaden los métodos propios. Estas clases heredan todos los métodos de la clase padre correspondiente, pero no le afectan las modificaciones en el esquema.

3.2-Modelo de diseño

El modelo de diseño está muy cercano al de implementación, lo que es natural para guardar y mantener el modelo de diseño a través del ciclo de vida completo del software. El modelo de diseño constituye un acercamiento a la arquitectura final de la aplicación y un esquema por el cual se registrarán los desarrolladores.

3.2.1-Descripción de las clases del diseño

Descripción del CU" gestionar edificio"

Nombre de la clase: Gestionar Edificio_class_Actions
Nombre del método: executeInsertarDatosDelEdificio()
Descripción: Este método se encarga de que el usuario inserte los datos del edificio.
Nombre del método: executeModificarDatosDelEdificio ()
Descripción: Este método se encarga de modificar la información del edificio que haya sido seleccionado.
Nombre del método: executeVisualizarDatosDelEdificio ()
Descripción: Este método se encarga de visualizar la información de los edificios existentes.
Nombre del método: executeEliminarDatosDelEdificio ()

Descripción: Este método se encarga de eliminar los datos del edificio seleccionado.
Nombre del método: executeGenerarReporteDeCapacidadesPorEdificio ()
Descripción: Este método es el encargado de generar un reporte de las capacidades libres por los apartamentos del edificio seleccionado.

Tabla 21 Descripción del CU" gestionar edificio"

Descripción del CU" Gestionar apartamento"

Nombre de la clase: Gestionar Apartamento_class_Actions
Nombre del método: executeInsertarDatosDeLosApartamentos()
Descripción: Este método se encarga de insertar los datos del apartamento.
Nombre del método: executeModificarDatosDelApartamento ()
Descripción: Este método se encarga de modificar la información del apartamento que haya sido seleccionado.
Nombre del método: executeVisualizarDatosDelApartamento ()
Descripción: Este método se encarga de visualizar la información de los Apartamentos existentes.
Nombre del método: executeEliminarDatosDeLosApartamentos ()
Descripción: Este método es el encargado de eliminar los datos del apartamento seleccionado.

Tabla 22 Descripción del CU" Gestionar apartamento"

Descripción del CU" Gestionar ubicación del estudiante"

Nombre de la clase: Gestionar Ubicación_class_Actions
Nombre del método: executeInsertarDatosDelEstudianteEnLaResidencia()
Descripción: Este método se encarga de insertar los datos de la ubicación del estudiante en la residencia.
Nombre del método: executeModificarDatosDelEstudianteEnLaResidencia ()
Descripción: Este método se encarga de modificar los datos de la ubicación del estudiante que haya sido seleccionado.
Nombre del método: executeVisualizarDatosDeUbicaciónDelEstudiante ()
Descripción: Este método se encarga de visualizar la información de ubicación del

estudiante seleccionado.
Nombre del método: executeEliminarDatosDeUbicaciónDelEstudiante ()
Descripción: Este método es el encargado de eliminar los datos de la ubicación del estudiante seleccionado.
Nombre del método: executeBuscarUbicaciónDelEstudiante ()
Descripción: Este método es el encargado de buscar los datos de la ubicación del estudiante seleccionado.

Tabla 23 Descripción del CU” Gestionar ubicación del estudiante”

Descripción del CU” Gestionar datos de la instructora”

Nombre de la clase: Gestionar datos instructora_class_Actions
Nombre del método: executeInsertarDatosDeLaInstructora()
Descripción: Este método se encarga de insertar los datos de la instructora.
Nombre del método: executeModificarDatosDeLaInstructora ()
Descripción: Este método se encarga de modificar los datos de la instructora que haya sido seleccionado.
Nombre del método: executeVisualizarDatosDeLaInstructora ()
Descripción: Este método se encarga de visualizar la información de la instructora seleccionada.
Nombre del método: executeEliminarDatosDeLaInstructora ()
Descripción: Este método es el encargado de eliminar los datos de la instructora seleccionada.

Tabla 24 Descripción del CU” Gestionar datos de la instructora”

Descripción del CU” Gestionar cuartería”

Nombre de la clase: Gestionar Cuartería_class_Actions
Nombre del método: executeInsertarPlanificaciónDeLaCuartería()
Descripción: Este método se encarga de insertar los datos de la planificación de la cuartería.
Nombre del método: executeModificarPlanificaciónDeLaCuartería()
Descripción: Este método se encarga de modificar los datos de la planificación de la cuartería seleccionada.
Nombre del método: executeVisualizarPlanificaciónDeLaCuartería ()

Descripción: Este método se encarga de visualizar la información de la planificación seleccionada.
Nombre del método: executeEliminarPlanificaciónDeLaCuarteria ()
Descripción: Este método es el encargado de eliminar los datos de la planificación seleccionada.
Nombre del método: executeInsertarAsistenciaAlaCuarteria ()
Descripción: Este método es el encargado de insertarle la asistencia a la cuartería seleccionada.
Nombre del método: executeModificarAsistenciaAlaCuarteria ()
Descripción: Este método es el encargado de modificar la asistencia a la cuartería seleccionada.
Nombre del método: executeGenerarReporteDeLaCuarteria ()
Descripción: Este método es el encargado de generar un reporte de las Cuarterías realizadas. Utiliza el método mostrarAusentes() que este muestra un listado con los ausentes a la cuartería de la fecha seleccionada.
Nombre del método: mostrarAusentes()
Descripción: Este método es el encargado de buscar y mostrar los estudiantes que se ausentaron a la cuartería.

Tabla 25 Descripción del CU" Gestionar cuartería"

Descripción del CU" Gestionar Guardia Obrera Estudiantil"

Nombre de la clase: Gestionar GOE_class_Actions
Nombre del método: executeInsertarPlanificaciónDeLaGuardiaObreraEstudiantil ()
Descripción: Este método se encarga de insertar los datos de la planificación de la Guardia Obrera Estudiantil.
Nombre del método: executeModificarPlanificaciónDeLaGuardiaObreraEstudiantil ()
Descripción: Este método se encarga de modificar los datos de la planificación de la Guardia Obrera Estudiantil seleccionada.
Nombre del método: executeVisualizarPlanificaciónDeLaGuardiaObreraEstudiantil ()
Descripción: Este método se encarga de visualizar la información de la planificación seleccionada.

Nombre del método: executeEliminarPlanificaciónDeLaGuardiaObreraEstudiantil ()
Descripción: Este método es el encargado de eliminar los datos de la planificación seleccionada.
Nombre del método: executeInsertarAsistenciaAlaGuardiaObreraEstudiantil ()
Descripción: Este método es el encargado de insertarle la asistencia a la Guardia Obrera Estudiantil seleccionada.
Nombre del método: executeModificarAsistenciaAlaGuardiaObreraEstudiantil ()
Descripción: Este método es el encargado de modificar la asistencia a la Guardia Obrera Estudiantil seleccionada.
Nombre del método: executeGenerarReporteDeLaGuardiaObreraEstudiantil ()
Descripción: Este método es el encargado de generar un reporte de las Guardia Obrera Estudiantil realizadas hasta el momento. Utiliza el método mostrarAusentes() que este muestra un listado con los ausentes a la guardia de la fecha seleccionada.
Nombre del método: mostrarAusentes()
Descripción: Este método es el encargado de buscar y mostrar los estudiantes que se ausentaron a la guardia.

Tabla 26 Descripción del CU" Gestionar Guardia Obrera Estudiantil"

Descripción del CU" Gestionar Trabajo Socialmente Útil"

Nombre de la clase: Gestionar TSU_class_Actions
Nombre del método: executeInsertarPlanificaciónDelTSU ()
Descripción: Este método se encarga de insertar los datos de la planificación del TSU.
Nombre del método: executeModificarPlanificaciónDelTSU ()
Descripción: Este método se encarga de modificar los datos de la planificación del TSU.
Nombre del método: executeVisualizarPlanificaciónDelTSU ()
Descripción: Este método se encarga de visualizar la información de la planificación seleccionada.
Nombre del método: executeEliminarPlanificaciónDelTSU ()
Descripción: Este método es el encargado de eliminar los datos de la planificación seleccionada.

Nombre del método: executeInsertarAsistenciaAITSU ()
Descripción: Este método es el encargado de insertarle la asistencia al TSU.
Nombre del método: executeModificarAsistenciaAITSU ()
Descripción: Este método es el encargado de modificar la asistencia al TSU seleccionado.
Nombre del método: executeGenerarReporteDelTSU ()
Descripción: Este método es el encargado de generar un reporte del tsu, este utiliza el método mostrarAusentes() que este muestra un listado con los ausentes al TSU de la fecha seleccionada.
Nombre del método: mostrarAusentes()
Descripción: Este método es el encargado de buscar y mostrar los estudiantes que se ausentaron al TSU.

Tabla 27 Descripción del CU” Gestionar Trabajo Socialmente Útil”

Descripción del CU” Gestionar evaluación del estudiante”

Nombre de la clase: Gestionar Evaluación_class_Actions
Nombre del método: executeInsertarEvaluaciónDelEstudiante()
Descripción: Este método se encarga de insertar los datos de la evaluación del estudiante en la residencia.
Nombre del método: executeModificarEvaluaciónDelEstudiante()
Descripción: Este método se encarga de modificar los datos de la evaluación en la residencia del estudiante seleccionado.
Nombre del método: executeVisualizarEvaluaciónDelEstudiante()
Descripción: Este método se encarga de visualizar la información de la evaluación del estudiante seleccionado.
Nombre del método: executeEliminarEvaluaciónDelEstudiante ()
Descripción: Este método es el encargado de eliminar los datos de la evaluación del estudiante seleccionado.
Nombre del método: executeGenerarReporteDeLaEvaluaciónDelEstudiante ()
Descripción: Este método es el encargado de generar un reporte de la evaluación del estudiante hasta la fecha.

Nombre del método: mostrarEvaluacion()
Descripción: Este método es el encargado de buscar y mostrar las evaluaciones del estudiante.

Tabla 28 Descripción del CU” Gestionar evaluación del estudiante”

Descripción del CU” Gestionar afectaciones de los apartamentos”

Nombre de la clase: Gestionar Afectaciones_class_Actions
Nombre del método: executeInsertarAfectacionesDeLosApartamentos ()
Descripción: Este método se encarga de insertar los datos de las afectaciones del apartamento.
Nombre del método: executeModificarAfectacionesDeLosApartamentos ()
Descripción: Este método se encarga de modificar los datos de las afectaciones del apartamento seleccionado.
Nombre del método: executeVisualizarAfectacionesDeLosApartamentos ()
Descripción: Este método se encarga de visualizar la información de las afectaciones del apartamento seleccionado.
Nombre del método: executeEliminarAfectacionesDeLosApartamentos ()
Descripción: Este método es el encargado de eliminar las afectaciones del apartamento seleccionado.
Nombre del método: executeGenerarReporteDeLasAfectacionesDeLosApartamentos ()
Descripción: Este método es el encargado de generar un reporte de las afectaciones existentes del tipo especificado.
Nombre del método: mostrarAfectación()
Descripción: Este método es el encargado de buscar y mostrar las afectaciones por apartamento.

Tabla 29 Descripción del CU” Gestionar afectaciones de los apartamentos”

Descripción del CU” Gestionar parada de beca”

Nombre de la clase: Gestionar Parada Beca_class_Actions
Nombre del método: executeInsertarPuntuacionEdificio ()
Descripción: Este método se encarga de insertar la puntuación del edificio en la parada de beca.

Nombre del método: executeModificarPuntuacionEdificio ()
Descripción: Este método se encarga de modificar la puntuación en la parada de beca del apartamento seleccionado.
Nombre del método: executeVisualizarPuntuacionEdificio ()
Descripción: Este método se encarga de visualizar la puntuación alcanzada por el edificio en la parada de beca, además de permitir modificar o eliminar la puntuación escogida.
Nombre del método: executeEliminarPuntuacionEdificio ()
Descripción: Este método se encarga de eliminar la puntuación en la parada de beca del apartamento seleccionado.
Nombre del método: executeInsertarFecha ()
Descripción: Este método es el encargado de insertar fecha de la parada de beca.
Nombre del método: executeModificarFecha ()
Descripción: Este método es el encargado de modificar la fecha de la parada de beca.
Nombre del método: executeVisualizarFecha ()
Descripción: Este método se encarga de visualizar la fecha de la parada de beca, además de permitir modificar o eliminar la fecha escogida.
Nombre del método: executeEliminarFecha ()
Descripción: Este método es el encargado de eliminar la fecha de la parada de beca.
Nombre del método: executeInsertarPuntuacionApartamento ()
Descripción: Este método se encarga de insertar la puntuación del apartamento en la parada de beca.
Nombre del método: executeModificarPuntuacionApartamento ()
Descripción: Este método se encarga de modificar la puntuación en la parada de beca del apartamento seleccionado.
Nombre del método: executeVisualizarPuntuacionApartamento ()
Descripción: Este método se encarga de visualizar la puntuación alcanzada por el apartamento en la parada de beca, además de permitir modificar o eliminar la puntuación escogida.
Nombre del método: executeEliminarPuntuacionApartamento ()
Descripción: Este método se encarga de eliminar la puntuación en la parada de beca

del apartamento seleccionado.
Nombre del método: mostrarParadaBeca ()
Descripción: Este método es el encargado de buscar y mostrar las paradas de becas.

Tabla 30 Descripción del CU" Gestionar parada de beca"

3.2.2-Diagramas de clases del Diseño

Los diagramas de clases del diseño se utilizan para modelar la vista de diseño estática de un sistema, siendo la vista la representación a través de los diagramas de interacción. Para modelar las clases del diseño del sistema empleamos el framework Symfony. Todos los diagramas se modelaron siguiendo los mismos significados, las diferencias que existen entre ellos se debe a las características independientes de cada caso de uso.

A continuación se muestran los diagramas de clases del diseño por cada caso de uso del sistema.

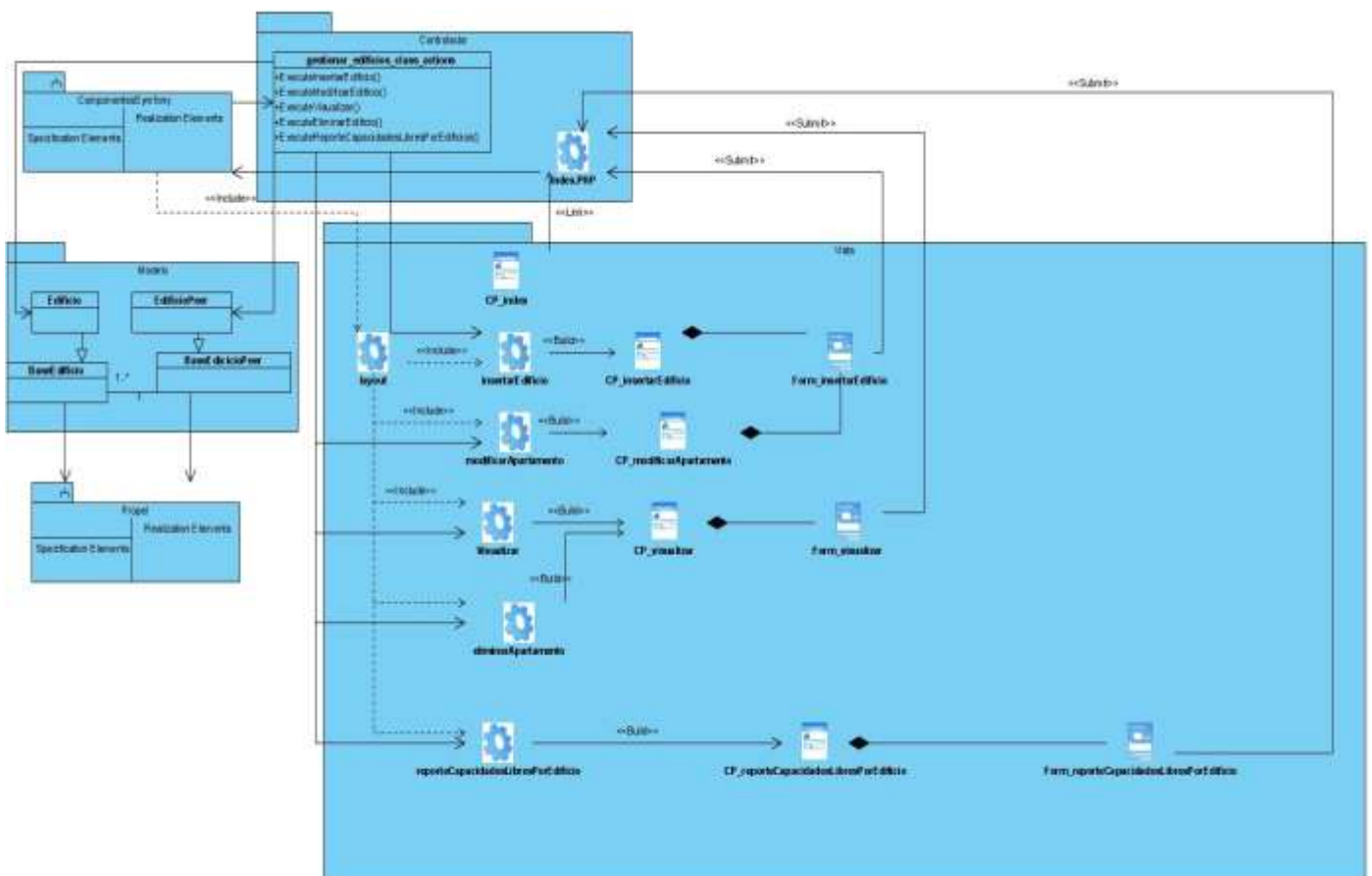


Figura 17 Diagrama de clases del diseño del CU:"Gestionar Edificio"

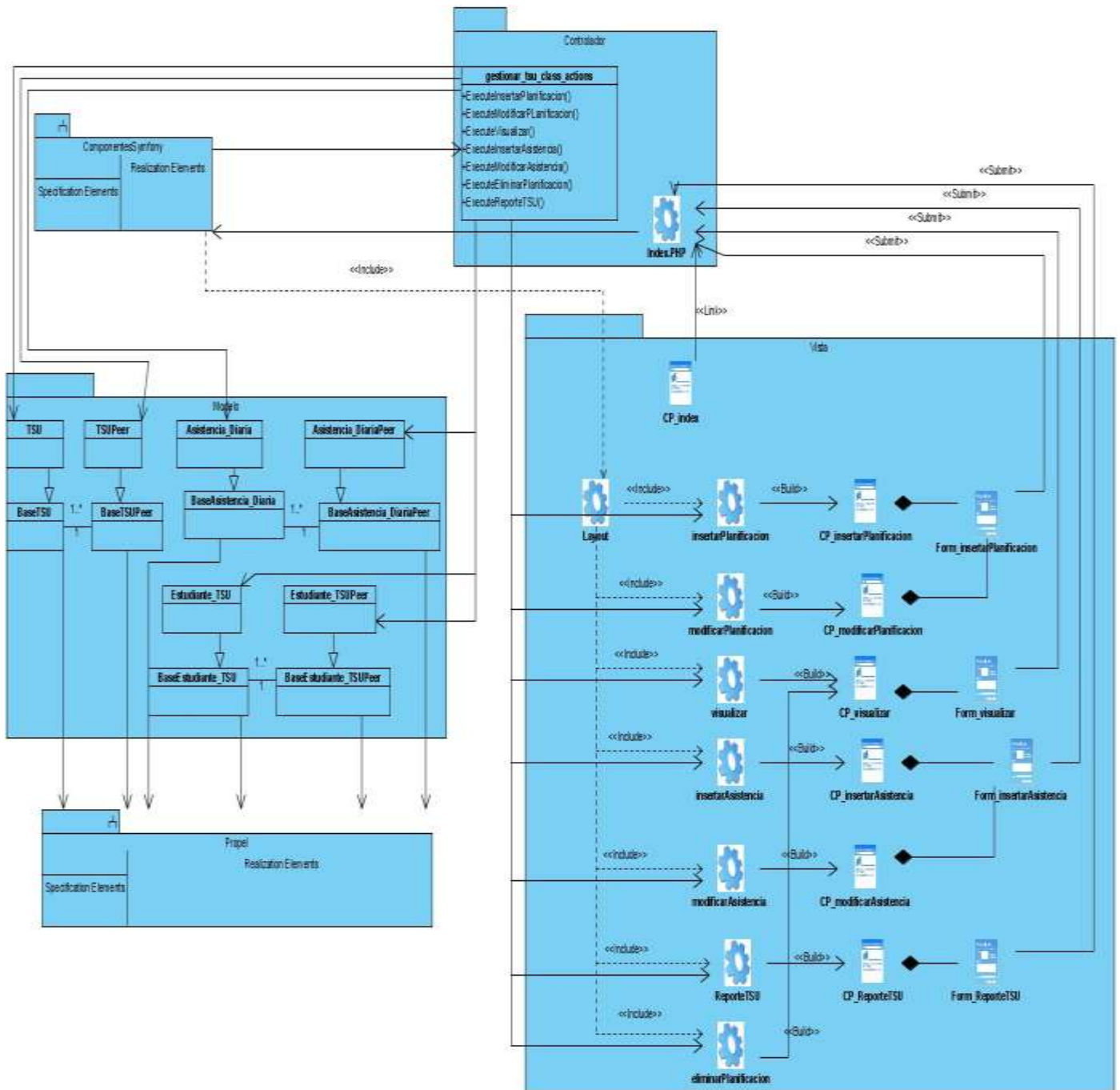


Figura 18 Diagrama de clases del diseño del CU: "Gestionar TSU"

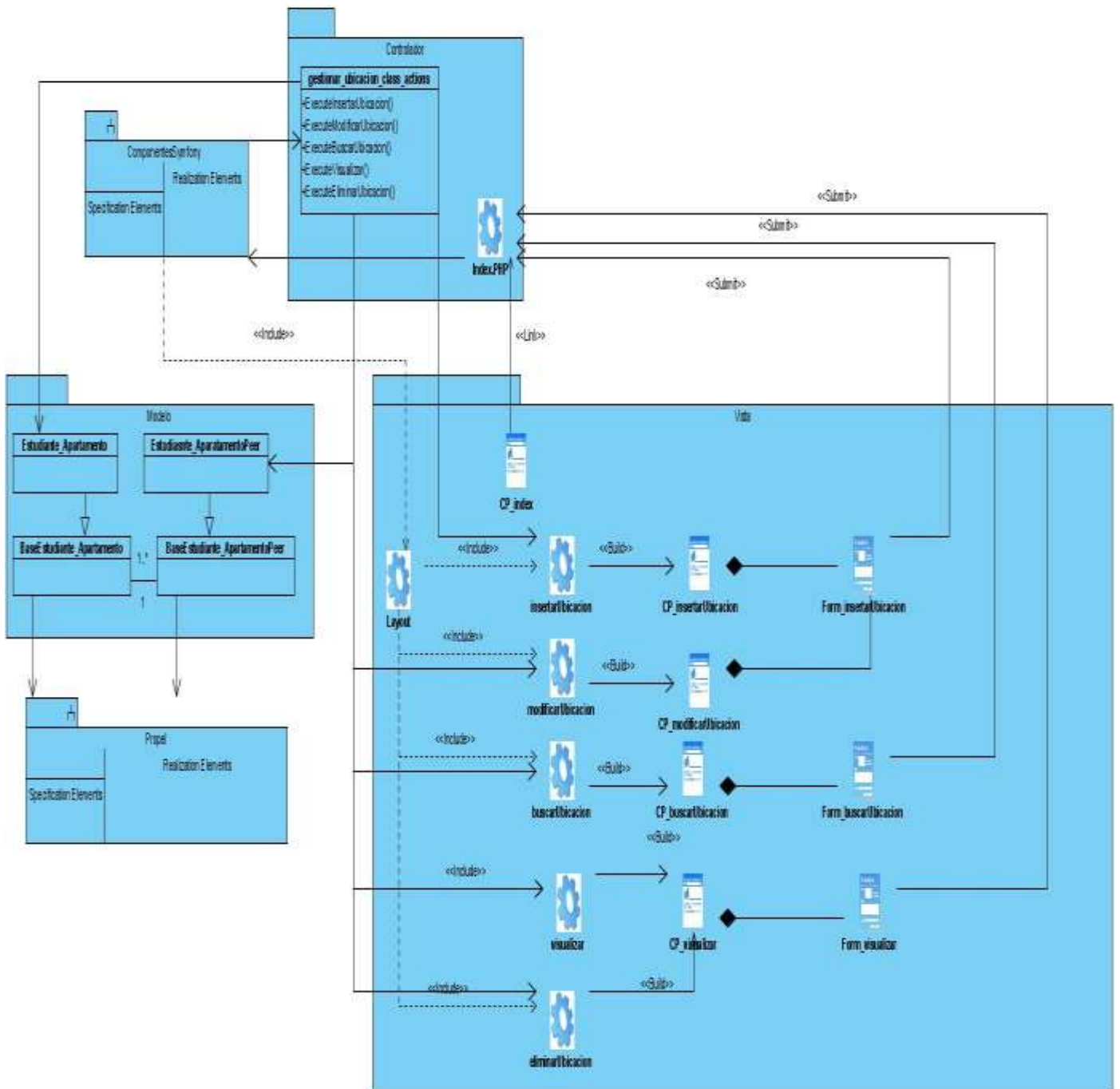


Figura 19 Diagrama de clases del diseño: “Gestionar Ubicación del estudiante en la residencia”

3.3-Diagramas de Interacción

El diagrama de interacción, representa la forma en cómo un Cliente (Actor) u Objetos (Clases) se comunican entre sí en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de

donde se obtienen las responsabilidades claramente. [17] Estos diagramas muestran objetos, así como los mensajes que se pasan entre ellos dentro del caso de uso.

Los diagramas de interacción capturan el comportamiento de los casos de usos. [18] Estos se dividen en dos categorías: Diagramas de Secuencia y Diagramas de Colaboración. Un diagrama de secuencia destaca la ordenación temporal de los mensajes. [19] En nuestro diseño realizamos los diagramas de interacción utilizándose para modelar la vista dinámica del diseño, destacando la ordenación temporal de los mensajes y mostrando las relaciones que se establecen entre los objetos y los diferentes mensajes que se pueden enviar entre ellos.

3.3.1-Diagramas Secuencia CU” Gestionar edificio”

Escenario: Insertar datos del edificio

El propósito de este escenario es mostrar la interacción entre las clases para lograr que el actor inserte los datos del edificio y el sistema los guarde en la base datos.

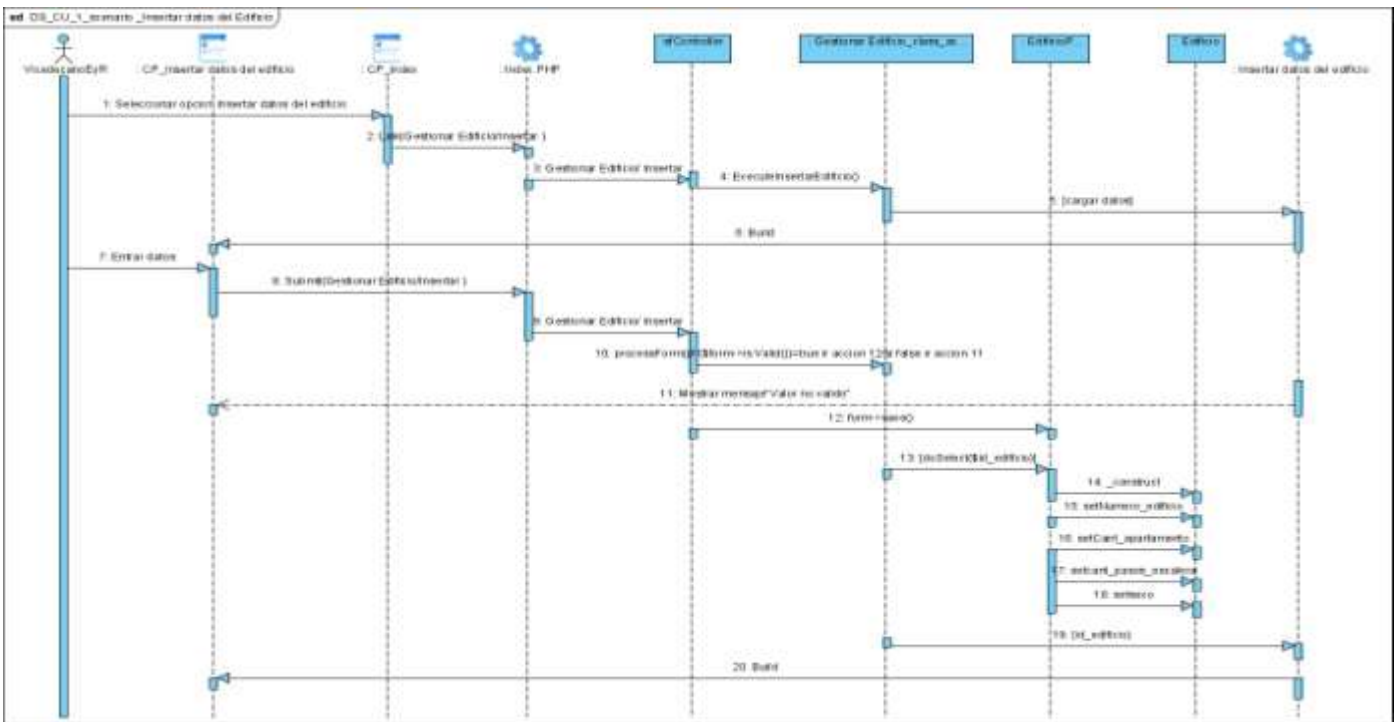


Figura 20 Escenario: Insertar datos del edificio

Escenario: **Generar reporte de los edificios:** En este escenario se muestra la secuencia de pasos para poder generar reporte de las capacidades libres por edificios.

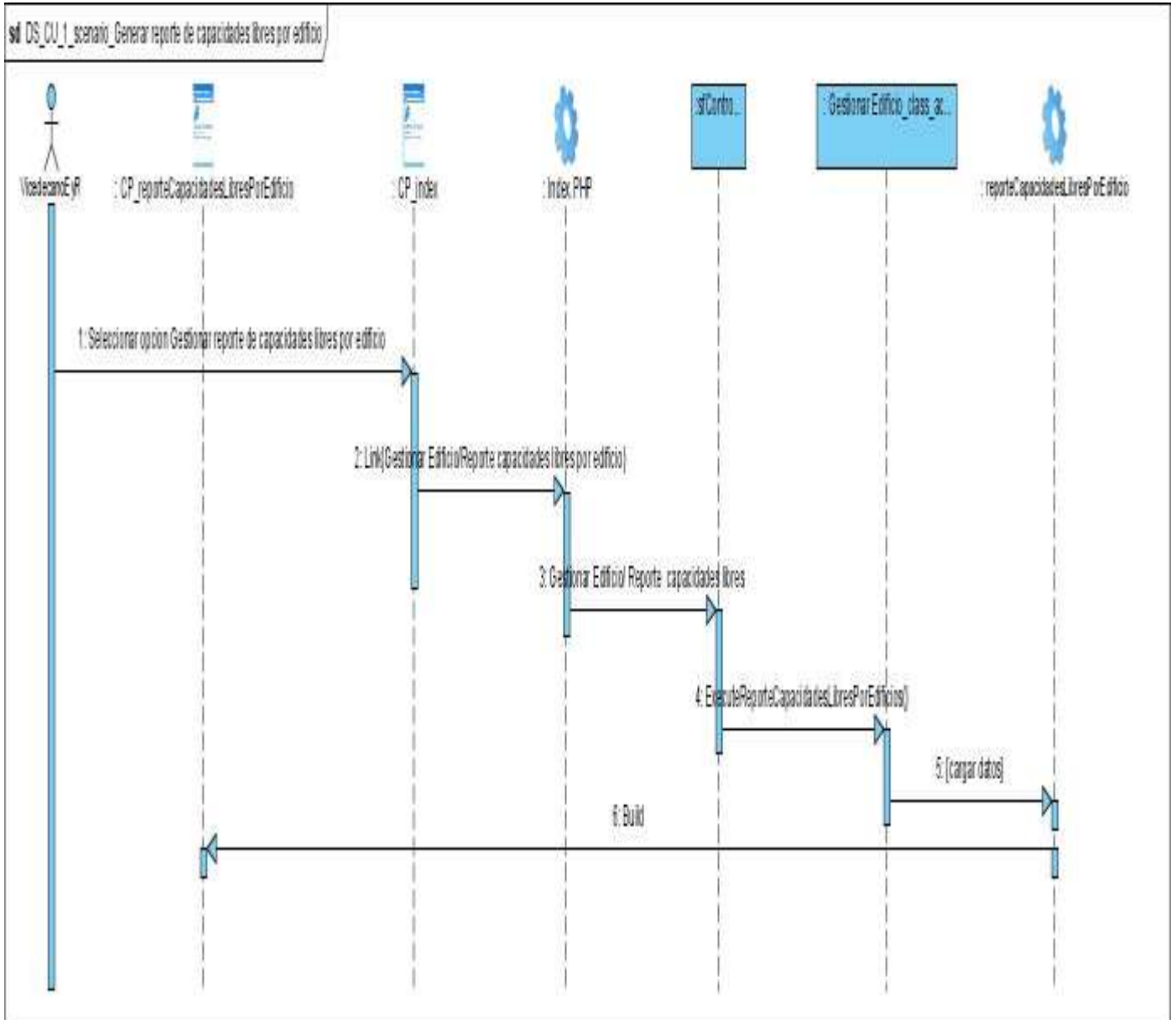


Figura 21 Escenario: Generar reporte de los edificios

3.3.2-Diagramas Secuencia CU "Gestionar Trabajo Socialmente Útil"

Escenario: “Modificar planificación del Trabajo Socialmente Útil”: En la figura 22 se muestran la secuencia a seguir para que el actor modifique la planificación del TSU y se guarde los cambios en la base de datos.

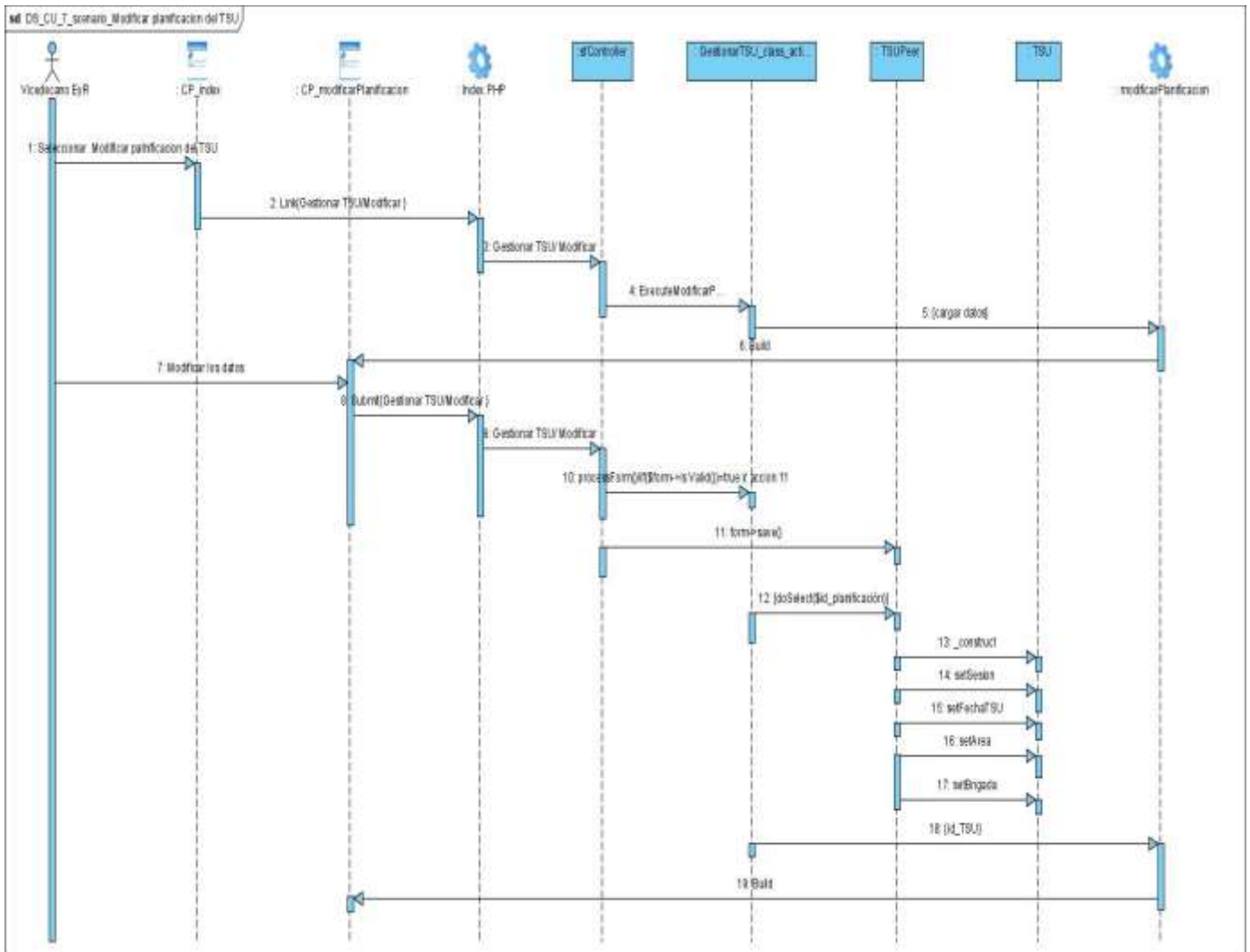


Figura 22 Escenario: Modificar planificación del Trabajo Socialmente Útil

Escenario: Eliminar planificación del Trabajo Socialmente Útil

Este escenario describe los pasos a seguir para poder eliminar la planificación, al escoger la planificación el sistema muestra un mensaje pidiendo afirmación, si el actor acepta se elimina la planificación de la base de datos y si no el sistema muestra el listado de las planificaciones.

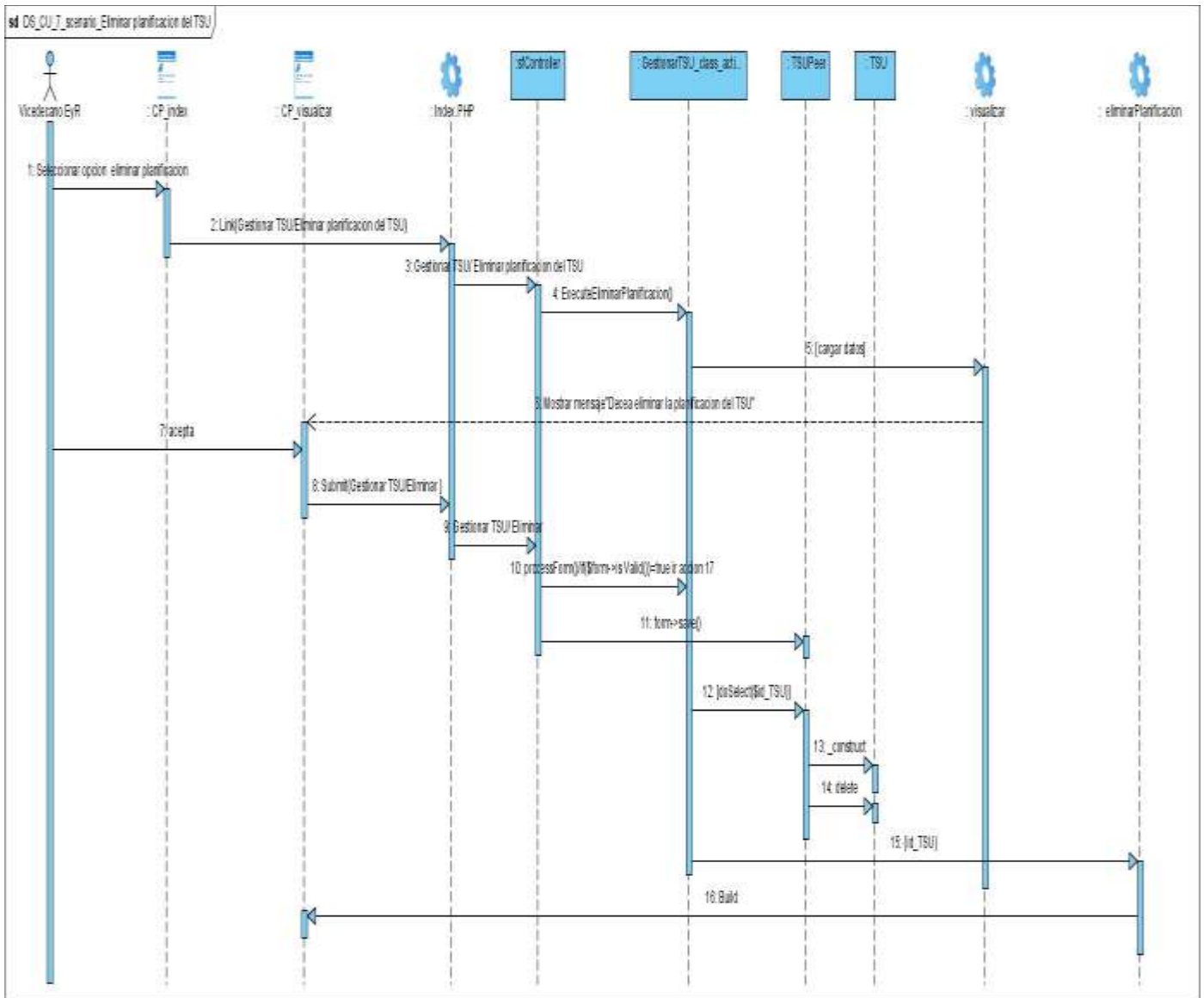


Figura 23 Escenario: Eliminar planificación del Trabajo Socialmente Útil

3.3.3-Diagramas de iteración CU “Gestionar datos de la ubicación del estudiante”

Escenario” Insertar datos de la ubicación del estudiante”: Este escenario muestra los pasos a seguir por el actor y cómo funciona el sistema para que se inserten los datos de la ubicación del estudiante en la residencia estudiantil.

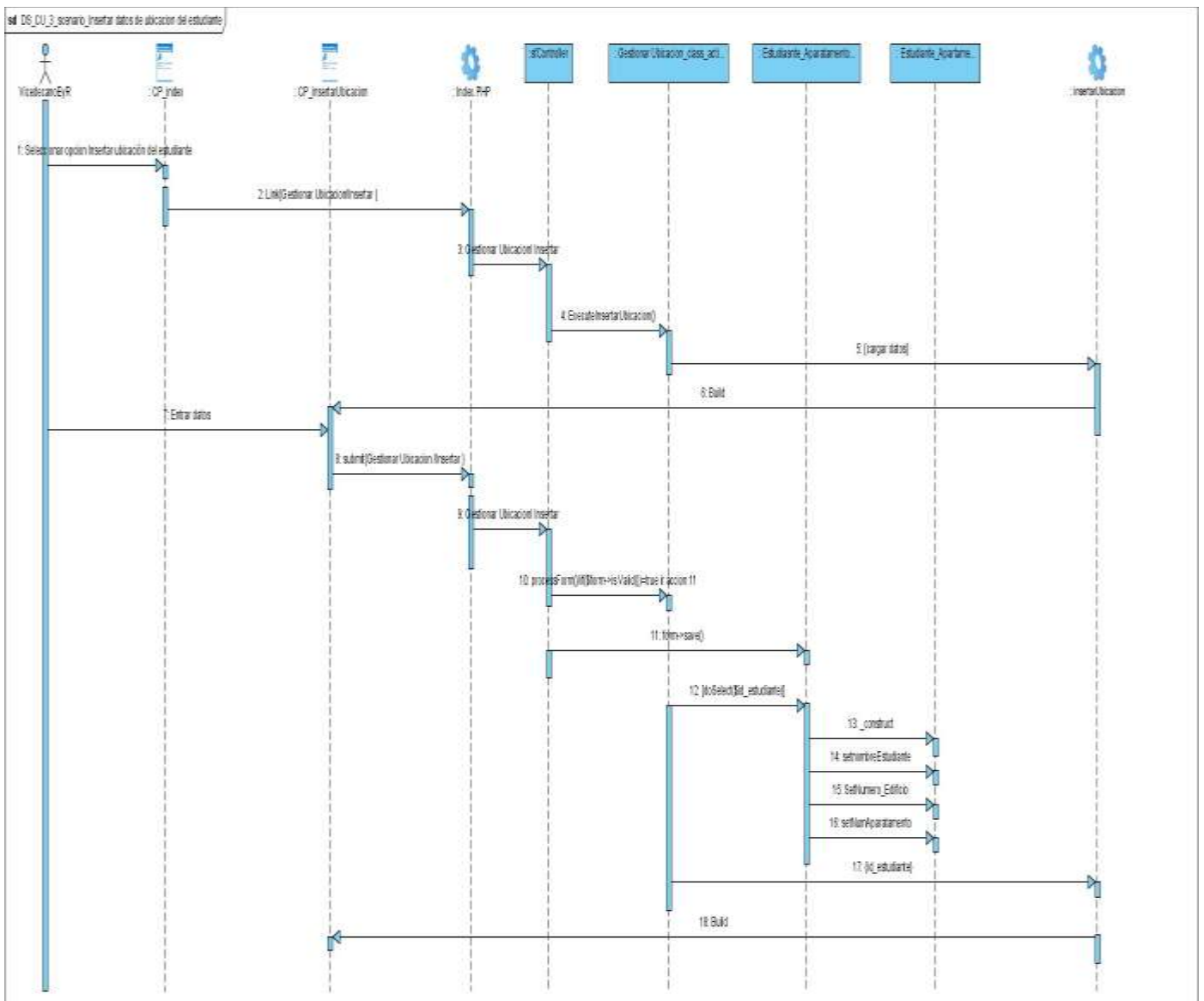


Figura 24 Escenario” Insertar datos de la ubicación del estudiante”

Escenario” Modificar datos de la ubicación del estudiante”: Este escenario muestra los pasos a seguir por el actor y cómo funciona el sistema para que se modifiquen los datos de la ubicación del estudiante en la residencia estudiantil.

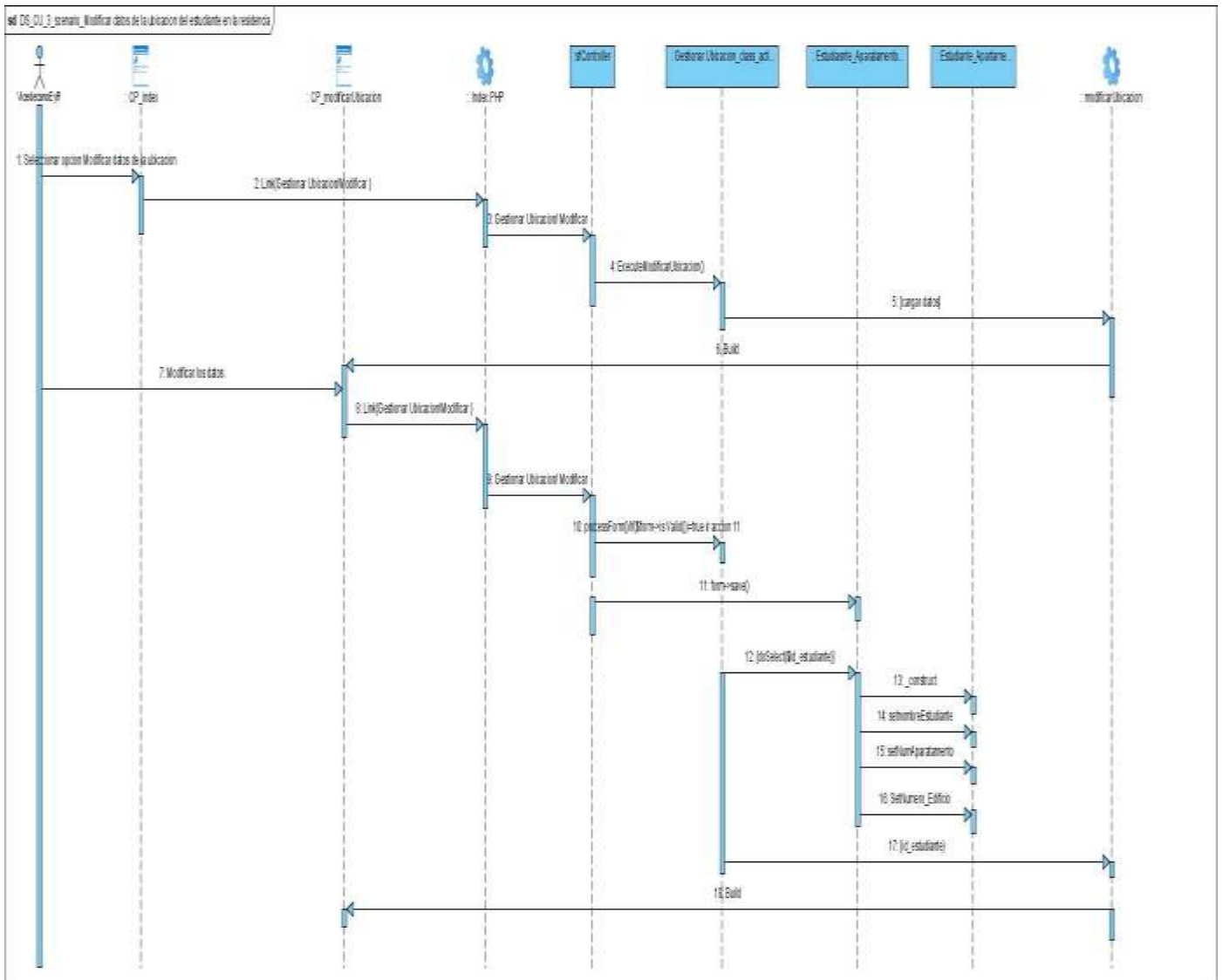


Figura 25 Escenario” Modificar datos de la ubicación del estudiante”

Escenario” Eliminar datos de la ubicación del estudiante”: Este escenario muestra los pasos a seguir por el actor y cómo funciona el sistema para que se eliminen los datos de la ubicación del estudiante en la residencia.

