

Universidad de las Ciencias Informáticas

Facultad 6



TÍTULO: “SISTEMA PARA LA GESTIÓN DE LA INFORMACIÓN DE LOS LABORATORIOS DE LA DIRECCIÓN DE CALIDAD DEL CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA: DISEÑO DEL MÓDULO DE LA SECCIÓN DE MEJORAMIENTO DE LA CALIDAD”

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

Autores: Marielis Sparis Chil.

Jorge Iván Fernández Oro.

Tutor: Ing. Alieski Sarmiento Almenares.

Co-tutor: Lic. Miulkenia Navarro Reyes.

Ciudad de La Habana, Cuba

Junio, 2008

“Año 50 de la Revolución”

“Haz de tu vida un sueño, y de tu sueño una realidad.”

Antoine De Saint Exupery

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_____.

Marielis Sparis Chil.

Jorge Iván Fernández Oro

Firma del Autor

Firma del Autor

Alieski Sarmiento Almenares

Miulkenia Navarro Reyes

Firma del Tutor

Firma del Co-Tutor

DATOS DE CONTACTO.

Tutor:

Ing. Alieski Sarmiento Almenares.

Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba.

E-mail: asarmiento@uci.cu

Co-tutor: Miulkenia Navarro Reyes

Universidad de las Ciencias Informáticas

E-mail: mnavarror@uci.cu

Dedicatorias

Dedico este trabajo a mis padres que tanto quiero y admiro y que son mi ejemplo a seguir en la vida.

A mi hermana que la adoro.

A mi familia en sentido general, que tanto ha hecho por mi para que pudiera llegar este día (Mima, Piqui, abuelo y abuela, tía Dania, Naimita, tía María, tío Pepe, Irán, tío Gui, Leo).

A mi amiga del alma, Meivys, por ser tan especial.

A Betty y a Jorgito mis amigos de la Universidad, por estar siempre a mi lado y compartir tantos momentos a lo largo de toda la carrera, nunca los voy a olvidar.

Marielis

Dedico primero que todo este trabajo de diploma a mis padres Mireya Oro y Fidel Fernández por siempre estar ahí para mí, por ser siempre mis guías y ejemplo a seguir, por su paciencia y dedicación y por su apoyo en todo.

A mis abuelos Teresa, Tomasa y Mario, por la ayuda inigualable que me brindaron, por darme ese amor que solo ellos saben ofrecer.

A mi hermana Yaremi.

A toda mi familia en general (Mamita, Pancho, Joseito, Yuli, Papito, Tony, Cristian, Libia, al negro, a Miriam, Mayito) por ser ellos también de una forma u otra motivo de inspiración y dedicación.

A Marielis, amiga incondicional y dúo de tesis por estar siempre a mi lado desde los inicios.

A mis amigos, porque más que amigos son mis hermanos y siempre hemos estado trabajando, compartiendo y riendo juntos, por siempre darme su apoyo y ayudarme en los momentos que los necesitaba, a todos ellos, reciban mi cariño.

Y muy en especial a Jessica.

Jorge Iván

Agradecimientos

Durante la elaboración de este trabajo influyeron determinadas personas que de una forma u otra propiciaron su realización con mucho esmero y esfuerzo.

Es por ello que quisiéramos aprovechar este momento para expresarles nuestros más sinceros agradecimientos:

Agradecemos primeramente a nuestros padres que siempre han estado pendientes de nosotros y que han sabido guiarnos por el camino correcto con su amor y dedicación.

A todas nuestras amistades de la Universidad que siempre nos han dado ánimo en los momentos más difíciles y que han sido parte de una etapa inolvidable de nuestras vidas.

A nuestro tutor Alieski Sarmiento por su apoyo y su asesoramiento continuo.

A los que han sido nuestros profesores aquí en la UCI por la formación e instrucción que nos han dado.

A todos, muchas gracias.

Resumen

El Centro de Ingeniería Genética y Biotecnología cuenta con una Dirección de Calidad, en la cual se manipula un elevado volumen de documentación, lo que dificulta la búsqueda y el intercambio de información. Con el objetivo de mejorar la gestión de ésta, se está desarrollando un Sistema para la Gestión de la Información de los Laboratorios (LIMS) en esta área de la calidad.

Este LIMS está dividido en varios módulos y uno de ellos es el de la Sección de Mejoramiento de la Calidad. En el presente trabajo se aborda el proceso de diseño de dicho módulo, siguiendo la metodología RUP, elaborando algunos de los artefactos obtenidos por los roles de diseñador, diseñador de interfaz de usuario y arquitecto de software que propone dicha metodología, haciendo uso de herramientas de actualidad como la herramienta de modelado Visual Paradigm y el framework Symfony.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1 INTRODUCCIÓN	5
1.2 LA GESTIÓN DE LA INFORMACIÓN	5
1.3 SISTEMA DE GESTIÓN DE INFORMACIÓN	7
1.4 SISTEMA DE GESTIÓN DE LA INFORMACIÓN DE LOS LABORATORIOS	9
1.5 PROCESO DE DESARROLLO DE SOFTWARE A UTILIZAR: RATIONAL UNIFIED PROCESS (RUP)	10
1.6 DISEÑO SEGÚN EL RUP	11
1.6.1 Propósito del diseño.....	11
1.7 RoL	12
1.7.1 Roles desempeñados.....	12
1.8 ARTEFACTOS	13
1.9 VISUAL PARADIGM 6.1: HERRAMIENTA DE MODELADO	15
1.10 LENGUAJE DE PROGRAMACIÓN: PERSONAL HOME PAGE (PHP)	16
1.11 FRAMEWORK A UTILIZAR: SYMFONY 1.0.11	16
1.12 PATRONES DE DISEÑO	17
1.12.1 Ventajas de los patrones de diseño.....	20
1.13 CONCLUSIONES	20
CAPÍTULO 2. DISEÑO DEL SISTEMA	21
2.1 INTRODUCCIÓN	21
2.2 REFERENCIA A LA ARQUITECTURA DEL SISTEMA	21
2.2.1 Vista Lógica.....	22
2.2.2 Vista de Despliegue.....	24
2.3 DISEÑO DEL SISTEMA	26
2.3.1 Mapa de Navegación.....	26
2.3.2 Prototipos de interfaz de usuario por caso de uso del sistema.....	29
2.3.3 Diagramas de clases del diseño por caso de uso del sistema.....	32
2.3.4 Diagramas de interacción por escenario de caso de uso del sistema.....	45
2.4 VALIDACIÓN DEL DISEÑO REALIZADO	69

2.5 CONCLUSIONES.....	89
CONCLUSIONES.....	90
RECOMENDACIONES.....	91
REFERENCIAS BIBLIOGRÁFICAS.....	92
BIBLIOGRAFÍA.....	94
ANEXOS	96
GLOSARIO DE TÉRMINOS	104

ÍNDICE DE FIGURAS

Fig. 1 Representación de RUP con sus fases, iteraciones y flujos de trabajo.	11
Fig. 2 Plantilla decorada con un layout.....	18
Fig. 3 Funcionamiento del patrón MVC.	21
Fig. 4 Vista Lógica.	23
Fig. 5 Diagrama de Despliegue.....	25
Fig. 6 Mapa de navegación.....	26
Fig. 7 Mapa de navegación: Diagrama #2.....	27
Fig. 8 Mapa de navegación: Diagrama #3.....	28
Fig. 9 Prototipo de Crear Registro de Control de Cambios.....	29
Fig. 10 Prototipo de Buscar Registro de Control de Cambios.....	29
Fig. 11 Prototipo de Visualizar Registro de Control de Cambios	30
Fig. 12 Prototipo de Modificar Registro de Control de Cambios	30
Fig. 13 Prototipo de Registrar Registro de Control de Cambios	31
Fig. 14 Prototipo de Generar Reporte del Registro de Control de Cambios	31
Fig. 15 Prototipo de Visualizar Reporte del Registro de Control de Cambios	32
Fig. 16 CU Gestionar Registro de Control de Cambios (SIC-0816).....	34
Fig. 17 CU Gestionar Reporte de Límites (SIC-0963).	35
Fig. 18 CU Gestionar Programa de Monitoreo Ambiental (SIC-0098).	36
Fig. 19 CU Gestionar Acciones Correctivas (SIC-0848).....	37
Fig. 20 CU Gestionar Seguimiento de Acciones Correctivas (SIC-0858).....	38
Fig. 21 CU Gestionar Programa de Informes de Tendencias (SIC-0919).....	39
Fig. 22 CU Gestionar Registro de Control de las Desviaciones (SIC-0851)	40
Fig. 23 CU Gestionar Registro de Devoluciones (SIC-0023).....	41
Fig. 24 CU Gestionar Registro de Retirada de Productos del Mercado (SIC-0024).....	42
Fig. 25 CU Gestionar Notificación de Queja (SIC-0042).....	43
Fig. 26 CU Gestionar Cierre de Queja (SIC-0959)	44
Fig. 27 Paquete Modelo.	45
Fig. 28 Escenario Buscar Programa de Monitoreo Ambiental	47
Fig. 29 Escenario Modificar Programa de Monitoreo Ambiental	48