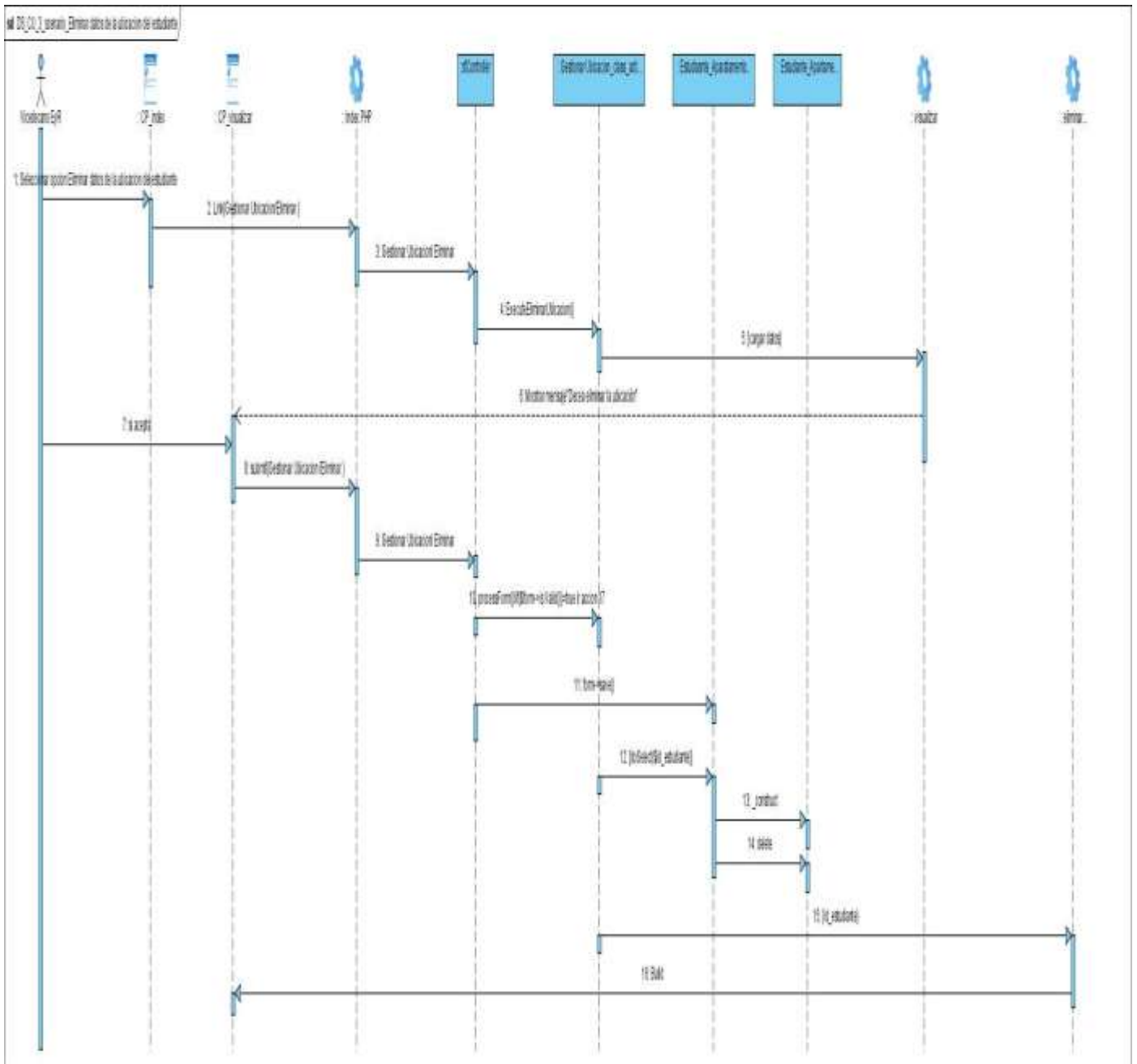


Figura 26 Escenario” Eliminar datos de la ubicación del estudiante”

Escenario” Visualizar datos de la ubicación del estudiante”: Este escenario muestra los pasos a seguir por el actor y cómo funciona el sistema para que se visualicen los datos de la ubicación del estudiante en la residencia estudiantil.

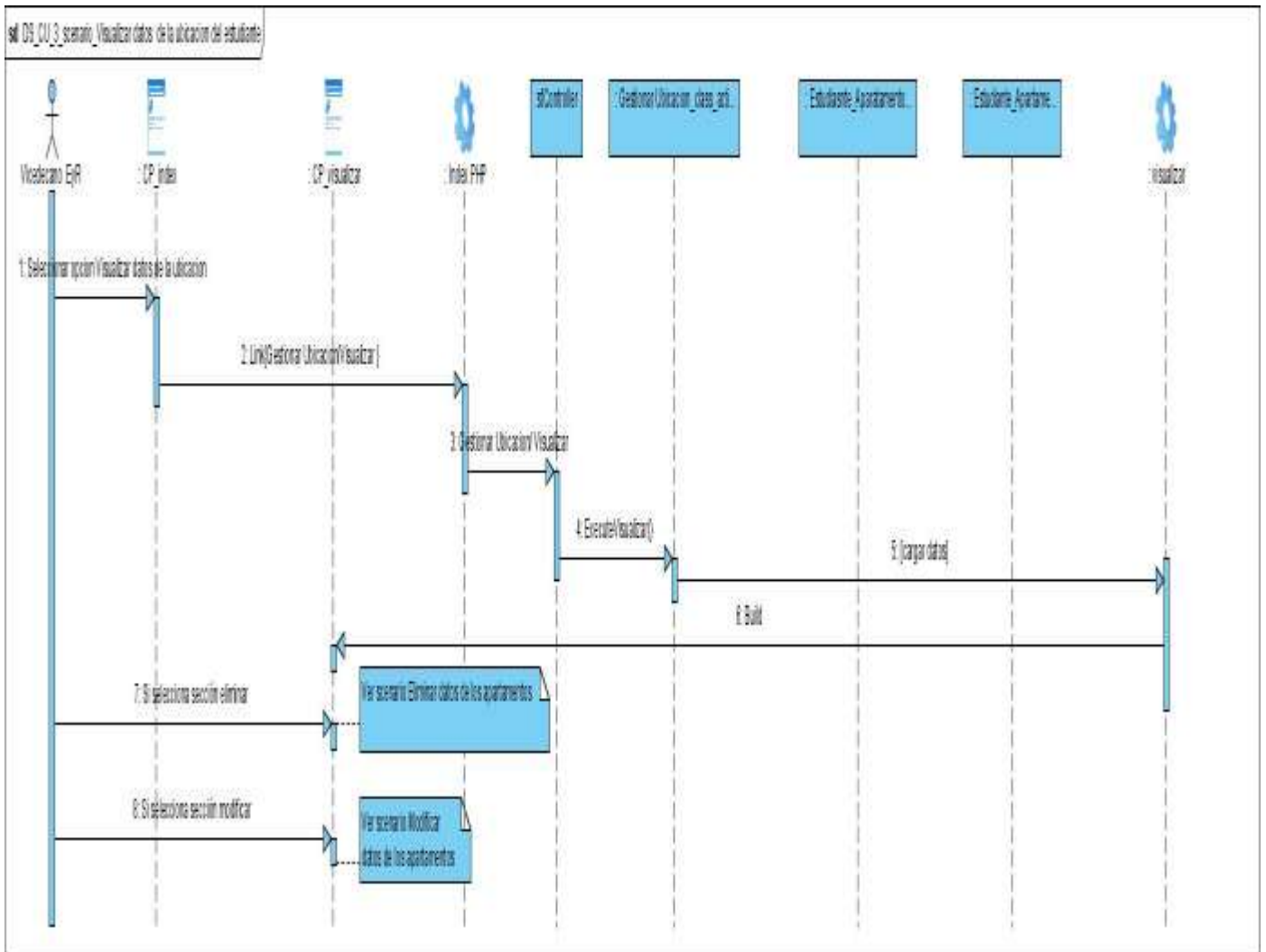


Figura 27 Escenario” Visualizar datos de la ubicación del estudiante”

Escenario” Buscar datos de la ubicación del estudiante”: Este escenario muestra los pasos a seguir por el actor y cómo funciona el sistema para buscar los datos de la ubicación del estudiante en la residencia

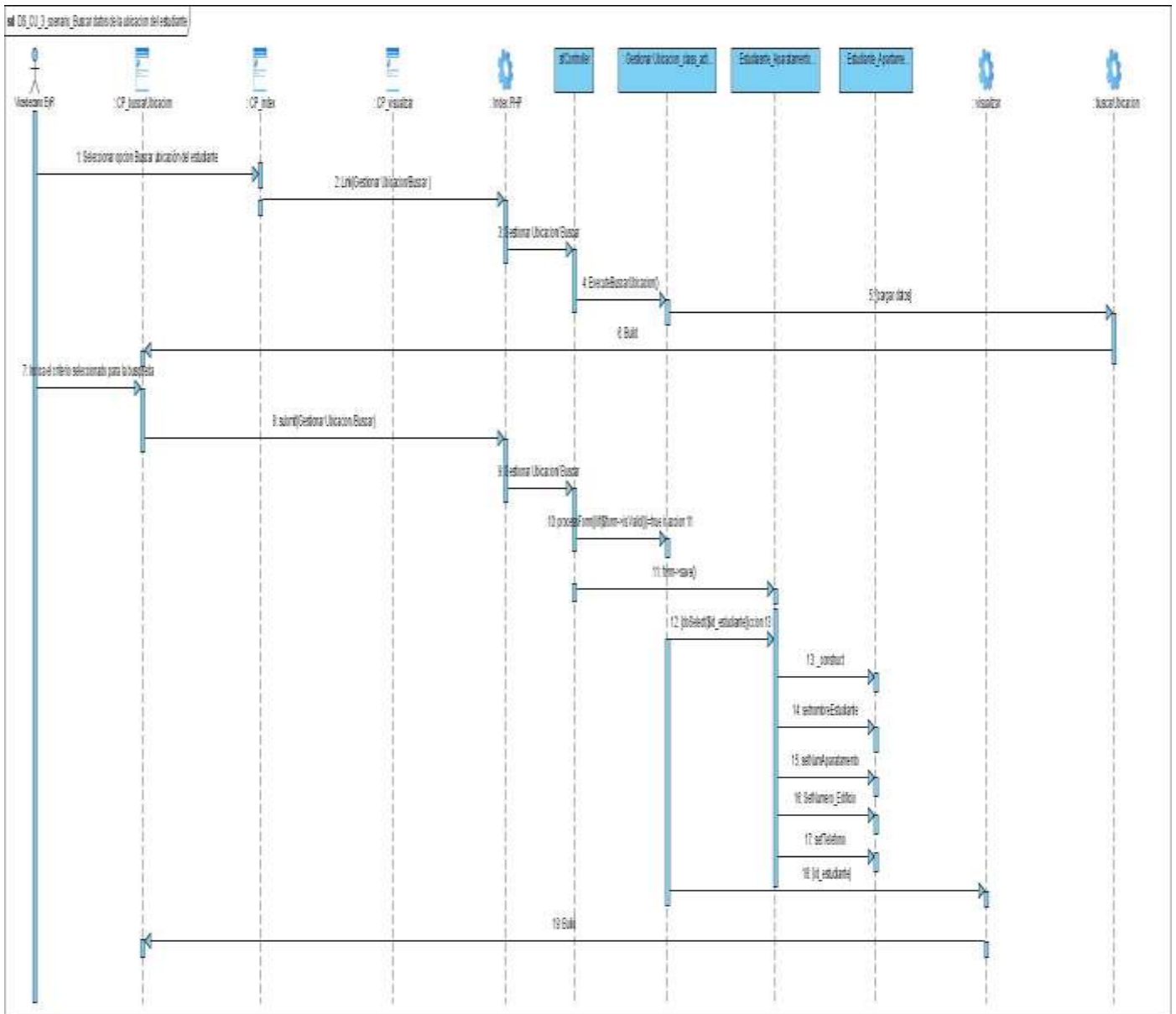
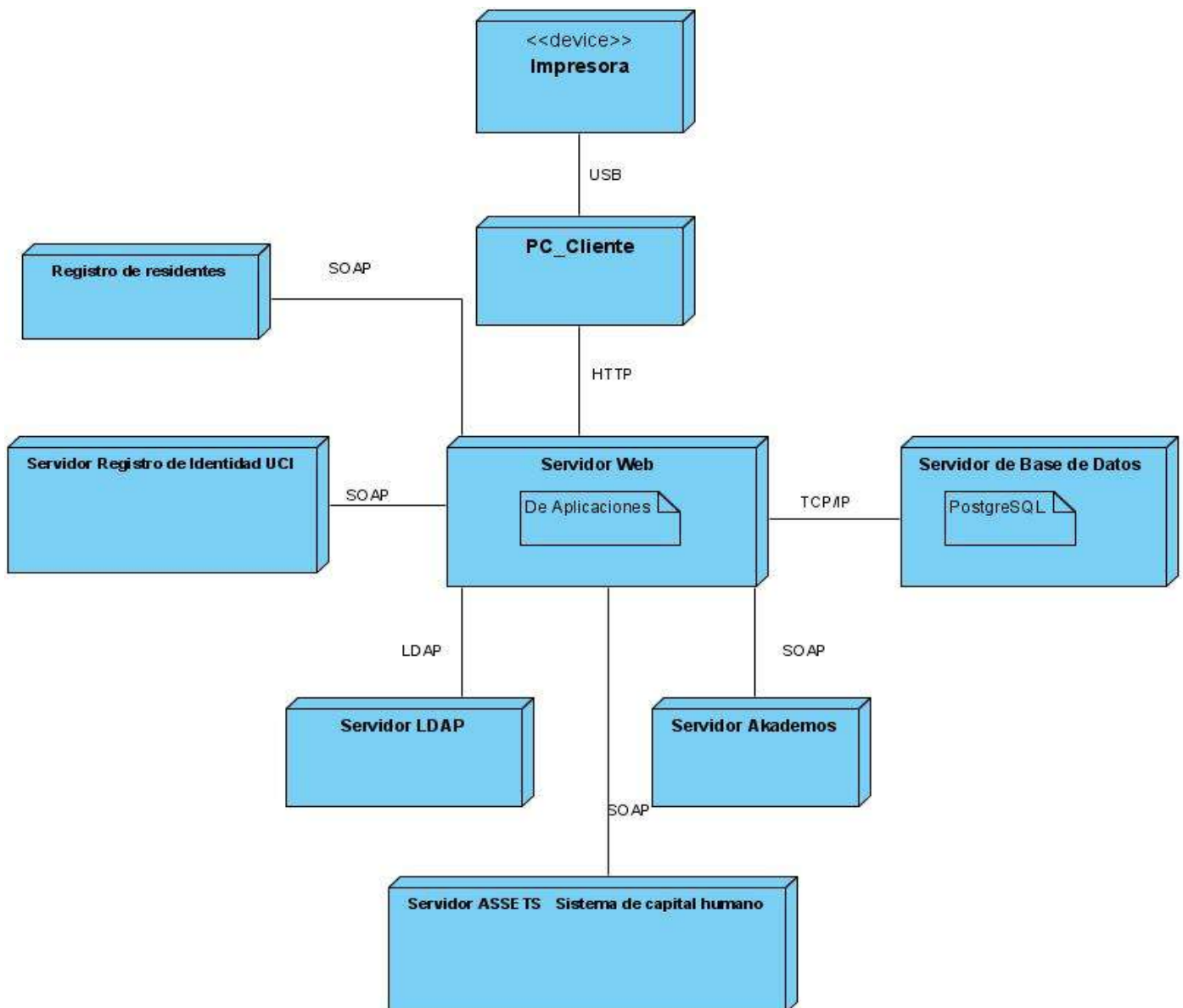


Figura 28 Escenario "Buscar datos de la ubicación del estudiante"

3.4-Diagrama Despliegue

El modelo de despliegue muestra la configuración de los nodos de proceso en el tiempo de ejecución, los enlaces de comunicación entre ellos, y las instancias de componentes y objetos que residen en ellas.



Descripción de los nodos del modelo:

- PC Cliente: Estación de trabajo desde donde interactúa el cliente con la aplicación.
- Servidor Web: Contiene la aplicación Web.
- Servidor Base de Datos PostgreSQL: Servidor donde se almacenan los datos de la aplicación.
- Impresora: Dispositivo que permite editar los reportes generados en la aplicación.

3.5-Diagrama de clases persistentes

Este diagrama muestra las tablas de la base de datos que persistirán a lo largo de todo el desarrollo del software conteniendo información de valor así como las relaciones que se establecen entre ellas.

Todas las clases identificadas en el dominio del análisis no son persistentes. La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Lo contrario son las clases temporales (**transient**) que son manejadas y almacenadas por el sistema en tiempo de ejecución por lo que dejan de existir cuando termina el programa. [20]

Es responsabilidad del diseñador definir cuáles clases son las que deben ser persistentes. En este proceso se recomienda aplicar algunas reglas, ellas son:

- Cuando una clase que está formada por otras clases es persistente, automáticamente las clases componentes también son persistentes. Lo contrario no se cumple necesariamente.
- Cuando una clase hija de una jerarquía es persistente, automáticamente son persistentes sus ancestros en el árbol de jerarquía. Lo contrario no se cumple necesariamente.
- Cuando se define como persistente a una clase que agrupa a objetos de un mismo tipo de clase base (se refiere a las clases listas, colecciones, registros), entonces automáticamente son persistentes todas las clases hijas a partir de la clase base, incluyendo a la clase base.
- Cuando hay herencia múltiple, esta debe ser resuelta antes si el medio de almacenamiento a utilizar no soporta este concepto. La solución más factible es que la clase hija herede de la clase, de la que redefine sus métodos y añada un atributo pasivo del tipo de la otra clase de la que heredaba. Si se redefinía comportamiento de más de una clase padre, hay que escoger de cual se quedaría heredando y añadir un atributo pasivo por cada una de las clases padres de las que no hereda. Los métodos que se redefinen que ya no se reciben por herencia, en su implementación incluirán las relaciones con las clases padres.

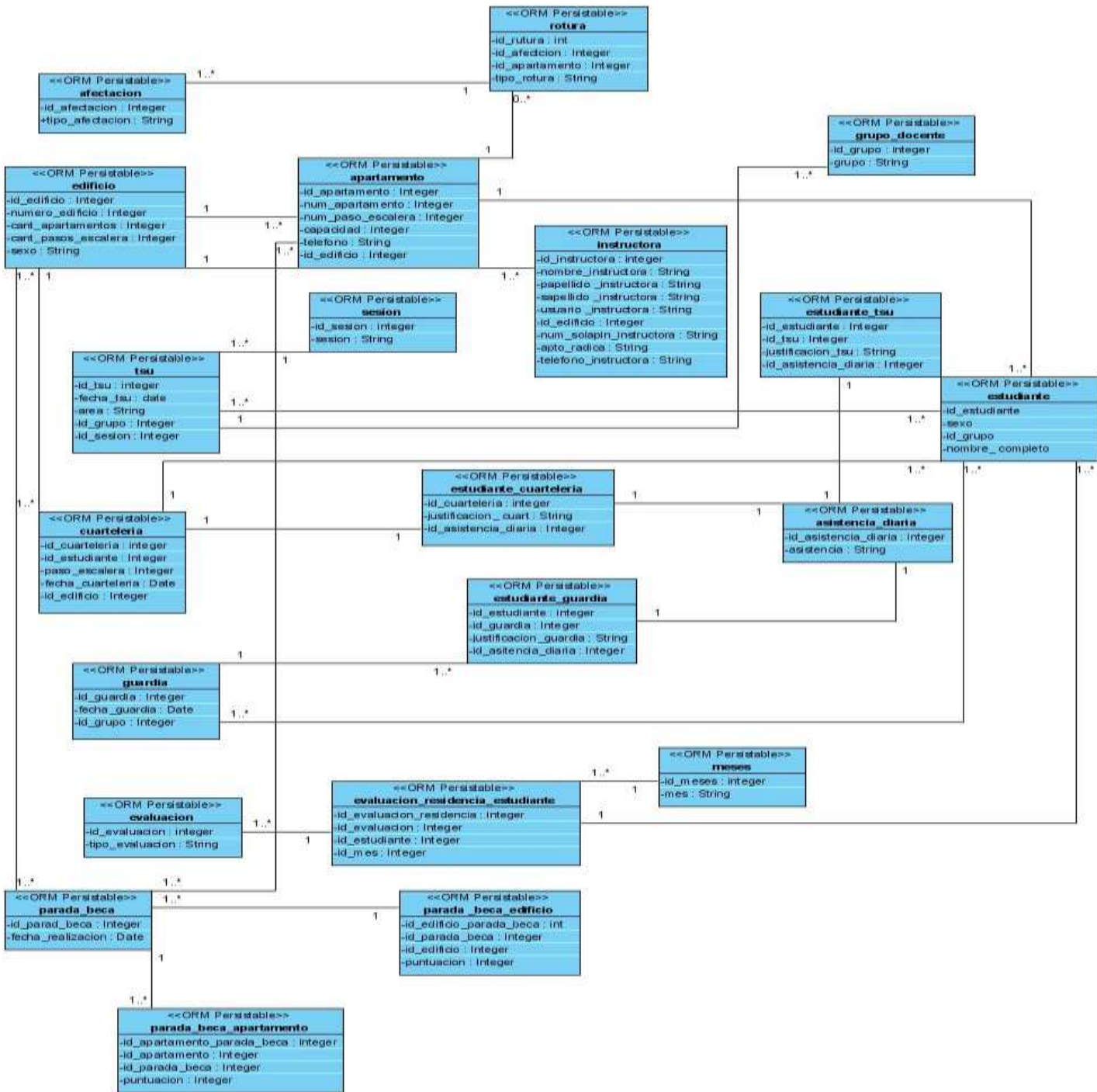


Figura 29 Diagrama de clases persistentes

3.7-Mapa de navegación.

Un mapa de navegación es la representación gráfica de la organización de la información de una estructura web. Expresa todas las relaciones de jerarquía y secuencia y permite elaborar escenarios de comportamiento de los usuarios. [21]

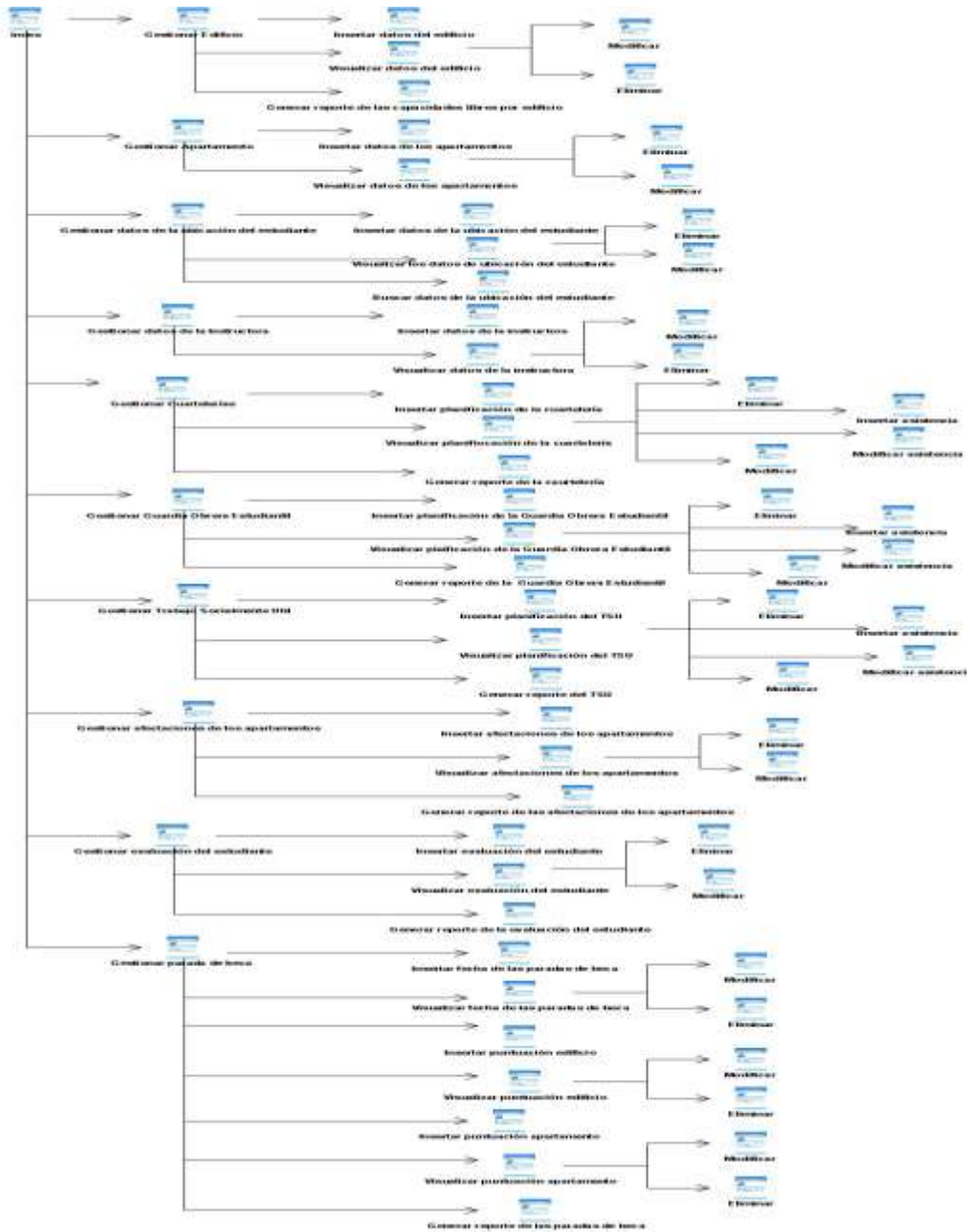


Figura 31 Mapa de navegación

Conclusiones

En este capítulo se definieron las clases del diseño y se muestra una breve explicación. Se realizaron los diagramas de clases del diseño y los diagramas de interacción realizados en el diseño, así como el diagrama de clases persistentes y el modelo de datos. Se construyó el diagrama de despliegue con la distribución de los componentes físicos necesarios para la implantación del sistema.

CAPITULO 4: IMPLEMENTACIÓN DEL SISTEMA.

Introducción

En el presente capítulo se desarrolla el modelo de implementación, analizando los diagramas de implementación que lo conforman. Estos diagramas detallan en componentes las diferentes clases y objetos utilizados en la implementación, teniendo una idea del funcionamiento de las mismas.

El propósito fundamental es lograr una mejor y más detallada vista de la aplicación en relación a la propuesta final del sistema así como obtener una implementación que satisfaga las necesidades de los usuarios.

4.1-Diagramas de componentes

Un diagrama de componentes representa las dependencias entre componentes software, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Un módulo de software se puede representar como componente. Algunos componentes existen en tiempo de compilación, algunos en tiempo de enlace y algunos en tiempo de ejecución, otros en varias de éstas. Los diagramas de componentes de este trabajo fueron realizados por casos de uso, con el objetivo de simplificar la comprensión de los mismos. Aplicando el framework Symfony y el patrón MVC el diagrama se divide en paquetes de la siguiente manera:

Modelo el cual tiene las clases de la base de datos relacionados con el caso de uso correspondiente, este paquete se relaciona con un subsistema Propel encargándose del mapeo de objetos a bases de datos (conocido como ORM, de "*object-relational mapping*") [11] y este a su vez con la base de datos SIGIPE.

Vista contiene todas las páginas clientes.php relacionadas con el layout que va a contener toda la información semejante de los clientes y este a su vez con un componente `sfFrontWebController.class.php` que se relaciona con el controlador.

El componente **sfFrontWebController** o *Controlador Frontal* maneja todas las peticiones web, *siendo* el punto de entrada de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL escrita o pinchada por el usuario. El controlador frontal se encarga de despachar las peticiones, lo que implica algo más que detectar la acción que se ejecuta. De hecho, ejecuta el código común a todas las acciones. En pocas palabras el controlador frontal es el encargado

de determinar qué combinación de módulo-acción se ejecutará. [11]

Controlador contiene la clase action.php que es la que contiene toda la implementación de las acciones que va a realizar el caso de uso, se relaciona con un subsistema sfValidate que es el encargado de las validaciones.

Security.YML es un archivo de configuración que permite restringir el acceso a determinadas acciones del Módulo.

A continuación se muestran algunos de estos diagramas correspondiente al CU: Gestionar edificios y el CU: Gestionar Trabajo Socialmente Útil.

Diagrama Componente CU: Gestionar edificios

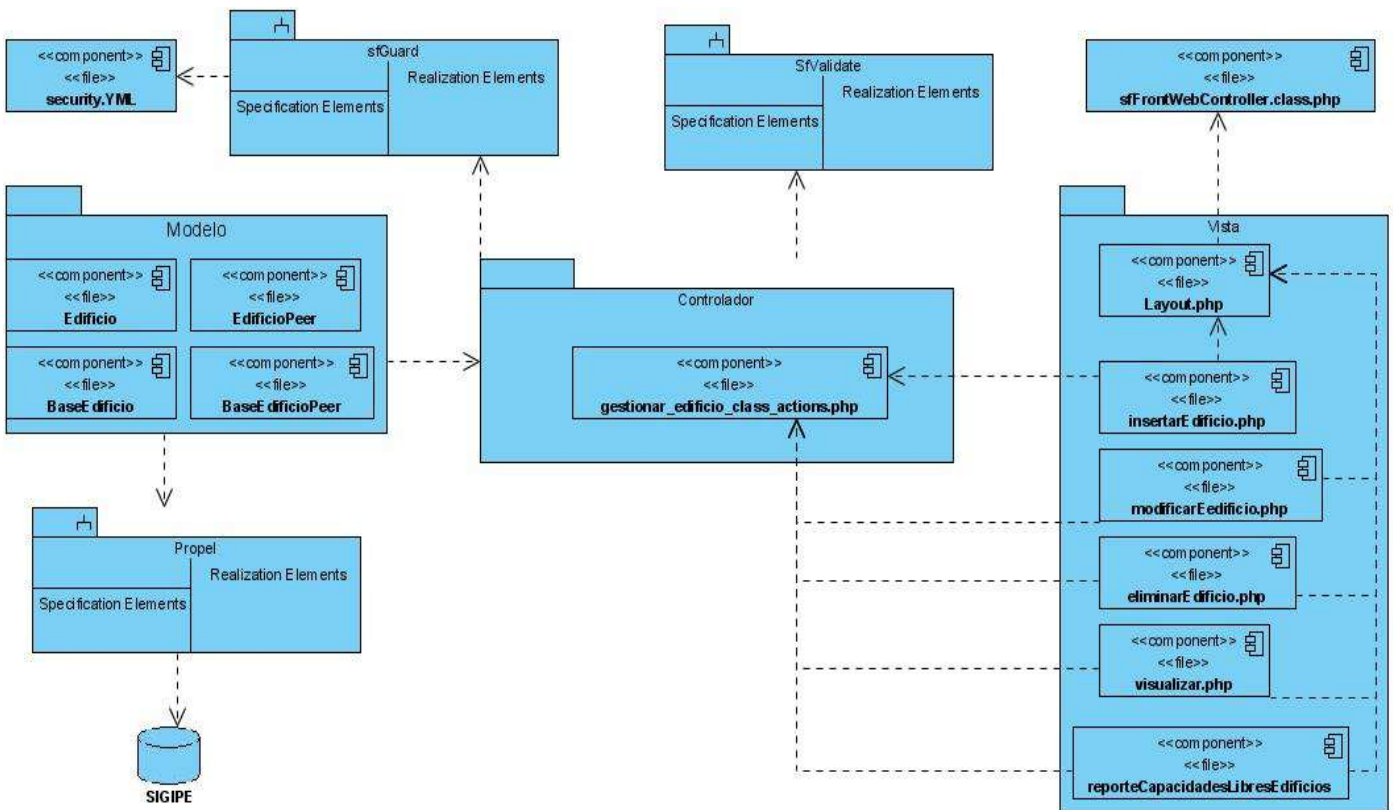


Figura 32 Diagrama de Componente CU: Gestionar edificios.

Diagrama de Componente CU: Gestionar Trabajo Socialmente Útil

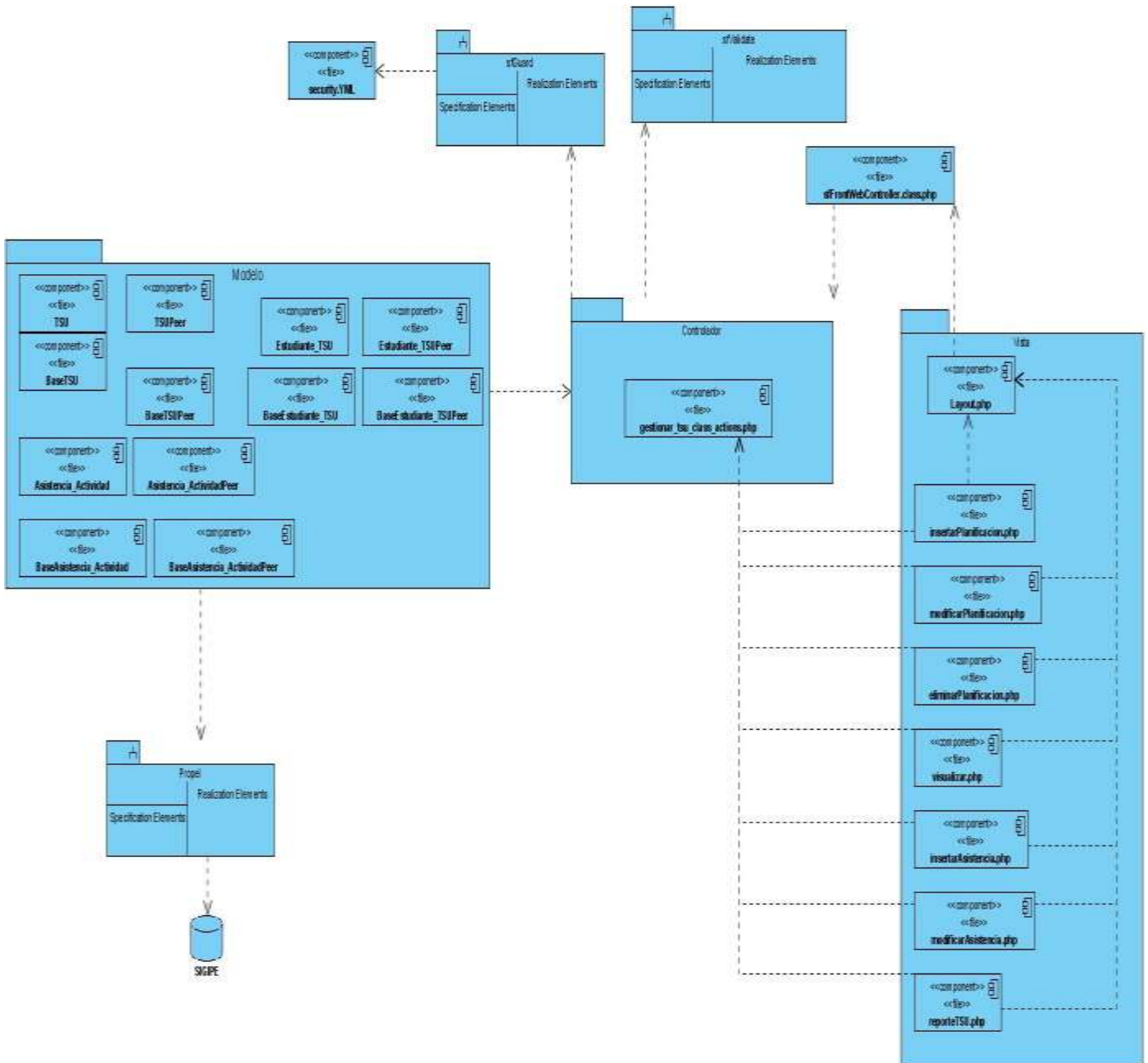


Figura 33 Diagrama de Componente CU: Gestionar Trabajo Socialmente Útil

Diagramas de componentes CU” Gestionar datos de la ubicación del estudiante Ubicación”

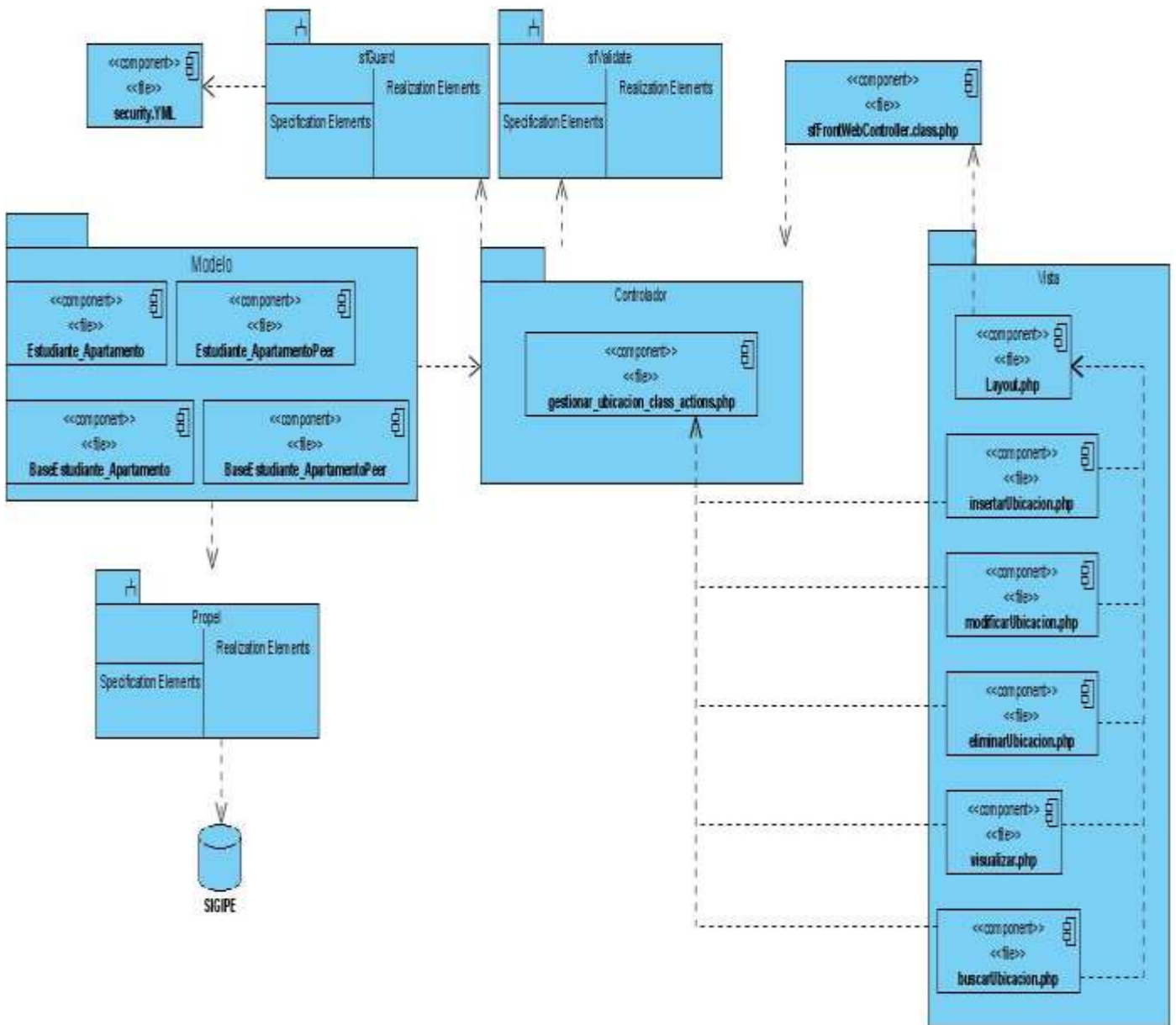


Figura 34 Diagramas de componentes CU:” Gestionar datos de la ubicación del estudiante Ubicación”

4.2-Ejemplos de métodos y su descripción

En este espacio se dará una breve descripción de algunos métodos de gestión que se consideran significativos para el desarrollo de la aplicación.

Caso de uso: Gestionar datos de ubicación del estudiante

Para insertar los datos de ubicación del estudiante fue necesaria la implementación de otros métodos de importancia antes del método principal para esta funcionalidad, a continuación se mostrará un ejemplo de estos métodos:

```
public function executeObtenerEstudiantesGrupo()
{
    $el_grupo=$this->getRequestParameter('grupo');
    $arreglo=array('-1'=>'[Seleccione Estudiantes]');
    $u=new Criterias();
    $u->add(GrupoDocentePeer::GRUPO,$el_grupo);
    $grup=GrupoDocentePeer::doSelectOne($u);
    if($grup)
    $id_grupo=$grup->getIdGrupo();
    else
    $id_grupo=-1;
    $e=new Criterias();
    $e->addJoin(EstudiantePeer::ID_ESTUDIANTE,EstudianteApartamentoPeer::ID_ESTUDIANTE,Criterias::INNER_JOIN);
    $all=EstudiantePeer::doSelect($e);
    $el=new Criterias();
    $el->add(EstudiantePeer::ID_GRUPO,$id_grupo);
    $this->idUbicados=array();
    foreach ($all as $u)
    array_push($this->idUbicados,$u->getIdEstudiante());
    $el->add(EstudiantePeer::ID_ESTUDIANTE,$this->idUbicados,Criterias::NOT_IN);
    $ubicados=EstudiantePeer::doSelect($el);
    if(count($ubicados)>0)
    {
        foreach ($ubicados as $E)
        {
            $arreglo[$E->getIdEstudiante()]=$E->getNombreCompleto();
        }
    }

    $this->listEstudiantes=$arreglo;
}
}
```

Figura 35 Método: executeObtenerEstudianteGrupo ()

El método anterior permite después de haber seleccionado un grupo docente, visualizar los estudiantes pertenecientes al mismo, y luego de ser ubicado el estudiante en un apartamento cuando se vuelva a seleccionar el mismo grupo docente nos muestre el estudiante ya ubicado.

Finalmente para insertar los datos de ubicación del estudiante se implementa el método:

```

public function executeInsertarUbicacionEstudiante ()
{
    if($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $g=GrupoDocentePeer::doSelect(new Criteria());
        $arreglo=array('-1'=>' [Seleccione Grupos] ');
        foreach ($g as $grupo)
        {
            $arreglo[$grupo->getGrupo()]= $grupo->getGrupo();
        }
        $this->listgrupos=$arreglo;
    }
}

else
{
    $msg;
    $apart=$this->getRequestParameter('apartamento');
    $a=new Criteria();
    $a->add(ApartamentoPeer::NUM_APARTAMENTO, $apart);
    $apt=ApartamentoPeer::doSelectOne($a);
    if(count($apt->getEstudianteApartamentos()) < $apt->getCapacidad())
    {
        $apartamento=$this->getRequestParameter('apartamento');
        $apto=new Criteria();
        $apto->add(ApartamentoPeer::NUM_APARTAMENTO, $apartamento);
        $el_apartamento=ApartamentoPeer::doSelectOne($apto);
        if($el_apartamento)
        $id_apartamento=$el_apartamento->getIdApartamento();
        else
        $this->redirect('gestionar_ubicacion/insertarUbicacionEstudiante');
        $estudiante=$this->getRequestParameter('estudiante');
        $estu=new Criteria();
        $estu->add(EstudiantePeer::ID_ESTUDIANTE, $estudiante);
        $el_estudiante=EstudiantePeer::doSelectOne($estu);
        if($el_estudiante)
        $id_estudiante=$el_estudiante->getIdEstudiante();
        $estApt = new EstudianteApartamento();
        $estApt->setIdApartamento($id_apartamento);
        $estApt->setIdEstudiante($id_estudiante);
        $this->redirect('gestionar_ubicacion/insertarUbicacionEstudiante');
    }
    else $msg = 'Límite de estudiantes para ese apartamento alcanzado!';
    $this->getUser()->setFlash('msg', $msg);
    $this->redirect('gestionar_ubicacion/index');
}
}

```

Figura 36 Método: executeInsertarUbicaciónEstudiante ()

El método anterior primeramente permite listar los grupos docentes de la facultad mediante la función doSelect(new Criteria()), luego realiza la captura del número del apartamento y el identificador del estudiante seleccionados mediante la función getRequestParameter('Parametro'), y lo guarda en la base de datos, antes verifica que el apartamento seleccionado todavía tenga capacidades libres, sino lanza un mensaje.

Caso de uso: Gestionar trabajo socialmente útil

```
public function executeMostrarAusentes(sfWebRequest $request)
{
    $this->getUser()->Auditoria();
    $this->fecha=$this->getRequestParameter('tsu');
    $All=0;
    $this->All=$All;
    if (!empty($this->fecha))
    {
        $c=new Criteria();
        $c->add(TsuPeer::FECHA_TSU,$this->fecha);
        $tsu=TsuPeer::doSelect($c);
        $this->tsu=$tsu;
        if (!empty($tsu))
        {
            $c->add(TsuPeer::FECHA_TSU,$this->fecha);
            $c->addJoin(EstudianteTsuPeer::ID_TSU,TsuPeer::ID_TSU);
            $c->addOr(EstudianteTsuPeer::ID_ASISTENCIA_DIARIA,2);
            $c->addOr(EstudianteTsuPeer::ID_ASISTENCIA_DIARIA,3);
            $c->addJoin(EstudiantePeer::ID_ESTUDIANTE,EstudianteTsuPeer::ID_ESTUDIANTE);
            $this->All=EstudiantePeer::doCount($c);
            $this->est=EstudiantePeer::doSelect($c);
        }
    }
}
```

Figura 37 Método: executeMostrarAusentes()

El método anterior permite generar un reporte de los estudiantes que no asistieron al trabajo socialmente útil, dada una fecha que se captura con la función `getRequestParameter('Parámetro')` verifica que exista en la tabla `Tsu` para saber si hay una planificación con esa fecha y busca en la tabla `EstudianteTsu` los estudiantes que no asistieron esa fecha al `Tsu` y los lista.

4.3-Validación de la aplicación

Una forma de verificar la consistencia de los datos de la aplicación es introduciendo datos erróneos o dejando campos en blanco que son de carácter obligatorio, esto podría considerarse como un paso de avance y ayuda para el flujo de trabajo de prueba.

Insertar planificación de la cuartelería	Insertar datos del apartamento
<p>Introduzca el edificio</p> <p>Edificio: << Seleccione >> *</p>	<p>Valor incorrecto</p> <p>Apartamento: prueba *</p>
<p>Introduzca el edificio</p> <p>Estudiante: << Seleccione >> *</p>	<p>Teléfono: 3005 *</p>
<p>Introduzca la fecha</p> <p>Fecha: [] ... *</p>	<p>Edificio: 23 *</p>
<p>Introduzca el paso de escalera</p> <p>Paso de escalera: << Seleccione >> *</p>	<p>Paso de escalera: 1 *</p>
<p>Capacidad: 6 *</p>	
<p>Aceptar Cancelar</p>	<p>Aceptar Cancelar</p>

Otro de los casos es en el caso de uso buscar ubicación, no permite que dejen todos los criterios vacíos.

Microsoft Internet Explorer

Introduzca al menos un criterio.

Aceptar

Buscar Cancelar

Este caso es el de insertar datos de la instructora en él se muestra que el sistema no permite que inserten números donde debe haber letras y en el caso del teléfono muestra la mejor manera de introducir este campo.

The screenshot shows a web form titled "Insertar datos de la instructora". The form contains several input fields with the following values: "Nombre: 345345", "1er. apellido: peres", "2do. apellido: peres", "Usuario: yperes", "Solapin: 1235", "Edificio. atiende: 87" (with a dropdown arrow), "Apto. radica: 1031", and "Teléfono: 123". There are two validation messages in red text: "Inserte solo letras por favor" near the name field and "El teléfono requiere 7 dígitos" near the phone field. At the bottom, there are two buttons: "Aceptar" and "Cancelar".

Conclusiones

En el capítulo se realizó la vista de implementación, dentro de ella se construyeron los diagramas de componentes de cada uno de los casos de usos, guiados por las definiciones realizadas durante el diseño, el diagrama de clases persistentes y a partir de este se elaboró el modelo de datos. Por último se presentaron ejemplos de código de algunas de las principales funcionalidades de la aplicación.

CONCLUSIONES GENERALES

En este trabajo de diploma se realizó un amplio estudio de la residencia estudiantil de la facultad 6, se profundizó en sus características principales y su funcionamiento, logrando entender a plenitud las actividades que se desarrollan en la misma, identificándose las funcionalidades que cumpliría el sistema y seleccionándose una mejor vía de planificación y distribución. Se realizó la selección de las herramientas, lenguajes y tecnologías necesarias para la confección de un sistema, que cumpliera con los requerimientos definidos y para lograr un mejor control y seguimiento de las actividades realizadas en la residencia estudiantil.

Se realizó el análisis del negocio, efectuándose en este la descripción del negocio, los diagramas de casos de usos del negocio y sistema, diagramas de actividades, y modelo de objetos. De igual manera en el diseño se realizaron los diagramas de clases del diseño, de componentes y de iteraciones, en este caso de secuencia, se desarrolló el mapa de navegación para una mejor comprensión de la aplicación.

El producto final obtenido facilitará el control de las actividades que se realizan en la residencia estudiantil, permitiendo que el manejo de la información se realice con una mejor calidad.

RECOMENDACIONES

Teniendo en cuenta el estado que alcanzó el sistema implementado se recomienda:

- Realizar las pruebas exploratorias al sistema implementado.
- Generalizar la aplicación a otras facultades.
- Realizar una nueva versión de la aplicación, incorporándole la información de los profesores de la facultad.

REFERENCIAS BIBLIOGRÁFICAS

1. Ministerio de la informática y las comunicaciones de cuba, [en línea]. [Consultado: 2 de febrero de 2009]. Disponible en: <http://www.mic.gov.cu/hinfosoc.aspx>
2. Anónimo, sitio de Preguntas más frecuentes.[en línea].[Consultado:2 de febrero de 2009]. Disponible en :<http://www.tech-faq.com/lang/es/data-management.shtml>
3. Bartle, Phil.Titulo: Información para la gestión y gestión de la información.[en línea].[Consultado 5 de febrero de 2009]. Disponible en: <http://www.scn.org/mpfc/modules/mon-miss.htm>
4. Milán, Tellería, Oliver, López Vázquez, Manuel, Salazar Alea, Caridad. *Sistema automatizado informativo para la comunidad de residencia estudiantil de la Universidad de Pinar del Río "Hermanos Saiz Montes de Oca"*. Pinar del Río, 2008
5. Dávila,Pérez, Yuniesky, Martínez, Padrón, Yadrian. *Sistema de Gestión de Información de la facultad 8.Módulo para la gestión de la residencia estudiantil versión 2.Universidad de la Ciencias Informáticas, 2008.*
6. Addison Wesley Ed. James Rumbaugh, Ivar Jacobson y Grady Booch. *El Lenguaje Unificado de Modelado. Manual de Referencia. 2000.*
7. RUP_Help_ES. [Consultado: 2 de marzo de 2009].
8. Anónimo, Javascript Patzcuaro MC.[En línea].[Consultado 20 de marzo de 2009]. Disponible en: http://www.articulosinformativos.com.mx/Javascript_Patzcuaro_MC-r1106253-Patzcuaro_MC.html
9. Alvarez, Angel, Miguel. *Una introducción meramente conceptual al potente lenguaje de script del lado del cliente.*[En línea].[Consultado 5 de abril de 2009]. Disponible en: <http://www.desarrolloweb.com/articulos/25.php>
10. Anónimo, Soporte Técnico,[En línea].[Consultado 5 de abril de 2009]. Disponible en : http://www.iprc.org.co/aspectos_tecnicos.shtml
11. Anónimo, Qué es el PHP, [En línea].[Publicado 23 de mayo de 2001].[Consultado 8 de abril de 2009]. Disponible en : <http://www.maestrosdelweb.com/editorial/phpintro/>
12. Eguíluz, Javier. *Guía definitiva Versión 6.0.*[En línea].[Consultado 8 de febrero de 2009]. Disponible en: http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html
13. Sanz Bermejo, Laura y Monreal Gómez, Enrique. *Eclipse como IDE.* [En línea] [Consultado: 12 de enero de 2009]. Disponible en : <http://kybele.escet.urjc.es/documentos/HC/Exposiciones/EclipseIDE.pdf>

14. Anónimo, Introducción a Patrones.[en línea].[Consultado 10 de febrero de 2009].Disponible en:
http://agamenon.uniandes.edu.co/~pfiguero/soo/Magister_Patrones/intropatrones
15. Garcia, Joaquin. *Diseño de software Orientado a Objeto, Patrones de diseño*. [Publicado: 27 de mayo de 2005]. [En línea]. [Consultado 10 de febrero de 2009]. Disponible en:
<http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>
16. Potencier, Fabien & Zaninotto, François. *Symfony 1.2, la guía definitiva*. [En línea]. 30 de diciembre 2008 [Consultado 1 de marzo 2009]. Disponible en: http://www.librosweb.es/symfony_1_2
17. Anónimo, Diagramas de interacción. [en línea]. [Consultado 10 de febrero de 2009]. Disponible en:
<http://www.dcc.uchile.cl/~psalinas/uml/interaccion.html>
18. Anónimo, Diagramas de interacción. [en línea]. Curso 2005-2006 [Consultado 10 de febrero de 2009]. Disponible en:
<http://proyectos.uemc.edu/eps/profesores/gbartolome/AIS/Teor%C3%ADa/8%20Modelado%20de%20la%20din%C3%A1mica/Tema%206%20-%20Diagrama%20de%20interaccion.pdf>
19. Anónimo, Diagramas de interacción. [en línea]. [Consultado 10 de febrero de 2009]. Disponible en:
<http://tvdi.det.uvigo.es/~avilas/UML/node41.html>
20. Anónimo, *Conferencia 3 Flujo de trabajo de análisis y diseño*, curso 2006-2007, [Consultado 20 de abril de 2009].
21. Anónimo, *Arquitectura de información*. [En línea]. [Consultado 21 de abril de 2009]. Disponible en:
<http://arquitecturadeinformacion.cl/como/mapa.html>

BIBLIOGRAFÍA


1. **Anónimo.** [En línea] [Citado el: 20 de 04 de 2009.]
http://httpd.apache.org/docs/2.0/new_features_2_0.html.
2. **Anónimo.** [En línea] [Citado el: 5 de 04 de 2009.] <http://www.dbrunas.com.ar/postgres/migrapg.pdf>.
3. **Baeza, Baeza, Pablo, Nicolás.** *Visual Parading DB Visual Architect SQL*.
4. Conferencia 3Flujo de trabajo de análisis y diseño, curso 2006-2007. Ciudad de la Habana : s.n.
5. **Pecos, Daniel.** PostgreSQL vs. MySQL. [En línea] [Citado el: 23 de 04 de 2009.]
http://www.netpecos.org/docs/mysql_postgres/index.html.
6. **García, Luis.** Sistema de control de versiones: SUBVERSION . [En línea] 17 de 1 de 2008. [Citado el: 23 de 04 de 2009.]
<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=548> .
7. **Pressman, Roger S.** [En línea] <http://bibliodoc.uci.cu/pdf/reg02689.pdf>.
8. **Oktaba, Hanna.** Introducción a Patrones. [En línea] [Citado el: 25 de 04 de 2009.]
<http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>..
9. **Sanz Bermejo, Laura y Monreal Gómez, Enrique.** Eclipse como IDE. [En línea] [Citado el: 12 de 02 de 2009.] <http://kybele.escet.urjc.es/documentos/HC/Exposiciones/EclipseIDE.pdf>..
10. **UCI, Colectivo de desarrollo de la.** *Patrones de diseño*. Ciudad de la Habana : s.n., 2007-2008.
11. **Soriano, Ing. Fernando.** Diseño Orientado a Objetos Patrones GoF. 2008.
12. **Zaninotto, Fabien Potencier y Francois.** Symfony. [En línea] 20 de 12 de 2007. [Citado el: 10 de 04 de 2008.] <http://www.librosweb.es/symfony/>..

Anexos

Anexo.1-Descripciones extendidas de CUS











Tabla 31 Descripción del CUS” Gestionar edificios”

Caso de Uso:	Gestionar edificios	
Actores:	Vicedecano de residencia y extensión	
Resumen:	El caso de uso comienza cuando El Vicedecano de residencia y extensión decide Insertar datos del edificio, Modificar datos del edificio, Visualizar datos del edificio, Eliminar datos del edificio o Generar reporte de las capacidades libres por edificio. Luego de que El Vicedecano de residencia y extensión indique los datos del edificio a insertar, modificar o visualizar el sistema realiza la opción seleccionada.	
Referencia:	RF2, RF2.1, RF2.2, RF2.3	
Precondiciones:	El Vicedecano de residencia y extensión debe estar autenticado en el sistema.	
Flujo Normal de Eventos		
Acción del Actor		
Respuesta del Sistema		
1-El Vicedecano de residencia y extensión se dirige al menú principal y selecciona el submenú Gestionar edificios.	2-El sistema muestra varias opciones entre las que se encuentran: Insertar datos del edificio, Visualizar datos del edificio y Generar reporte de las capacidades libres por edificio.	
3-El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.	4-Si selecciona “Insertar datos de los edificios” ver sección Insertar datos del edificio . - Si selecciona “Visualizar datos de los edificios” ver sección Visualizar datos del edificio . - Si selecciona “Generar reporte de las capacidades libres por edificios” ver sección Generar reporte de las capacidades libres por edificio .	


Sección "Insertar datos del edificio"	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción "Insertar datos de los edificios".	2-El sistema muestra un formulario en los que pide llenar los datos del edificio a insertar (número del edificio, sexo, cantidad de apartamentos, cantidad de pasos de escalera).
3- El Vicedecano de residencia y extensión inserta los datos del edificio.	4-El sistema verifica que no haya campos vacíos.
	5-El sistema verifica que no exista un edificio con el mismo número.
	6-Si los datos introducidos están correctos los guarda en la base de datos.
Prototipo interfaz	
	
Sección "Visualizar datos del edificio"	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción "Visualizar datos de los edificios".	2- El sistema muestra todos los edificios insertados con sus datos correspondientes y dos opciones para cada uno: - Modificar datos de los edificios - Eliminar datos de los edificios
3-El Vicedecano de residencia y extensión selecciona una de las opciones.	4- Si selecciona "Modificar datos de los edificios" ver sección Modificar datos del

	<p>edificio.</p> <p>- Si selecciona “Eliminar datos de los edificios” ver sección Eliminar datos del edificio.</p>
--	--

Prototipo interfaz

11 Edificios 1 - 5				
No. del edificio	Cantidad de apartamentos	Cantidad de pasos de escalera	Sexo	
70	6	1	m	  Capacidades Libres
71	9	2	m	  Capacidades Libres
72	5	2	f	  Capacidades Libres
73	5	2	f	  Capacidades Libres
74	8	2	m	  Capacidades Libres

« < 1 - 2 - 3 > »




Sección “Modificar datos del edificio”

Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Modificar datos de los edificios”.	2-El sistema muestra los datos del edificio seleccionado (número, sexo, cantidad de apartamentos, cantidad de pasos de escalera).
3- El Vicedecano de residencia y extensión modifica los datos necesarios.	4-El sistema verifica que no haya campos vacíos.
	5- Si las modificaciones de los datos fueron correctas el sistema guarda los cambios efectuados en la base de datos.

Prototipo interfaz


Modificar datos de los edificios	
Edificio:	<input type="text" value="70"/>
No. apartamentos:	<input type="text" value="6"/>
No. pasos de escalera:	<input type="text" value="1"/>
Sexo:	<input type="text" value="m"/>
<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>	

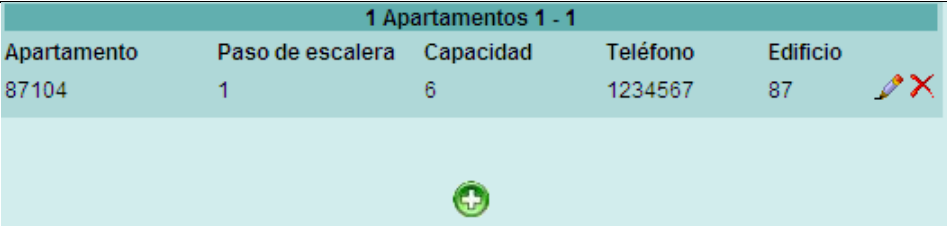
Sección "Eliminar datos del edificio"	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción "Eliminar datos del edificio".	2-El sistema muestra un mensaje de alerta indicando si desea o no eliminar el edificio seleccionado.
3-El Vicedecano de residencia y extensión indica que desea eliminar el edificio seleccionado.	4-El sistema elimina el edificio seleccionado y actualiza la página.
Sección "Generar reporte de las capacidades libres por edificio"	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la sección "Generar reporte de las capacidades libres por edificios".	2- El sistema muestra todos los edificios con sus capacidades libres y el total de las capacidades libres.
Prototipo Interfaz	
	
Flujos Alternos	
Sección "Insertar datos del edificio"	
Acción del Actor	Respuesta del Sistema
3.1- El Vicedecano de residencia y extensión al insertar los datos deja campos vacíos en el formulario.	4.1- El sistema muestra un mensaje de error para cada campo indicando que debe introducir el dato que falta.
3.2- El Vicedecano de residencia y extensión introduce un edificio con el número igual a uno ya insertado.	5.1-El sistema muestra un mensaje de error indicando que ya existe un edificio con ese número.
3.3- El Vicedecano de residencia y extensión introduce algún dato mal.	6.1- El sistema muestra un mensaje de error indicando que es lo que debe introducir.
Sección "Modificar datos del edificio"	
Acción del Actor	Respuesta del Sistema

3.1-El Vicedecano de residencia y extensión al modificar los datos dejó campos vacíos	4.1- El sistema muestra un mensaje de error para cada campo indicando que debe introducir el dato que falta.
3.2-El Vicedecano de residencia y extensión al modificar los datos introdujo algún campos mal.	5.1- El sistema muestra un mensaje de error para el campo indicando la forma correcta.
Sección “Eliminar datos del edificio”	
Acción del Actor	Respuesta del Sistema
5.1-El Vicedecano de residencia y extensión indica que no desea eliminar el edificio seleccionado.	6.1- El sistema no elimina el edificio seleccionado.
Poscondiciones:	Quedan insertados los edificios con los datos correspondientes.
Prioridad:	Crítico

Tabla32 Descripción del CUS” Gestionar apartamentos”

Caso de Uso:	Gestionar apartamentos
Actores:	Vicedecano de residencia y extensión
Resumen:	El caso de uso comienza cuando el Vicedecano de residencia y extensión decide Insertar datos del apartamento, Modificar datos del apartamento, Visualizar datos del apartamento, Eliminar datos del apartamento. Luego que El Vicedecano de residencia y extensión indique los datos del apartamento a insertar, modificar, visualizar, eliminar, el sistema realiza la opción seleccionada.
Referencia:	RF3, RF3.1, RF3.2, RF3.3, RF3.4, RF2.5
Precondiciones:	El Vicedecano de residencia y extensión debe estar autenticado en el sistema.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión se dirige al menú principal y selecciona el submenú Gestionar	2-El sistema muestra las opciones: Insertar datos del apartamento y Visualizar datos del apartamento.

apartamentos.	
3- El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.	4-Si selecciona “Insertar datos del apartamento” ver sección Insertar datos del apartamento . - Si selecciona “Visualizar datos del apartamento” ver sección Visualizar datos del apartamento .
Sección “Insertar datos del apartamento”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Insertar datos de los apartamentos”.	2-El sistema muestra un formulario en los que pide llenar los datos del apartamento a insertar (edificio, número del apartamento, teléfono, capacidad, número del paso de escalera).
3- El Vicedecano de residencia y extensión inserta los datos del apartamento.	4-El sistema verifica que no haya campos vacíos.
	5-El sistema verifica que no exista un apartamento con el mismo número.
	6-Si los datos introducidos están correctos los guarda en la base de datos.
Prototipo interfaz	
	
Sección “Visualizar datos del apartamento”	

Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción "Visualizar datos de los apartamentos".	2- El sistema muestra todos los apartamentos insertados hasta el momento con sus datos correspondientes y dos opciones para cada uno: - Modificar datos del apartamento - Eliminar datos de los apartamentos
3-El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.	4-Si selecciona "Modificar datos del apartamento" ver sección Modificar datos del apartamento . -Si selecciona "Eliminar datos del apartamento" ver sección Eliminar datos del apartamento .
Prototipo interfaz	
	
Sección "Modificar datos del apartamento"	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción "Modificar datos del apartamento".	2- El sistema muestra los datos del apartamento seleccionado (número del apartamento, teléfono, capacidad, número del paso de escalera y edificio al que pertenece).
3- El Vicedecano de residencia y extensión modifica los datos necesarios.	4-El sistema verifica que no haya campos vacíos.
	5-Si las modificaciones de los datos fueron correctas el sistema guarda los cambios efectuados en la base de datos.
Prototipo interfaz	



Modificar datos del apartamento	
Edificio:	25
Paso de escalera:	1
Apartamento:	105
Capacidad:	6
Teléfono:	1236547
<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>	
Sección "Eliminar datos del apartamento"	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción "Eliminar datos del apartamento".	2- El sistema muestra un mensaje de alerta indicando si desea o no eliminar el apartamento.
3-El Vicedecano de residencia y extensión indica que desea eliminar el apartamento seleccionado.	4-El sistema elimina el apartamento indicado.
Flujos Alternos	
Sección "Insertar datos del apartamento"	
Acción del Actor	Respuesta del Sistema
3.1- El Vicedecano de residencia y extensión al insertar los datos deja campos vacíos en el formulario.	4.1- El sistema muestra un mensaje de error para cada campo indicando que debe introducir el dato que falta.
3.2 El Vicedecano de residencia y extensión introduce un apartamento con el número igual a uno ya insertado.	5.1-El sistema muestra un mensaje de error indicando que ya existe un apartamento con ese número.
3.2 El Vicedecano de residencia y extensión introduce un algún campo con un dato incorrecto.	6.1-El sistema muestra un mensaje de error indicando la manera correcta de insertar el campo.
Sección "Modificar datos del apartamento"	
Acción del Actor	Respuesta del Sistema
3.1-El Vicedecano de residencia y	4.1- El sistema muestra un mensaje de error

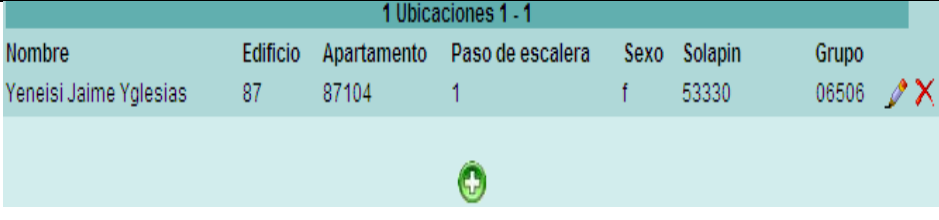
extensión al modifica los datos deja campos vacíos en el formulario.	para cada campo indicando que debe introducir el dato que falta.
3.2-El Vicedecano de residencia y extensión al modificar los datos introduce algún campo incorrecto.	5.1- El sistema muestra un mensaje de error para el campo indicando la forma correcta de introducir el dato.
Sección “Eliminar datos del apartamento”	
Acción del Actor	Respuesta del Sistema
3.1-El Vicedecano de residencia y extensión indica que no desea eliminar el apartamento.	4.1- El sistema no elimina el apartamento.
Poscondiciones:	Quedan insertados los apartamentos con los datos correspondientes.
Prioridad:	Crítico

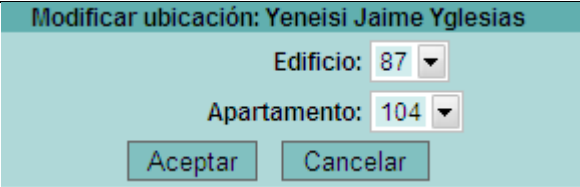
Tabla 33 Descripción del CUS” Gestionar datos de ubicación del estudiante”

Caso de Uso:	Gestionar datos de ubicación del estudiante
Actores:	Vicedecano de residencia y extensión
Resumen:	El caso de uso comienza cuando El Vicedecano de residencia y extensión decide Insertar, Modificar, Visualizar, Buscar o Eliminar datos de ubicación del estudiante. Luego que el Vicedecano de residencia y extensión indique los datos de ubicación a insertar, modificar, visualizar o eliminar el sistema realiza la opción seleccionada.
Referencia:	RF3, RF3.1, RF3.2, RF3.3, RF3.4, RF3.5
Precondiciones:	El Vicedecano de residencia y extensión debe estar autenticado en el sistema.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión se dirige al menú principal y selecciona el submenú Gestionar datos de ubicación del estudiante.	2-El sistema muestra varias opciones entre las que se encuentran: Insertar datos de ubicación, Buscar ubicación del estudiante y Visualizar listado de estudiantes con los

	datos de ubicación.
3-El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.	4-Si selecciona la opción Insertar datos de ubicación del estudiante en la residencia ver sección “Insertar datos de ubicación del estudiante en la residencia” . - Si selecciona la opción Buscar ubicación del estudiante ver sección “Buscar ubicación del estudiante” . - Si selecciona la opción Visualizar listado de estudiantes con los datos de ubicación ver sección “Visualizar listado de estudiantes con los datos de ubicación” .
Sección “Insertar datos de ubicación del estudiante en la residencia”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Insertar datos de ubicación del estudiante”.	2- El sistema muestra un formulario en los que pide llenar los datos de ubicación a insertar (grupo, nombre del estudiante, número del edificio y número del apartamento).
3-El Vicedecano de residencia y extensión inserta los datos.	4-El sistema verifica que no haya campos vacíos.
	5-Si los datos introducidos están correctos el sistema los guarda en la base de datos.
Prototipo interfaz	

	
Sección “Buscar ubicación del estudiante”	
Acción del Actor	Respuesta del Sistema
<p>1-El Vicedecano de residencia y extensión selecciona la opción “Buscar datos de ubicación del estudiante”.</p> <p>3- El Vicedecano de residencia y extensión indica el criterio(s) seleccionado para buscar al estudiante.</p>	<p>2-El sistema muestra un formulario con los criterios de búsqueda (usuario, edificio, sexo, grupo, nombre).</p> <p>4- El sistema verifica que al menos un criterio sea escogido.</p> <p>5-El sistema busca en la base de datos el estudiante(s) que coincida con el criterio(s) especificado(s).</p> <p>6-El sistema muestra el estudiante(s) con los datos correspondientes.</p>
Prototipo interfaz	
	
Sección “Visualizar los datos de ubicación del estudiante”	
Acción del Actor	Respuesta del Sistema
<p>1-El Vicedecano de residencia y extensión selecciona la opción “Visualizar los datos de</p>	<p>2- El sistema muestra un listado con los estudiantes insertados hasta el momento, visualizando (nombre, 1er apellido, 2do</p>

ubicación del estudiante”.	<p>apellido, sexo, edificio, apartamento, paso de escalera y grupo) y dos opciones para cada uno:</p> <p>-Modificar datos de ubicación del estudiante en la residencia</p> <p>-Eliminar datos de ubicación del estudiante</p>
3-El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.	<p>4- Si selecciona la opción Modificar datos de ubicación del estudiante en la residencia ver sección “Modificar datos de ubicación del estudiante en la residencia”.</p> <p>-Si selecciona la opción Eliminar datos de ubicación del estudiante en la residencia ver sección “Eliminar datos de ubicación del estudiante en la residencia”.</p>
Prototipo Interfaz	
	
Sección “Modificar datos de ubicación del estudiante en la residencia”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Modificar datos de ubicación del estudiante”.	2- El sistema muestra los datos del estudiante seleccionado (grupo, nombre del estudiante, número del edificio, número del apartamento).
3-El Vicedecano de residencia y extensión modifica los datos necesarios.	4-El sistema verifica que no haya campos vacíos.
	5-El sistema guarda los cambios efectuados en la base de datos.

Prototipo Interfaz	
	
Sección “Eliminar datos de ubicación del estudiante”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Eliminar datos de ubicación del estudiante”.	2- El sistema muestra un mensaje de alerta indicando si desea o no eliminar la ubicación.
3-El Vicedecano de residencia y extensión indica que desea eliminar la ubicación seleccionada.	4-El sistema elimina la ubicación indicada.
Flujos Alternos	
Sección “Insertar datos de ubicación del estudiante en la residencia”	
Acción del Actor	Respuesta del Sistema
3.1- El Vicedecano de residencia y extensión al insertar los datos dejó campos vacíos.	4.1 El sistema muestra un mensaje de error:”Debe introducir todos los datos.”
3.2- El Vicedecano de residencia y extensión al insertar los datos insertos algún dato incorrectamente.	5.1 El sistema muestra un mensaje indicando la mejor manera de llenar el campo.
Sección “Modificar datos de ubicación del estudiante en la residencia”	
Acción del Actor	Respuesta del Sistema
3.1- El Vicedecano de residencia y extensión al modificar los datos dejó campos vacíos.	4.1- El sistema muestra un mensaje de error:”Debe introducir todos los datos.”
Sección” Buscar ubicación del estudiante”	
Acción del Actor	Respuesta del Sistema
3.1-El Vicedecano de residencia y extensión dejó los campos vacíos.	4.1-El sistema muestra un mensaje de error “Introduzca al menos un criterio”.
	5.1-El sistema no encontró coincidencias con el dato(s) insertado(s).
	6.1-El sistema muestra un mensaje: No


	hubo resultados para esa búsqueda.
Sección” Eliminar datos de ubicación del estudiante ”	
Acción del Actor	Respuesta del Sistema
3.1-El Vicedecano de residencia y extensión indica que no desea eliminar la ubicación.	4.1-El sistema no elimina la ubicación.
Poscondiciones:	Quedan insertados los datos de ubicación de los estudiantes.
Prioridad:	Crítico

Tabla 34 Descripción del CUS” Gestionar datos de la instructora”

Caso de Uso:	Gestionar datos de la instructora
Actores:	Vicedecano de residencia y extensión
Resumen:	El caso de uso comienza cuando el Vicedecano de residencia y extensión decide Insertar datos de la instructora, Modificar datos de la instructora, Visualizar datos de la instructora y Eliminar datos de la instructora. Luego de que El Vicedecano de residencia y extensión indique los datos de la instructora a insertar, modificar, visualizar o eliminar, el Sistema realiza la opción seleccionada.
Referencia:	RF4, RF4.1, RF4.2, RF4.3, RF4.4
CU asociados:	
Precondiciones:	El Vicedecano de residencia y extensión debe estar autenticado en el Sistema.
Flujo Normal de Eventos	
Acción del Actor	
Respuesta del Sistema	
1-El Vicedecano de residencia y extensión se dirige al menú principal y selecciona el submenú Gestionar datos de la instructora.	2-El sistema muestra varias opciones entre las que se encuentran: Insertar datos de la instructora y Visualizar datos de la instructora.

3-El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.	4-Si selecciona la opción Insertar datos de la instructora ver sección “Insertar datos de la instructora” . - Si selecciona la opción Visualizar datos de la instructora ver sección “Visualizar datos de la instructora” . - Si selecciona la opción Modificar datos de la instructora ver sección “Modificar datos de la instructora” . -Si selecciona la opción Eliminar datos de la instructora ver sección “Eliminar datos de la instructora” .
Sección “Insertar datos de la instructora”	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción “Insertar datos de la instructora”.	2- El sistema muestra un formulario en los que pide llenar los datos de la instructora a insertar (nombre, 1er apellido, 2do apellido, usuario, número de edificio, número de solapín, apartamento en el que radica y teléfono).
3-El Vicedecano de residencia y extensión inserta los datos.	4-El sistema verifica que no haya campos vacíos.
	5- El sistema verifica que no exista una instructora con el mismo número de solapín.
	6- Si los datos introducidos están correctos el Sistema los guarda en la base de datos.
Prototipo interfaz	

<div style="background-color: #e0f2f1; padding: 5px; border: 1px solid black;"> <p style="text-align: center; margin: 0;">Insertar datos de la instructora</p> <p>Nombre: <input style="width: 100%;" type="text"/> *</p> <p>1er. apellido: <input style="width: 100%;" type="text"/> *</p> <p>2do. apellido: <input style="width: 100%;" type="text"/> *</p> <p>Usuario: <input style="width: 100%;" type="text"/></p> <p>Solapín: <input style="width: 100%;" type="text"/> *</p> <p>Edificio <input style="width: 100%;" type="text" value=" << Seleccione >> "/> *</p> <p>Apto. radica: <input style="width: 100%;" type="text"/> *</p> <p>Teléfono: <input style="width: 100%;" type="text"/> *</p> <p style="text-align: center;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p> </div>															
<p>Sección “Visualizar datos de la instructora”</p>															
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>														
<p>1-El Vicedecano de residencia y extensión selecciona la opción “Visualizar datos de la instructora”.</p>	<p>2- El sistema muestra un listado con las instructoras insertadas hasta el momento, visualizando (nombre, 1er apellido, 2do apellido, usuario, número de edificio, número de solapín, apartamento en el que radica y teléfono) y dos opciones para cada uno:</p> <ul style="list-style-type: none"> -Modificar datos de la instructora -Eliminar datos de la instructora 														
<p>3-El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.</p>	<p>4- Si selecciona la opción Modificar datos de la instructora ver sección “Modificar datos de la instructora”.</p> <p>-Si selecciona la opción Eliminar datos de la instructora ver sección “Eliminar datos de la instructora”.</p>														
<p>Prototipo interfaz</p>															
<div style="background-color: #e0f2f1; padding: 5px; border: 1px solid black;"> <p style="text-align: center; margin: 0;">1 instructoras 1 - 1</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Edificio</th> <th style="width: 20%;">Nombre</th> <th style="width: 10%;">Usuario</th> <th style="width: 10%;">Solapín</th> <th style="width: 10%;">Apto. Radica</th> <th style="width: 10%;">Teléfono</th> <th style="width: 10%;"></th> </tr> </thead> <tbody> <tr> <td>87</td> <td>maritza.perez.perez</td> <td>mperez</td> <td>12345</td> <td>87102</td> <td>1234567</td> <td style="text-align: right;">✎ ✕</td> </tr> </tbody> </table> <p style="text-align: center; margin-top: 10px;"> <input type="button" value="➕"/> </p> </div>		Edificio	Nombre	Usuario	Solapín	Apto. Radica	Teléfono		87	maritza.perez.perez	mperez	12345	87102	1234567	✎ ✕
Edificio	Nombre	Usuario	Solapín	Apto. Radica	Teléfono										
87	maritza.perez.perez	mperez	12345	87102	1234567	✎ ✕									

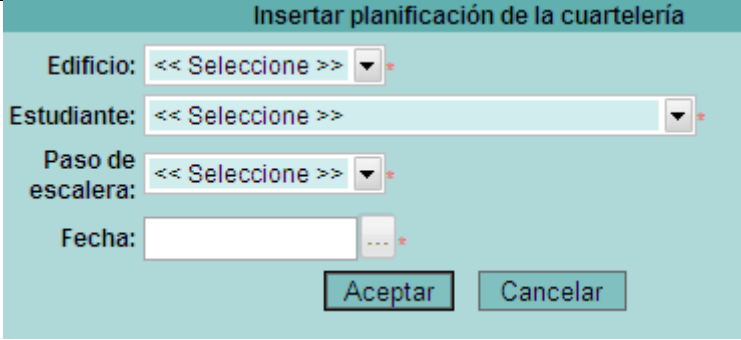
Sección “Modificar datos de la instructora”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Modificar datos de la instructora”.	2-El sistema muestra los datos de la instructora seleccionada (nombre, 1er apellido, 2do apellido, usuario, número de edificio, número de solapín, apartamento en el que radica y teléfono).
3-El Vicedecano de residencia y extensión modifica los datos necesarios de la instructora seleccionada.	4-Si las modificaciones de los datos fueron correctas el Sistema guarda los cambios efectuados en la base de datos.
Prototipo interfaz	
	
Sección “Eliminar datos de la instructora”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Eliminar datos de la instructora”.	2- El sistema muestra un mensaje de alerta indicando si desea o no eliminar la instructora.
3-El Vicedecano de residencia y extensión indica que desea eliminar a la instructora.	4-El sistema elimina la instructora indicada.
Flujos Alternos	
Sección “Insertar datos de la instructora”	
Acción del Actor	Respuesta del Sistema
3.1-El Vicedecano de residencia y	4.1-El sistema muestra un mensaje de

extensión al insertar los datos dejó campos vacíos.	error para cada campo indicando que debe introducir el dato que falta.
3.2-El solapín de la instructora insertada por el Vicedecano de residencia y extensión coincide con uno ya insertado.	5.1-El sistema muestra un mensaje “Existe una instructora con ese solapín.”
3.3-El Vicedecano de residencia y extensión al llenar los campos, introdujo alguno incorrectamente.	6.1-El sistema muestra un mensaje indicando la mejor manera de llenar el campo.
Sección “Modificar datos de la instructora”	
Acción del Actor	Respuesta del Sistema
3.1-El Vicedecano de residencia y extensión al modificar los datos lleno algún campo incorrectamente.	4.1-El sistema muestra un mensaje de error para cada campo indicando la mejor manera de llenar este campo.
Sección “Eliminar datos de la instructora”	
Acción del Actor	Respuesta del Sistema
3.1-El Vicedecano de residencia y extensión indica que no desea eliminar el estudiante.	4.1- El sistema no elimina la instructora.
Poscondiciones:	Quedan insertados los datos de la instructora.
Prioridad:	Secundario

Tabla 35 Descripción del CUS” Gestionar Cuartelería”

Caso de Uso:	Gestionar cuartelería
Actores:	Instructora
Resumen:	El caso de uso comienza cuando la instructora decide Insertar planificación de la cuartelería, Modificar planificación de la cuartelería y Visualizar la planificación de la cuartelería, así como Insertar y Modificar la asistencia de los estudiantes a la misma, y además permite generar reportes.
Referencia:	RF5, RF5.1, RF5.2, RF5.3, RF5.4, RF5.5, RF5.6, RF5.7
CU asociados:	
Precondiciones:	La instructora debe estar autenticada en el Sistema.

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-La instructora se dirige al menú principal y selecciona el submenú Gestionar cuartelería.	2-El sistema muestra varias opciones entre las que se encuentran: Insertar planificación de la cuartelería, Visualizar planificación de la cuartelería y Generar reportes de la cuartelería.
3-La instructora selecciona una de las opciones presentadas.	4-Si selecciona la opción Insertar planificación de la cuartelería ver sección Insertar planificación de la cuartelería . -Si selecciona la opción Visualizar planificación de la cuartelería ver sección Visualizar planificación de la cuartelería . -Si selecciona la opción Generar reportes de la cuartelería ver sección Generar reportes de la cuartelería .
Sección “Insertar planificación de la cuartelería”	
Acción del Actor	Respuesta del Sistema
1-La instructora selecciona la sección Insertar planificación de la cuartelería.	2-El sistema muestra un formulario en el que pide llenar los campos de la planificación a insertar(nombre del cuartelero, fecha, número de edificio, número de paso de escalera)
3-La instructora inserta los datos de la planificación.	4-El sistema verifica que no haya campos vacíos.
	5- El sistema verifica que no exista una planificación con los mismos datos.
	6- Si los datos introducidos están correctos el sistema los guarda en la

	base de datos.
Prototipo interfaz	
	
Sección “Visualizar planificación de la cuartería”	
Acción del Actor	Respuesta del Sistema
1-La instructora selecciona la opción “Visualizar planificación de la cuartería”.	2-.El sistema muestra un listado de las planificaciones insertadas hasta el momento con los datos correspondientes y cuatro opciones para cada una: <ul style="list-style-type: none"> - Modificar planificación de la cuartería - Eliminar planificación de la cuartería - Insertar asistencia a la cuartería - Modificar asistencia a la cuartería
3-La instructora selecciona una de las opciones presentadas.	4-Si selecciona la opción Modificar planificación de la cuartería ver sección Modificar planificación de la cuartería . <ul style="list-style-type: none"> -Si selecciona la opción Eliminar planificación de la cuartería ver sección Eliminar planificación de la cuartería.

	<p>-Si selecciona la opción Insertar asistencia a la cuartelería ver sección Insertar asistencia a la cuartelería.</p> <p>-Si selecciona la opción Modificar asistencia a la cuartelería ver sección Modificar asistencia a la cuartelería.</p>
--	---

Prototipo interfaz

4 Planificaciones de la cuartelería 1 - 4			
Fecha cuartelería	Estudiante	Edificio	Paso escalera
16-06-2009	Humberto González Boza	25	1
16-06-2009	Ernesto Aviles González	25	1
16-06-2009	Antonio Pascual Rivera Matos	25	1
23-06-2009	Ernesto Aviles González	25	1

Sección “Modificar planificación de la cuartelería”

Acción del Actor	Respuesta del Sistema
1-La instructora selecciona la opción “Modificar planificación de la cuartelería”.	2-El sistema muestra los datos de la planificación seleccionada (nombre del cuartelero, fecha, número de edificio, número de paso de escalera).
3-La instructora modifica los datos necesarios.	4-Si las modificaciones de los datos fueron correctas el sistema guarda los cambios efectuados en la base de datos.

Prototipo interfaz

Modificar planificación de la cuartelería

Edificio: 87 *

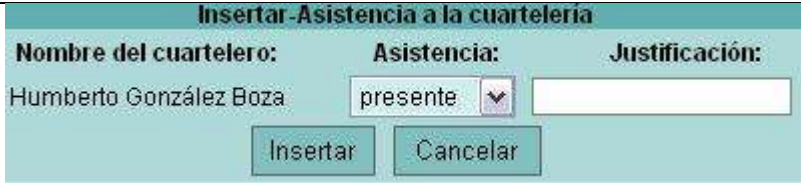
Estudiante: Ernesto Aviles González *

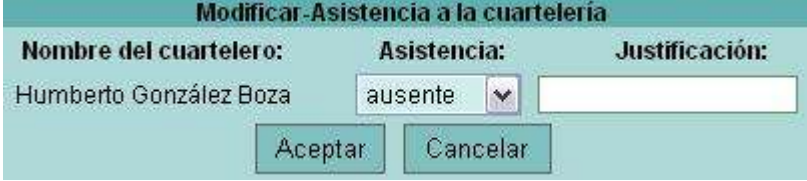

Paso de escalera: 1 *

Fecha: 2011-06-2009 *

Sección ”Eliminar planificación de la cuartelería”

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

1-La instructora selecciona la opción "Eliminar planificación de la cuartelería".	2- El sistema muestra un mensaje de alerta indicando si desea o no eliminar la planificación.
3-La instructora indica que desea eliminar la planificación.	4-El sistema elimina la planificación indicada.
Sección "Insertar asistencia a la cuartelería"	
Acción del Actor	Respuesta del Sistema
1-La instructora selecciona la opción Insertar asistencia a la cuartelería.	2-El sistema muestra el estudiante relacionado con la cuartelería seleccionada y dos campos, uno con opciones a seleccionar (presente, ausente, justificado) y otro para en caso de que la opción seleccionada en el campo anterior sea justificado poder insertar la justificación.
3-La instructora inserta la asistencia.	4-El sistema verifica que se inserten los datos correctos.
	5-El sistema guarda los datos en la base de datos.
Prototipo interfaz	
	
Sección "Modificar asistencia a la cuartelería"	
Acción del Actor	Respuesta del Sistema
1-La instructora selecciona la opción Modificar asistencia a la cuartelería.	2-El sistema muestra al estudiante relacionado con la cuartelería seleccionada y los campos con los datos correspondientes para modificar la asistencia.
3-La instructora modifica la asistencia.	4-El sistema verifica que se inserten los

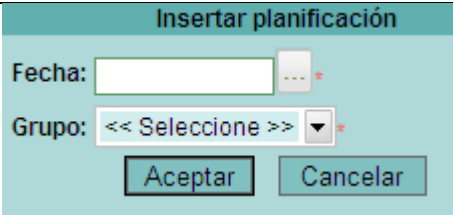
	datos correctos.
	5-El sistema guarda los datos en la base de datos.
Prototipo interfaz	
	
Sección “Generar reportes de la cuartería.”	
Acción del Actor	Respuesta del Sistema
1-La instructora selecciona la opción Generar reportes de la cuartería.	2-El sistema muestra un formulario con un campo para seleccionar una fecha de una planificación.
3-La instructora selecciona la fecha.	4-El sistema muestra un listado con todos los estudiantes que no realizaron la cuartería en esa fecha, el edificio, el paso de escalera, el tipo de asistencia.
Prototipo interfaz	
	
Flujos Alternos	
Sección “Insertar planificación de la cuartería”	
Acción del Actor	Respuesta del Sistema
3.1-La instructora al insertar los datos dejó campos vacíos.	4.1-El sistema muestra un mensaje de error para cada campo indicando que debe introducir el dato que falta.
3.2-La instructora al insertar algún campo igual.	5.1-El sistema muestra un mensaje de error indicando el campo.
3.3-La instructora al insertar los datos introdujo algún campo incorrectamente.	6.1-El sistema muestra un mensaje de error para el campo indicando la forma de introducir el dato que correctamente.