Fig. 30 Escenario Generar Reporte del Programa de Monitoreo Ambiental	49
Fig. 31 Escenario Crear Registro de Control de Cambios	50
Fig. 32 Escenario Buscar Registro de Control de Cambios	51
Fig. 33 Escenario Registrar Registro de Control de Cambios.....	52
Fig. 34 Escenario Modificar Registro de Control de Cambios.....	53
Fig. 35 Escenario Generar Reporte del Registro de Control de Cambios.....	54
Fig. 36 Escenario Crear Programa de Informe de Tendencias.....	55
Fig. 37 Escenario Buscar Programa de Informe de Tendencias.....	56
Fig. 38 Escenario Registrar Programa de Informe de Tendencias	57
Fig. 39 Escenario Modificar Programa de Informe de Tendencias	58
Fig. 40 Escenario Generar Reporte del Programa de Informe de Tendencias	59
Fig. 41 Escenario Crear Registro de Control de las Desviaciones	60
Fig. 42 Escenario Modificar Registro de Control de las Desviaciones.....	61
Fig. 43 Escenario Buscar Registro de Control de las Desviaciones	62
Fig. 44 Escenario Generar Reporte del Registro de Control de las Desviaciones	63
Fig. 45 Escenario Crear Reporte de Límites	64
Fig. 46 Escenario Buscar Reporte de Límites	65
Fig. 47 Escenario Registrar Reporte de Límites	66
Fig. 48 Escenario Modificar Reporte de Límites	67
Fig. 49 Escenario Generar Reporte de Reporte de Límites.....	68
Fig. 50 Escenario Crear Notificación de quejas.....	71
Fig. 51 Escenario Buscar Notificación de quejas.....	74
Fig. 52 Escenario Modificar Notificación de quejas	77
Fig. 53 Escenario Generar Reporte de la Notificación de quejas	80
Fig. 54 Representación de la Estructura jerárquica de la Dirección de Calidad del CIGB.	96
Fig. 55 Representación de la Vista Lógica.	97
Fig. 56 Prototipo de Crear Programa de Monitoreo Ambiental	98
Fig. 57 Prototipo de Buscar Programa de Monitoreo Ambiental	98
Fig. 58 Prototipo de Visualizar Programa de Monitoreo Ambiental.....	99
Fig. 59 Prototipo de Modificar Programa de Monitoreo Ambiental.....	99
Fig. 60 Prototipo de Crear Programa de Informes de Tendencias.....	100
Fig. 61 Prototipo de Buscar Programa de Informes de Tendencias.....	100
Fig. 62 Prototipo de Visualizar Programa de Informes de Tendencias	101
Fig. 63 Prototipo de Modificar Programa de Informes de Tendencias	101
Fig. 64 Prototipo de Registrar Programa de Informes de Tendencias	102
Fig. 65 Prototipo de Generar Reporte del Programa de Informes de Tendencias	102
Fig. 66 Prototipo de Visualizar Reporte del Programa de Informes de Tendencias	103
Fig. 67 El flujo de trabajo de Symphony.....	103

Introducción

En la actualidad las Tecnologías de la Información y las Comunicaciones (TIC) ocupan un lugar importante en el desarrollo de la sociedad y la economía.

Cuba ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las TIC, así como lograr una cultura digital como una de las características imprescindibles del hombre.

A partir de la década del 80 del siglo pasado, se intensificó en el país la actividad científica aplicada a la salud, que tuvo como colofón la inauguración, el 1ro de julio de 1986, del Centro de Ingeniería Genética y Biotecnología (CIGB), y la creación en 1992 del Polo Científico en el oeste de La Habana. Este Centro marcó la mayoría de edad del pujante movimiento científico de la Isla.

La calidad, es la imagen del CIGB, los productos desarrollados y elaborados en este centro se caracterizan por su eficacia y seguridad. Para lograr este objetivo el CIGB cuenta con una **Dirección de Calidad** compuesta por los Departamentos de **Control de la Calidad y Aseguramiento de la Calidad**.

El **Departamento de Control de la Calidad** tiene entre sus funciones fundamentales las relacionadas con el muestreo, las especificaciones, los ensayos y la evaluación de la calidad de los productos que se generan en la empresa. Además debe comprobar y poner en práctica todos los procedimientos de control, evaluar, mantener y almacenar los materiales de referencia, asegurar que se controle la estabilidad de los ingredientes farmacéuticos activos y los productos terminados. El Departamento de Control de la Calidad es el responsable de autorizar o rechazar las materias primas y los productos intermedios. Para acometer estas funciones, el Departamento está compuesto por dos secciones y dos grupos de trabajo: [1]

- Grupo de Desarrollo.
- Grupo de Liberación Analítica.
- Sección biológica compuesta por cinco laboratorios:
 - Laboratorio de Microbiología.
 - Laboratorio de Biología Molecular.
 - Laboratorio de Ensayos Biológicos I.

- Laboratorio de Ensayos Biológicos II.
- Laboratorio de Inmunoquímica.
- Sección físico-química compuesta por tres laboratorios:
 - Laboratorio Análisis Químico.
 - Laboratorio de Cromatografía y Electroforesis.
 - Laboratorio de Sistemas Críticos.

El **Departamento de Aseguramiento de la Calidad** garantiza que se lleven a cabo las acciones planificadas y sistemáticas que son necesarias para proporcionar la confianza de que nuestros productos y servicios satisfacen los requisitos de calidad establecidos. Vela por el cumplimiento de las Buenas Prácticas de Producción (BPP), Buenas Prácticas de Laboratorio (BPL) y Buenas Prácticas Clínicas (BPC). Este Departamento está compuesto por dos Secciones y dos grupos de trabajo: [2]

- **Sección de Mejoramiento de la Calidad (SMC).**
- Sección de Inspección, Auditoría y Liberación de lotes.
 - Grupo de Inspección y Auditorías.
 - Grupo de Liberación de Lotes.
- Grupo de Documentación.
- Grupo de Metrología.

(Ver Anexo#1) Representación de la Estructura jerárquica de la Dirección de Calidad del CIGB.

La **Sección de Mejoramiento de la Calidad** tiene como función principal el mejoramiento de la calidad de las producciones del centro.

En esta sección se realizan los trámites de las quejas, reclamaciones y devoluciones, retirada de productos del mercado, así como de insatisfacción de los clientes.

Con el objetivo de analizar el comportamiento de las características de los diferentes productos en un período de tiempo determinado, se realizan los estudios de tendencias de los productos intermedios y finales.

Otras funciones desarrolladas en esta sección son: [2]

- Dictar los lineamientos generales para la tarea de validación.
- Proceso de evaluación, aprobación y selección de los proveedores.
- Control de productos no conformes.
- Control de cambios de los procesos y las técnicas analíticas.
- Elaboración y coordinación de los Programas de Monitoreo Ambiental de las áreas limpias de las diferentes producciones del CIGB.

Este proceso genera grandes volúmenes diarios de información impresa, lo que dificulta el almacenamiento organizado, confiable y duradero de esta última.

El gran volumen de la documentación generada, en la mayoría de los casos, es revisado, supervisado y aprobado por las personas con rangos superiores, lo que en ocasiones puede provocar pérdida de tiempo considerable, cuando se necesita de un tiempo de respuesta mínimo, para la toma de decisiones.

Es importante decir que la cantidad de información generada es constantemente consultada, se emplean numerosos medios de oficina por parte de los trabajadores de la SMC en la obtención y generación de diversos reportes, propios de la labor de supervisión. Todo lo anteriormente expuesto dificulta la búsqueda y el intercambio de información.