Sección “Eliminar planificación de la cuartería”	
Acción del Actor	Respuesta del Sistema
3.1-La instructora indica que no desea eliminar la planificación.	4.1-El sistema no elimina la planificación.
Sección “Modificar planificación de la cuartería”	
Acción del Actor	Respuesta del Sistema
3.1-La instructora al modificar los datos introduce algún dato incorrecto.	4.1-El sistema muestra un mensaje de error indicando la mejor forma de llenar el campo.
Sección “Insertar asistencia a la cuartería”	
Acción del Actor	Respuesta del Sistema
3.1-Al insertar la asistencia la instructora inserto algún campo incorrecto.	4.1-El sistema muestra un mensaje indicando el tipo de dato que se debe introducir.
Sección “Modificar asistencia a la cuartería”	
Acción del Actor	Respuesta del Sistema
3.1-Al insertar la asistencia la instructora inserto algún campo incorrecto.	4.1-El sistema muestra un mensaje indicando el tipo de dato que se debe introducir.
Poscondiciones:	Quedan insertados los datos de la cuartería.
Prioridad:	Crítico


Tabla 36 Descripción del CUS” Gestionar Guardia Obrera Estudiantil”

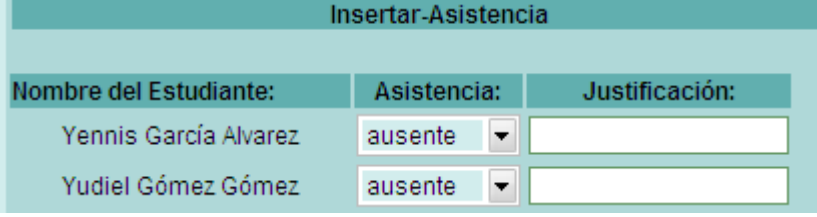

Caso de Uso:	Gestionar guardia obrera estudiantil
Actores:	Vicedecano de residencia y extensión
Resumen:	El caso de uso comienza cuando El Vicedecano de residencia y extensión decide Insertar planificación de la guardia obrera estudiantil, Modificar planificación de la guardia obrera estudiantil, Eliminar planificación de la guardia obrera estudiantil y Visualizar la planificación de la guardia obrera estudiantil, así

	como Insertar y Modificar la asistencia de los estudiantes a la misma, y además permite generar reportes.
Referencia:	RF6, RF6.1, RF6.2, RF6.3, RF6.4, RF6.5, RF6.6, RF6.7
CU asociados:	
Precondiciones:	El Vicedecano de residencia y extensión debe estar autenticado en el sistema.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión se dirige al menú principal y selecciona el submenú Gestionar guardia obrera estudiantil.	2-El Sistema muestra varias opciones entre las que se encuentran: Insertar, Visualizar planificación de la guardia obrera estudiantil, y Generar reportes de la guardia obrera estudiantil.
3- El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.	4-Si selecciona la opción Insertar planificación de la guardia obrera estudiantil ver sección Insertar planificación de la guardia obrera estudiantil . -Si selecciona la opción Visualizar planificación de la guardia obrera estudiantil ver sección Visualizar planificación de la guardia obrera estudiantil . -Si selecciona la opción Generar reportes de la guardia obrera estudiantil ver sección Generar reportes de la guardia obrera estudiantil .
Sección "Insertar planificación de la guardia obrera estudiantil"	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y	2-El sistema muestra un formulario en el

extensión selecciona la sección Insertar planificación de la guardia obrera estudiantil.	que pide llenar los campos de la planificación a insertar (fecha, brigada).
3- El Vicedecano de residencia y extensión inserta los datos de la planificación.	4-El sistema verifica que no haya campos vacíos.
	5-El sistema verifica que no exista una planificación con la misma fecha.
	6-Si los datos introducidos están correctos el Sistema los guarda en la base de datos.
Prototipo interfaz	
	
Sección “Visualizar planificación de la guardia obrera estudiantil”	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción Visualizar planificación de la guardia obrera estudiantil.	<p>2-El sistema muestra un listado de las planificaciones insertadas hasta el momento con 4 opciones para cada una:</p> <ul style="list-style-type: none"> -Modificar planificación de la guardia obrera estudiantil -Eliminar planificación de la guardia obrera estudiantil -Insertar asistencia a la guardia obrera estudiantil -Modificar asistencia a la guardia obrera estudiantil

<p>3- El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.</p>	<p>4-Si selecciona la opción Modificar planificación de la guardia obrera estudiantil ver sección Modificar planificación de la guardia obrera estudiantil.</p> <p>-Si selecciona la opción Eliminar planificación de la guardia obrera estudiantil ver sección Eliminar planificación de la guardia obrera estudiantil.</p> <p>-Si selecciona la opción Insertar asistencia a la guardia obrera estudiantil ver sección Insertar asistencia a la guardia obrera estudiantil.</p> <p>-Si selecciona la opción Modificar asistencia a la guardia obrera estudiantil ver sección Modificar asistencia a la guardia obrera estudiantil.</p>
Prototipo interfaz	
	
Sección “Modificar planificación de la guardia obrera estudiantil”	
Acción del Actor	Respuesta del Sistema
<p>1-El Vicedecano de residencia y extensión selecciona la opción “Modificar planificación de la guardia obrera estudiantil”.</p>	<p>2-El sistema muestra los datos de la planificación seleccionada (fecha, brigada).</p>

3- El Vicedecano de residencia y extensión modifica los datos necesarios.	4-El sistema verifica que no haya campos vacíos. 5-Si las modificaciones de los datos fueron correctas el sistema guarda los cambios efectuados en la base de datos.
Prototipo interfaz	
	
Sección "Eliminar planificación de la guardia obrera estudiantil"	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción "Eliminar planificación de la guardia obrera estudiantil".	2-El sistema muestra un mensaje de alerta indicando si desea o no eliminar la planificación.
3- El Vicedecano de residencia y extensión indica que desea eliminar la planificación.	4-El sistema elimina la planificación indicada.
Sección "Insertar asistencia a la guardia obrera estudiantil"	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción Insertar asistencia a la guardia obrera estudiantil.	2-El sistema muestra un listado con los estudiantes relacionados con la planificación seleccionada y dos campos, uno con opciones a seleccionar (presente, ausente, justificado) y el otro para en caso de que la opción seleccionada en el campo anterior sea justificado poder insertar la justificación.
3- El Vicedecano de residencia y	4-El sistema guarda los datos en la base

extensión le inserta la asistencia a cada estudiante.	de datos.
Prototipo interfaz	
	
Sección "Modificar asistencia a la guardia obrera estudiantil"	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción Modificar asistencia a la guardia obrera estudiantil.	2-El sistema muestra un listado con los estudiantes relacionados con la planificación seleccionada y los campos con la asistencia a modificar.
3- El Vicedecano de residencia y extensión modifica la asistencia a los alumnos que desee.	4-El sistema guarda los cambios efectuados.
Prototipo interfaz	
	
Sección "Generar reportes de la guardia obrera estudiantil."	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción Generar reportes de la Guardia Obrera Estudiantil	2-El sistema muestra un formulario con un campo para seleccionar la fecha de la guardia que se va a reportar.
3-El Vicedecano de residencia y extensión selecciona la fecha.	4-El sistema muestra un formulario con los estudiantes que no asistieron ese día a la guardia.
Prototipo interfaz	


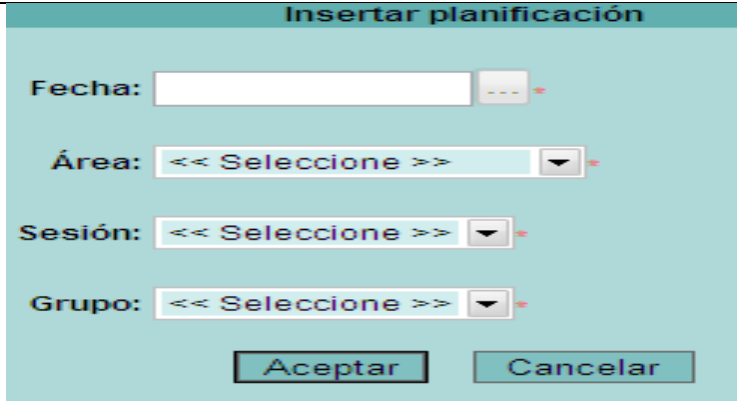
	
Flujos Alternos	
Sección “Insertar planificación de la guardia obrera estudiantil”	
Acción del Actor	Respuesta del Sistema
3.1- El Vicedecano de residencia y extensión al insertar los datos dejó campos vacíos.	4.1-El sistema muestra un mensaje de error para cada campo indicando que debe introducir el dato que falta.
3.2-La fecha de la planificación insertada por el vicedecano coincide con alguna ya insertada.	5.1-El sistema muestra un mensaje “Existe una planificación con esa fecha”.
Sección “Modificar planificación de la guardia obrera estudiantil”	
Acción del Actor	Respuesta del Sistema
5.1- El Vicedecano de residencia y extensión al modificar los datos dejó campos vacíos.	6.1-El sistema muestra un mensaje de error para cada campo indicando que debe introducir el dato que falta.
Sección “Eliminar planificación de la guardia obrera estudiantil”	
Acción del Actor	Respuesta del Sistema
3.1- El Vicedecano de residencia y extensión indica que no desea eliminar la planificación.	4.1-El sistema no elimina la planificación.
Poscondiciones:	Quedan insertados los datos de la guardia obrera estudiantil.
Prioridad:	Crítico















Tabla 37 Descripción del CUS” Gestionar Trabajo Socialmente Útil”


Caso de Uso:	Gestionar Trabajo Socialmente Útil
Actores:	Vicedecano de residencia y extensión

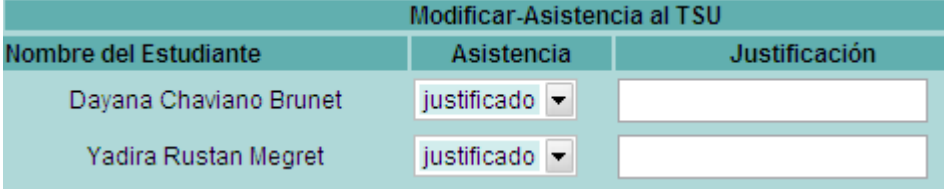

Resumen:	El caso de uso comienza cuando El Vicedecano de residencia y extensión decide Insertar planificación del trabajo socialmente útil, Modificar planificación del trabajo socialmente útil, Eliminar planificación del trabajo socialmente útil y Visualizar la planificación del trabajo socialmente útil, así como Insertar y Modificar la asistencia de los estudiantes al mismo, y además permite generar reportes.	
Referencia:	RF7, RF7.1, RF7.2, RF7.3, RF7.4, RF7.5, RF7.6, RF7.7	
CU asociados:		
Precondiciones:	El Vicedecano de residencia y extensión debe estar autenticado en el Sistema.	
Flujo Normal de Eventos		
Acción del Actor		
Respuesta del Sistema		
1-El Vicedecano de residencia y extensión se dirige al menú principal y selecciona el submenú Gestionar trabajo socialmente útil	2-El sistema muestra varias opciones entre las que se encuentran: Insertar planificación del trabajo socialmente útil, Visualizar planificación del trabajo socialmente útil, y Generar reportes del trabajo socialmente útil.	
3- El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.	4-Si selecciona la opción Insertar planificación del trabajo socialmente útil ver sección Insertar planificación del trabajo socialmente útil -Si selecciona la opción Visualizar planificación del trabajo socialmente útil ver sección Visualizar planificación del trabajo socialmente útil . -Si selecciona la opción Generar reportes del trabajo socialmente útil ver sección Generar reportes del trabajo	

	socialmente útil.
Sección “Insertar planificación del trabajo socialmente útil”	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la sección Insertar planificación del trabajo socialmente útil.	2-El sistema muestra un formulario en el que pide llenar los campos de la planificación a insertar (fecha, área, sesión, brigada).
3- El Vicedecano de residencia y extensión inserta los datos de la planificación.	4-El sistema verifica que no haya campos vacíos.
	5-El sistema verifica que no exista una planificación con los mismos datos.
	6-Si los datos introducidos están correctos el sistema los guarda en la base de datos.
Prototipo interfaz	
	
Sección “Visualizar planificación del trabajo socialmente útil”	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción Visualizar planificación del trabajo socialmente útil.	2-El sistema muestra un listado de las planificaciones insertadas hasta el momento con cuatro opciones a seleccionar para cada una: -Modificar planificación del trabajo

	<p>socialmente útil</p> <p>-Eliminar planificación del trabajo socialmente útil</p> <p>-Insertar asistencia al trabajo socialmente útil</p> <p>-Modificar asistencia al trabajo socialmente útil</p>
<p>3-El Vicedecano de residencia y extensión selecciona una de las opciones presentadas</p>	<p>4-Si selecciona la opción Modificar planificación del trabajo socialmente útil ver sección Modificar planificación del trabajo socialmente útil</p> <p>-Si selecciona la opción Eliminar planificación del trabajo socialmente útil ver sección Eliminar planificación del trabajo socialmente útil</p> <p>-Si selecciona la opción Insertar asistencia al trabajo socialmente útil ver sección Insertar asistencia al trabajo socialmente útil.</p> <p>-Si selecciona la opción Modificar asistencia al trabajo socialmente útil ver sección Modificar asistencia al trabajo socialmente útil.</p>
<p>Prototipo interfaz</p>	

4 Planificaciones del Tsu 1 - 4				
Fecha	Área	Grupo	Sesión	
15-06-2009	Comedor 2	06503	tarde	  
16-06-2009	Docente 4	06106	tarde	  
16-06-2009	Docente 4	06107	tarde	  
23-06-2009	Docente 4	06201	tarde	  
				
Sección “Modificar planificación del trabajo socialmente útil”				
Acción del Actor	Respuesta del Sistema			
1-El Vicedecano de residencia y extensión selecciona la opción “Modificar planificación del trabajo socialmente útil”.	2-El sistema muestra los datos de la planificación seleccionada (fecha, área, sesión, brigada).			
3-El Vicedecano de residencia y extensión modifica los datos necesarios.	4-El sistema verifica que no haya campos vacíos. 5-Si las modificaciones de los datos fueron correctas el sistema guarda los cambios efectuados, en la base de datos.			
Prototipo interfaz				
				
Sección “Eliminar planificación del trabajo socialmente útil”				
Acción del Actor	Respuesta del Sistema			
1-El Vicedecano de residencia y extensión selecciona la opción “Eliminar planificación del trabajo socialmente útil”.	2-El sistema muestra un mensaje de alerta indicando si desea o no eliminar la planificación.			
3-El Vicedecano de residencia y extensión indica que desea eliminar la planificación.	4-El sistema elimina la planificación indicada.			

Sección "Insertar asistencia al trabajo socialmente útil"	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción Insertar asistencia al trabajo socialmente útil.	2-El sistema muestra un listado con los estudiantes relacionados con la planificación seleccionada indicando de cada uno el nombre y dos campos, uno con opciones a seleccionar (presente, ausente, justificado)y otro para que en caso que la opción seleccionada en el campo anterior sea justificado poder insertar la justificación.
3-El Vicedecano de residencia y extensión le inserta la asistencia a cada estudiante.	4-El sistema guarda los datos en la base de datos.
Prototipo interfaz	
	
Sección "Modificar asistencia al trabajo socialmente útil"	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción Modificar asistencia al trabajo socialmente útil.	2-El sistema muestra un listado con los estudiantes que pertenecen a la brigada de la planificación seleccionada y los campos con la asistencia a modificar.


3- El Vicedecano de residencia y extensión modifica la asistencia a los estudiantes que desee.	4-El sistema guarda los cambios efectuados.
Prototipo interfaz	
	
Sección “Generar reportes del trabajo socialmente útil.”	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción Generar reportes del trabajo socialmente útil.	2-El sistema muestra un formulario con un campo para seleccionar la fecha del trabajo socialmente útil que se va a reportar.
3-El Vicedecano de residencia y extensión inserta la fecha del trabajo socialmente útil.	4-El sistema muestra los estudiantes que no asistieron al trabajo socialmente útil en esa fecha.
Prototipo interfaz	
	
Flujos Alternos	
Sección “Insertar planificación del trabajo socialmente útil”	
Acción del Actor	Respuesta del Sistema
3.1- El Vicedecano de residencia y extensión al insertar los datos dejó campos vacíos.	4.1-El sistema muestra un mensaje de error para cada campo indicando que debe introducir el dato que falta.
Sección “Modificar planificación del trabajo socialmente útil”	
Acción del Actor	Respuesta del Sistema
3.1-El Vicedecano de residencia y extensión al modificar los datos dejó	4.1-El sistema muestra un mensaje de error para cada campo indicando que

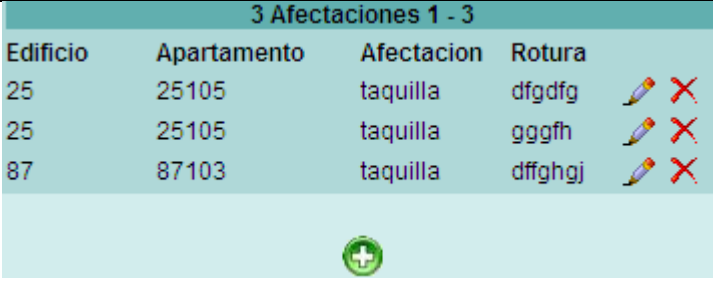
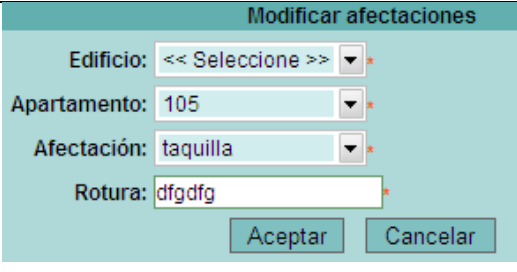
campos vacíos.	debe introducir el dato que falta.
Sección “Eliminar planificación del trabajo socialmente útil”	
Acción del Actor	Respuesta del Sistema
3.1- El Vicedecano de residencia y extensión indica que no desea eliminar la planificación.	4.1- El sistema no elimina la planificación.
Poscondiciones:	Quedan insertados los datos del trabajo socialmente útil.
Prioridad:	Crítico


Tabla 38 Descripción del CUS” Gestionar afectaciones de los apartamentos”

Caso de Uso:	Gestionar afectaciones de los apartamentos
Actores:	Vicedecano de residencia y extensión
Resumen:	El caso de uso comienza cuando El Vicedecano de residencia y extensión decide Insertar afectaciones de los apartamentos, Modificar afectaciones de los apartamentos, Eliminar afectaciones de los apartamentos o Generar reporte de afectaciones de los apartamentos. Luego de que El Vicedecano de residencia y extensión indique los datos del edificio a insertar, modificar o visualizar el sistema realiza la opción seleccionada.
Referencia:	RF8, RF8.1, RF8.2, RF8.3, RF8.4
Precondiciones:	
Flujo Normal de Eventos	
Acción del Actor	
Respuesta del Sistema	
1-El Vicedecano de residencia y extensión se dirige al menú principal y selecciona el submenú Gestionar afectaciones de los apartamentos.	2-El sistema muestra varias opciones entre las que se encuentran: Insertar afectaciones de los apartamentos y Generar reporte de las afectaciones de los apartamentos.

<p>3- El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.</p>	<p>4-Si selecciona “Insertar afectaciones de los apartamentos” ver sección Insertar afectaciones de los apartamentos.</p> <p>-Si selecciona “Visualizar afectaciones de los apartamentos” ver sección Visualizar afectaciones de los apartamentos.</p> <p>-Si selecciona” Generar reporte de las afectaciones de los apartamentos” ver sección Generar reporte de las afectaciones de los apartamentos.</p>
<p>Sección “Insertar afectaciones de los apartamentos”</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
<p>1- El Vicedecano de residencia y extensión selecciona la opción “Insertar afectaciones de los apartamentos”.</p>	<p>2-El sistema muestra un formulario en los que muestra los números de los apartamentos insertados hasta el momento un campo para insertar la rotura y diferentes opciones en las que se clasifican las afectaciones:</p> <ul style="list-style-type: none"> -TV -PC -Ventilador -Aire acondicionado -Filtración -Taquillas -Iluminación y corriente

	-Teléfono -Minibar -Muebles -Plomería
3- El Vicedecano de residencia y extensión inserta la(s) rotura(s).	4-El sistema verifica que no haya campos vacíos.
	5-Si los datos introducidos están correctos los guarda en la base de datos.
Prototipo interfaz	
	
Sección “Visualizar afectaciones de los apartamentos”	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción “Visualizar afectaciones de los apartamentos”.	2- El sistema muestra un formulario con las afectaciones que han sido insertadas hasta el momento con dos opciones a seleccionar para cada una: -Modificar afectaciones de los apartamentos -Eliminar afectaciones de los apartamentos
3-El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.	4-Si selecciona “Modificar afectaciones de los apartamentos” ver sección Modificar afectaciones de los

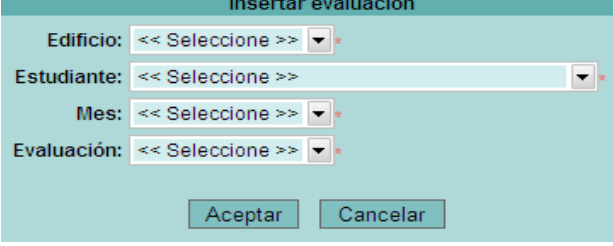
	<p>apartamentos.</p> <p>-Si selecciona “Eliminar afectaciones de los apartamentos” ver sección Eliminar afectaciones de los apartamentos.</p>																									
<p>Prototipo interfaz</p>																										
 <table border="1" style="background-color: #e0f2f1; width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="5" style="background-color: #4db6ac; color: white;">3 Afectaciones 1 - 3</th> </tr> <tr> <th>Edificio</th> <th>Apartamento</th> <th>Afectacion</th> <th>Rotura</th> <th></th> </tr> </thead> <tbody> <tr> <td>25</td> <td>25105</td> <td>taquilla</td> <td>dfgdfg</td> <td> </td> </tr> <tr> <td>25</td> <td>25105</td> <td>taquilla</td> <td>gggfh</td> <td> </td> </tr> <tr> <td>87</td> <td>87103</td> <td>taquilla</td> <td>dffghgj</td> <td> </td> </tr> </tbody> </table>		3 Afectaciones 1 - 3					Edificio	Apartamento	Afectacion	Rotura		25	25105	taquilla	dfgdfg		25	25105	taquilla	gggfh		87	87103	taquilla	dffghgj	
3 Afectaciones 1 - 3																										
Edificio	Apartamento	Afectacion	Rotura																							
25	25105	taquilla	dfgdfg																							
25	25105	taquilla	gggfh																							
87	87103	taquilla	dffghgj																							
<p>Sección “Modificar afectaciones de los apartamentos”</p>																										
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>																									
<p>1- El Vicedecano de residencia y extensión selecciona la opción “Modificar afectaciones de los apartamentos”.</p>	<p>2-El sistema muestra un formulario con los datos a modificar (número del apartamento, tipo de afectación y rotura) de la afectación seleccionada.</p>																									
<p>3-El Vicedecano de residencia y extensión modifica los datos necesarios.</p>	<p>4-El sistema verifica que no quede ningún campo vacío.</p> <p>5-Si las modificaciones de las afectaciones fueron correctas el Sistema guarda los cambios efectuados en la base de datos.</p>																									
<p>Prototipo interfaz</p>																										
																										

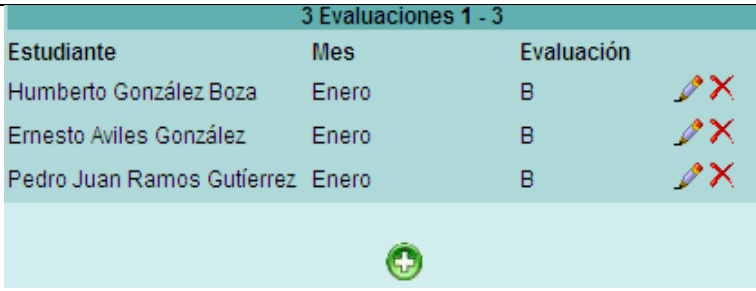
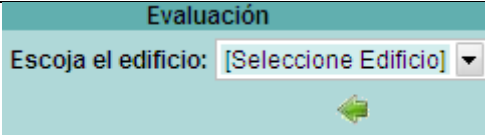
Sección “Eliminar afectaciones de los apartamentos”	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción “Eliminar afectaciones de los apartamentos”.	2- El sistema muestra un mensaje de alerta indicando si desea o no eliminar la afectación seleccionada.
3- El Vicedecano de residencia y extensión indica que desea eliminar la afectación.	4-El sistema elimina la afectación.
Sección “Generar reporte de las afectaciones de los apartamentos”	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y extensión selecciona la opción Generar reporte de las afectaciones de los apartamentos.	2- El Sistema muestra un formulario en el que debe escoger el tipo de afectación.
3- El Vicedecano de residencia y extensión selecciona el tipo de afectación.	4-El sistema muestra todas las afectaciones que existen del tipo seleccionado.
Prototipo Interfaz	
	
Flujos Alternos	
Sección “Insertar afectaciones de los apartamentos”	
Acción del Actor	Respuesta del Sistema
3.1 El Vicedecano de residencia y extensión al insertar los datos deja campos vacíos en el formulario.	4.1-El sistema muestra un mensaje de error para cada campo indicando que debe introducir el dato que falta.
3.1 El Vicedecano de residencia y extensión al insertar los datos introduce algún campos incorrectamente.	5.1-El sistema muestra un mensaje de error para el campo indicando la forma correcta de introducir el dato.
Sección “Modificar afectaciones de los apartamentos”	
Acción del Actor	Respuesta del Sistema

3.1 El Vicedecano de residencia y extensión al insertar los datos deja campos vacíos en el formulario.	4.1-El sistema muestra un mensaje de error para cada campo indicando que debe introducir el dato que falta.
3.1 El Vicedecano de residencia y extensión al insertar los datos introduce algún campos incorrectamente.	5.1-El sistema muestra un mensaje de error para el campo indicando la forma correcta de introducir el dato.
Sección “Eliminar afectaciones de los apartamentos”	
Acción del Actor	Respuesta del Sistema
3.1- El Vicedecano de residencia y extensión indica que no desea eliminar la afectación.	4.1-El Sistema no elimina la afectación.
Prototipo interfaz	
Poscondiciones:	Quedan insertadas las afectaciones de los apartamentos.
Prioridad:	Crítico

Tabla 39 Descripción del CUS” Gestionar evaluación del estudiante”

Caso de Uso:	Gestionar evaluación del estudiante
Actores:	Instructora
Resumen:	El caso de uso comienza cuando la instructora decide Insertar evaluación del estudiante, Visualizar evaluación del estudiante, Modificar evaluación del estudiante, Eliminar evaluación del estudiante o Generar reporte de la evaluación del estudiante. Luego de que la instructora indique la evaluación a insertar, modificar o generar reporte el sistema realiza la opción seleccionada.
Referencia:	RF9, RF9.1, RF9.2, RF9.3, RF9.4
Precondiciones:	El Vicedecano de residencia y extensión debe estar autenticado en el Sistema.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

1-La instructora se dirige al menú principal y selecciona el submenú Gestionar evaluación del estudiante.	2-El sistema muestra varias opciones entre las que se encuentran: Insertar evaluación del estudiante, Visualizar evaluación del estudiante, Generar reporte de la evaluación del estudiante.
3-La instructora selecciona una de las opciones presentadas.	<p>4-Si selecciona “Insertar evaluación del estudiante” ver sección Insertar evaluación del estudiante.</p> <p>-Si selecciona “Visualizar evaluación del estudiante” ver sección Visualizar evaluación del estudiante.</p> <p>-Si selecciona “Generar reporte de evaluación del estudiante” ver sección Generar reporte de la evaluación del estudiante</p>
Sección “Insertar evaluación del estudiante”	
Acción del Actor	Respuesta del Sistema
1- La instructora selecciona la opción “Insertar evaluación del estudiante”.	2-El sistema muestra un formulario en el que pide insertar (edificio, nombre completo del estudiante, mes y evaluación).
3- La instructora inserta los datos.	4-El sistema verifica que no haya campos vacíos y guarda la información en la base de datos.
Prototipo interfaz	
	
Sección “Visualizar evaluación del estudiante”	

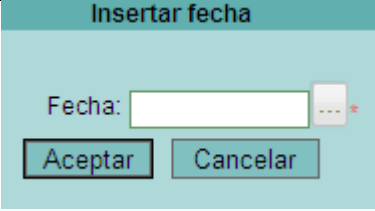
Acción del Actor	Respuesta del Sistema
1-La instructora selecciona la opción "Visualizar evaluación del estudiante".	2-El sistema un formulario para que se escoja el grupo del cual se quiere la evaluación.
3 -La instructora escoge el grupo.	4–El sistema muestra todos los estudiantes del grupo seleccionado con sus evaluaciones respectivas.
Prototipo interfaz	
	
Sección "Generar reporte de la evaluación del estudiante"	
Acción del Actor	Respuesta del Sistema
1- La instructora selecciona la opción "Generar reporte de evaluación del estudiante".	2- El sistema muestra un formulario en el que pide seleccionar el edificio a la que le va a generar reporte.
3- La instructora selecciona el edificio.	4-El sistema muestra todos los estudiantes del edificio y sus evaluaciones hasta el momento.
Prototipo interfaz	
	
Flujos Alternos	
Sección "Insertar evaluación del estudiante"	
Acción del Actor	Respuesta del Sistema
3.1 La instructora al insertar los datos deja campos vacíos en el formulario.	4.1-El sistema muestra un mensaje de error para cada campo indicando que debe introducir el dato que falta.

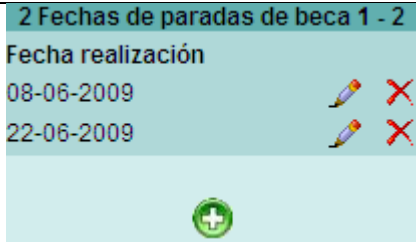

Prototipo interfaz	
Poscondiciones:	Quedan insertadas las evaluaciones de los estudiantes.
Prioridad:	Secundario

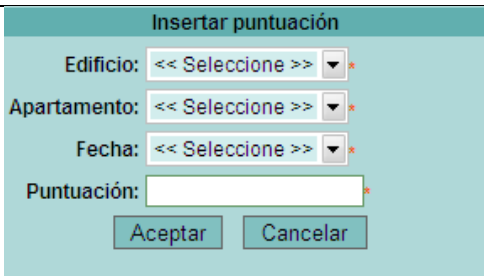
Tabla 40 Descripción del CUS” Gestionar parada de beca”





Caso de Uso:	Gestionar paradas de beca	
Actores:	Vicedecano de residencia y extensión	
Resumen:	El caso de uso comienza cuando el vicedecano de residencia y extensión decide Insertar, Modificar, Visualizar y Eliminar las fechas en que se efectuarán las paradas de beca así como las puntuaciones de los apartamentos y edificios en dichas paradas de beca, además permite generar reporte de la puntuación de las paradas de beca tanto para los edificios como para los apartamentos, luego de que el vicedecano de residencia y extensión indique la fecha o puntuación a insertar, modificar, visualizar, eliminar o generar reporte el Sistema realiza la opción seleccionada.	
Referencia:	RF10, RF10.1, RF10.2, RF10.3, RF10.4, RF10.5, RF10.6, RF10.7	
Precondiciones:	El Vicedecano de residencia y extensión debe estar autenticado en el Sistema.	
Flujo Normal de Eventos		
Acción del Actor		
Respuesta del Sistema		
1-El Vicedecano de residencia y extensión se dirige al menú principal y selecciona el submenú Gestionar paradas de beca.	2-El sistema muestra varias opciones entre las que se encuentran: Insertar fecha de las paradas de beca, Visualizar fechas de las paradas de beca, Insertar puntuación de las paradas de beca a los apartamentos, Visualizar puntuación de los apartamentos, Insertar puntuación de	


	<p>las paradas de beca a los edificios, Visualizar puntuación de los edificios, Generar reporte de la puntuación de las paradas de beca.</p>
<p>3- El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.</p>	<p>4-Si selecciona “Insertar fecha de las paradas de beca” ver sección Insertar fechas de las paradas de beca.</p> <p>-Si selecciona “Visualizar fechas de las paradas de beca” ver sección Visualizar fechas de las paradas de beca.</p> <p>-Si selecciona “Insertar puntuación de las paradas de beca a los apartamentos” ver sección Insertar puntuación de las paradas de beca a los apartamentos.</p> <p>-Si selecciona “Visualizar puntuación de los apartamentos” ver sección Visualizar puntuación de los apartamentos.</p> <p>-Si selecciona “Insertar puntuación de las paradas de beca a los edificios” ver sección Insertar puntuación de las paradas de beca a los edificios.</p> <p>-Si selecciona “Visualizar puntuación de los edificios” ver sección Visualizar puntuación de los edificios.</p> <p>- Si selecciona “Generar reporte de la puntuación de las paradas de beca” ver sección Generar reporte de la puntuación de las paradas de beca.</p>








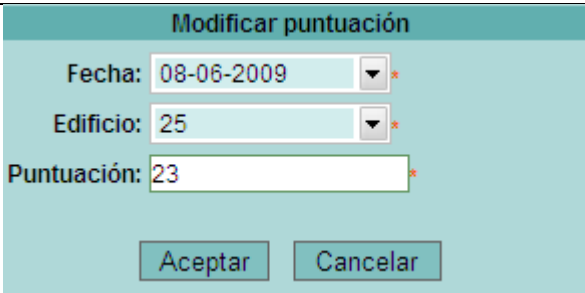
Sección "Insertar fechas de las paradas de beca"	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción "Insertar fechas de las paradas de beca".	2-El sistema muestra un formulario con un campo para insertar la fecha.
3- El Vicedecano de residencia y extensión inserta la fecha deseada.	4-El sistema verifica que el campo no esté vacío.
	5-El sistema guarda el dato en la base de datos.
Prototipo interfaz	
	
Sección "Visualizar fechas de las paradas de beca"	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción "Visualizar fechas de las paradas de beca".	2-El sistema muestra un listado de fechas que fueron insertadas hasta el momento con dos opciones a seleccionar para cada una: -Modificar fecha de la parada de beca - Eliminar fecha de la parada de beca
3- El Vicedecano de residencia y extensión selecciona una de las opciones presentadas.	4-- Si selecciona "Modificar fecha de las paradas de beca" ver sección Modificar fecha de las paradas de beca. - Si selecciona "Eliminar fecha de las paradas de beca" ver sección Eliminar fecha de las paradas de beca.
Prototipo interfaz	

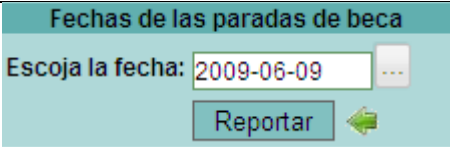
	
Sección “Modificar fecha de las paradas de beca”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Modificar fecha de las paradas de beca”.	2-El sistema muestra un formulario con un campo para modificar la fecha.
3- El Vicedecano de residencia y extensión modifica la fecha seleccionada.	4-Si las modificaciones fueron correctas el sistema las guarda en la base de datos.
Prototipo interfaz	
	
Sección “Eliminar fecha de las paradas de beca”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Eliminar fecha de las paradas de beca”.	2-El sistema muestra un mensaje de alerta indicando si desea o no eliminar la fecha seleccionada.
3-El Vicedecano de residencia y extensión indica que desea eliminar la fecha seleccionada.	4-El sistema elimina la fecha.
Sección “Insertar puntuación de las paradas de beca a los apartamentos”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Insertar puntuación de las paradas de beca a los apartamentos”.	2-El sistema muestra un formulario con los campos para insertar (número del apartamento, fecha de la parada de beca y la puntuación correspondiente a la parada de beca).

3-El Vicedecano de residencia y extensión inserta los datos correspondientes.	4-El sistema verifica que no haya campos vacíos.
	6-El sistema guarda los datos en la base de datos.
Prototipo interfaz	
	
Sección “Visualizar puntuación de los apartamentos”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Visualizar puntuación de los apartamentos”.	2-El sistema muestra un listado de las puntuaciones de los apartamentos insertadas hasta el momento con dos opciones a seleccionar para cada una: -Modificar puntuación de los apartamentos -Eliminar puntuación de los apartamentos
3-El Vicedecano de residencia y extensión selecciona la opción “Modificar puntuación de los apartamentos”.	4-Si selecciona “Modificar puntuación de los apartamentos” ver sección Modificar puntuación de los apartamentos . -Si selecciona “Eliminar puntuación de los apartamentos” ver sección Eliminar puntuación de los apartamentos .
Prototipo interfaz	

1 Puntuaciones 1 - 1	
Fecha	Apartamento Puntuación
08-06-2009	25105 12  
	
Sección “Modificar puntuación de los apartamentos”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Modificar puntuación de los apartamentos”.	2-El sistema muestra un formulario con los campos a modificar (número del apartamento, fecha de la parada de beca y la puntuación correspondiente a la parada de beca).
3-El Vicedecano de residencia y extensión modifica los datos necesarios.	4- Si las modificaciones fueron correctas el sistema las guarda en la base de datos.
Prototipo interfaz	
	
Sección “Eliminar puntuación de los apartamentos”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Eliminar puntuación de los apartamentos”.	2-El sistema muestra un mensaje de alerta indicando si desea o no eliminar la puntuación seleccionada.
3- El Vicedecano de residencia y extensión indica que desea eliminar la puntuación seleccionada.	4-El sistema elimina la puntuación seleccionada.
Sección “Insertar puntuación de las paradas de beca a los edificios”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Insertar	2-El sistema muestra un formulario con los campos para insertar (número del

puntuación de las paradas de beca a los edificios”.	edificio, fecha de la parada de beca y la puntuación correspondiente a la parada de beca).
3-El Vicedecano de residencia y extensión inserta los datos correspondientes.	4-El sistema verifica que no haya campos vacíos.
	6-El sistema guarda los datos en la base de datos.
Prototipo interfaz	
	
Sección “Visualizar puntuación de los edificios”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Visualizar puntuación de los edificios”.	2-El sistema muestra un listado de las puntuaciones de los edificios insertadas hasta el momento con dos opciones a seleccionar para cada una: -Modificar puntuación de los edificios -Eliminar puntuación de los edificios
3-El Vicedecano de residencia y extensión selecciona la opción “Modificar puntuación de los edificios”.	4-Si selecciona “Modificar puntuación de los edificios” ver sección Modificar puntuación de los edificios. -Si selecciona “Eliminar puntuación de los edificios” ver sección Eliminar puntuación de los edificios.
Prototipo interfaz	

3 Puntuaciones 1 - 3	
Fecha	Edificio Puntuación
08-06-2009	25 23  
08-06-2009	87 10  
22-06-2009	25 34  
	
Sección “Modificar puntuación de los edificios”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Modificar puntuación de los edificios”.	2- El sistema muestra un formulario con los campos a modificar (número del edificio, fecha de la parada de beca y la puntuación correspondiente a la parada de beca).
3-El Vicedecano de residencia y extensión modifica los datos necesarios.	4- Si las modificaciones fueron correctas el sistema las guarda en la base de datos.
Prototipo interfaz	
	
Sección “Eliminar puntuación de los edificios”	
Acción del Actor	Respuesta del Sistema
1-El Vicedecano de residencia y extensión selecciona la opción “Eliminar puntuación de los edificios”.	2- El sistema muestra un mensaje de alerta indicando si desea o no eliminar la puntuación seleccionada.
3- El Vicedecano de residencia y extensión indica que desea eliminar la puntuación seleccionada.	4-El sistema elimina la puntuación seleccionada.
Sección “Generar reporte de la puntuación de las paradas de beca”	
Acción del Actor	Respuesta del Sistema
1- El Vicedecano de residencia y	2-El sistema muestra un formulario con un

extensión selecciona la opción “Generar reporte de la puntuación de la parada de beca”.	campo para seleccionar la fecha de la parada de beca a la cual desea hacerle el reporte.
3- El Vicedecano de residencia y extensión selecciona la fecha deseada.	4-El sistema muestra un formulario que muestra todos los apartamentos y edificios con las puntuaciones de la parada de beca que correspondan con la fecha seleccionada
Prototipo interfaz	
	
Flujos Alternos	
Sección “Insertar fechas de las paradas de beca”	
Acción del Actor	Respuesta del Sistema
3.1 El Vicedecano de residencia y extensión al insertar la fecha deja algún campo vacío.	4.1- El sistema muestra un mensaje de error para el campo indicando que debe introducir el dato que falta.
Sección “Modificar fecha de las paradas de beca”	
Acción del Actor	Respuesta del Sistema
3.1 El Vicedecano de residencia y extensión al modificar la fecha deja algún campo vacío.	4.1-El sistema muestra un mensaje de error para el campo indicando que debe introducir el dato que falta.
Sección “Eliminar fecha de las paradas de beca”	
Acción del Actor	Respuesta del Sistema
3.1-La Vicedecano de Residencia y extensión indica que no desea eliminar la puntuación.	4.1-El sistema no elimina la evaluación.
Sección “Insertar puntuación de las paradas de beca a los apartamentos”	
Acción del Actor	Respuesta del Sistema
3.1 El Vicedecano de residencia y extensión al insertar la puntuación deja	4.1- El sistema muestra un mensaje de error para el campo indicando que debe

algún campo vacío.	introducir el dato que falta.
Sección “Modificar puntuación de las paradas de beca a los apartamentos”	
Acción del Actor	Respuesta del Sistema
3.1 El Vicedecano de residencia y extensión al modificar la puntuación deja algún campo vacío.	4.1-El sistema muestra un mensaje de error para el campo indicando que debe introducir el dato que falta.
Sección “Eliminar puntuación de las paradas de beca a los apartamentos”	
Acción del Actor	Respuesta del Sistema
3.1-La Vicedecano de Residencia y extensión indica que no desea eliminar la puntuación.	4.1-El sistema no elimina la evaluación.
Sección “Insertar puntuación de las paradas de beca a los edificios”	
Acción del Actor	Respuesta del Sistema
3.1 El Vicedecano de residencia y extensión al insertar la puntuación deja algún campo vacío.	4.1- El sistema muestra un mensaje de error para el campo indicando que debe introducir el dato que falta.
Sección “Modificar puntuación de las paradas de beca a los edificios”	
Acción del Actor	Respuesta del Sistema
3.1 El Vicedecano de residencia y extensión al modificar la puntuación deja algún campo vacío.	4.1-El sistema muestra un mensaje de error para el campo indicando que debe introducir el dato que falta.
Sección “Eliminar puntuación de las paradas de beca a los edificios”	
Acción del Actor	Respuesta del Sistema
3.1-La Vicedecano de Residencia y extensión indica que no desea eliminar la puntuación.	4.1-El sistema no elimina la evaluación.
Poscondiciones:	Quedan insertadas las puntuaciones de las paradas de beca.
Prioridad:	Secundario

Anexo.2-Diagrama de clases del diseño.

Diagrama de clases del diseño del CU: "Gestionar apartamento"

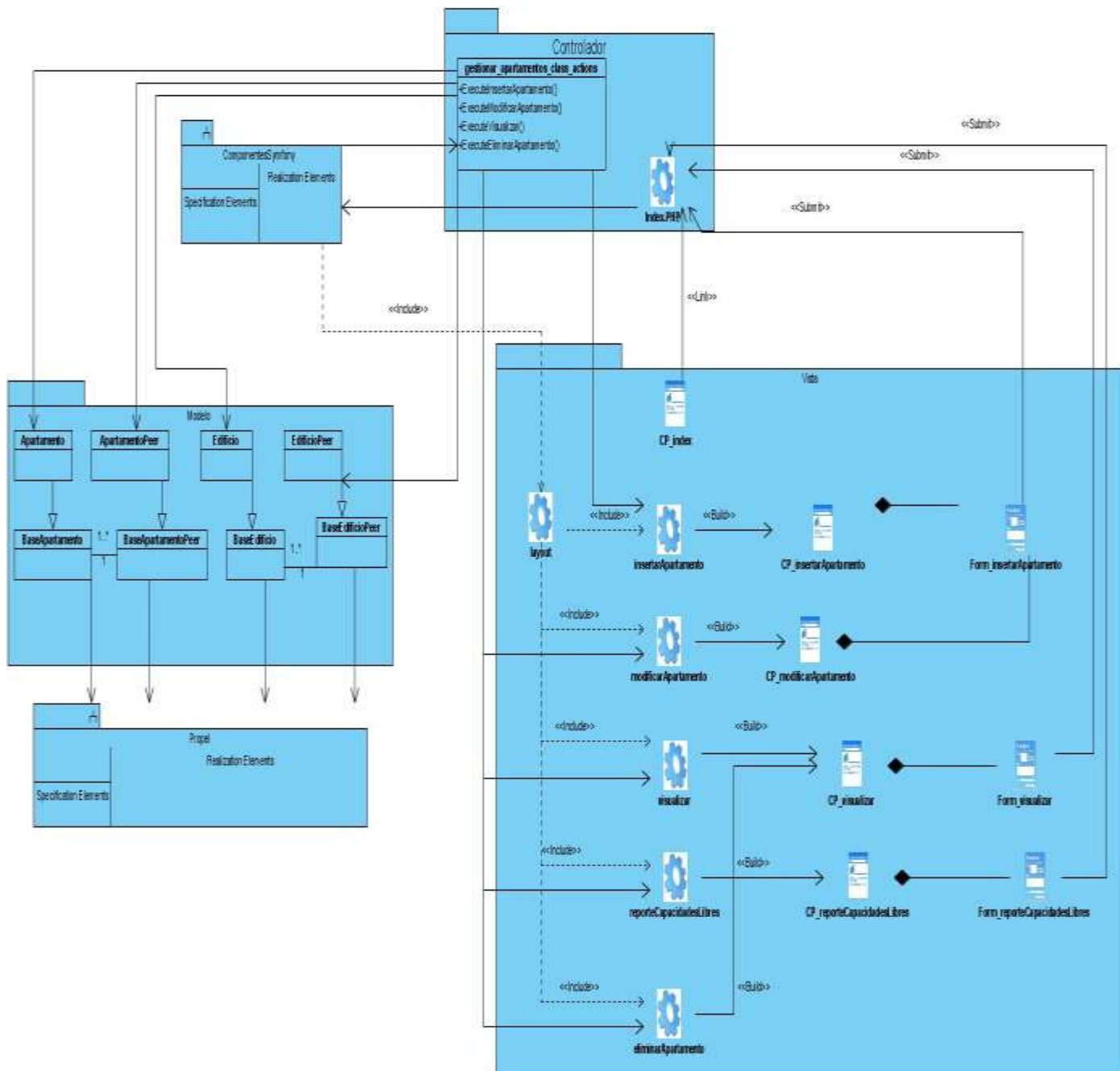


Diagrama de clases del diseño del CU: "Gestionar datos de la instructora"

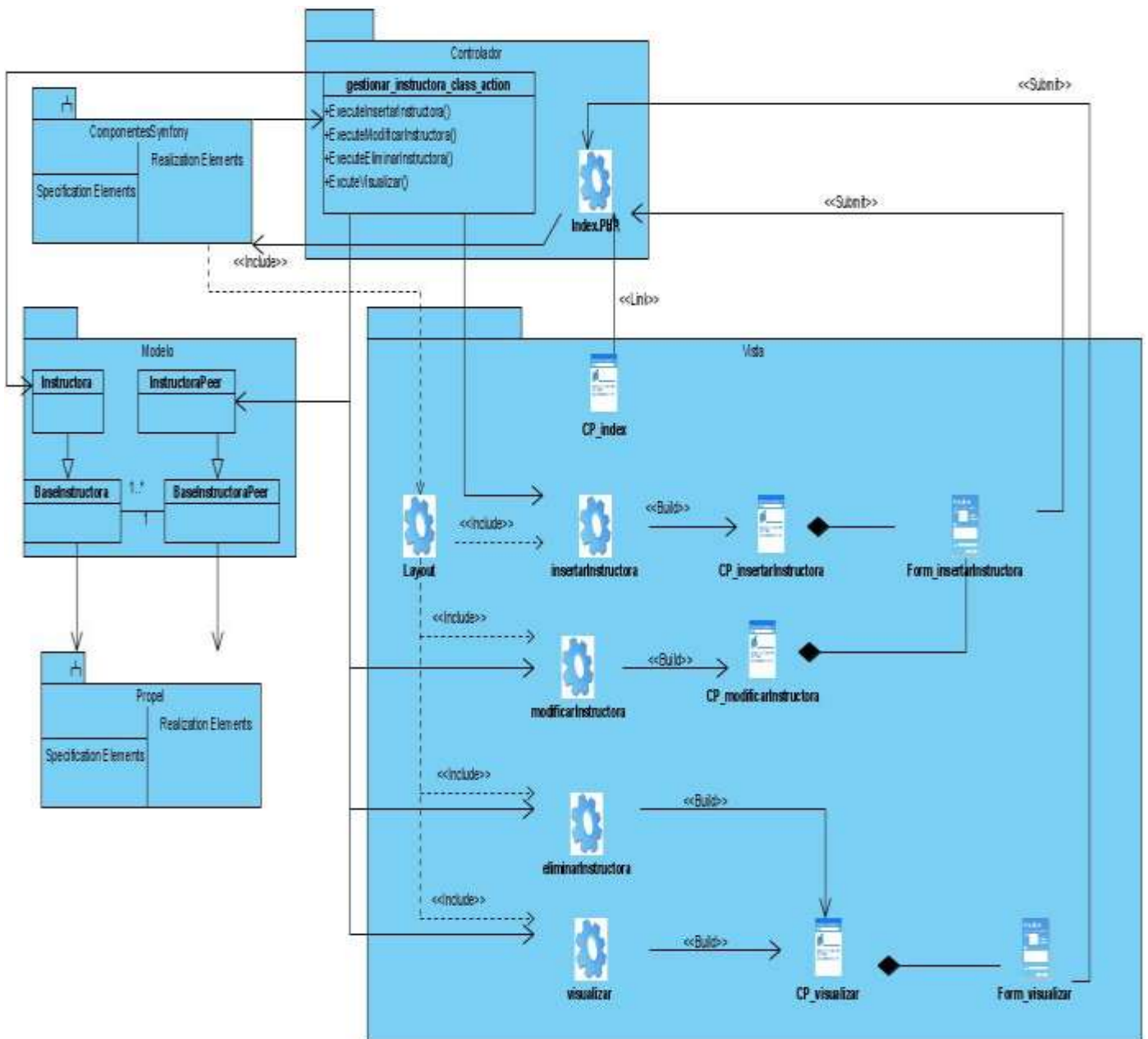


Diagrama de clases del diseño del CU: "Gestionar Cuarterería"

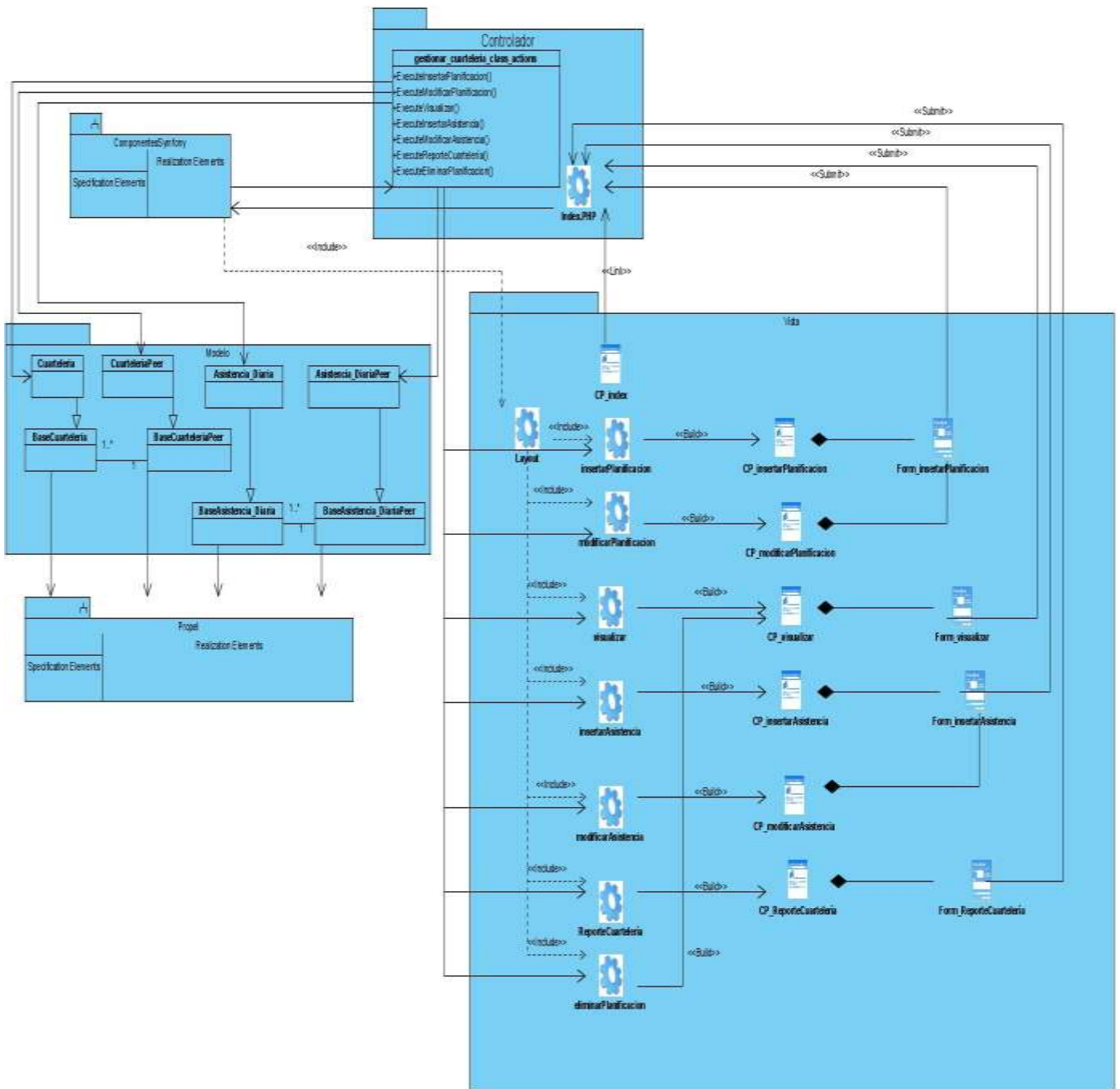


Diagrama de clases del diseño del CU: "Gestionar Guardia Obrera estudiantil"