En el CIGB, específicamente en el área de la Calidad de este Centro se hizo un estudio con el objetivo de incorporar nuevas tecnologías de la información y con esto lograr facilitar y agilizar el trabajo, para lo cual se decidió desarrollar un sistema de gestión de la información de los laboratorios (LIMS). Debido a que los miembros de la **Sección de Mejoramiento de la Calidad** tienen la obligación de manipular un gran volumen de documentación se decidió tratar de mejorar la gestión de la información en esta sección y se incorporó como un módulo a desarrollar para este LIMS. Hasta el mes julio del año 2007 se estuvieron realizando los flujos de trabajo **Modelamiento del negocio y Requerimientos** de este módulo, en este trabajo se le dará continuidad al proceso ya comenzado.

Basándonos en el trabajo realizado y el desempeño alcanzado anteriormente podemos identificar como **problema a resolver**: ¿Cómo traducir los requerimientos identificados para el módulo de la **Sección de Mejoramiento de la Calidad** del LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología en elementos que puedan ser implementados?

El problema planteado se enmarca en el **objeto de estudio**: Proceso de desarrollo de los sistemas de gestión de la información.

El campo de acción: Proceso de diseño de los sistemas de gestión de la información, para lo cual se trazó como **objetivo general:** Diseñar el módulo de la **Sección de Mejoramiento de la Calidad** para el LIMS de Control de la Calidad del CIGB.

Para el cumplimiento del objetivo planteado se desarrollarán las siguientes tareas:

- Estudio del Modelamiento del Negocio.
- Estudio del Modelo del Sistema.
- Realización de los CU-diseño: diagramas de clases del diseño y diagramas de interacción.
- Realización del diagrama de despliegue.
- Realización del mapa de navegación.

El presente trabajo consta de introducción, 2 capítulos, conclusiones, recomendaciones, bibliografía y anexos.

En el Capítulo 1 Fundamentación Teórica: se brinda una panorámica sobre el proceso de gestión de la información y los sistemas de gestión de la información, se brinda información sobre la metodología de desarrollo de software y de las herramientas a utilizar. Se realiza una breve descripción del diseño y de los roles a desempeñar así como de los artefactos a realizar.

En el Capítulo 2 Diseño del Sistema: se hace una breve referencia a la arquitectura, se realiza una modelación del sistema con la obtención de varios artefactos y se hace una validación del diseño realizado.

Capítulo 1. Fundamentación teórica.

1.1 Introducción

En el presente capítulo se brinda una panorámica sobre el proceso de gestión de la información y los sistemas de gestión de la misma. Se brinda información sobre el proceso de desarrollo de software RUP (Rational Unified Process) y sobre la herramienta CASE (Computer Aided Software Engineering) a utilizar, Visual Paradigm. Se realiza una breve descripción del diseño, así como de los roles a desempeñar y algunos de los artefactos obtenidos por dichos trabajadores del RUP. Además se realiza un estudio sobre el framework Symfony.

La metodología de desarrollo de software RUP, la herramienta de modelado Visual Paradigm, el lenguaje de programación PHP 5 y el framework de desarrollo Symfony han sido escogidos debido a sus diversas características y ventajas, por el proyecto LIMS de la Dirección de Calidad del CIGB, al cual pertenece este módulo.

1.2 La gestión de la información.

La gestión de la información es el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores (de todos los niveles) tomar decisiones documentadas. La información para la gestión es la información necesaria para tomar decisiones de gestión. [3] Es el proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información dentro y para la sociedad a la que sirve. Tiene como elemento básico la gestión del ciclo de vida de este recurso y se desarrolla en cualquier organización. [4]

En la era de la información, uno de los principales problemas es su exceso, es necesario invertir mucho tiempo en ella debido, entre otras causas, a la liberación de los mecanismos regulatorios existentes en materia de publicaciones, sobre todo como resultado del surgimiento y desarrollo de Internet; por esta razón, es frecuente encontrar un número significativo de publicaciones redundantes y de baja calidad mezcladas con otras importantes y sólidas, difíciles de hallar entre la información ruidosa que la "envuelve".