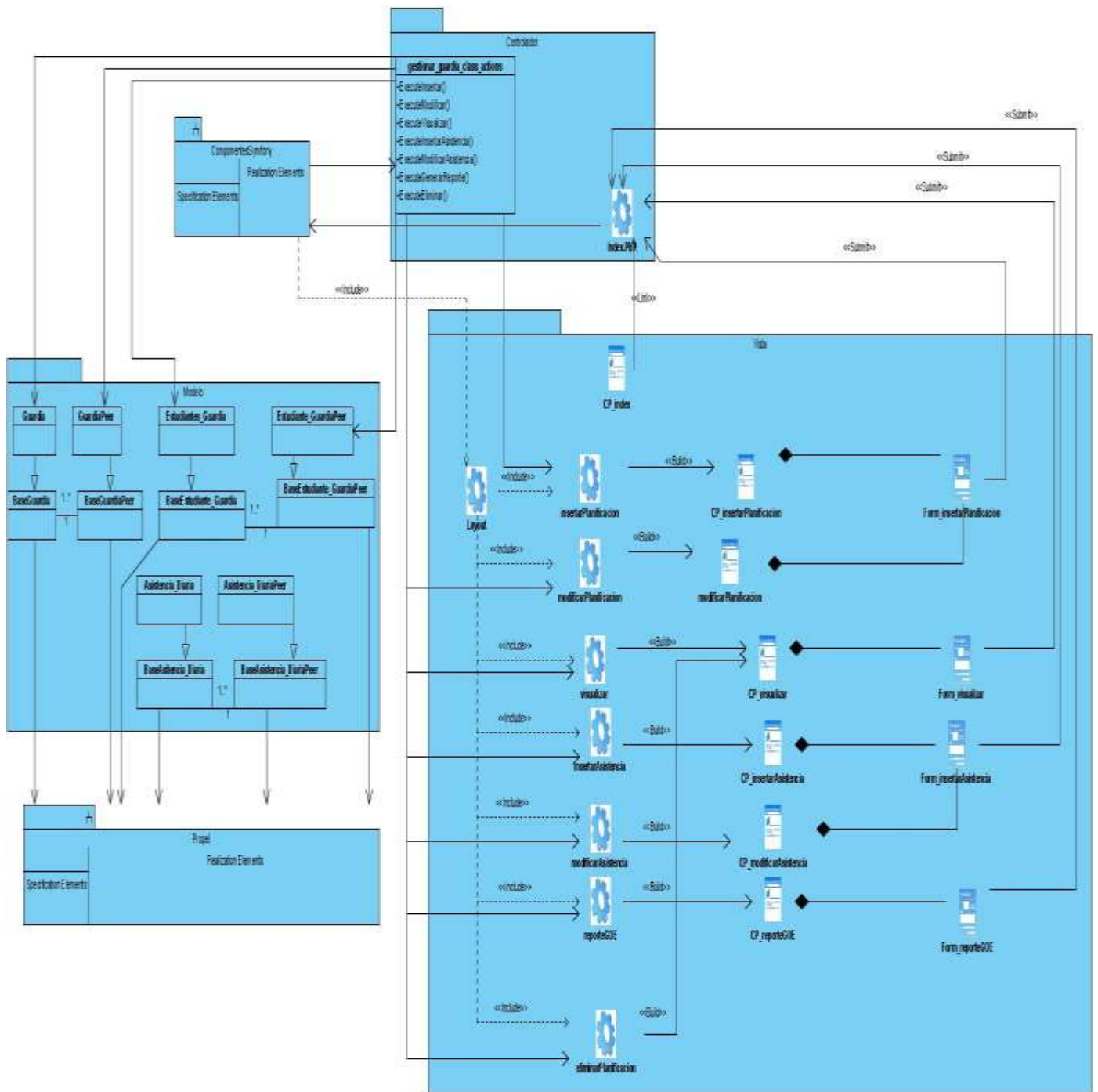


Diagrama de clases del diseño del CU: "Gestionar afectaciones de los apartamentos"

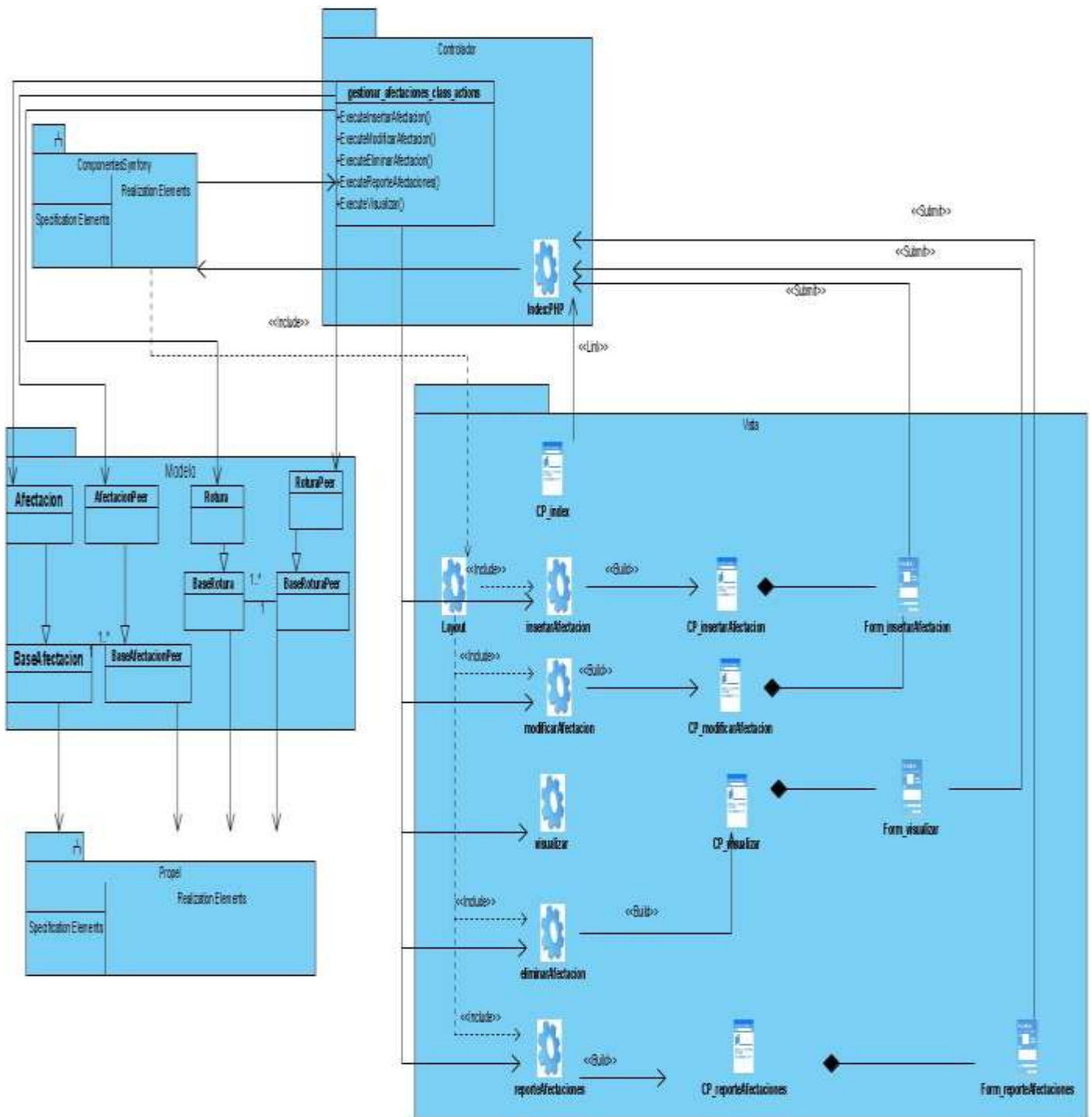


Diagrama de clases del diseño del CU: "Gestionar evaluación del estudiante"

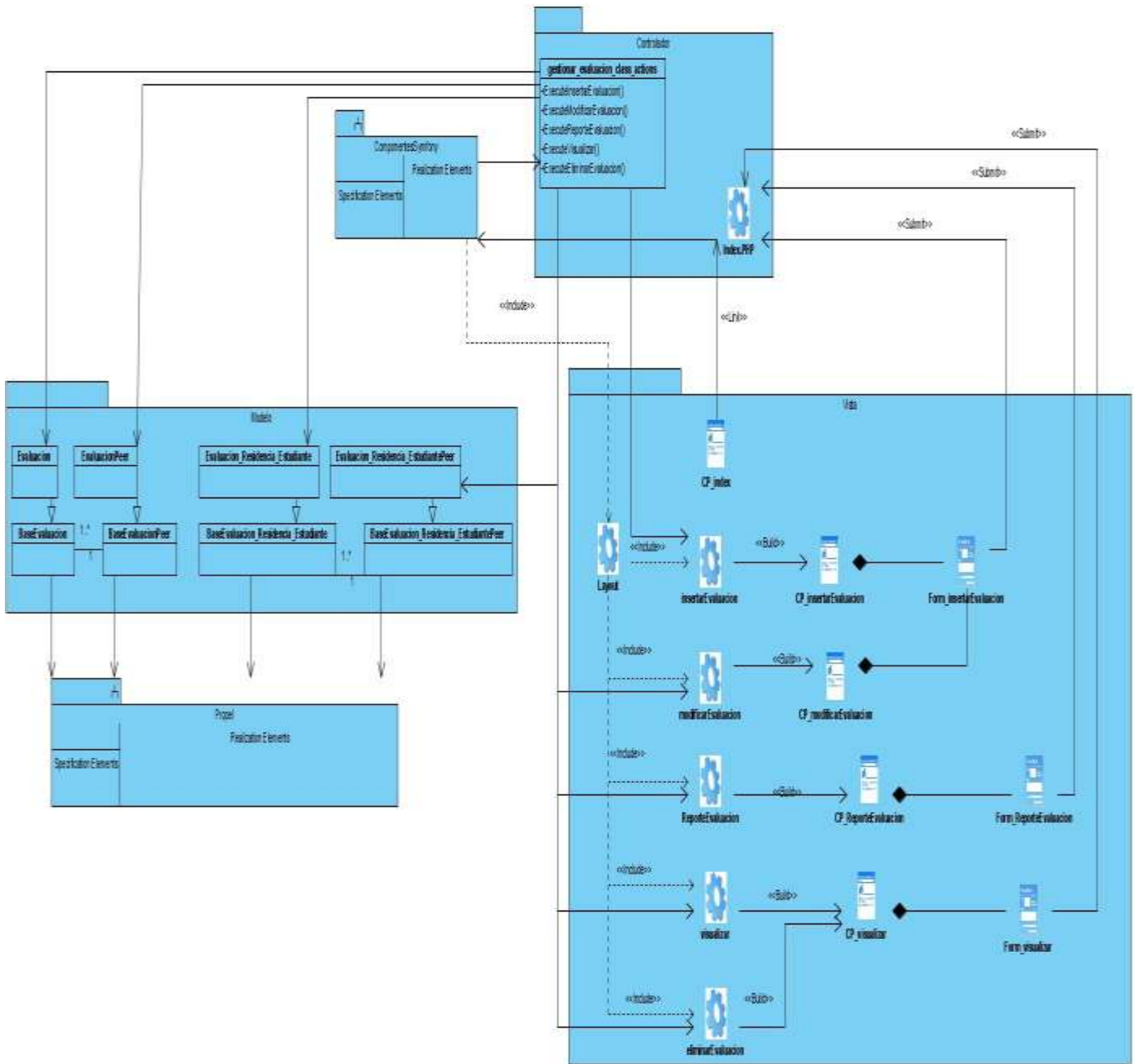
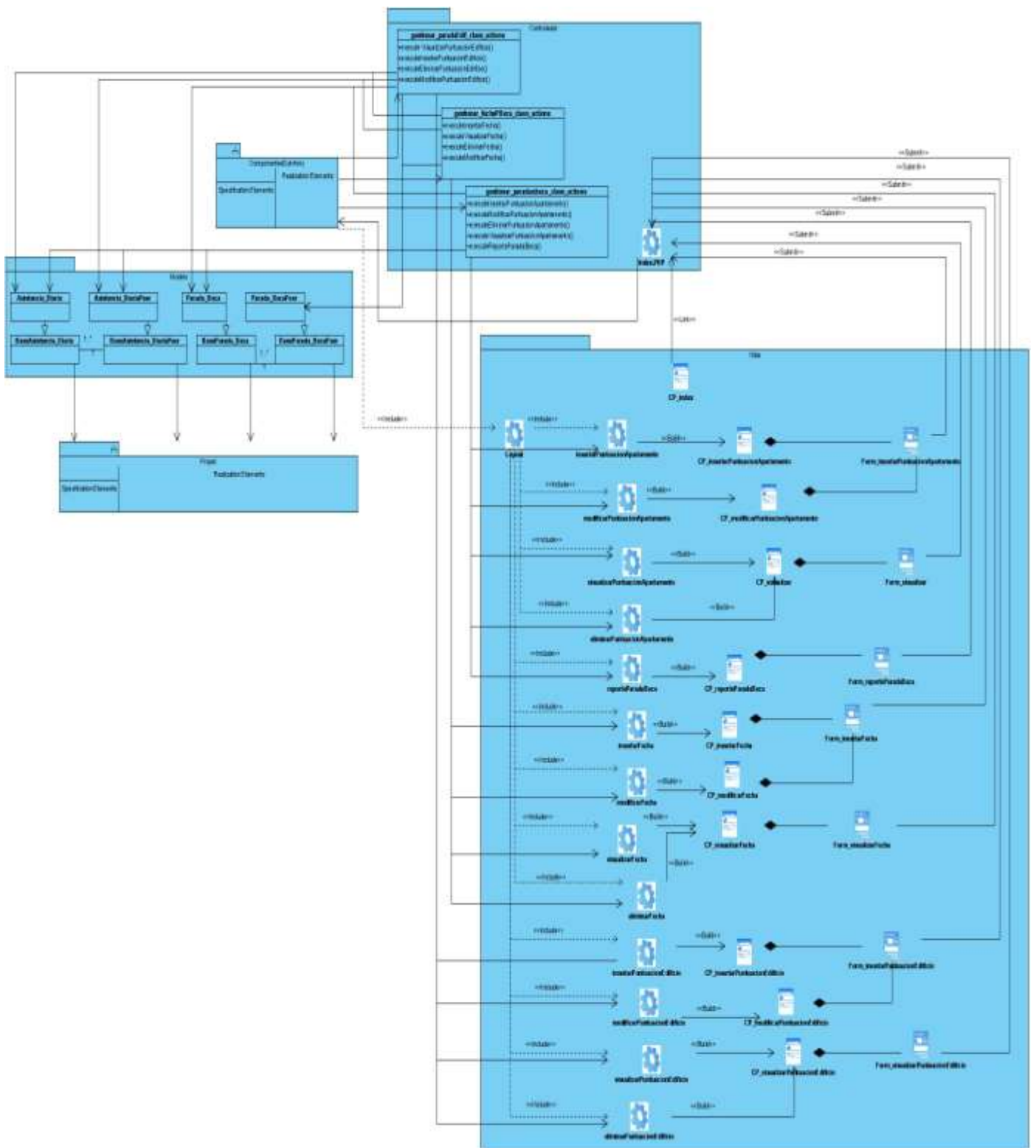


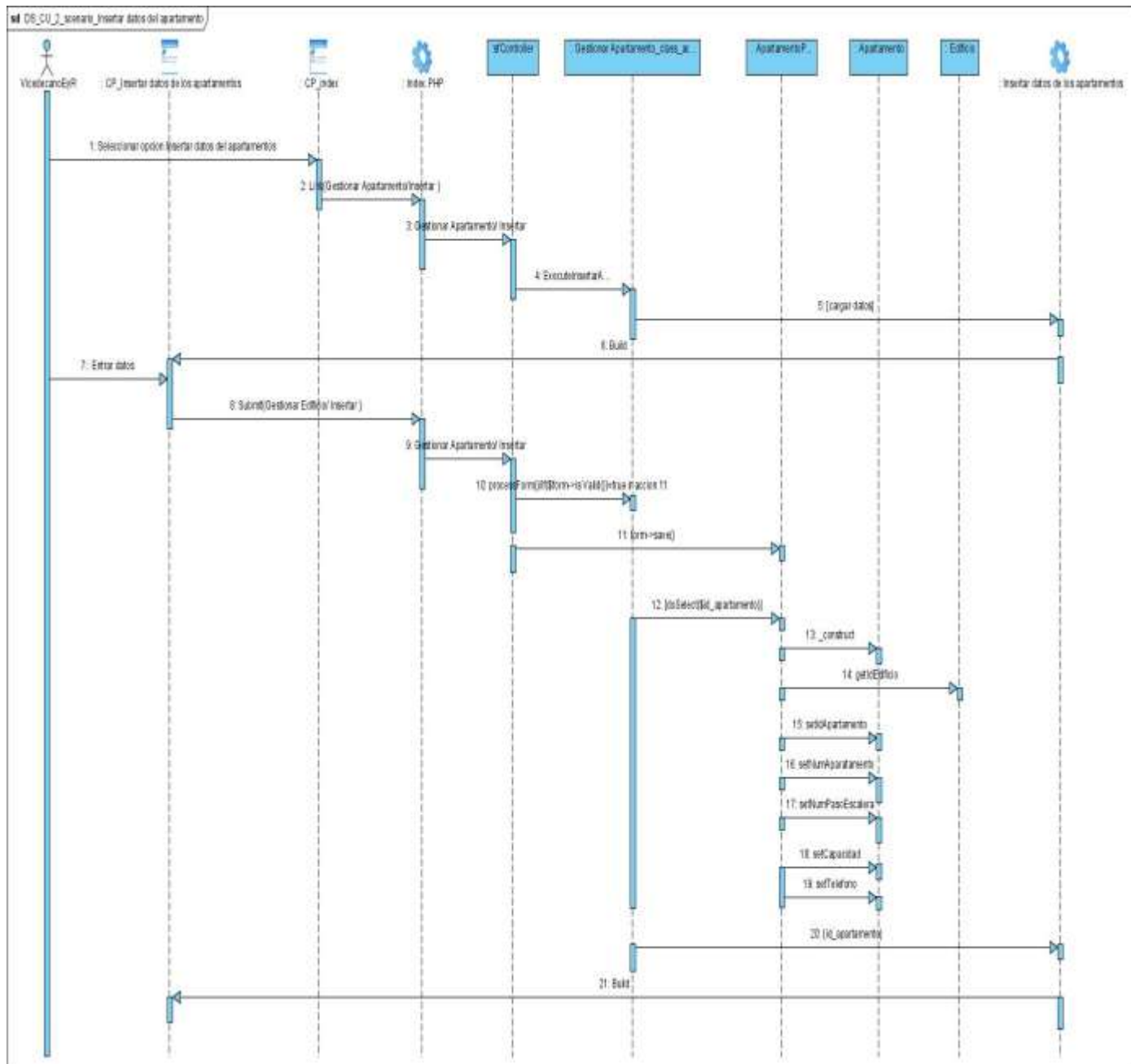
Diagrama de clases del diseño del CU: "Gestionar parada de beca"



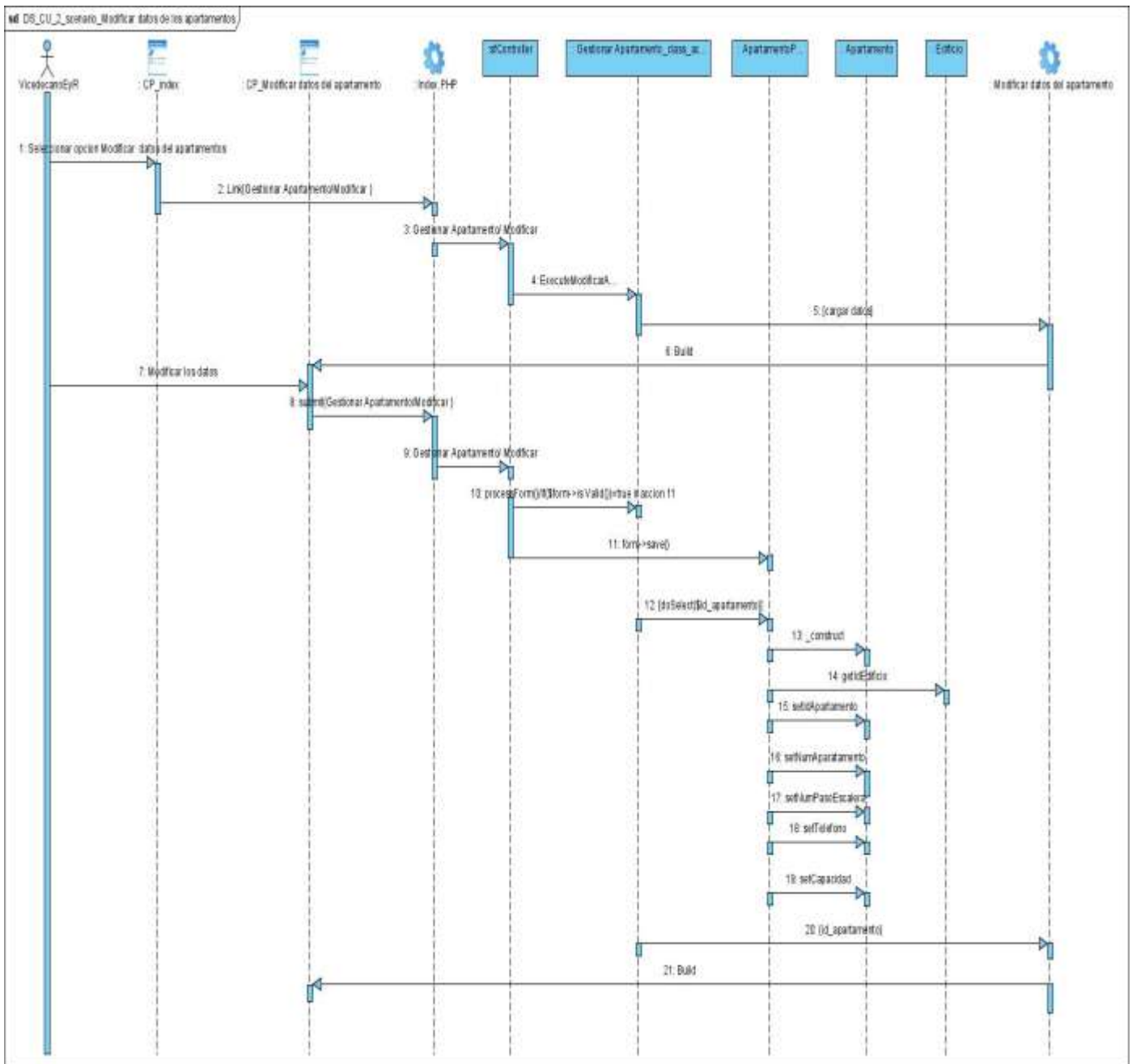
Anexo.3-Diagramas de secuencias

Diagramas de iteración CU “Gestionar apartamento”

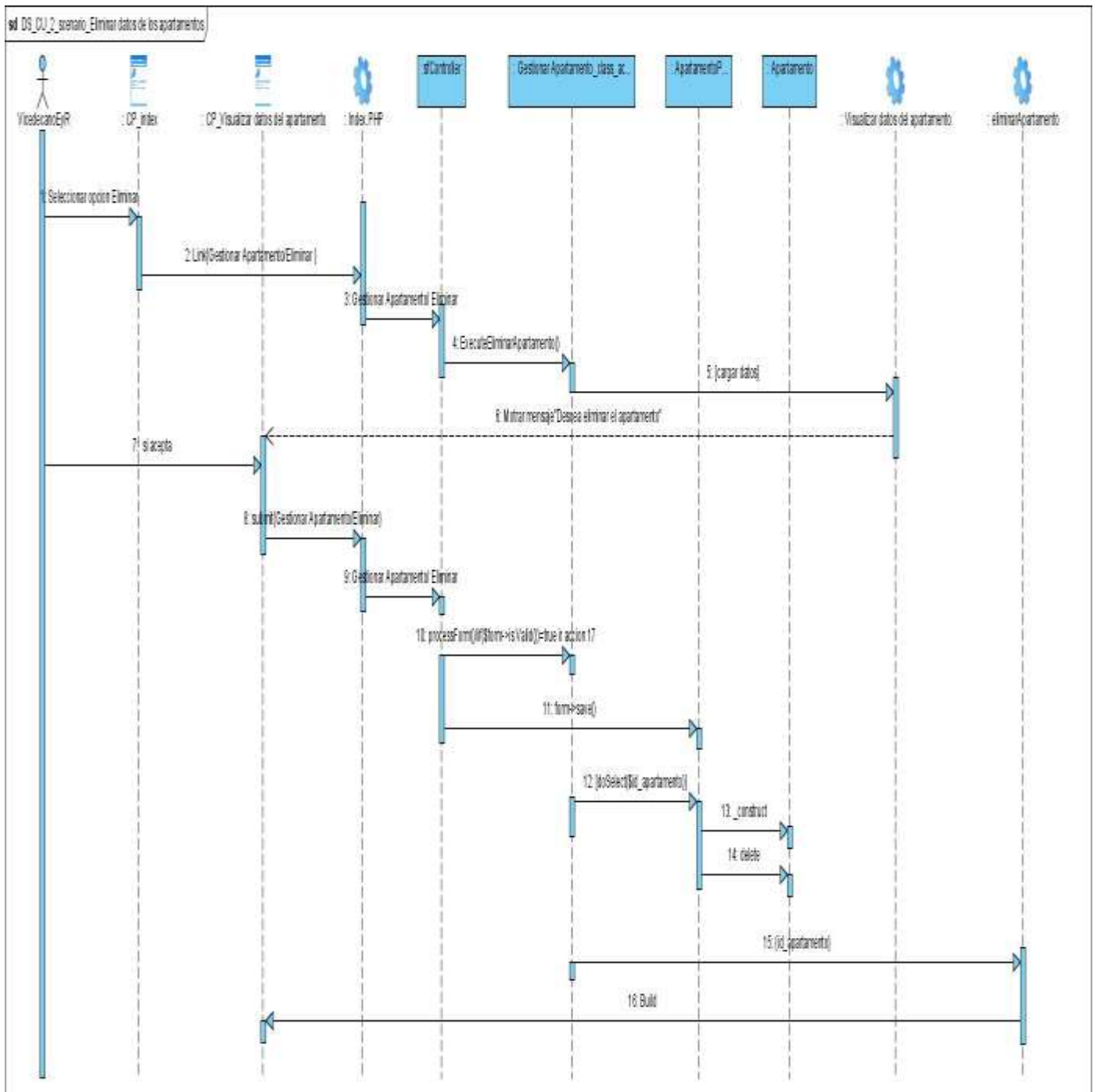
Escenario: “Insertar datos del apartamento”



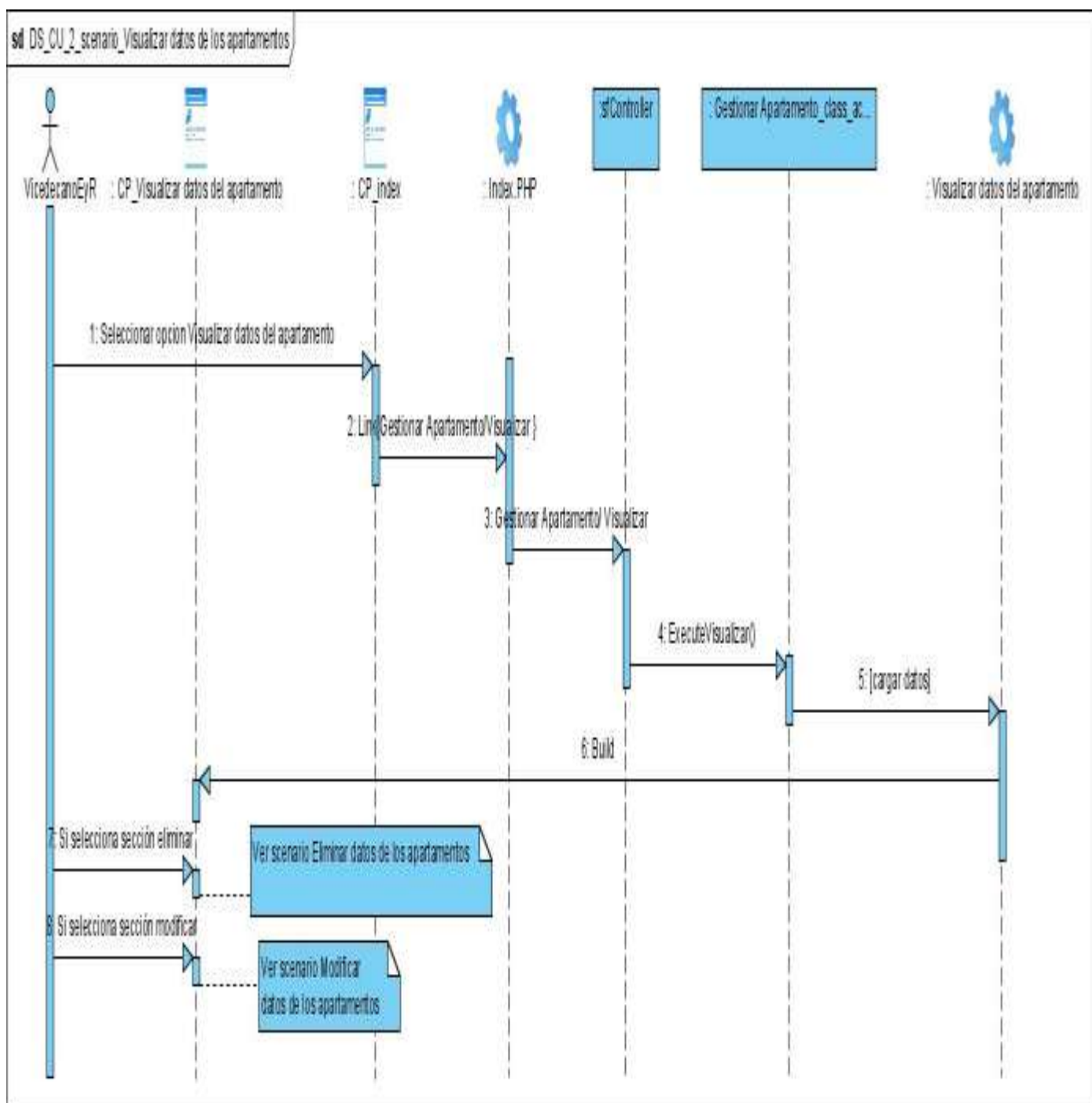
Escenario " Modificar datos del apartamento"



Escenario " Eliminar datos del apartamento "

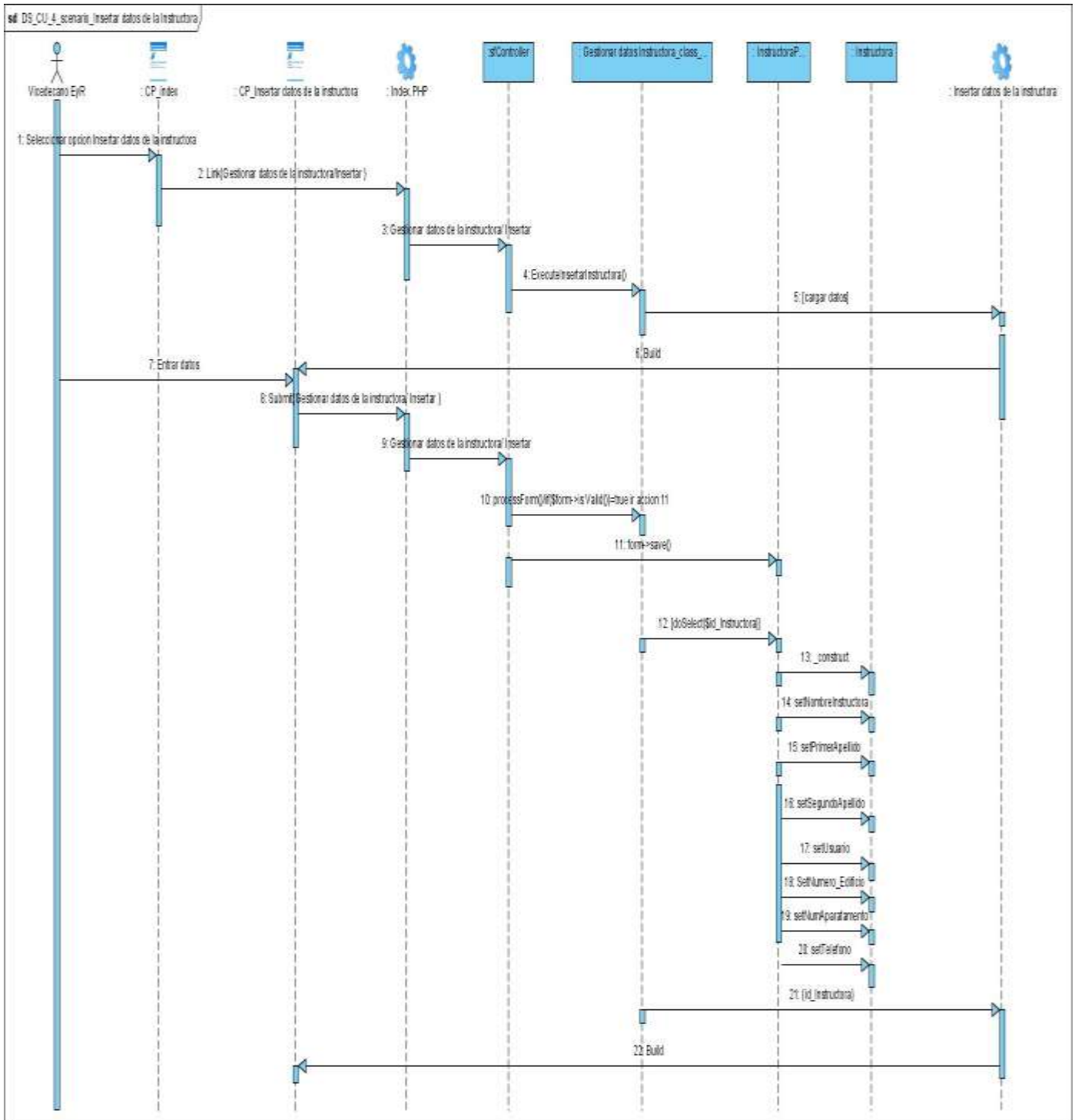


Escenario " Visualizar datos del apartamento "

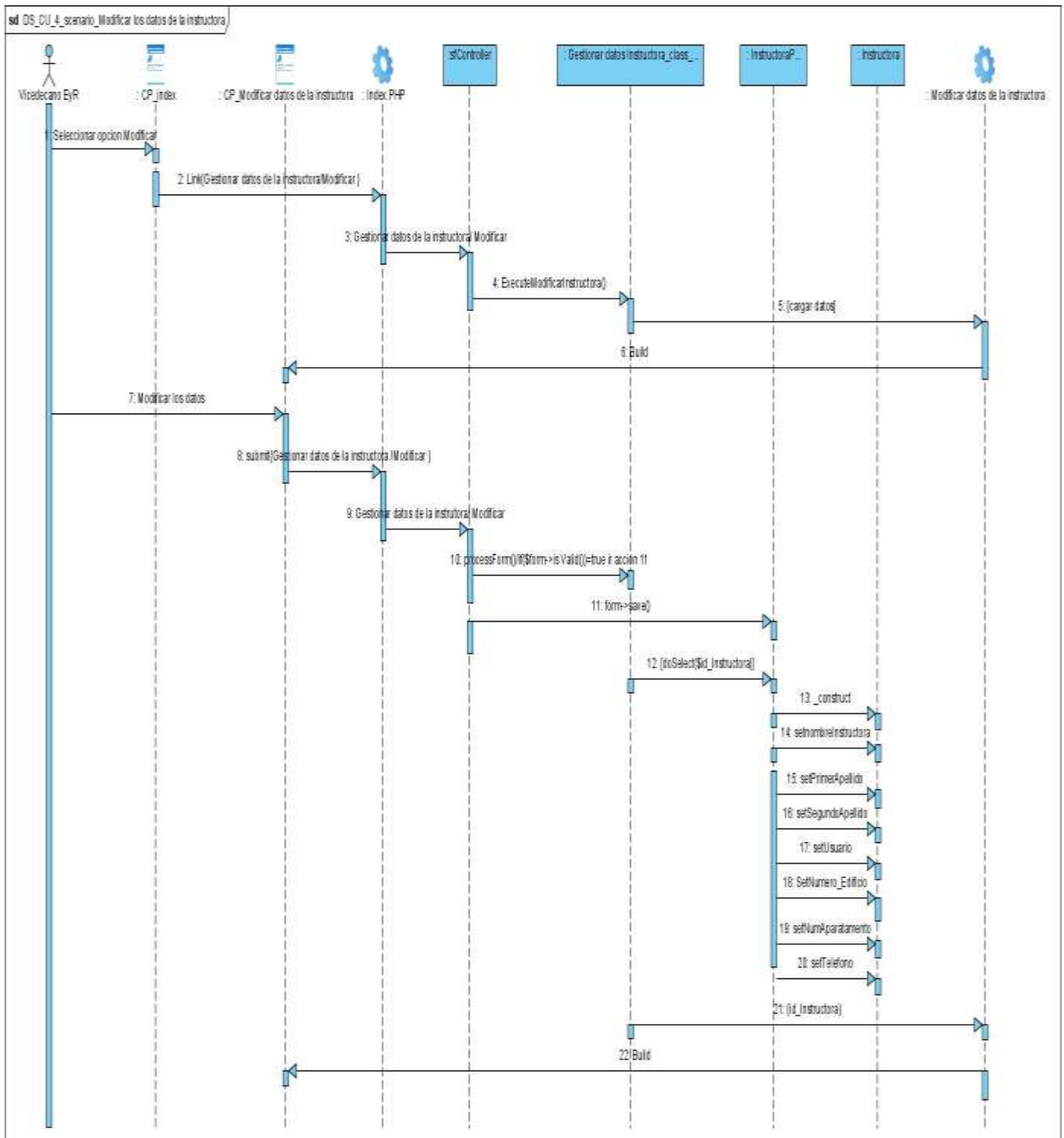


Diagramas de iteración CU “Gestionar datos de la instructora”

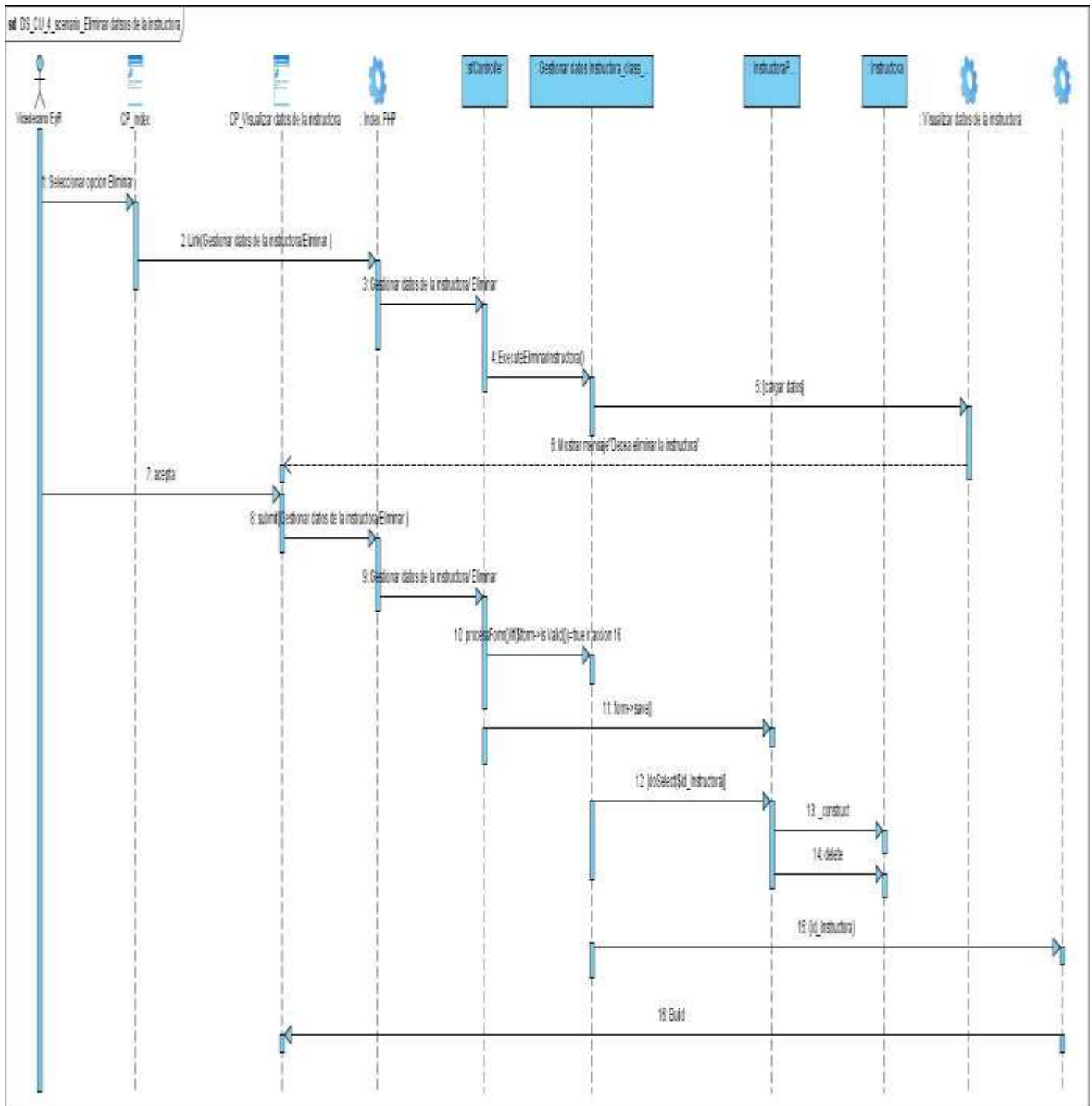
Escenario” Insertar datos de la instructora”



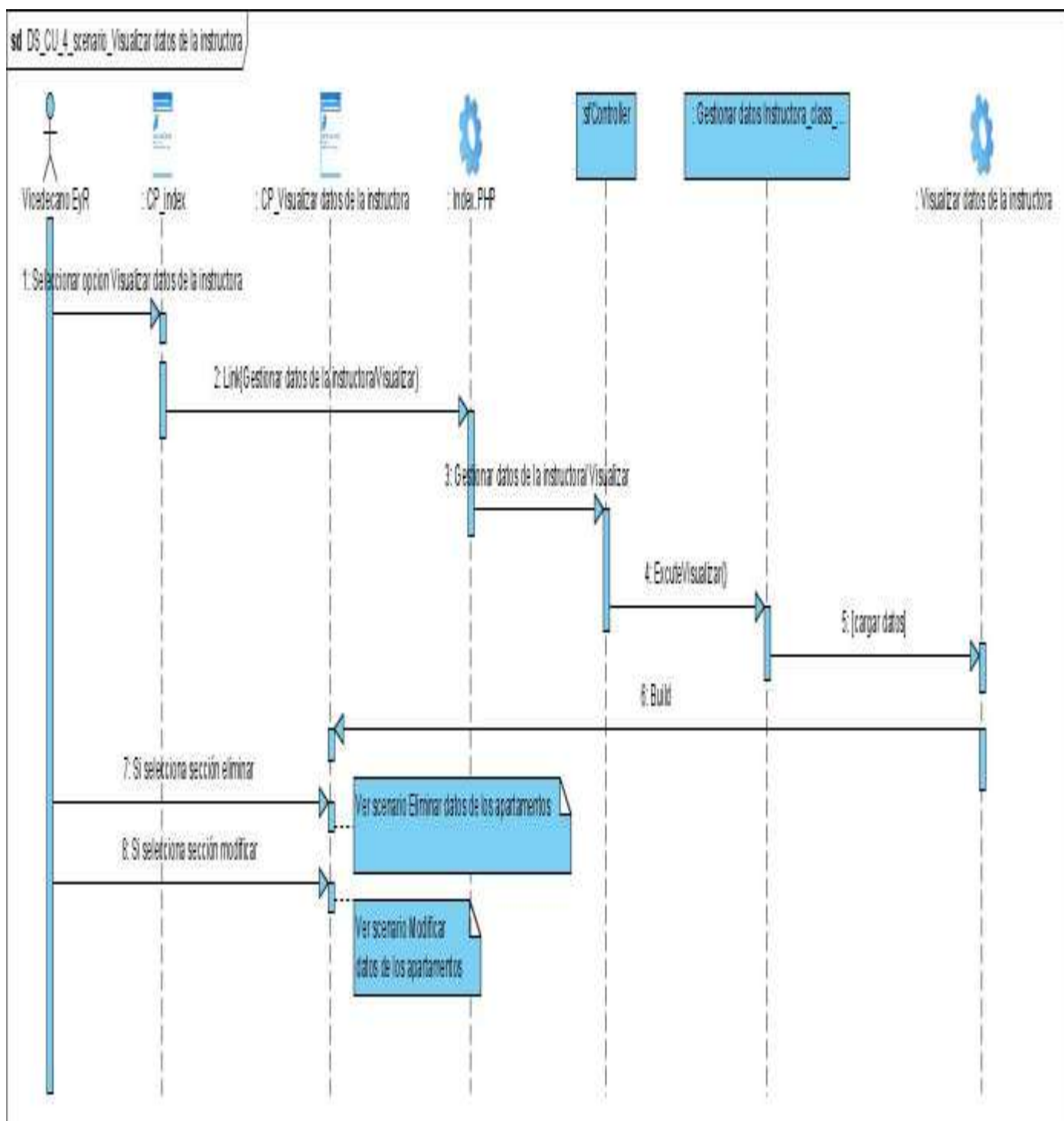
Escenario " Modificar datos de la instructora "



Escenario " Eliminar datos de la instructora "

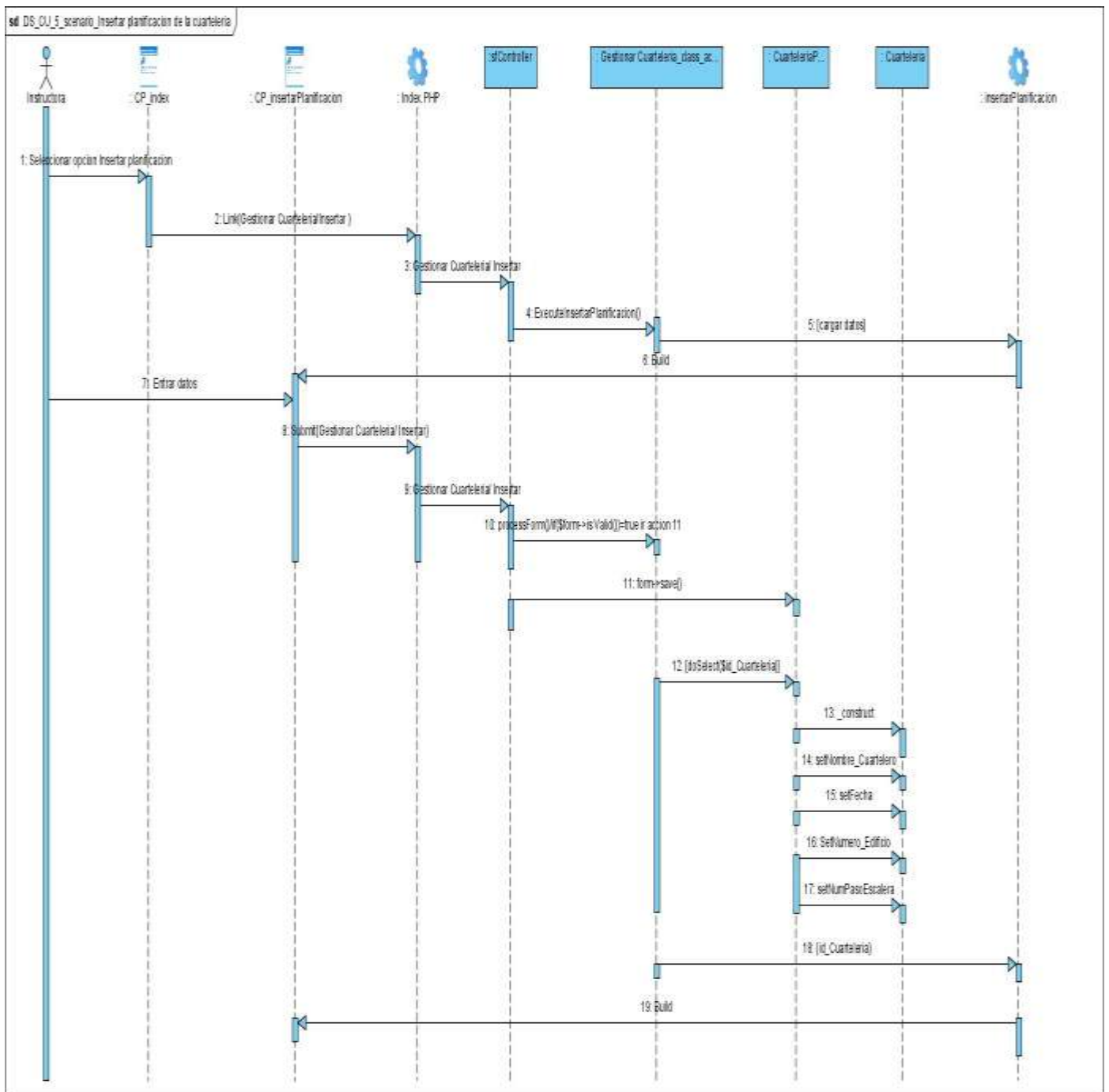


Escenario " Visualizar datos de la instructora "

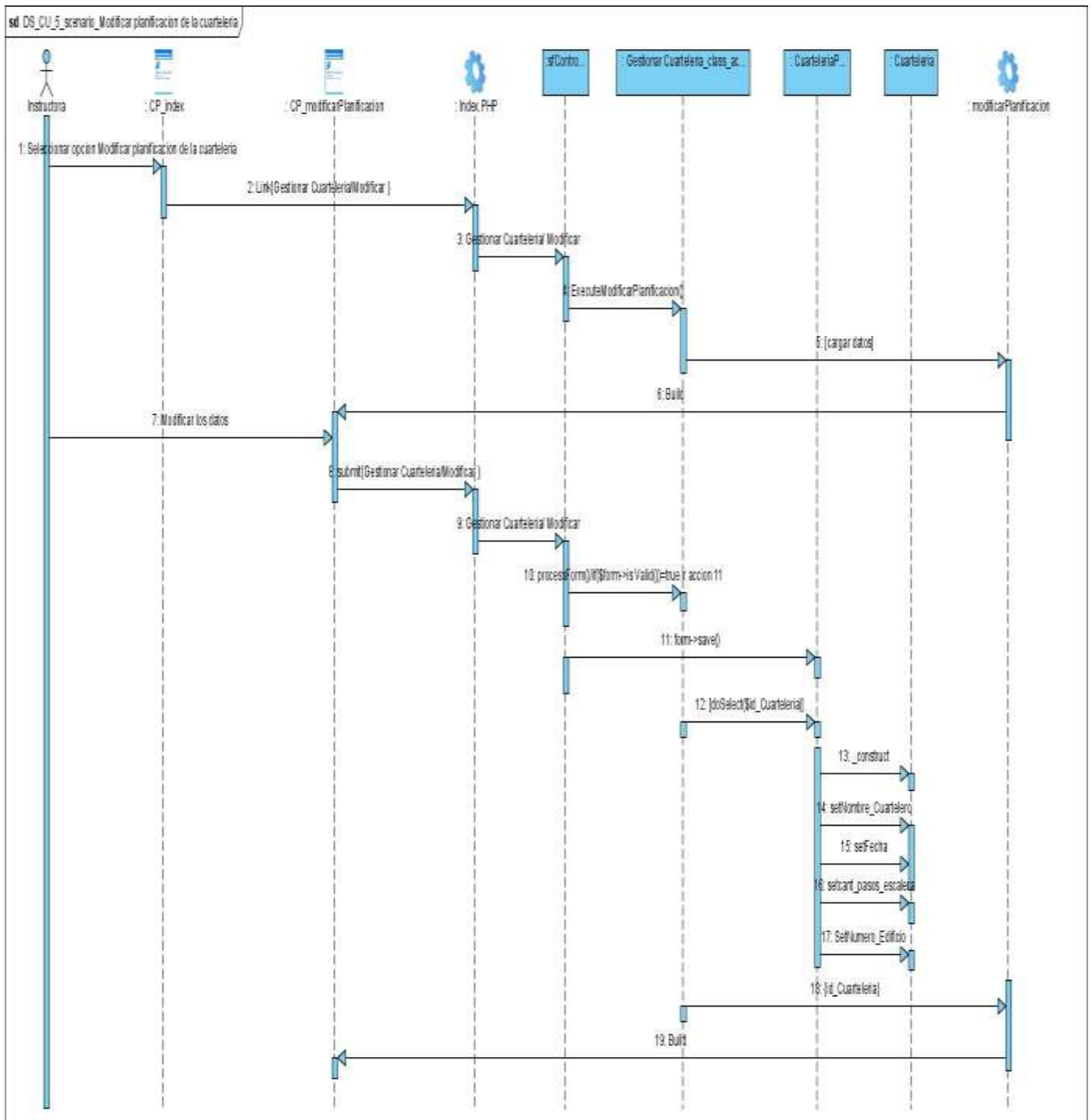


Diagramas de iteración CU “Gestionar Cuartelería”

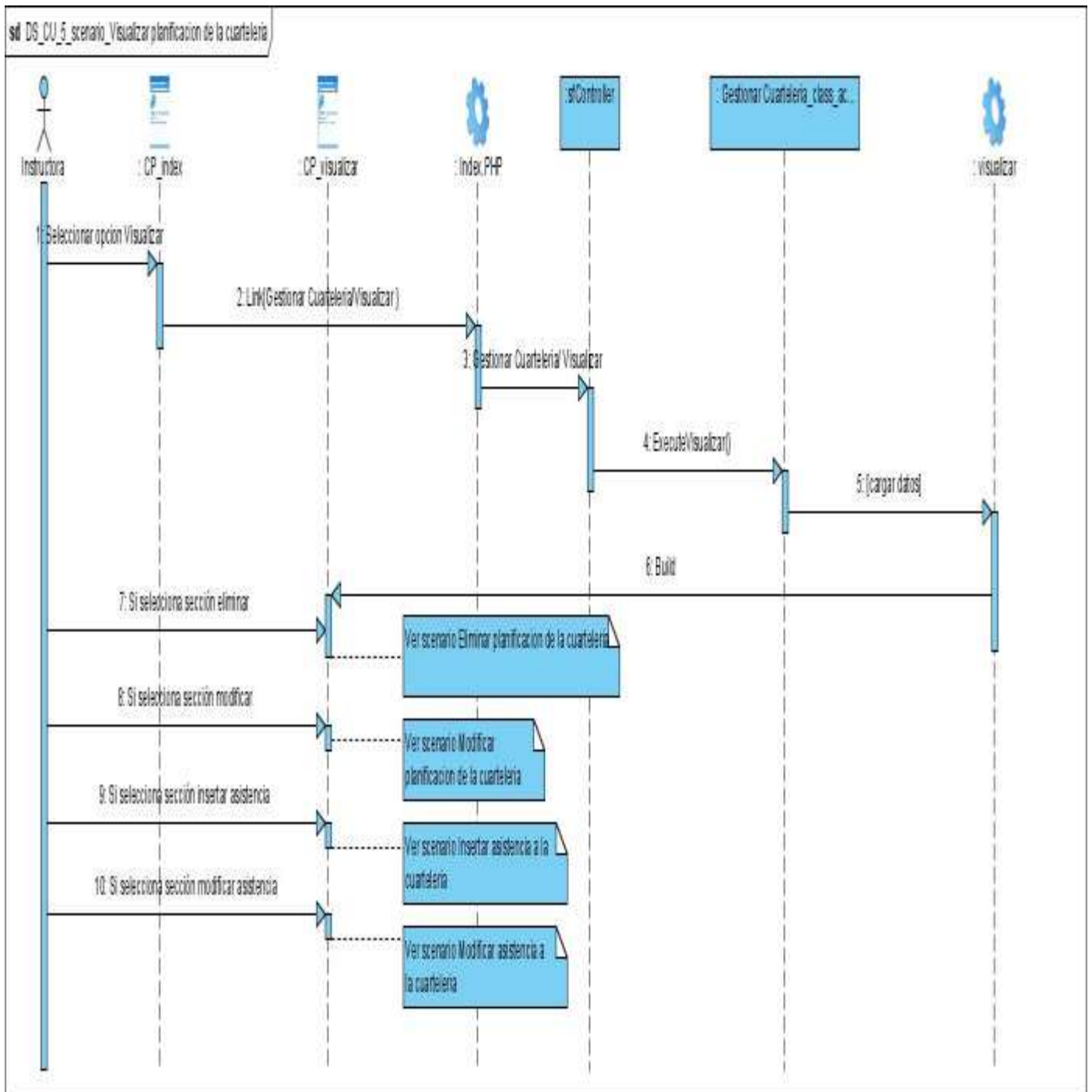
Escenario” Insertar planificación de la cuartelería”



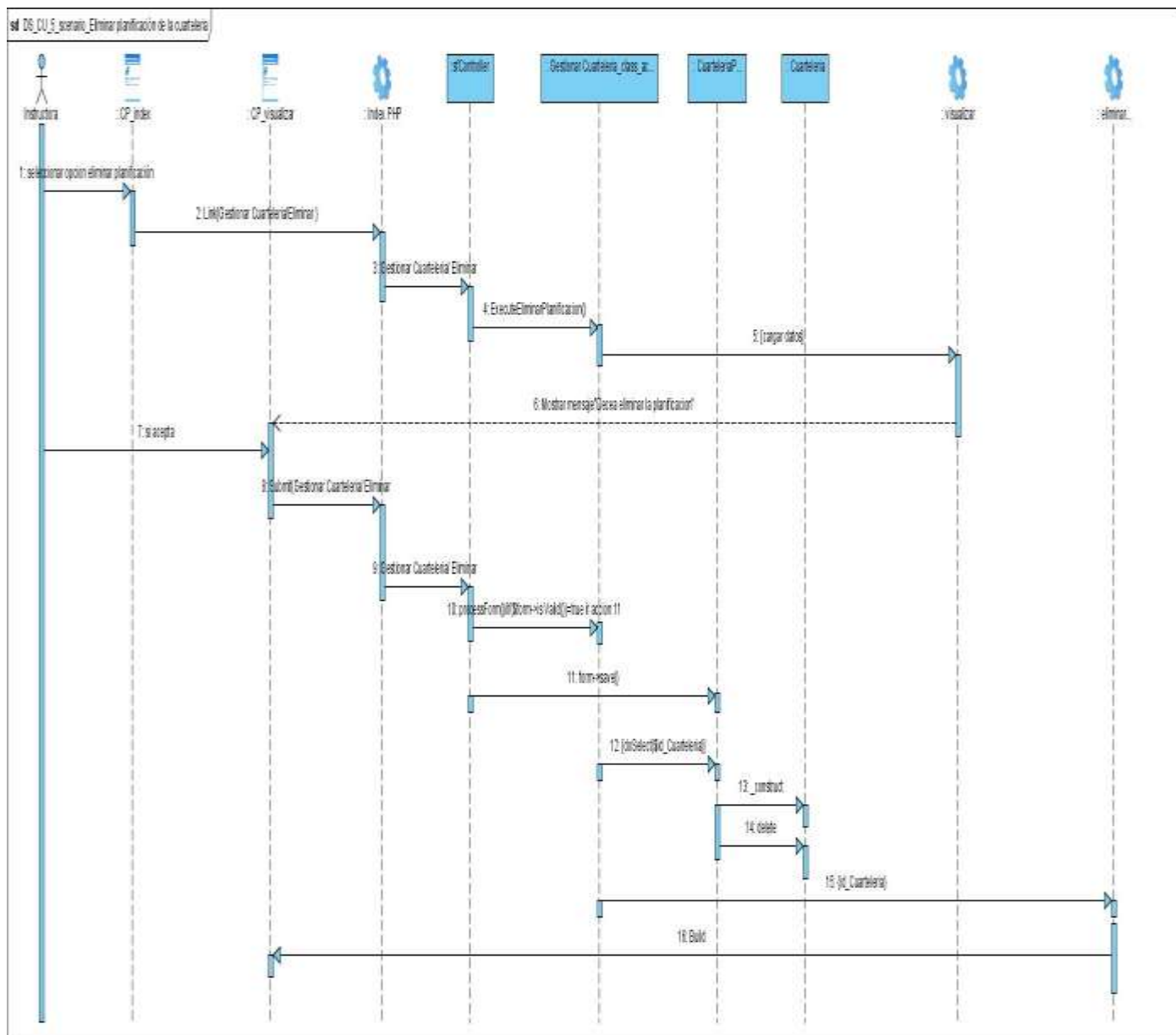
Escenario " Modificar planificación de la cuartelería "



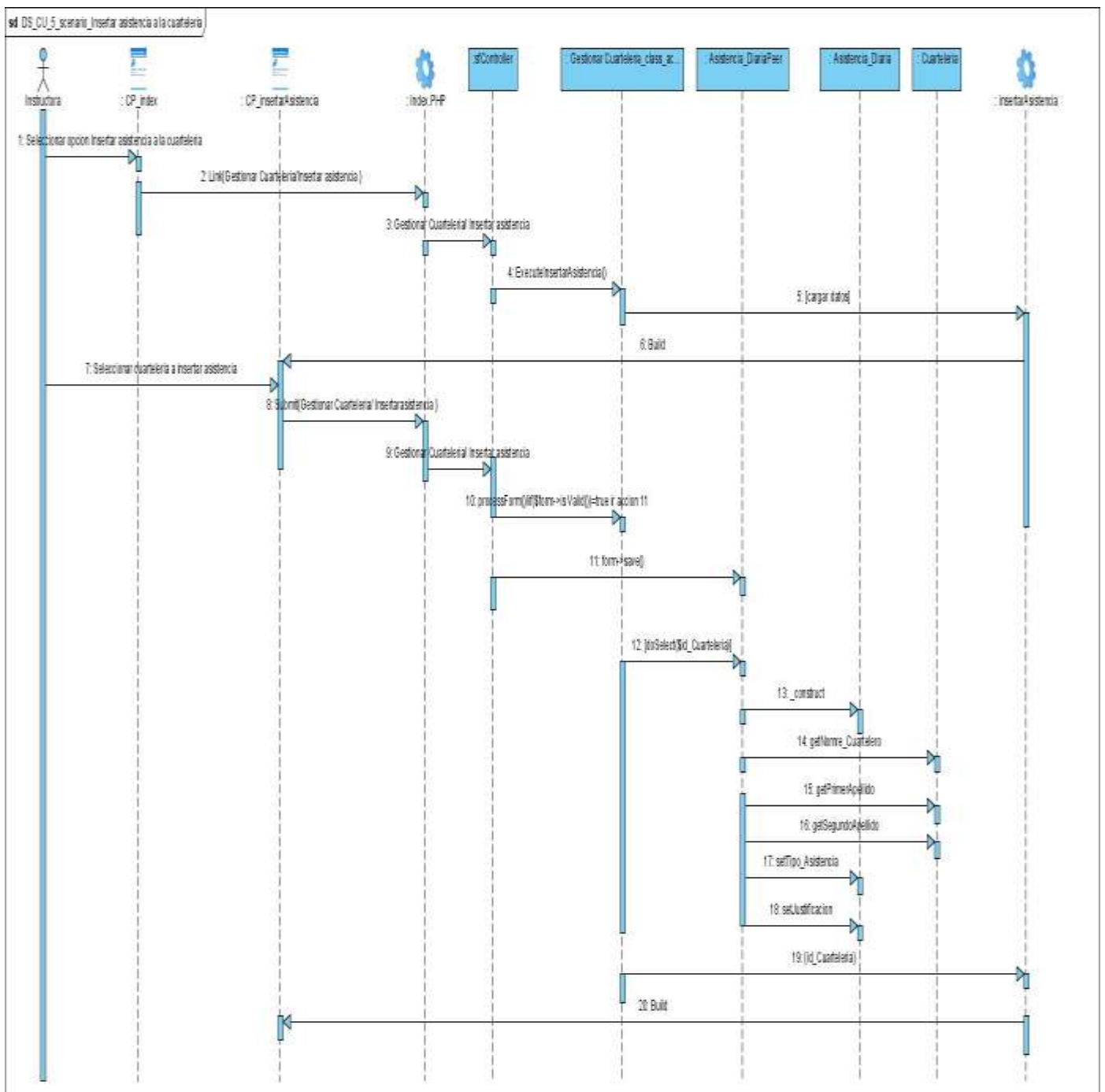
Escenario” Visualizar planificación de la cuartelería”



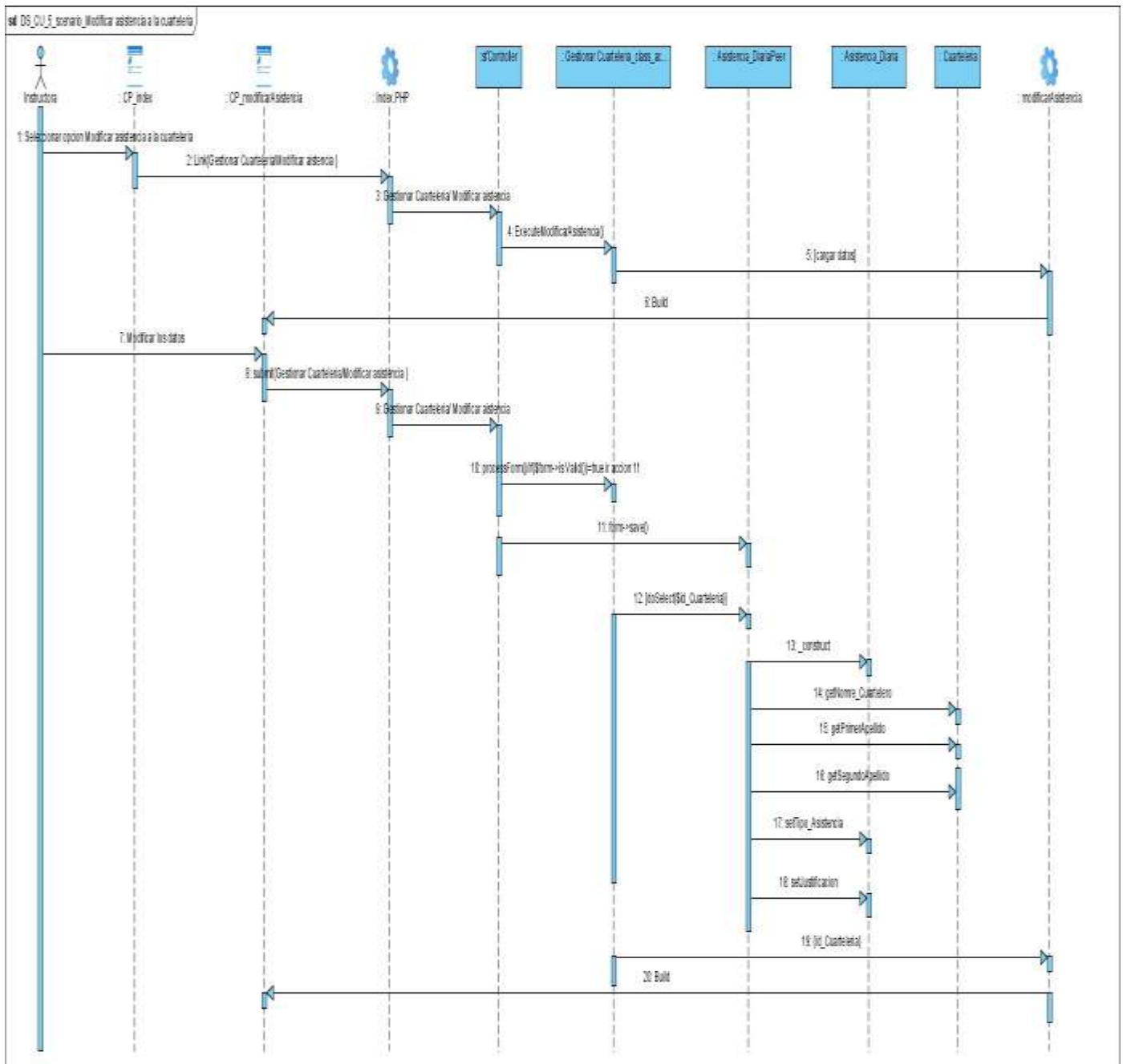
Escenario " Eliminar planificación de la cuartelería"



Escenario "Insertar asistencia a la cuartelería"



Escenario " Modificar asistencia a la cuartelería "



Escenario "Generar reporte de la cuartelería"

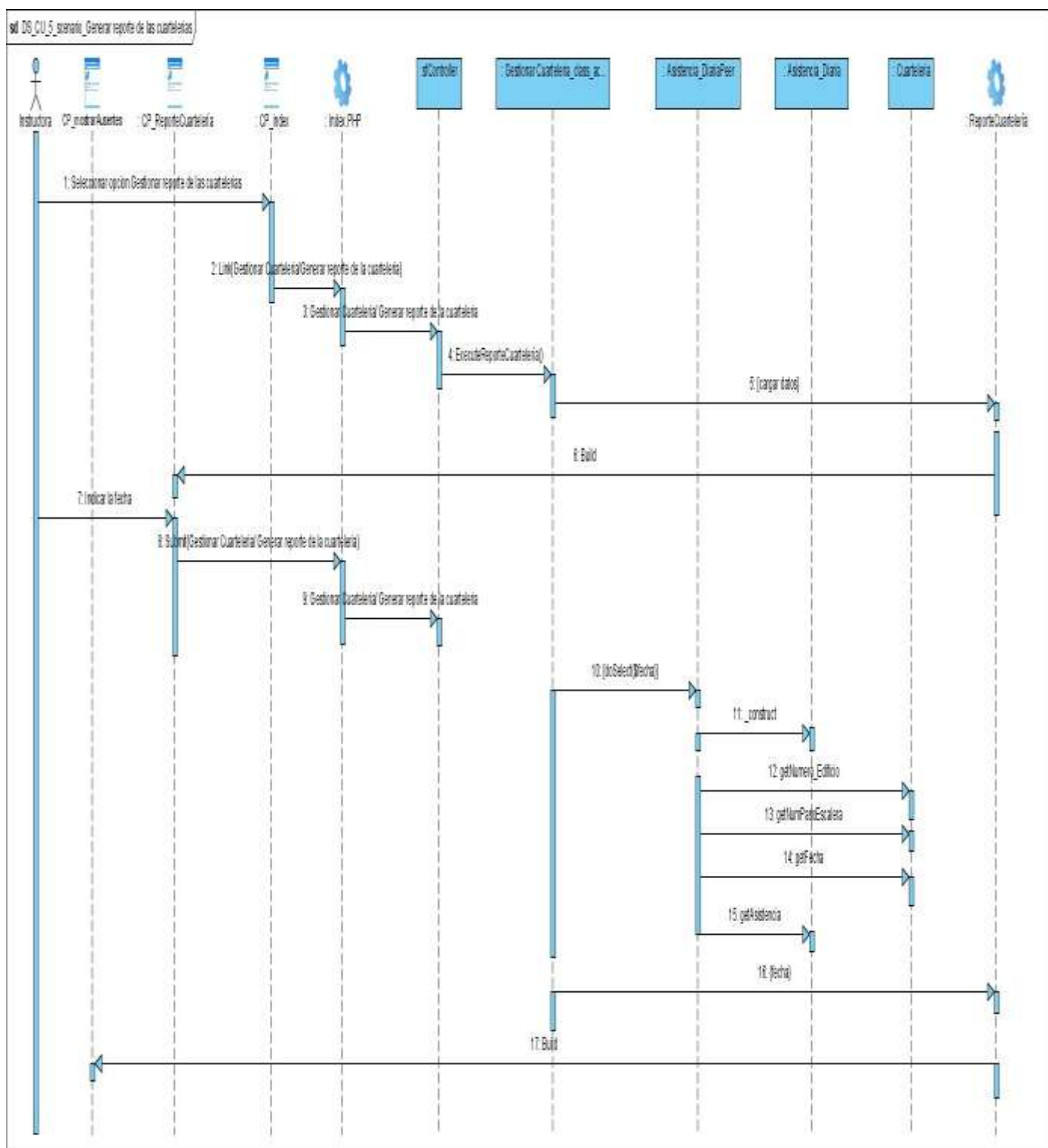
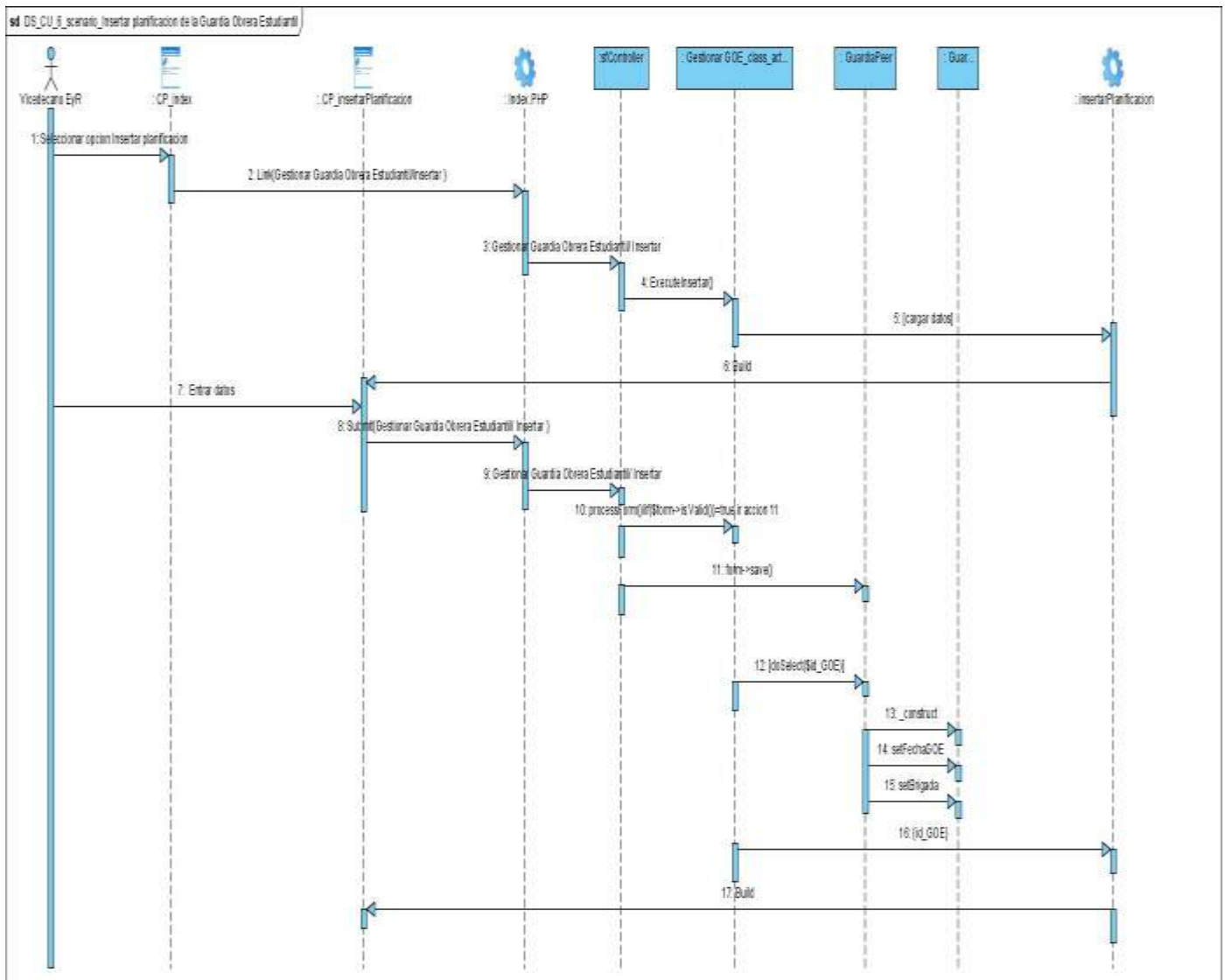
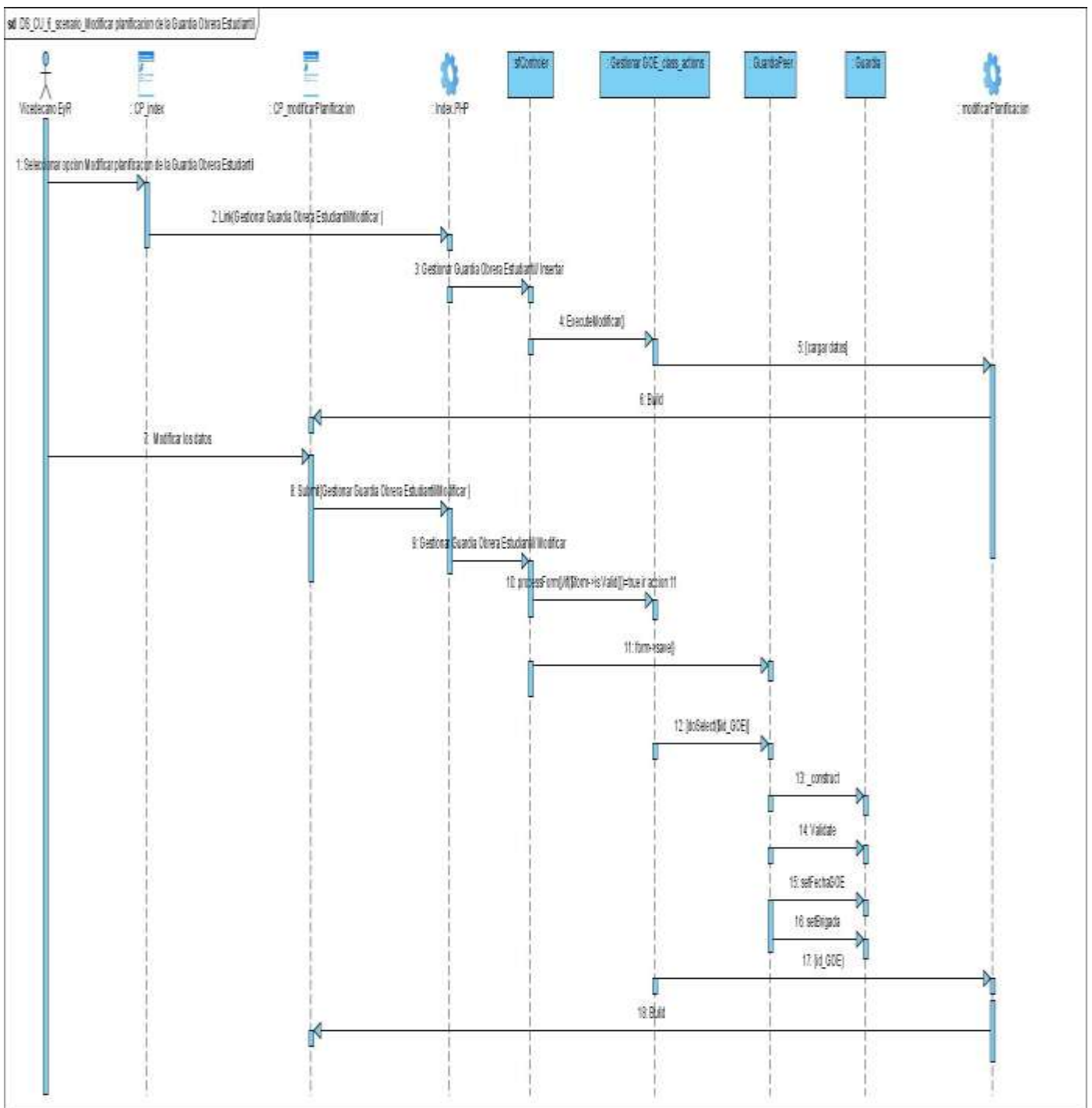


Diagrama de iteración CU “Gestionar Guardia Obrera estudiantil”

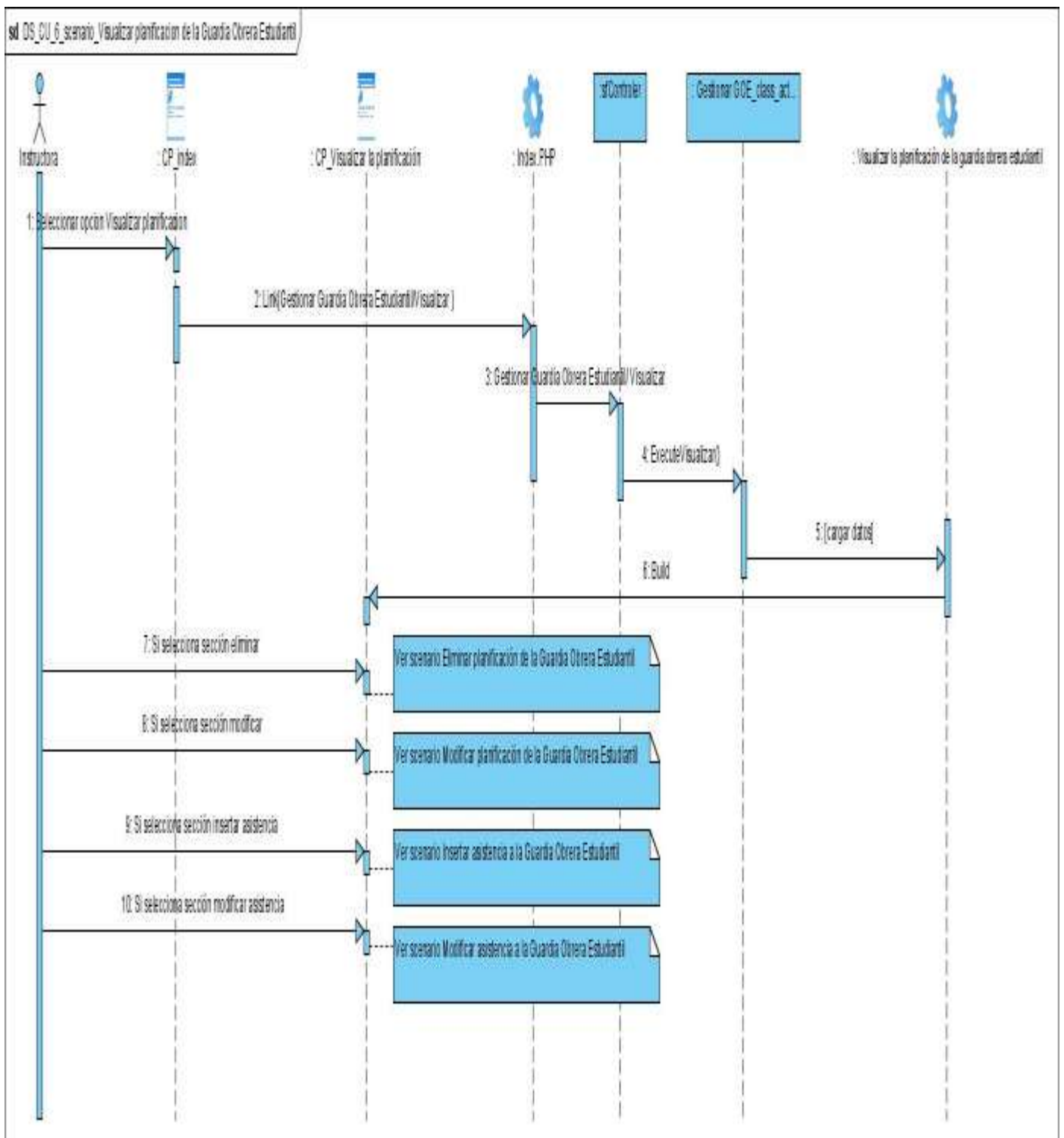
Escenario” Insertar planificación de la Guardia Obrera estudiantil”



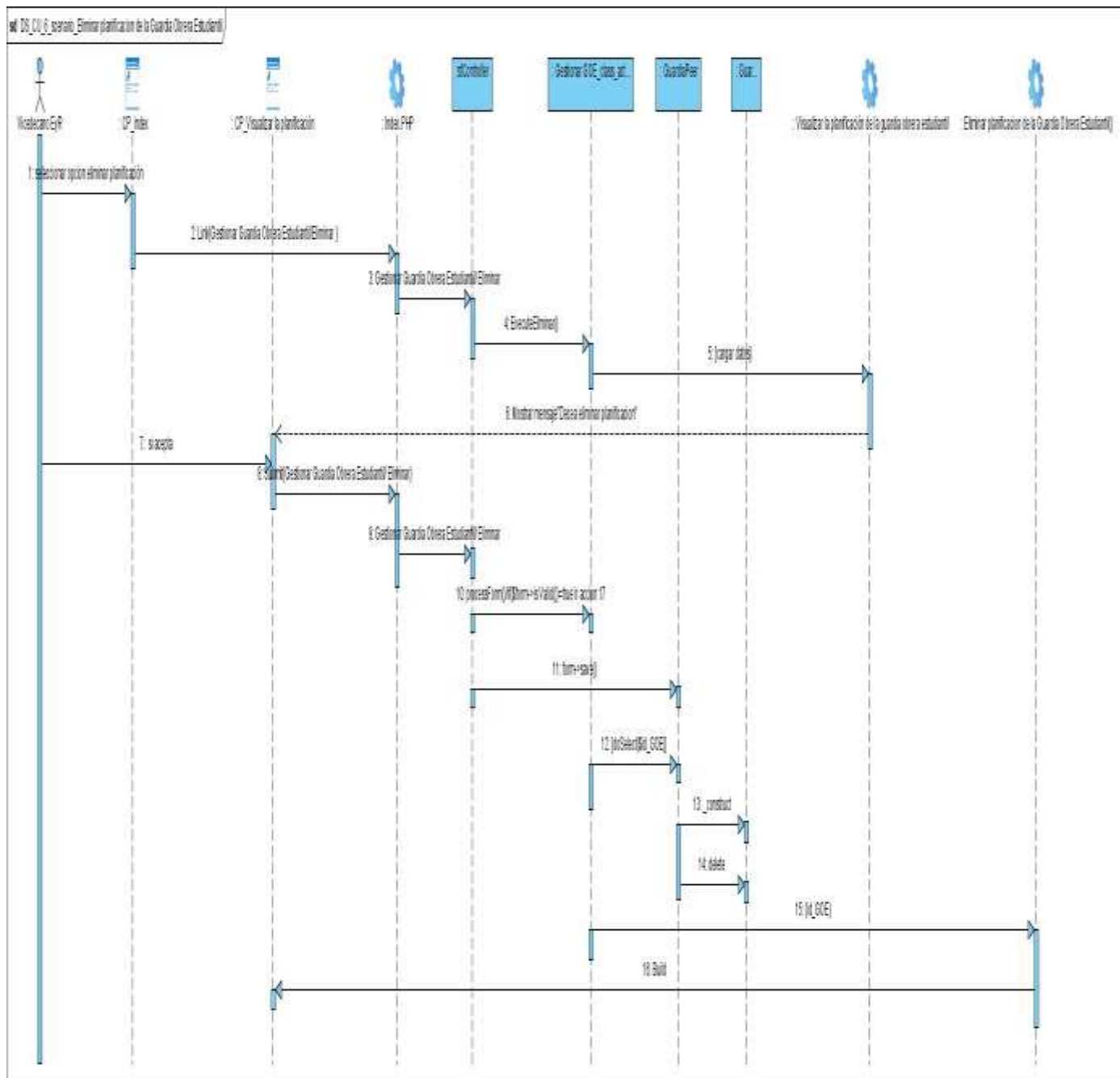
Escenario” Modificar planificación de la Guardia Obrera estudiantil”



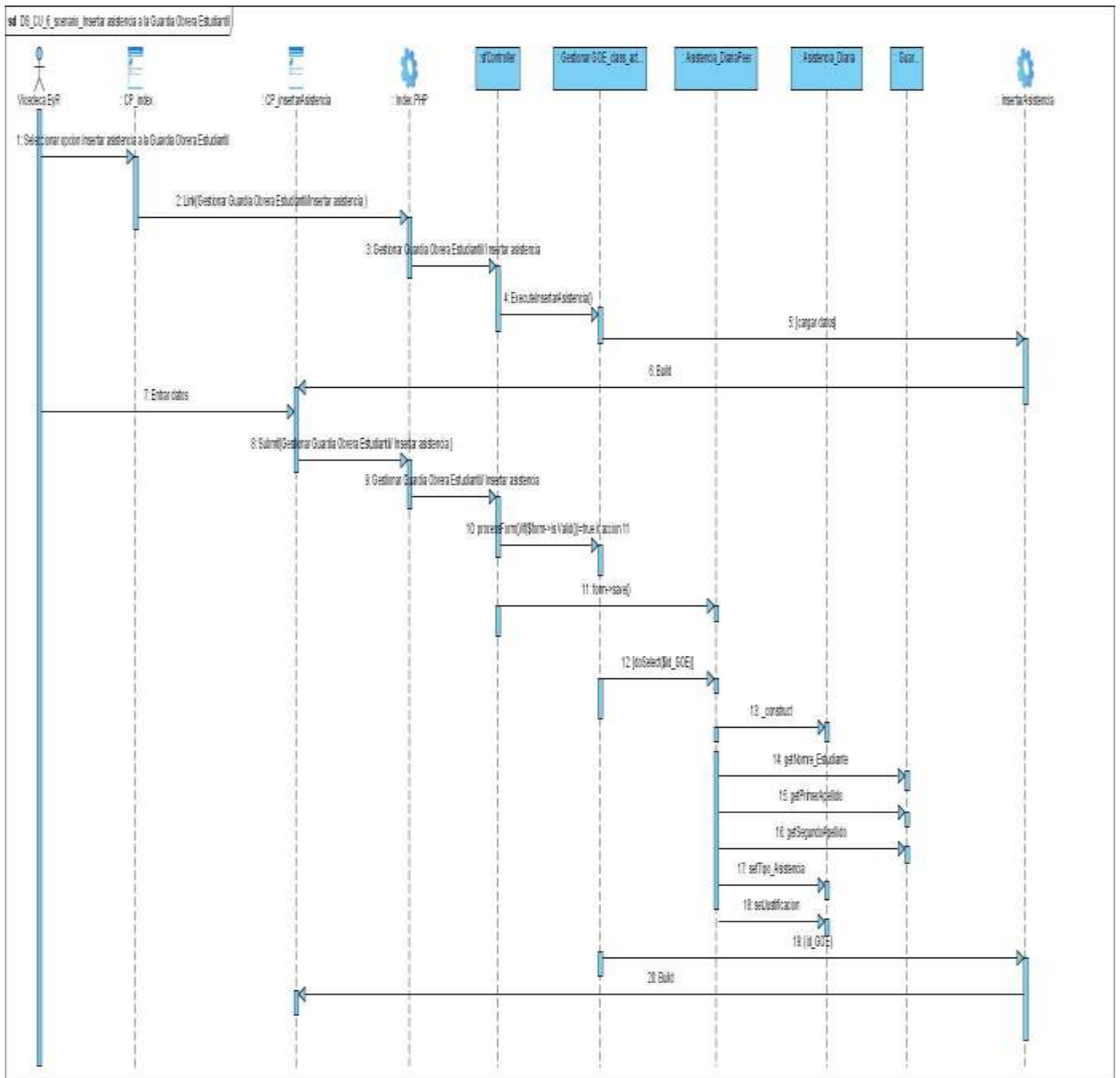
Escenario” Visualizar planificación de la Guardia Obrera estudiantil”



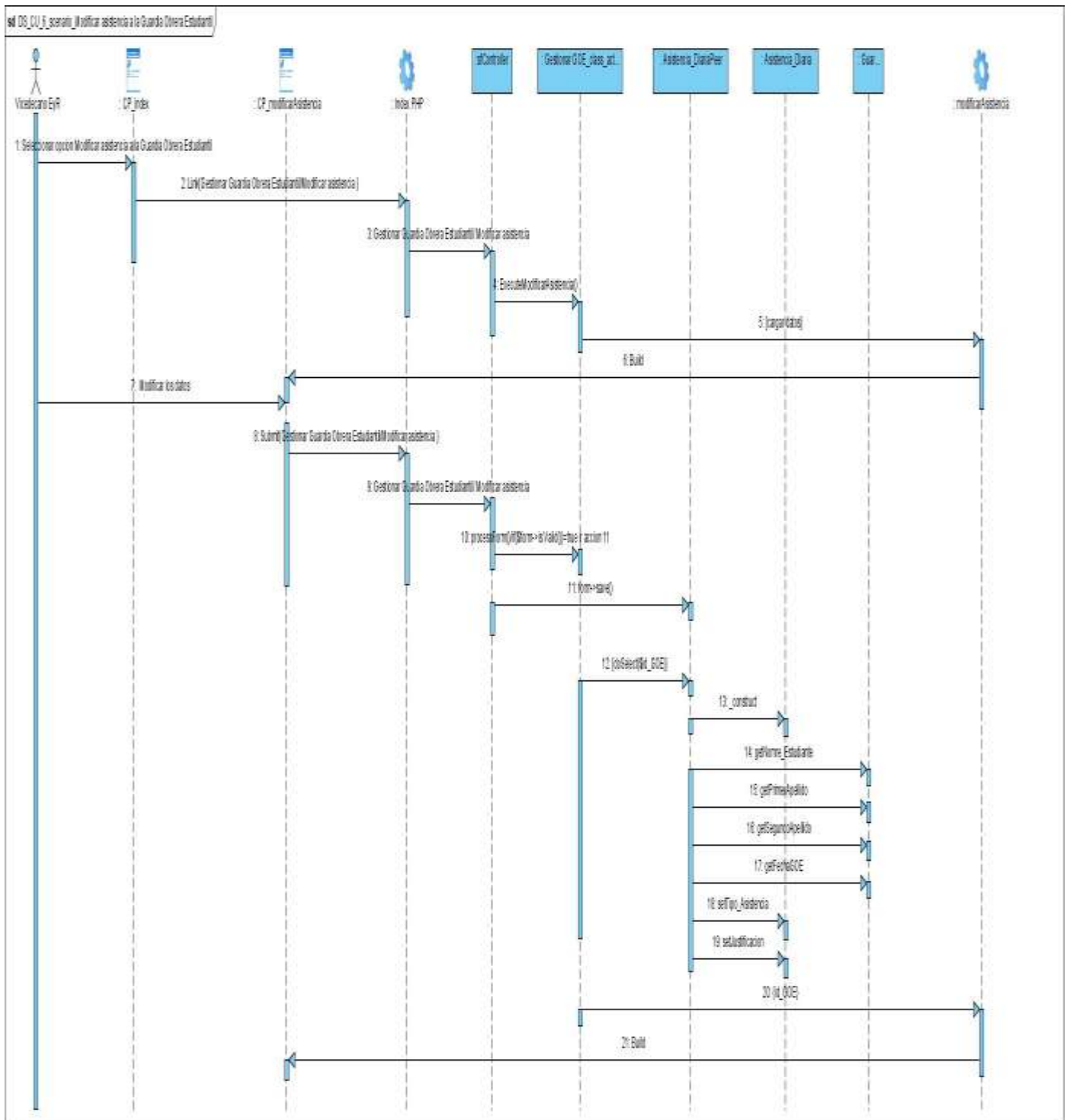
Escenario " Eliminar planificación de la Guardia Obrera estudiantil"



Escenario "Insertar asistencia a la Guardia Obrera estudiantil"



Escenario” Modificar asistencia a la Guardia Obrera estudiantil”



Escenario” Generar reporte de la Guardia Obrera estudiantil”