La información es un elemento fundamental para el desarrollo, con el pasar de los años, la gestión de la información ocupa, cada vez más, un espacio mayor en la economía de los países a escala mundial.

La información se diferencia por ser:

- Expandible
- Comprimible
- Sustituible
- Difusa
- Compartida

Los servicios de información, como parte esencial de la infraestructura para la gestión del conocimiento, suministran información, impulsan la generación del conocimiento para la búsqueda de soluciones a los problemas que enfrentan las organizaciones, analizan su impacto sobre los resultados de las empresas e influyen en el comportamiento de los individuos ante la información. La gestión de la información se vincula con la generación y la aplicación de estrategias, el establecimiento de políticas, así como con el desarrollo de una cultura organizacional y social dirigida al uso racional, efectivo y eficiente de la información en función de los objetivos y metas de las compañías en materia de desempeño y de calidad.

Las estrategias actuales para la gestión de la información y el conocimiento deben responder a los nuevos tipos de demandas, resultantes de la aparición de tendencias gerenciales más modernas en las organizaciones. En la creación de los nuevos sistemas de gestión de la información es imprescindible considerar las fuentes factográficas (datos), documentales y no documentales, los sistemas informáticos, la cultura de información, los modelos de comunicación, entre otros elementos. Según los requerimientos de los procesos internos de trabajo y los flujos de información propios, todos ellos deben propiciar la gestión del conocimiento organizacional y la implementación de sistemas de gestión de la calidad para la evaluación de los resultados y los proyectos de la institución. Se requiere, además, de la incorporación de nuevos valores a los productos y servicios de información, así como de una diseminación muy bien dirigida, con el fin de que ellos lleguen a aquellos individuos y secciones cuya actividad de generación o aplicación del conocimiento y de toma de decisiones es más importante para la empresa.

Una adecuada gestión de la información, en el contexto de una gerencia de la calidad, posibilita reducir los riesgos en la administración de la organización, como son la toma de decisiones apresuradas, tardías o inconsistentes, la entrada al mercado con productos no competitivos, entre otros, que ocasionan pérdidas y reducen su competitividad en el mercado. Obtener la información necesaria, con la calidad requerida, es una premisa indispensable para la supervivencia de las empresas, si se

considera que las organizaciones acortan cada vez más sus ciclos estratégicos y que la toma de decisiones, así como el cambio es continua.

La información es un agente importante en la modificación de las conductas existentes en la organización, su correcta gestión es una herramienta fundamental para la toma de decisiones, la formación del personal, la evaluación de los productos, la determinación de los errores y el control de los procesos. [5]

Funciones de la Gestión de Información: [4]

- Determinar necesidades internas de información, relativas a las funciones, actividades y procesos administrativos de la organización y a su satisfacción.
- Optimizar el flujo organizacional de la información y el nivel de la comunicación.
- Manejar eficientemente los recursos organizacionales de información, mejorar las inversiones sucesivas en los mismos y optimizar su aprovechamiento.
- Contribuir a modernizar u optimizar las actividades organizativas y los procesos administrativos relacionados con los mismos.
- Garantizar la calidad de los productos de la organización y asegurar su diseminación efectiva.
- Determinar las necesidades de información externa de la organización y satisfacerlas.

1.3 Sistema de Gestión de Información

¿Qué es un Sistema de Gestión de Información? (SGI) [4]

Un Sistema de Información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

Entrada de Información: Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas. Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáneres, la voz, los monitores sensibles al tacto, el teclado y el mouse, entre otras.

Almacenamiento de información: El almacenamiento 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. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).

Procesamiento de Información: Es la capacidad del Sistema 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. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

Salida de Información: La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interfaz automática de salida.

Los SGI permiten:

- Comprender la marcha de las organizaciones desde un enfoque analítico (donde queremos estar), evaluador (donde estamos) y creativo (donde podríamos estar).
- Develar oportunidades que merezcan ser explotadas y contrarrestar amenazas.
- Establecer los factores que resulten críticos y las necesidades asociadas al SGI.
- Estudiar el impacto de los SGI en la posición del negocio y buscar nuevas oportunidades

El Diseño de un Sistema de Gestión de Información precisa de:

- Un análisis previo de las necesidades de Información de la organización.
- Un diagnóstico de la situación.