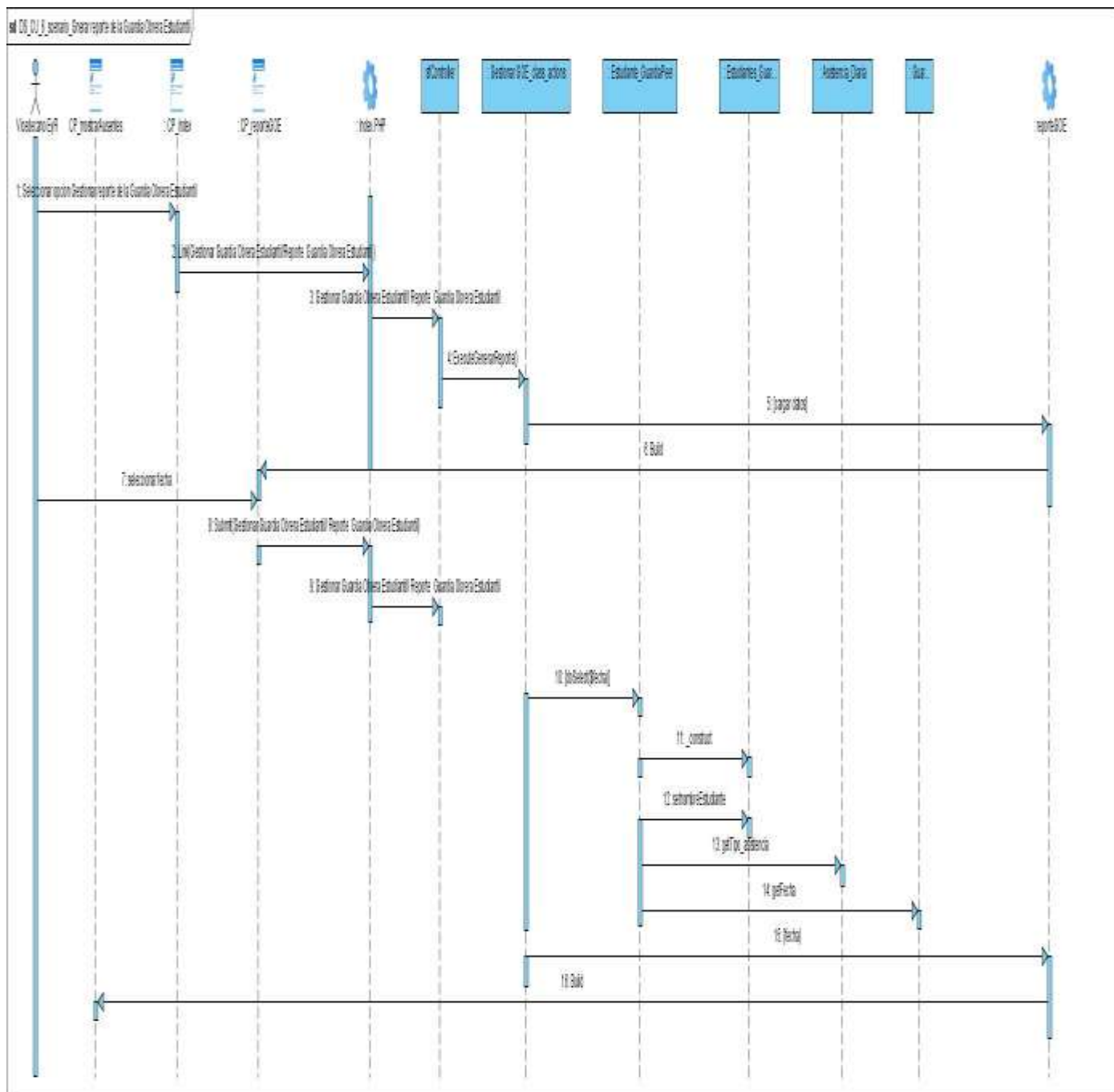
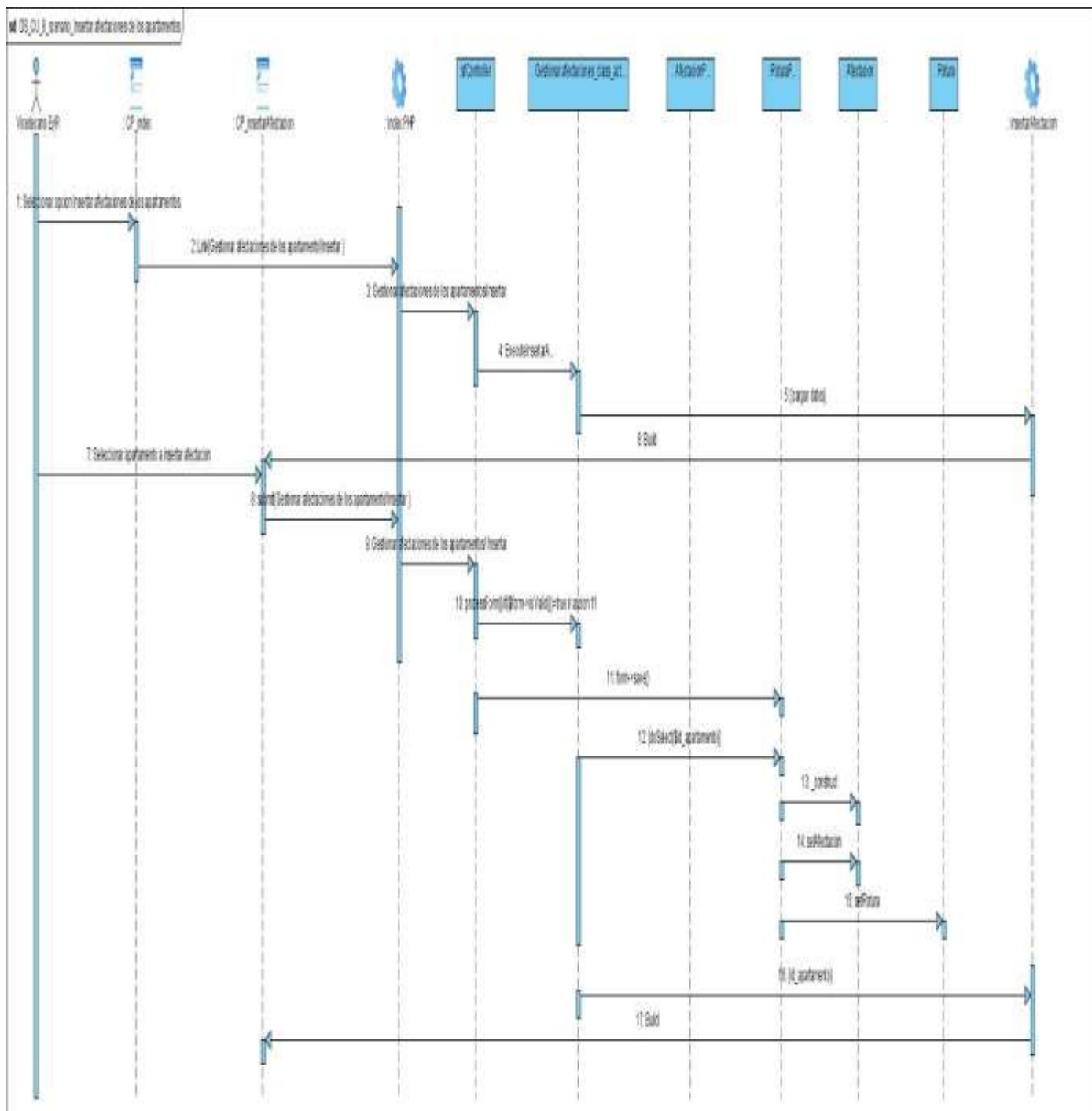
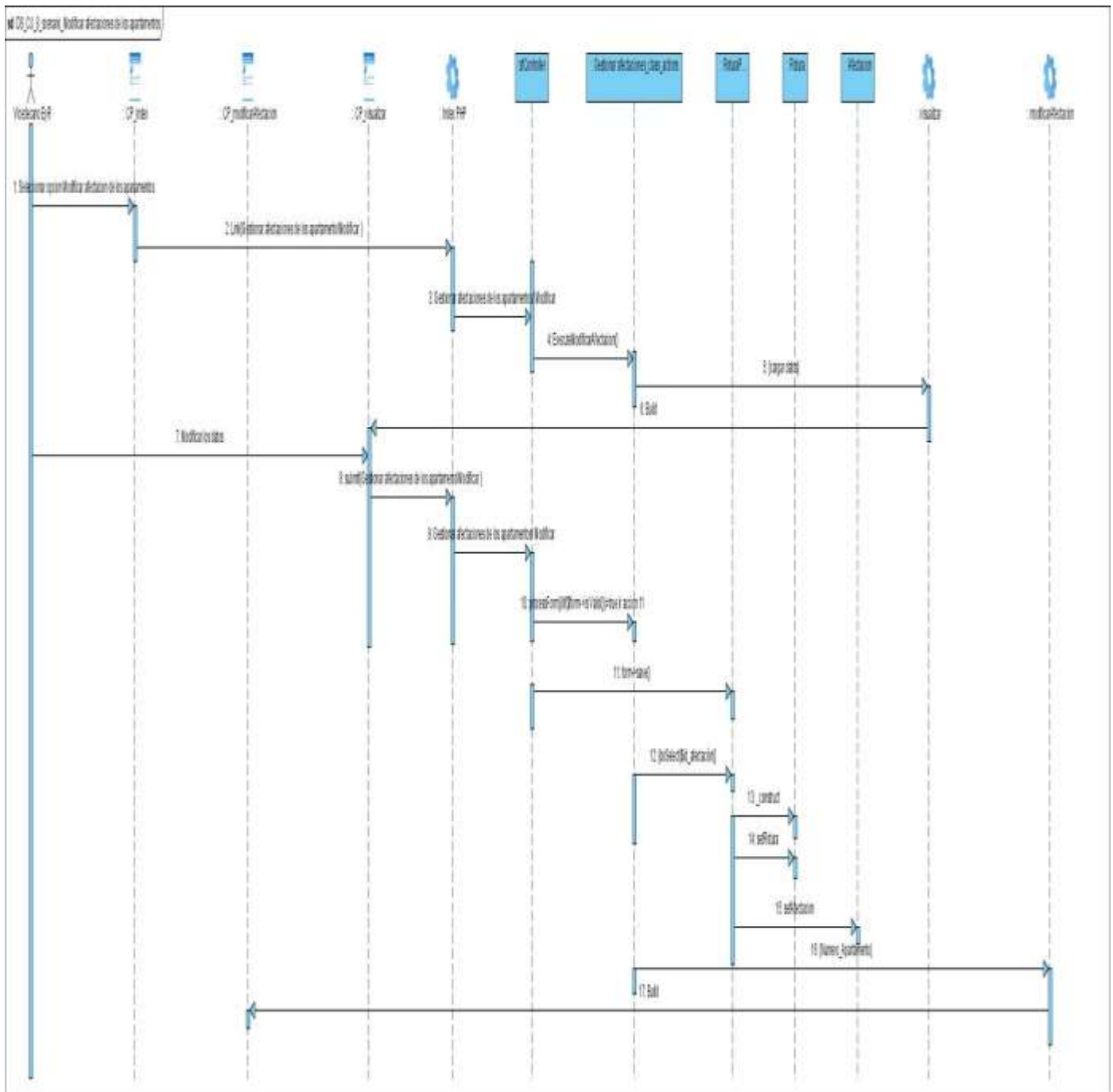


Diagrama de iteración CU “Gestionar afectación en los apartamentos”

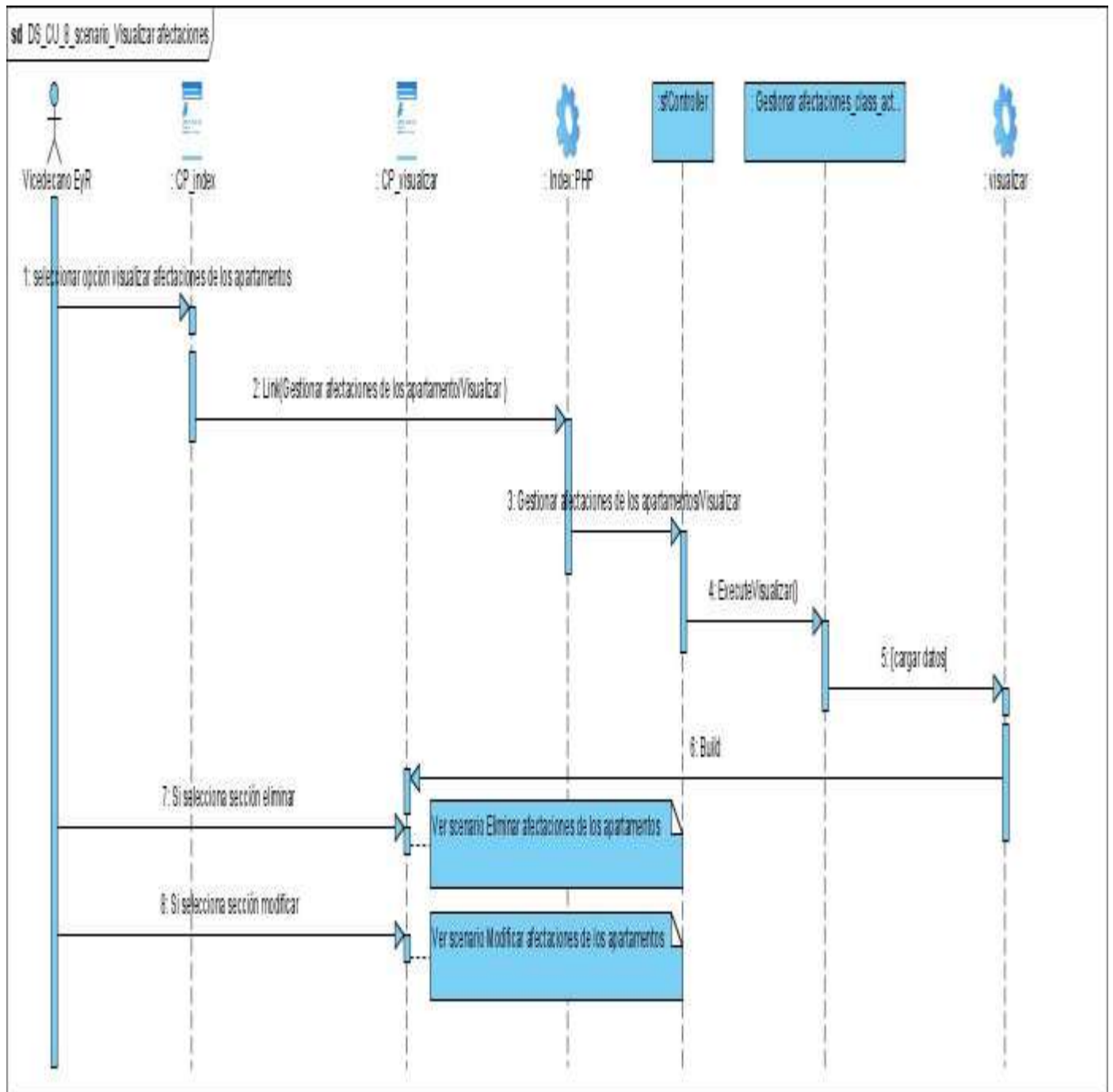
Escenario” Insertar afectación de los apartamentos”



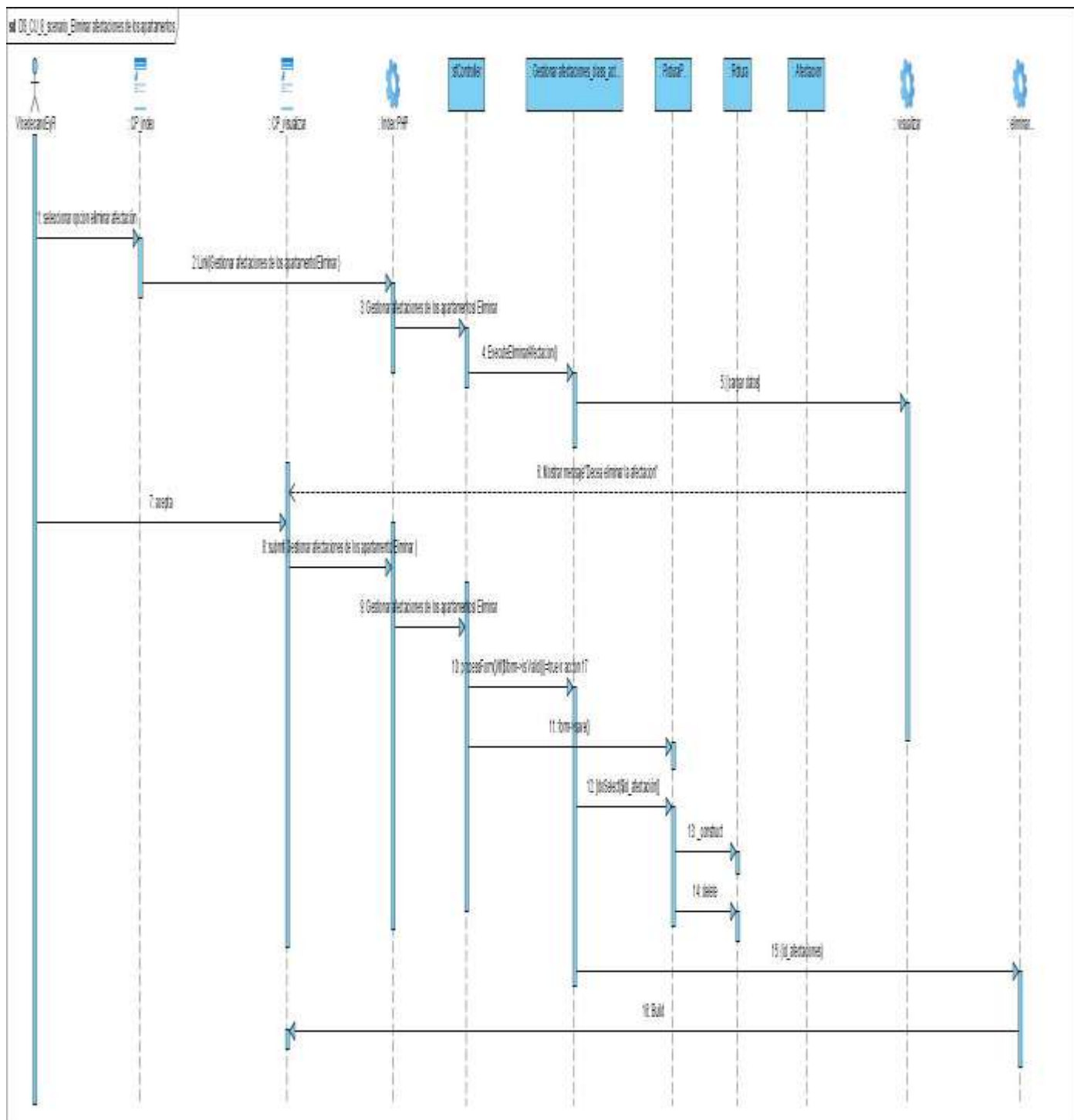
Escenario " Modificar afectación de los apartamentos"



Escenario " Visualizar afectación de los apartamentos"



Escenario " Eliminar afectación de los apartamentos"



Escenario” Generar reporte de la afectaciones de los apartamentos”

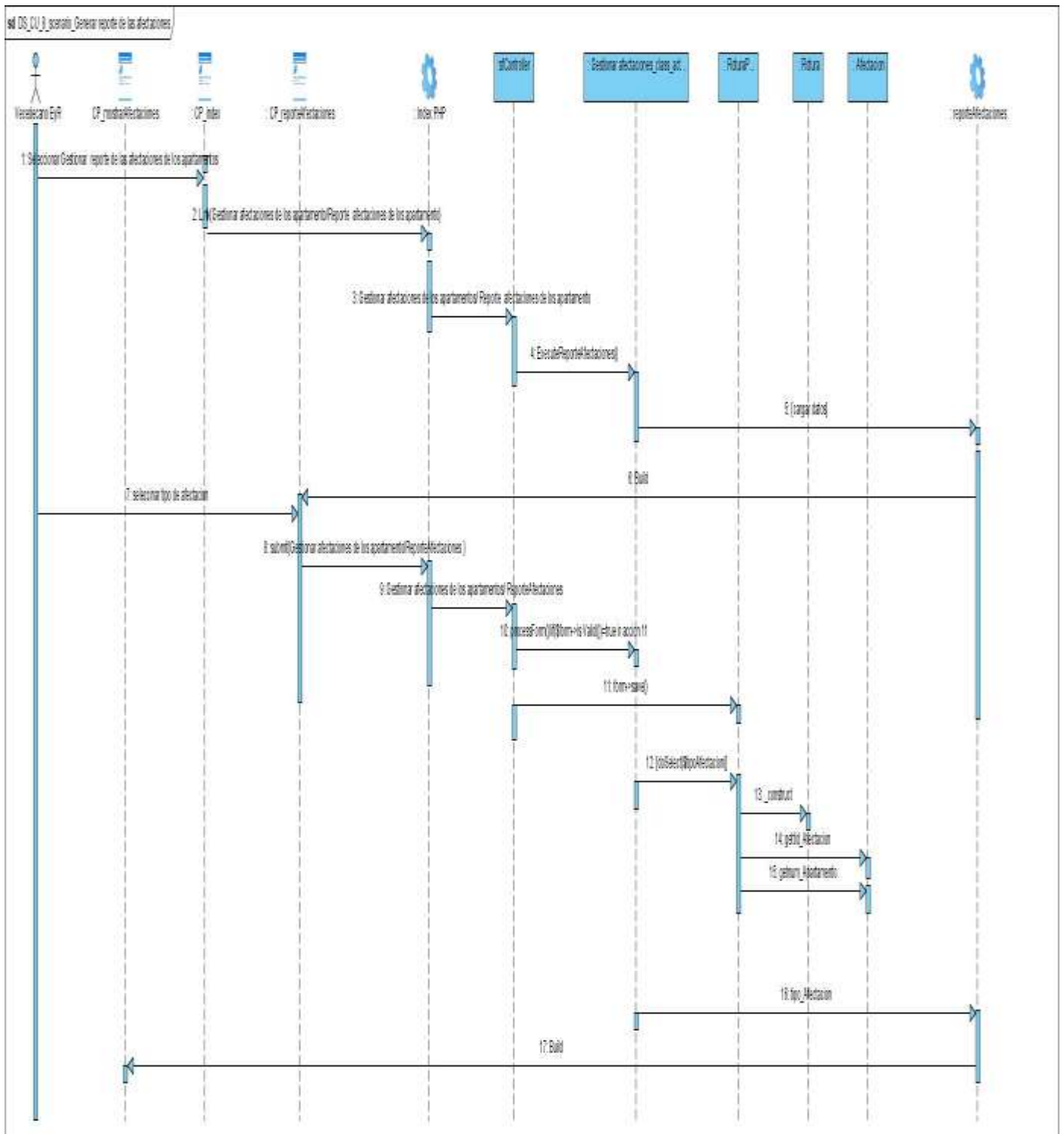
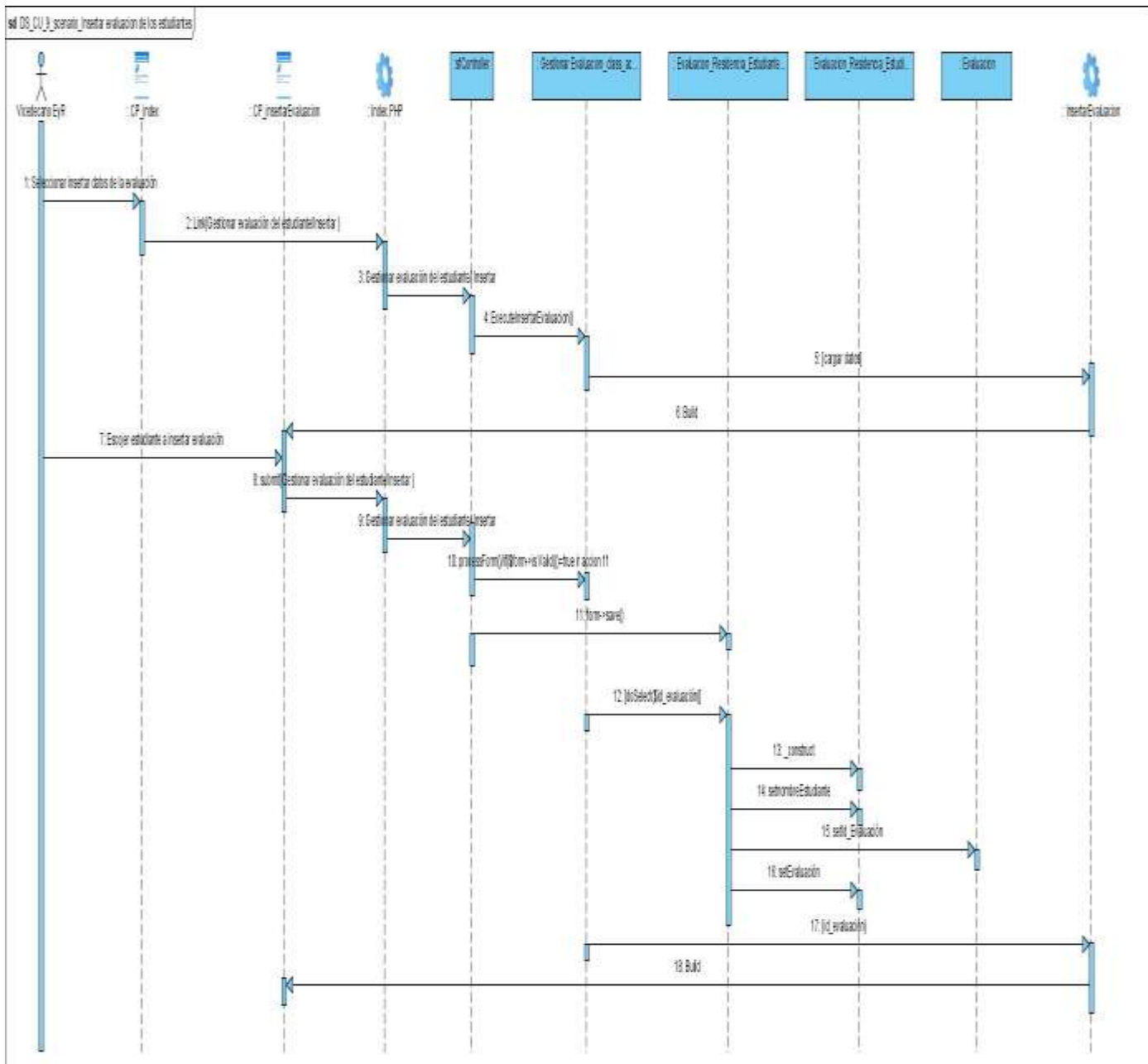
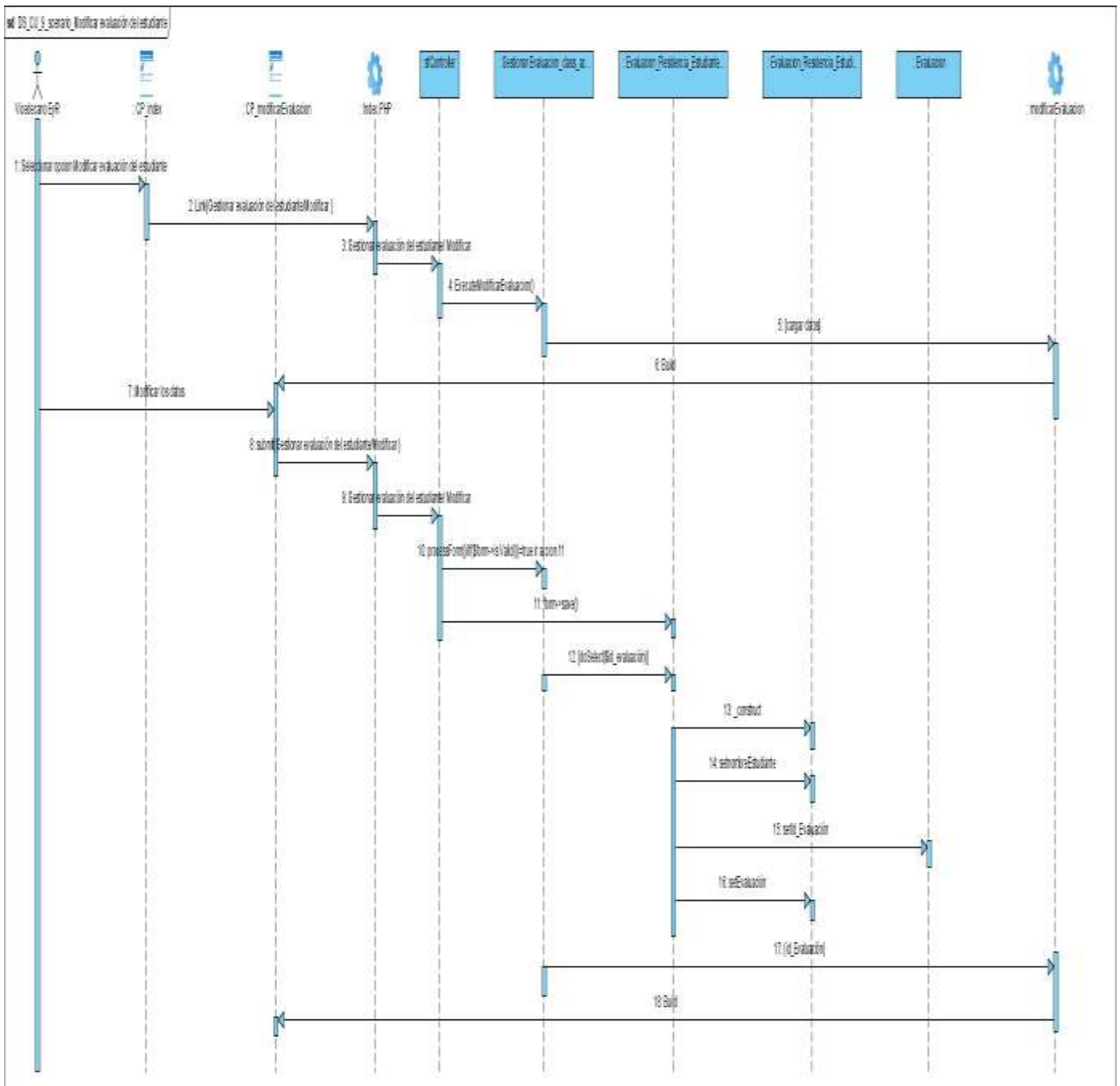


Diagrama de iteración CU “Gestionar evaluación de los estudiantes en la residencia”

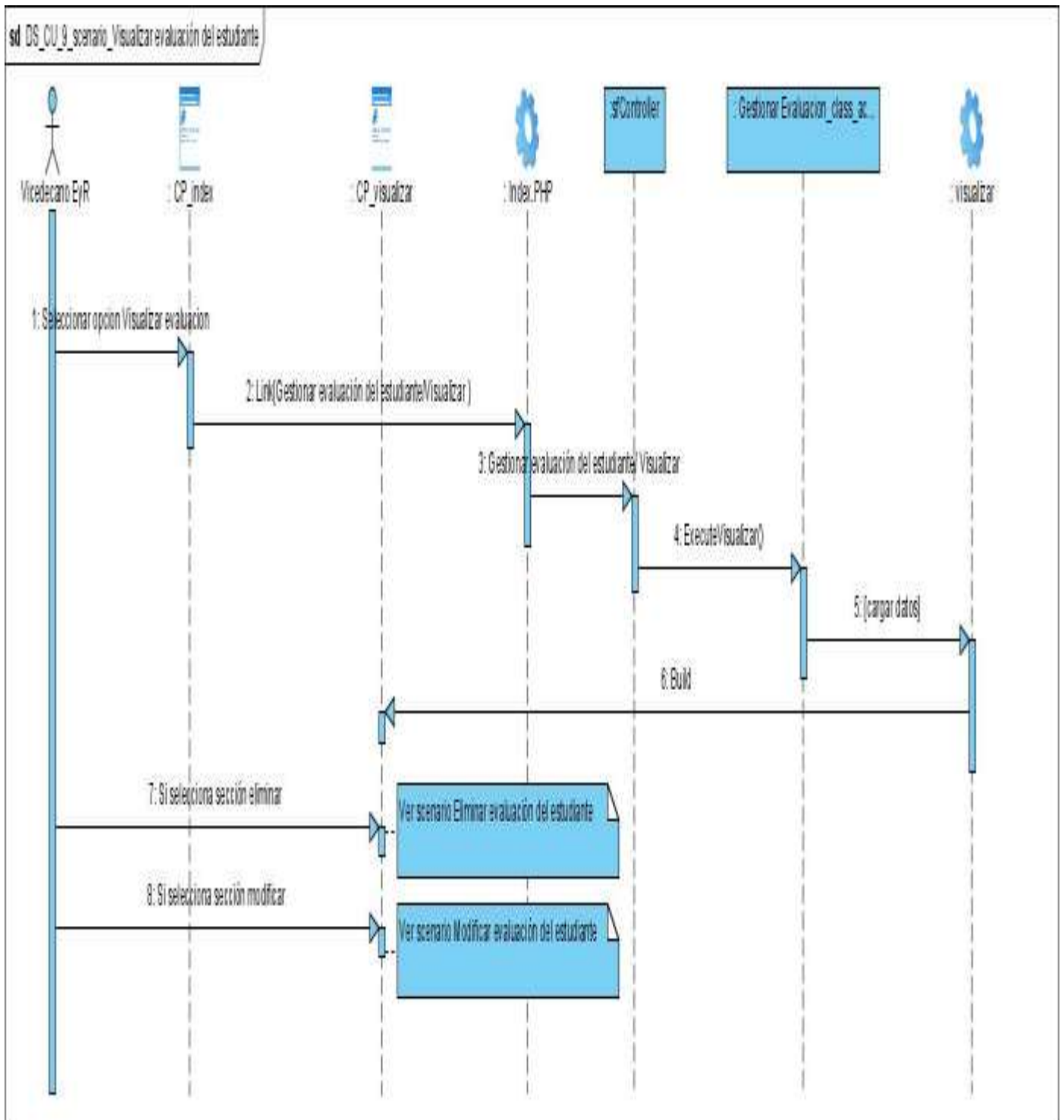
Escenario” Insertar evaluación de los estudiantes en la residencia”



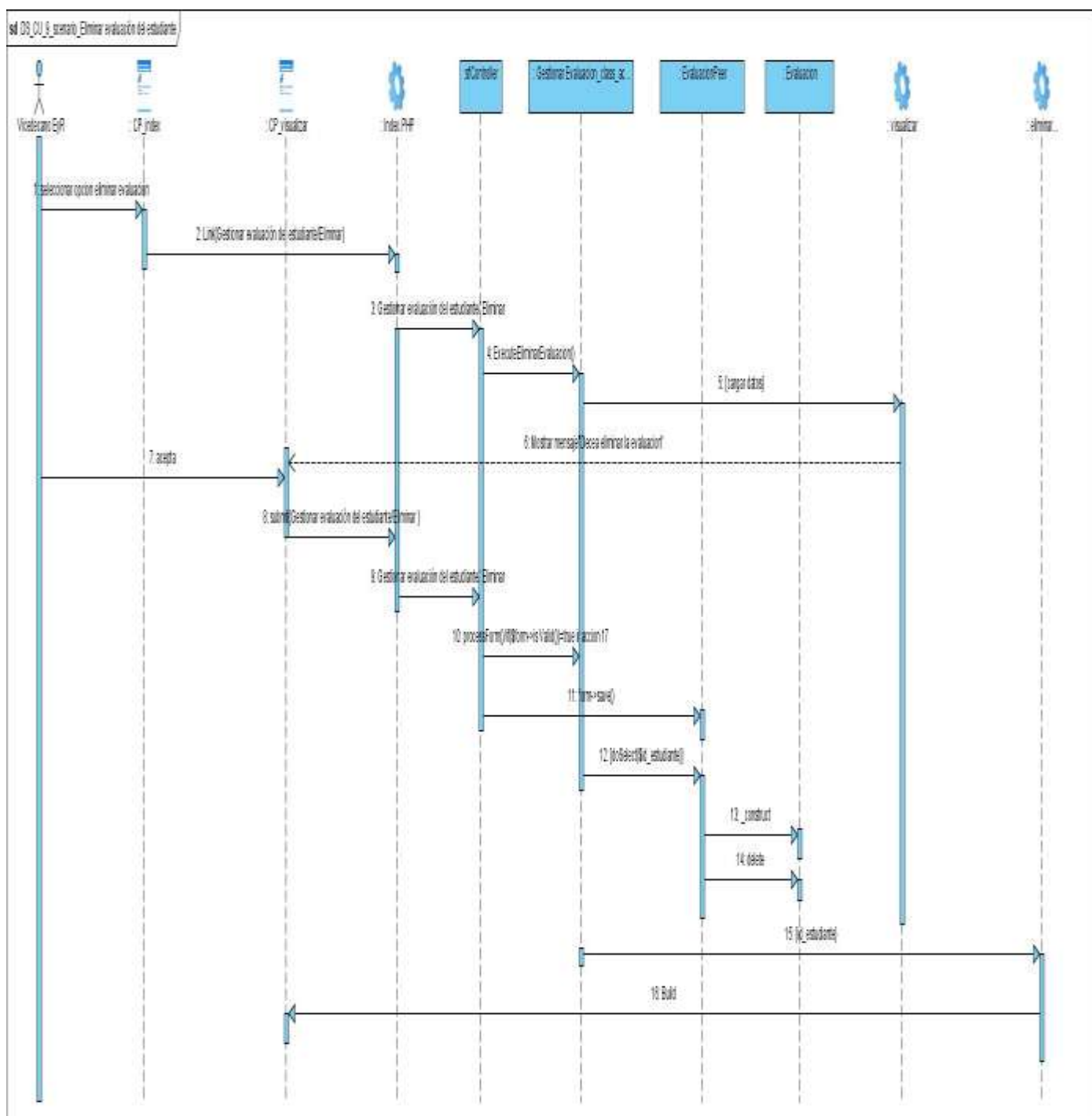
Escenario” Modificar evaluación de los estudiantes en la residencia”



Escenario " Visualizar evaluación de los estudiantes en la residencia "



Escenario” Eliminar evaluación de los estudiantes en la residencia”



Escenario” Generar reporte de las evaluaciones de los estudiantes en la residencia”

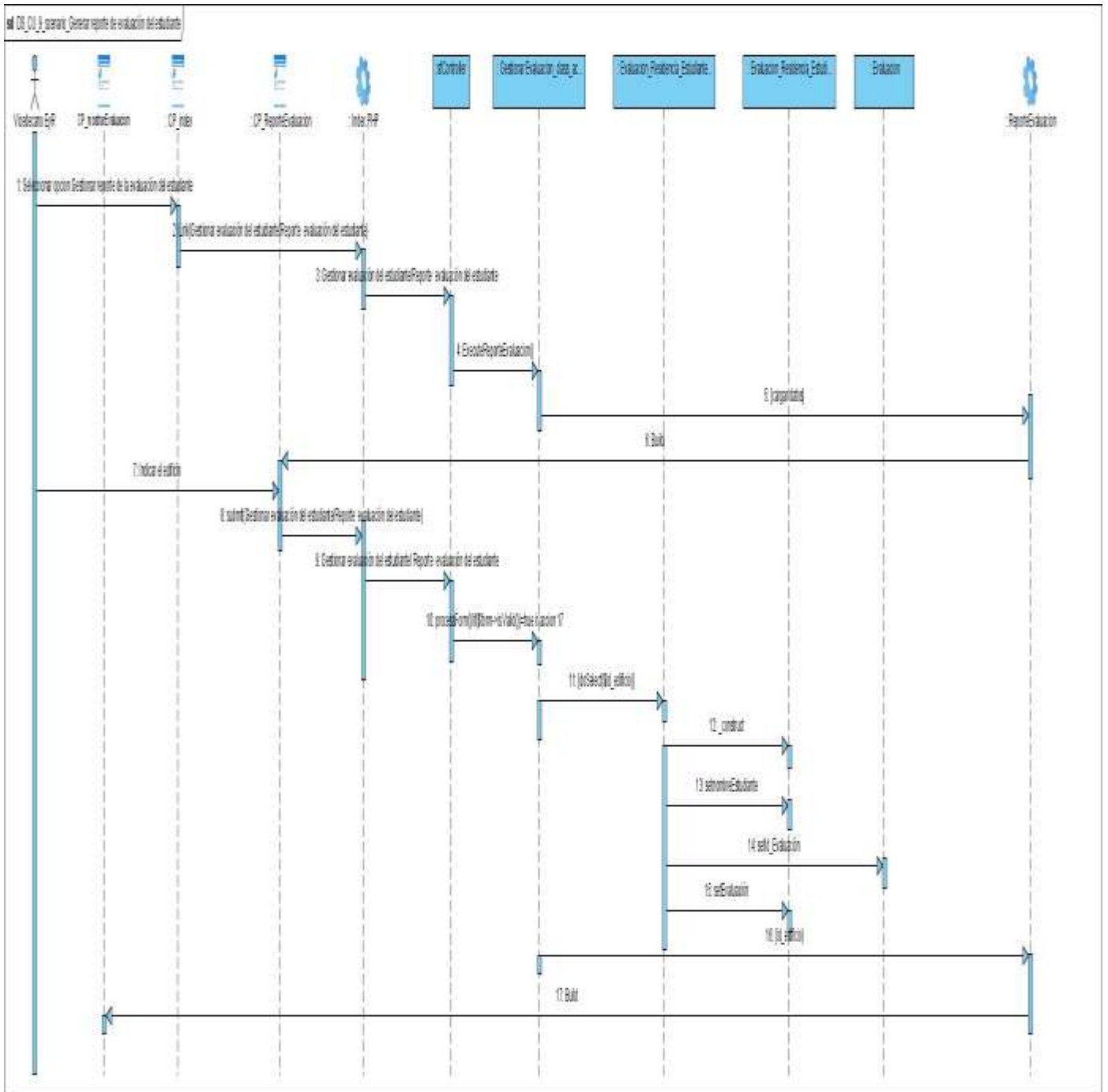
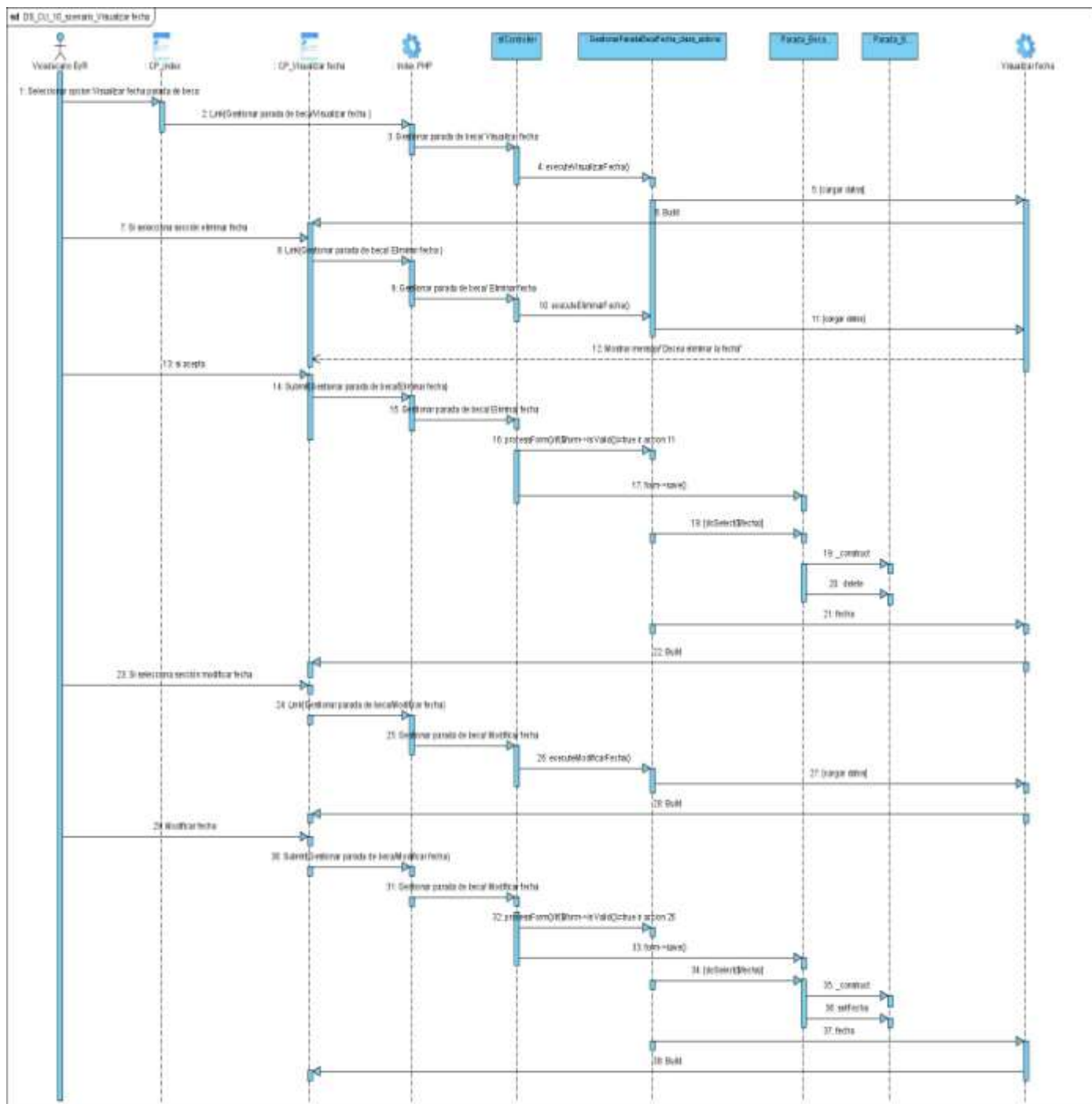
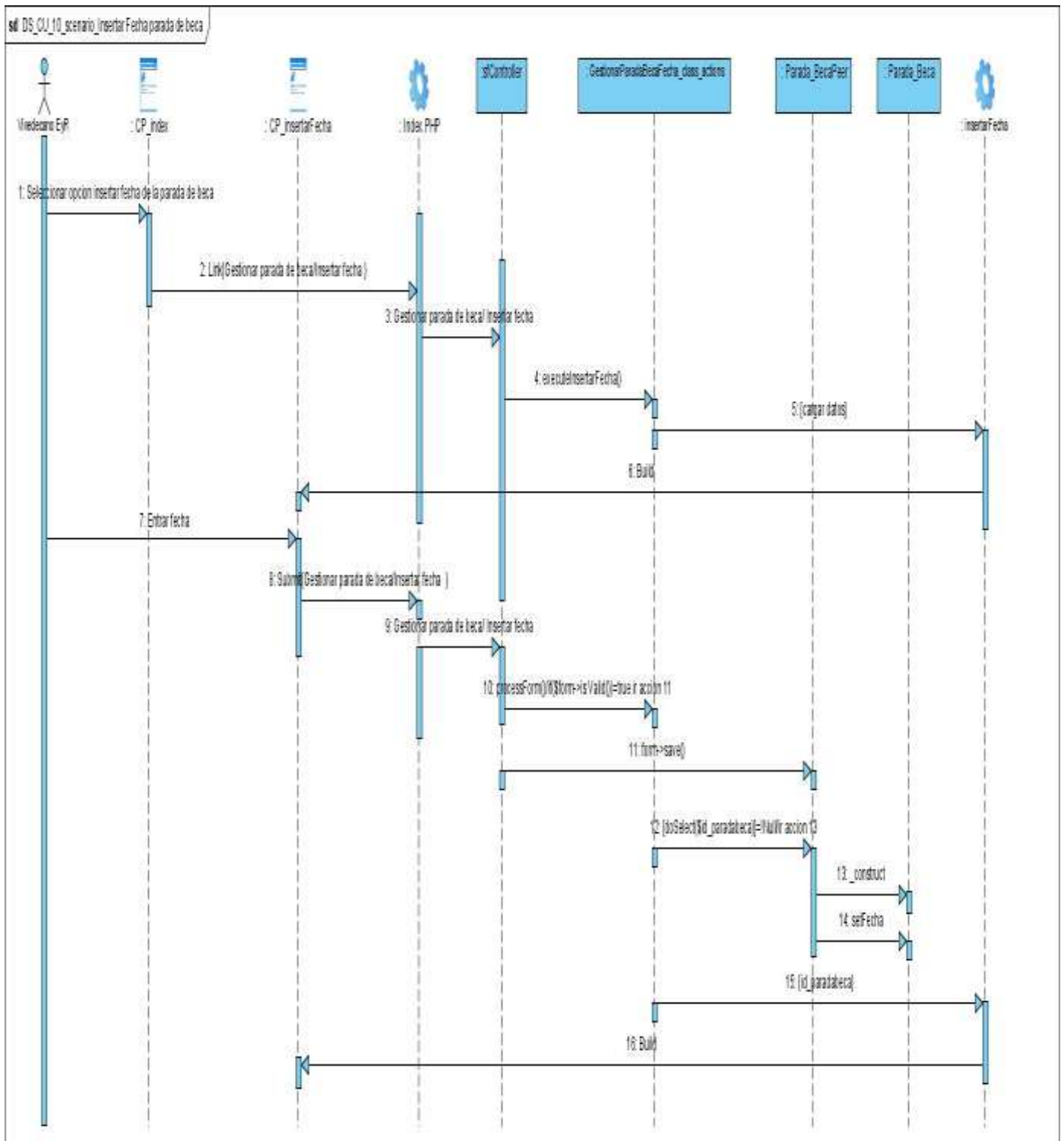