- Una auditoría de información que permita conocer los recursos de información disponibles y los que faltan, para qué y quienes lo utilizan, que valor se le añade en su uso, entre otros.

Es importante tener en cuenta que los resultados que se obtiene de un SGI dependen de la calidad de la información y su validez, por otro lado la información derivada del SGI dependerá de la exactitud y la eficiencia de los procedimientos usados en la recolección de la información sobre insumos procesos y resultados, por lo que en este sentido el diseño del sistema debe garantizar que los flujos de información en la organización sean lo más exactos posibles de manera que la información que salga del sistema sea correcta y lo mas ágil posible y pueda considerarse como inteligencia organizacional.

Los Sistemas de Gestión de Información constituyen hoy, no sólo soportes de los negocios, sino, además, un instrumento de ventajas competitivas sostenibles al permitir gestionar los activos tangibles e intangibles y convertirse en una herramienta integral de gerencia.

Considerar a la información base del conocimiento, la inteligencia y el talento organizacional, como única fuente de ventaja competitiva sostenible, coloca a las organizaciones en condiciones para el aprendizaje y la innovación.

1.4 Sistema de Gestión de la Información de los Laboratorios.

Un LIMS o "Laboratory Information Management System" es un programa de gestión de laboratorios que permite recoger, almacenar, calcular y gestionar datos en una amplia variedad de formas. Los LIMS representan una importante herramienta para la gestión global de un laboratorio en un entorno de calidad, agilizando temas de registro de datos primarios, archivo, trazabilidad, etc. y minimizando los errores debidos a la transferencia de información. [6]

Un LIMS maneja los datos pero necesita ser más que una simple hoja de cálculo. El objetivo de un LIMS es ayudar al funcionamiento de un laboratorio eficientemente, proporciona una forma conveniente de almacenar datos de laboratorio y el manejo en general de las funciones del mismo y puede ser usado para procesar resultados de instrumentos.

La mejora inmediata vista con la introducción de LIMS consiste en que ya no se requiere una memoria excelente ofreciendo una gran capacidad de almacenamiento de datos. La tecnología de la Base de Datos (DB) les permite a los usuarios de LIMS buscar por rasgos claves dentro de los datos e intercalar los resultados de esa búsqueda.

1.5 Proceso de Desarrollo de Software a utilizar: Rational Unified Process (RUP).

El proceso de desarrollo de software a utilizar es el Proceso Unificado de Desarrollo (RUP), el cual junto con el Lenguaje Unificado de Modelado UML, constituye el proceso estándar más utilizado para el análisis, implementación y documentación de sistemas orientados a objetos. RUP divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante.

Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible. [7]

Como características fundamentales de RUP se tiene que es: [7]

- Guiado por Casos de Uso

Un sistema de software se crea para servir a sus usuarios. Por lo tanto, para construir un sistema exitoso se debe conocer qué es lo que quieren y necesitan los usuarios prospectos. El término usuario se refiere no solamente a los usuarios humanos, sino a otros sistemas. En este contexto, el término usuario representa algo o alguien que interactúa con el sistema por desarrollar.

- Iterativo e Incremental

Desarrollar un producto de software comercial es una tarea enorme que puede continuar por varios meses o años. Es práctico dividir el trabajo en pequeños pedazos o mini-proyectos. Cada mini-proyecto es una iteración que finaliza en un incremento. Las iteraciones se refieren a pasos en el flujo de trabajo, los incrementos se refieren a crecimiento en el producto.

- Centrado en la Arquitectura

El papel del arquitecto de sistemas es similar en naturaleza al papel que el arquitecto desempeña en la construcción de edificios. El edificio se mira desde diferentes puntos de vista: estructura, servicios, plomería, electricidad, etc. Esto le permite al constructor ver una radiografía completa antes de empezar a construir. Similarmente, la arquitectura en un sistema de software es descrita como diferentes vistas del sistema que está siendo construido. La arquitectura es la vista del diseño completo con las características más importantes hechas más visibles y dejando los detalles de lado.