Diagrama de iteración CU “Gestionar parada de beca”

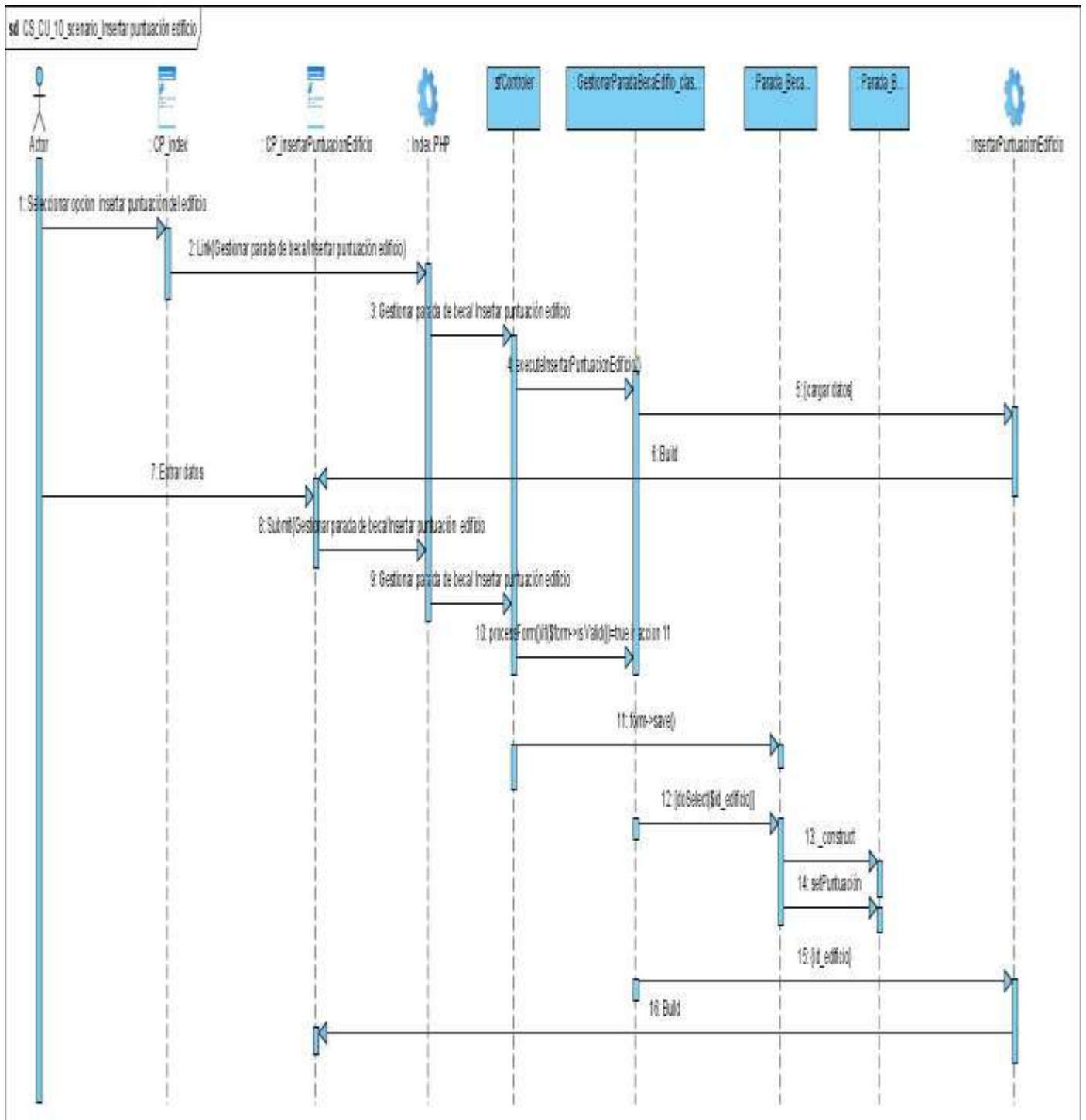
Escenario” Visualizar fecha de la parada de beca”



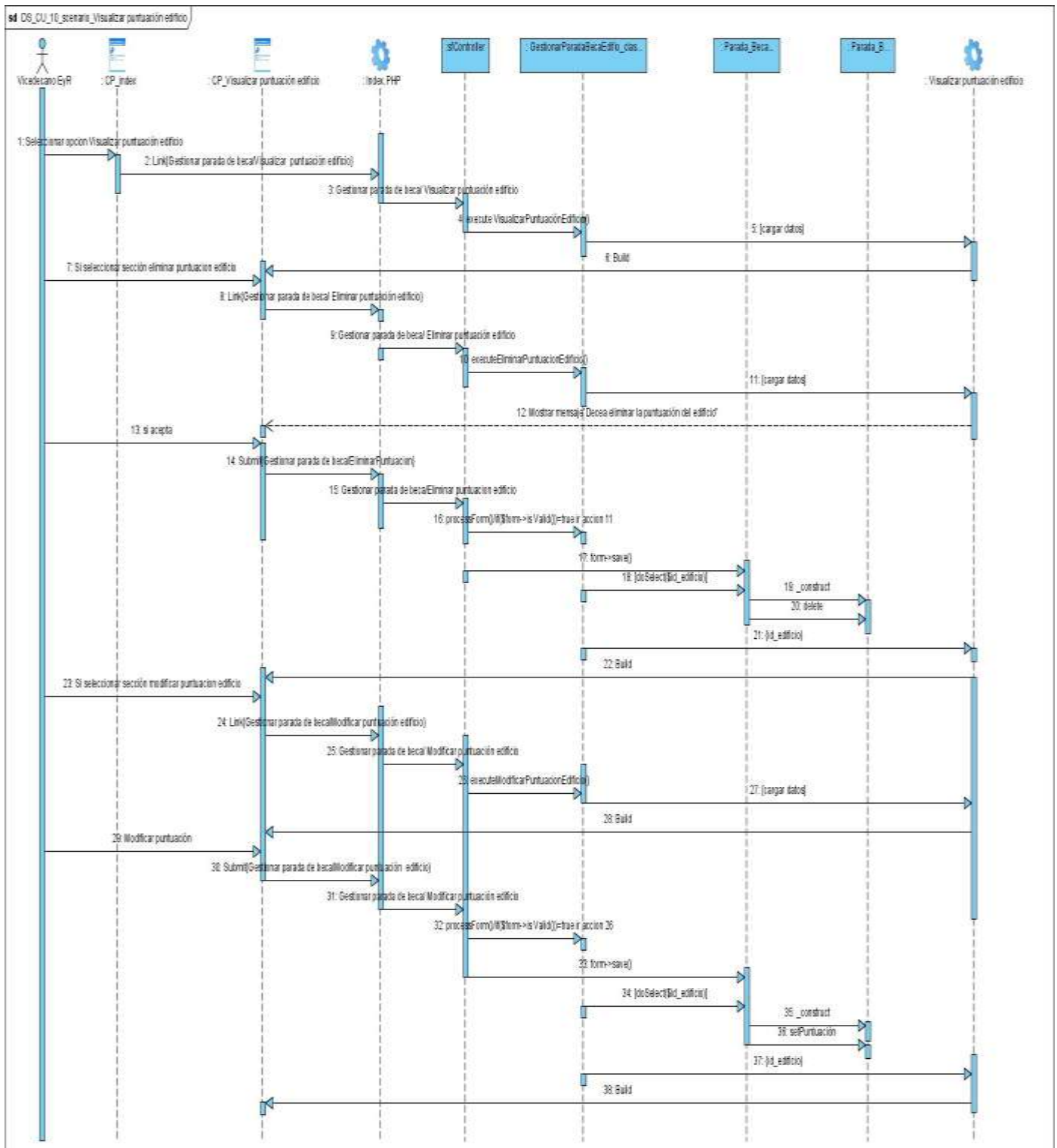
Escenario "Insertar fecha de la parada de beca"



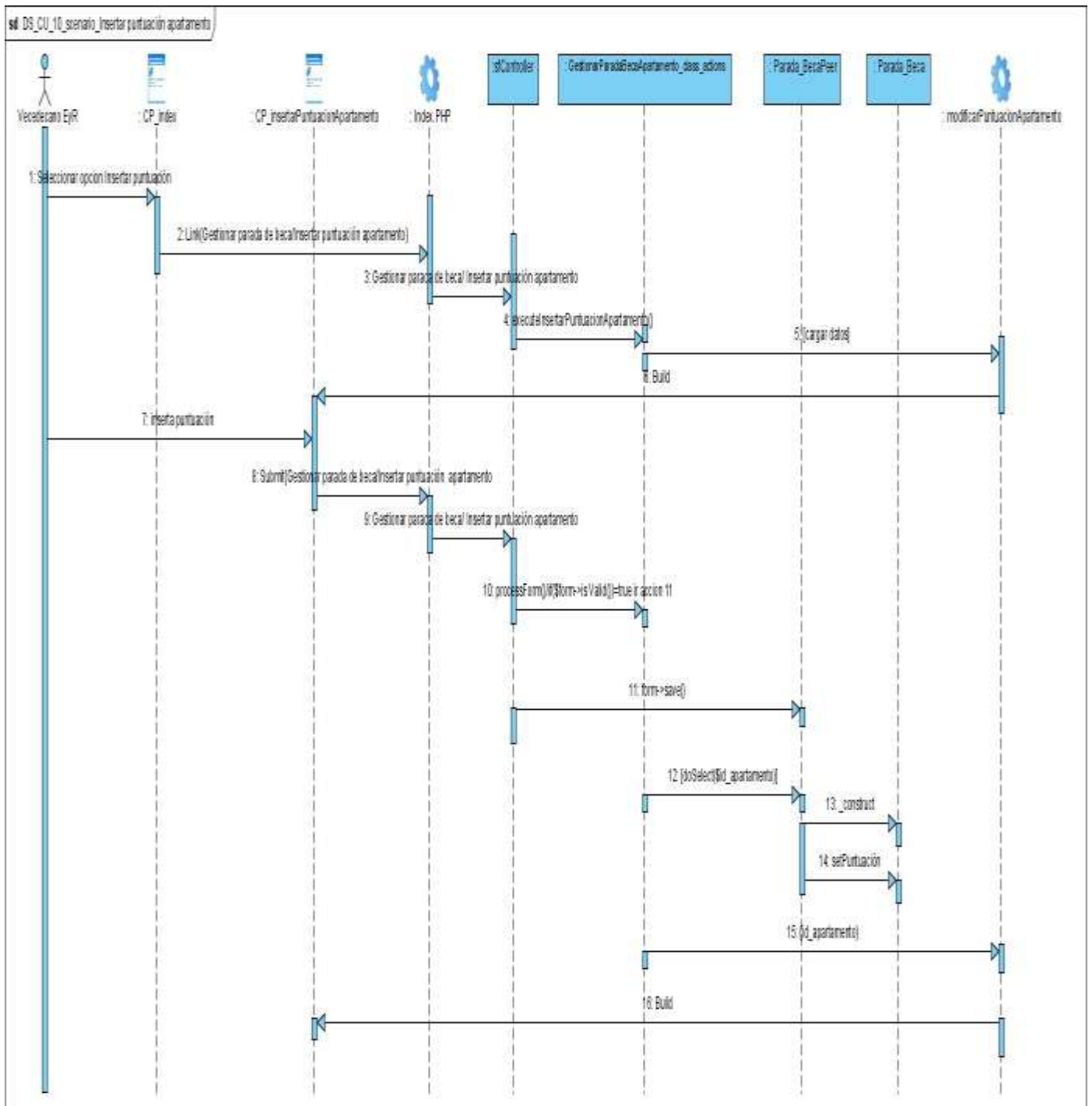
Escenario "Insertar puntuación edificio"



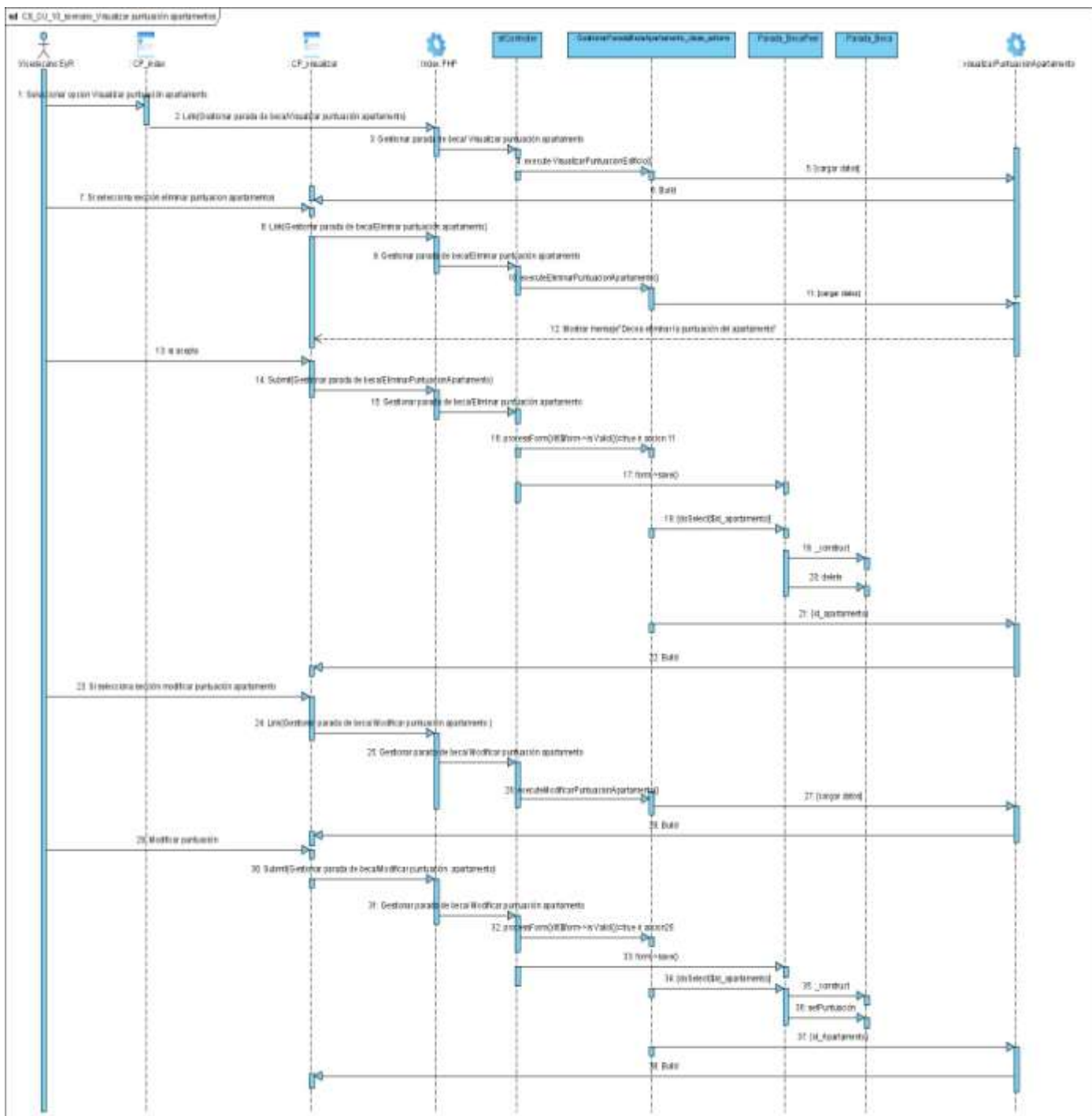
Escenario” Visualizar puntuación edificio”



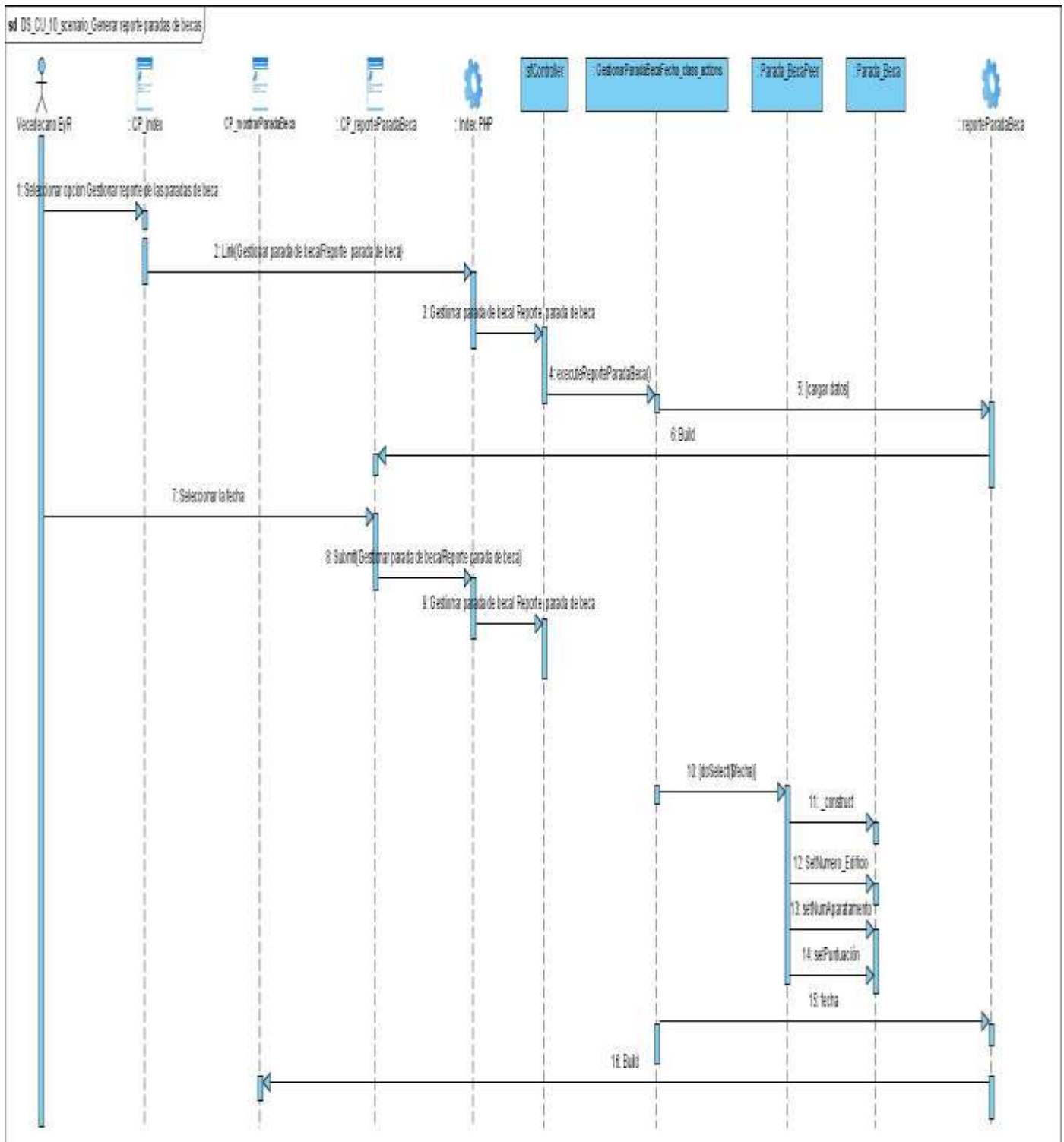
Escenario "Insertar puntuación apartamento"



Escenario " Visualizar puntuación apartamento "

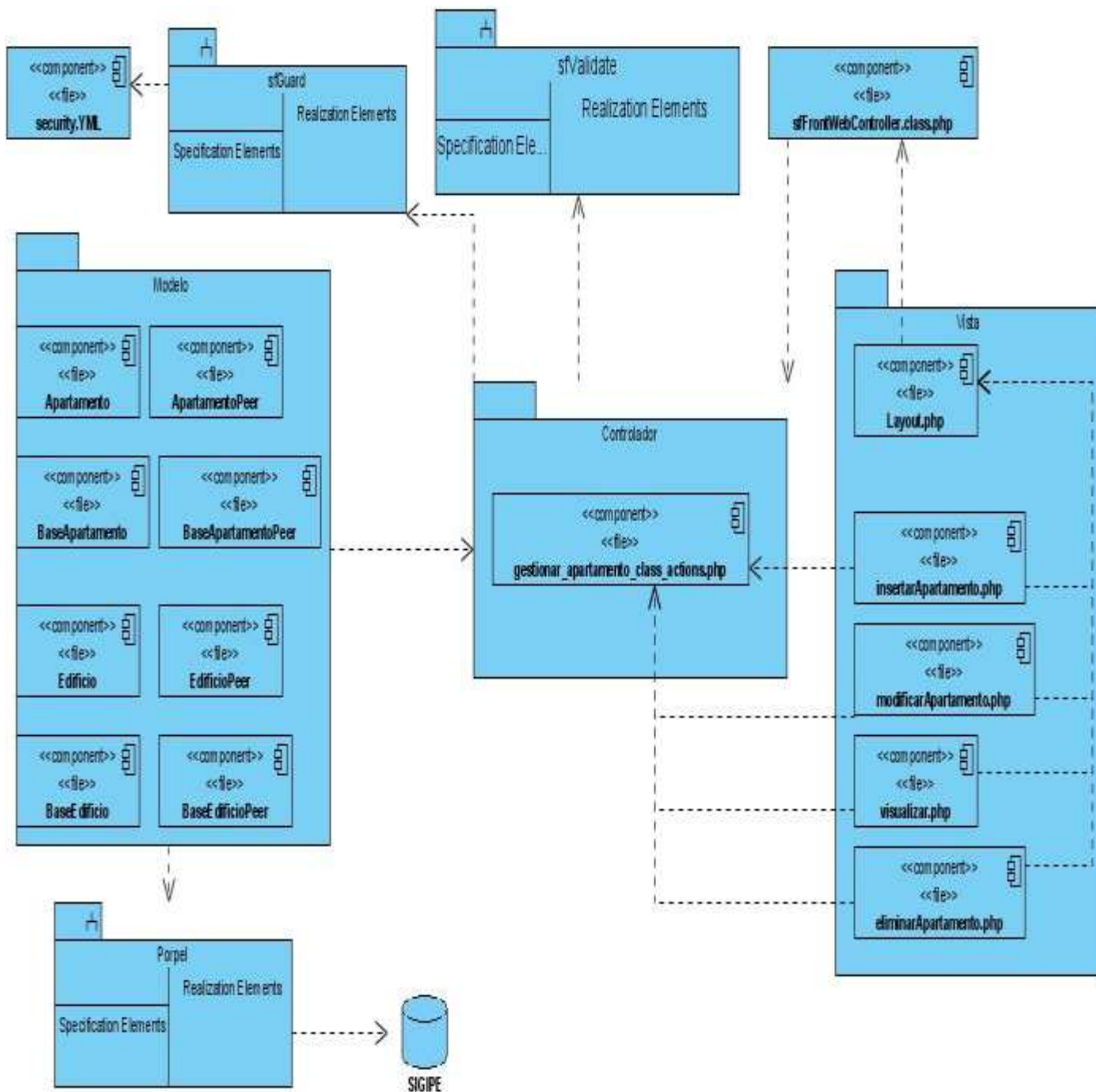


Escenario " Generar reporte de la parada de beca "

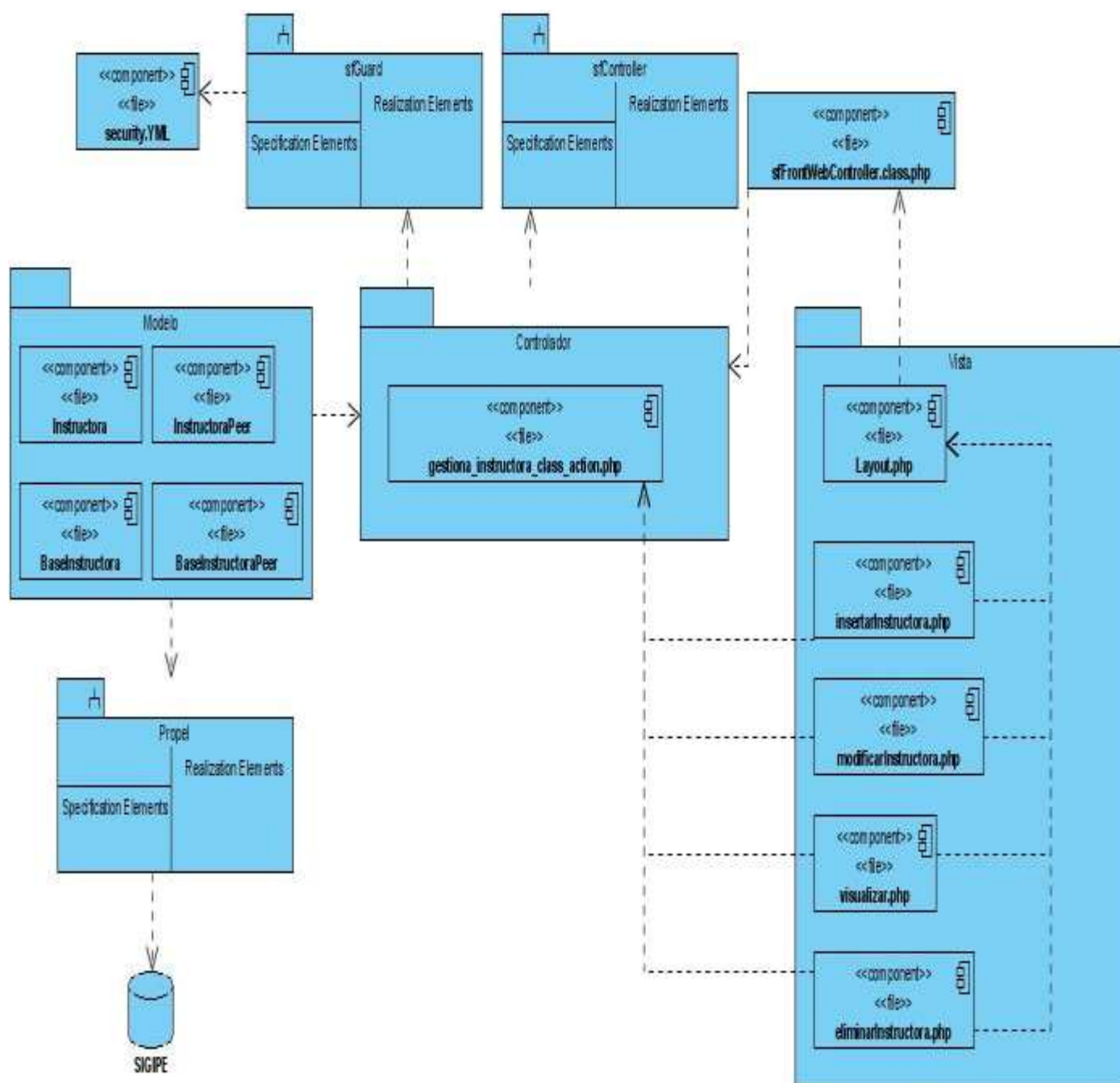


Anexo.4-Diagrama de componentes

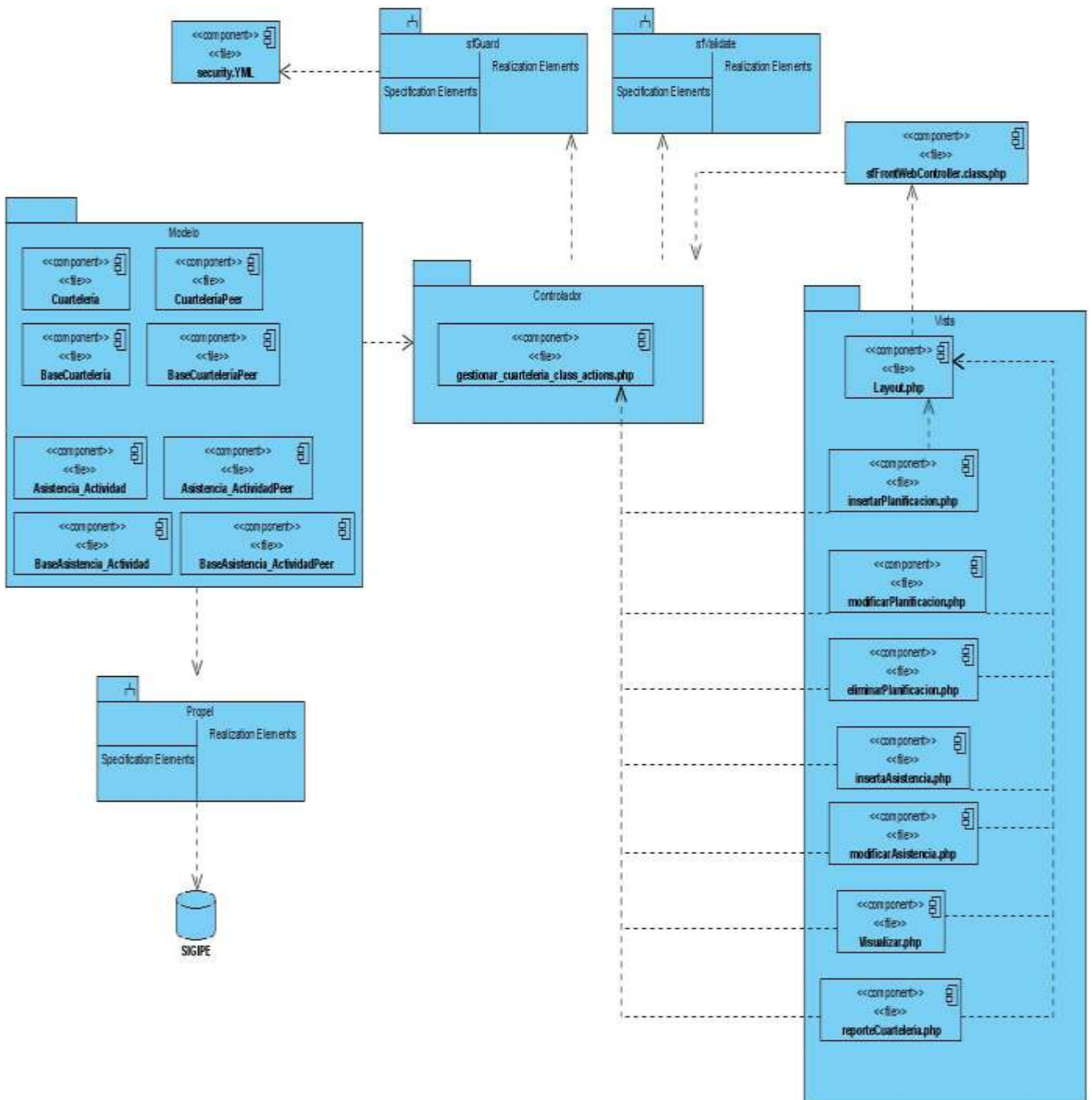
Diagramas de componentes CU” Gestionar datos de los apartamento”



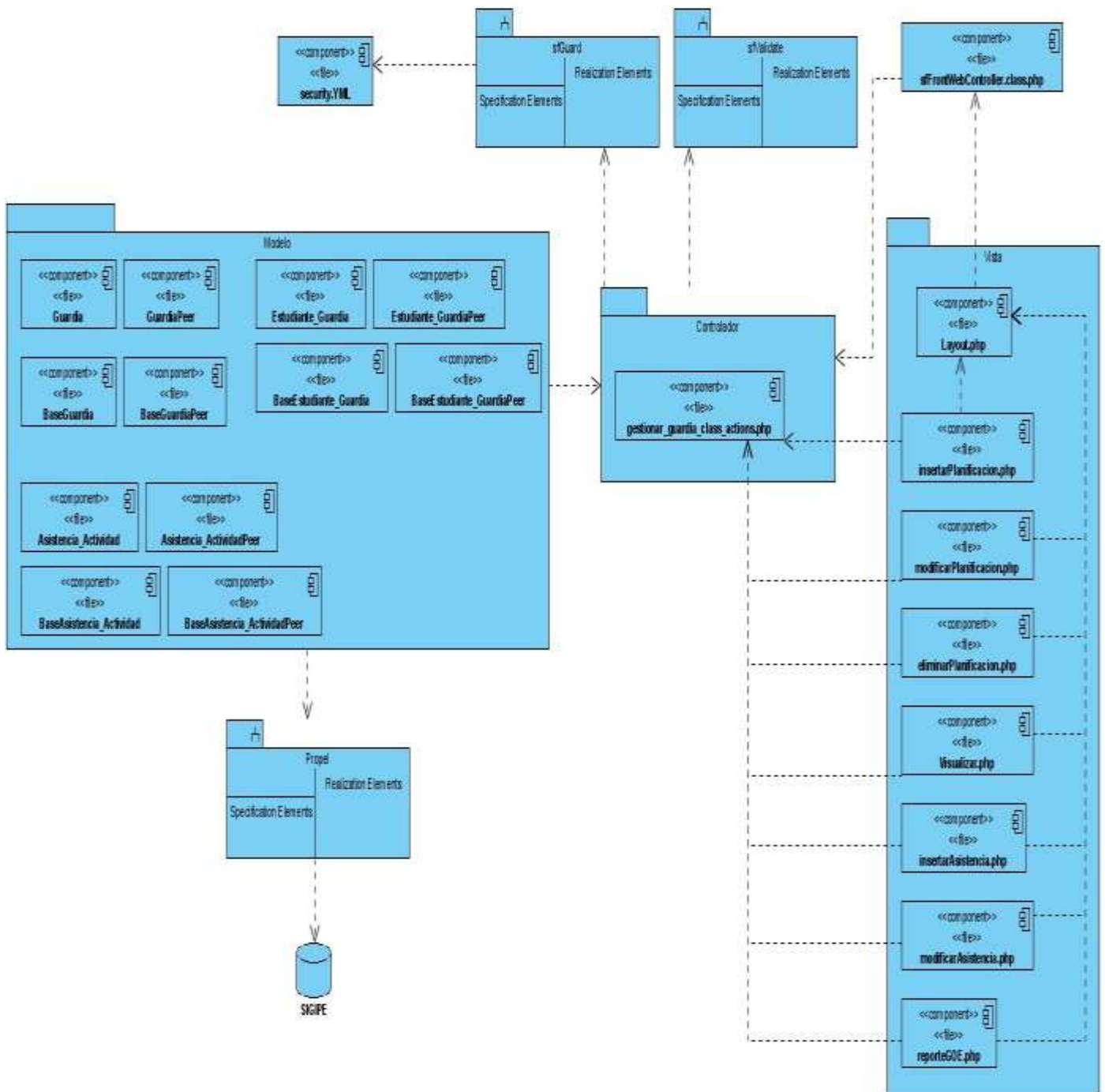
Diagramas de componentes CU "Gestionar datos de la instructora"



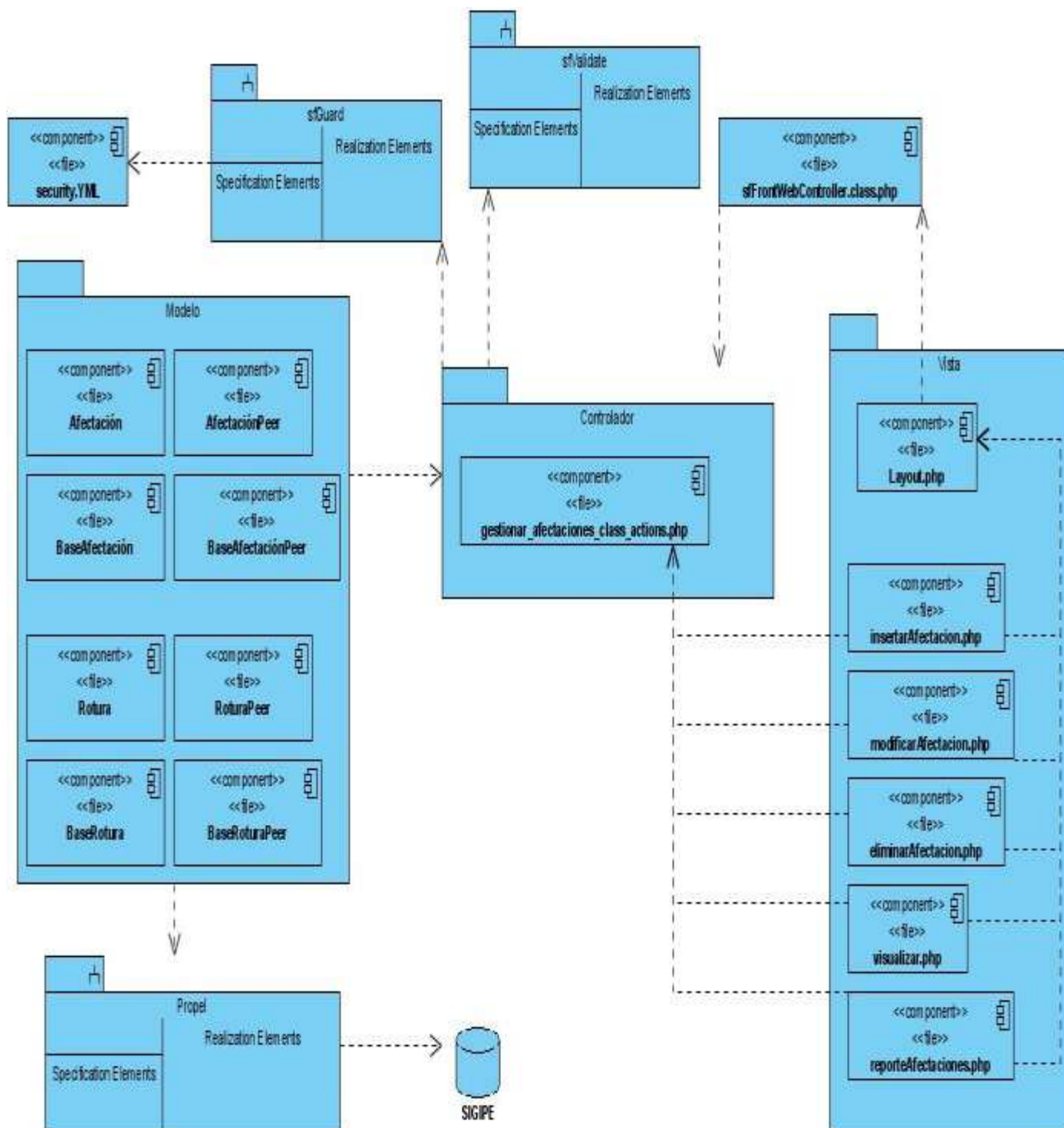
Diagramas de componentes CU "Gestionar Cuartelería"



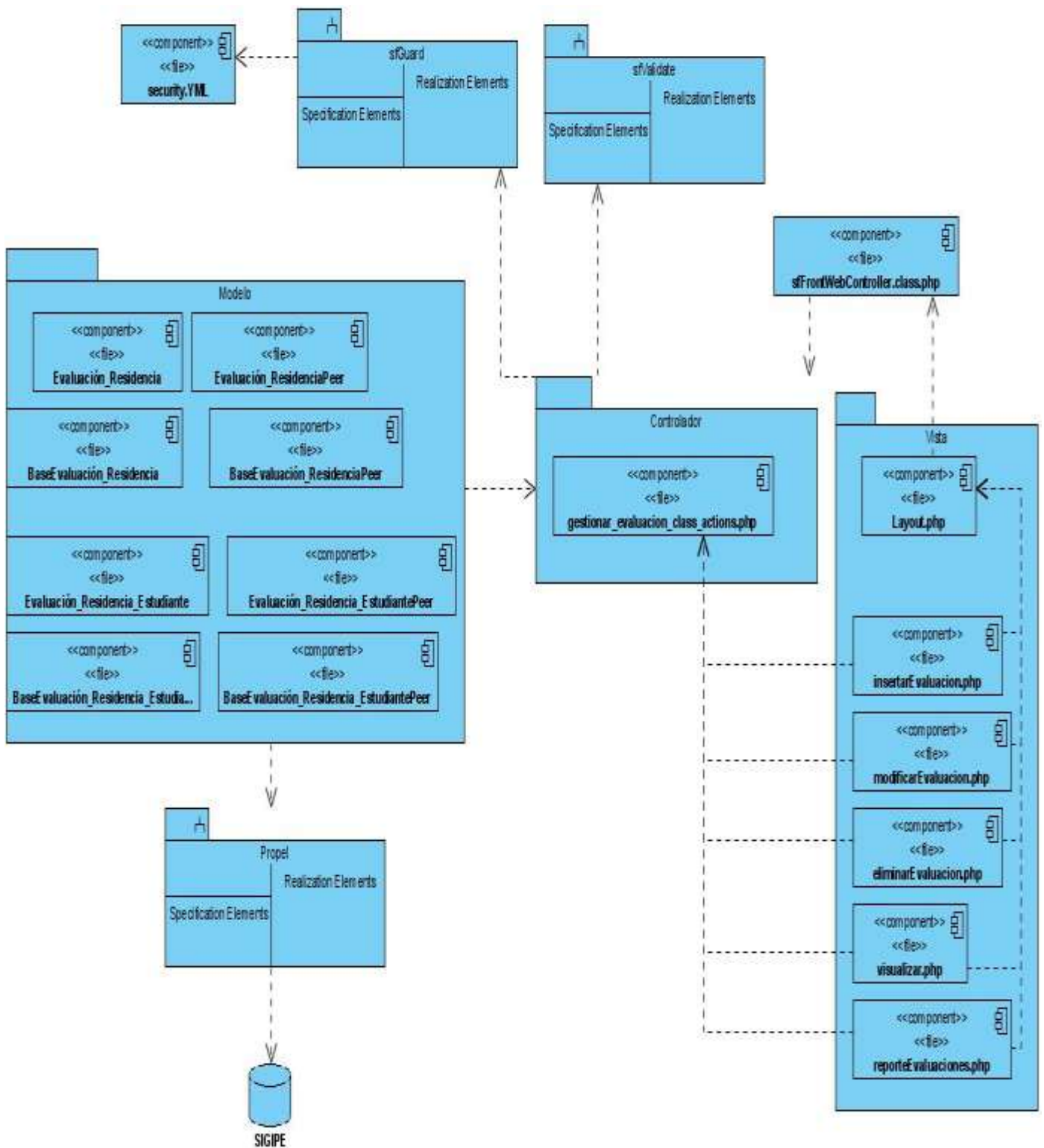
Diagramas de componentes CU" Gestionar Guardia Obrera Estudiantil"



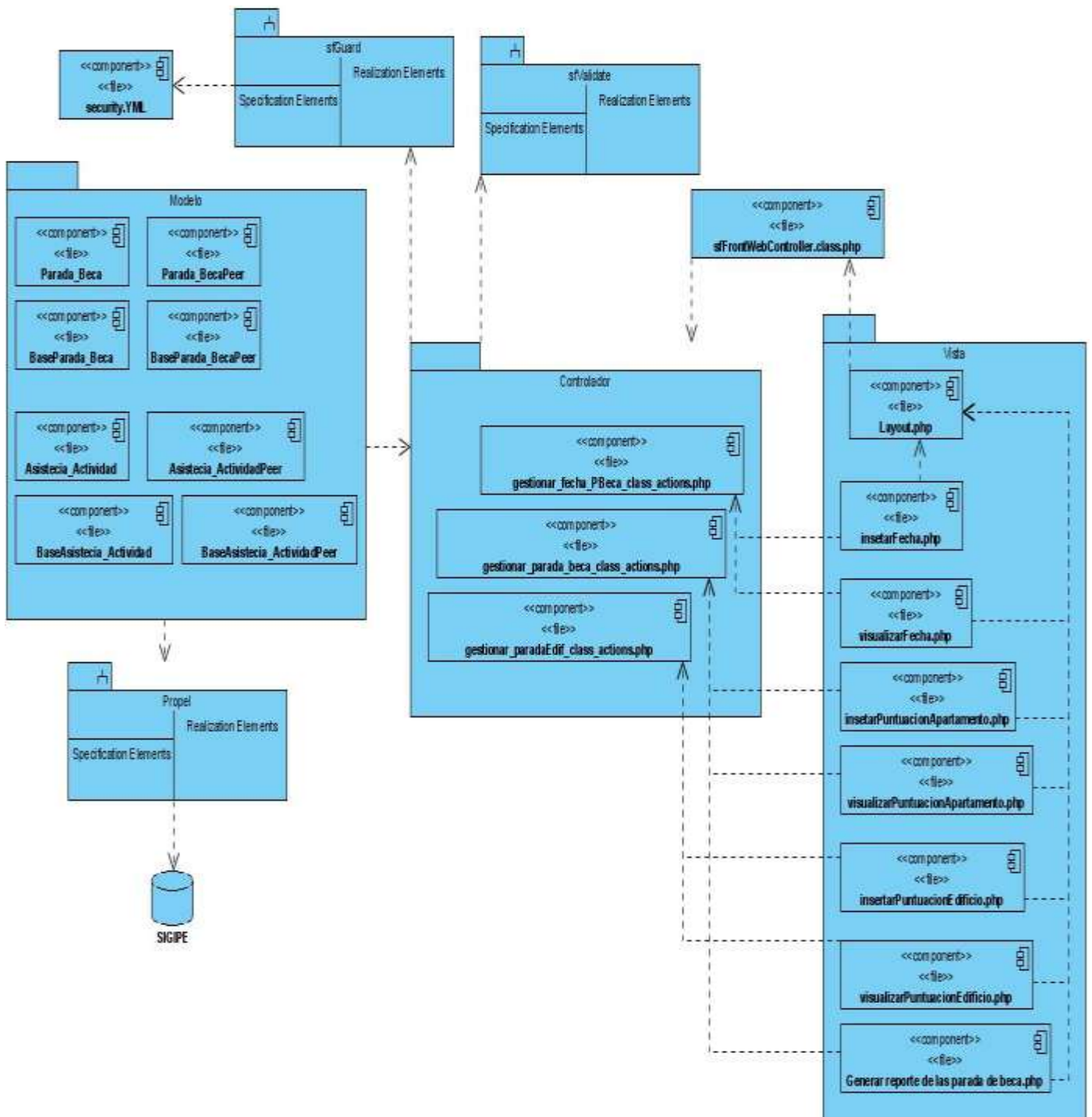
Diagramas de componentes CU "Gestionar afectaciones de los apartamentos"



Diagramas de componentes CU” Gestionar evaluación del estudiante en la residencia”



Diagramas de componentes CU” Gestionar parada de beca”



GLOSARIO DE TÉRMINOS

- Ingeniería de Software: Normas para el diseño de software que tiene como finalidad satisfacer las necesidades del cliente
- Casos de uso: Documentan el comportamiento de un sistema desde el punto de vista del usuario. Determinan los requisitos funcionales del sistema.
- Patrón de diseño: Son un conjunto de instrucciones predefinidas y comprobadas como soluciones a problemas muy específicos dentro del diseño de software.
- DOM (Document Object Model): es esencialmente un modelo computacional a través de la cual los programas y scripts pueden acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML.
- Framework: En el desarrollo de software, es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.
- Herramientas CASE: las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo y de dinero.
- HTML: es el acrónimo inglés de HyperText Markup Language, que se traduce al español como Lenguaje de Etiquetas de Hipertexto¹. Es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.
- GOF: Expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas.
- GRASP: Patrones generales de software para asignación de responsabilidades.
- Scripts: guión o conjunto de instrucciones. Permiten la automatización de tareas creando pequeñas utilidades.

- **Administrador:** es la persona que tiene privilegios para determinadas funcionalidades del sistema.
- **Requisito Funcional:** Requisito que especifica una acción que debe ser capaz de realizar el sistema, sin considerar restricciones físicas. Requisito que especifica comportamiento de entrada/salida de un sistema.
- **Requisito No Funcional:** Requisito que especifica propiedades del sistema, como restricciones del entorno o de implementación, rendimiento, dependencias de la plataforma, mantenibilidad, extensibilidad o fiabilidad. Requisito que especifica restricciones físicas sobre un requisito funcional.
- **Caso de uso:** es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.
- **Clase:** Elemento de software que describe o caracteriza una entidad del mundo real o un entidad del espacio de implementación.
- **Interfaz:** Es la parte de un programa informático que permite a este comunicarse con el usuario o con otras aplicaciones permitiendo el flujo de información.