De acuerdo al proceso de desarrollo de software a utilizar (RUP), anteriormente se estuvieron realizando los flujos de trabajo de **Modelamiento del negocio** y **Requerimientos** para el módulo de la

SMC, en este trabajo se va a realizar del flujo de trabajo de Análisis y Diseño la parte de **Diseño** de este módulo.

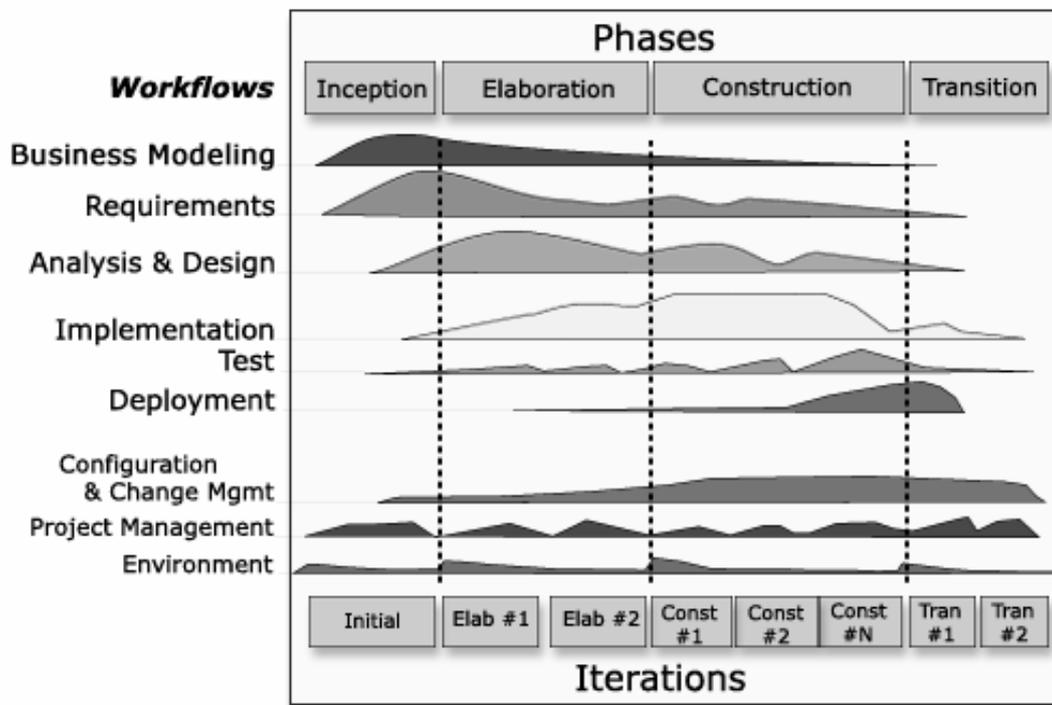


Fig. 1 Representación de RUP con sus fases, iteraciones y flujos de trabajo.

1.6 Diseño según el RUP.

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. En el diseño modelamos el sistema y encontramos su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requisitos. Además impone una estructura del sistema que debemos esforzarnos por conservar lo más fielmente posible cuando demos forma al sistema.

1.6.1 Propósito del diseño.

El diseño tiene como propósito:

- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables,

sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.

- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
- Capturar las interfaces entre los subsistemas antes en el ciclo de vida del software, lo cual es muy útil cuando utilizamos interfaces como elementos de sincronización entre diferentes equipos de desarrollo.

1.7 Rol.

Un rol define el comportamiento y responsabilidades de una persona, o un conjunto de individuos que trabajan juntos como un equipo, en el contexto de una organización de ingeniería de software.

1.7.1 Roles desempeñados.

Para el desarrollo de este trabajo los autores desempeñan los roles de diseñador, diseñador de interfaz de usuario y arquitecto de software definidos por el RUP.

Rol diseñador

El diseñador se encarga de diseñar una parte del sistema, dentro de las limitaciones de los requisitos, la arquitectura, y el proceso de desarrollo para el proyecto.

El diseñador identifica y define las responsabilidades, operaciones, atributos, y las relaciones de elementos de diseño. El diseñador asegura que el diseño es consistente con la arquitectura de software, y se detalla a un punto que se pueda proceder con la implementación.

Rol diseñador de interfaz de usuario

El diseñador de interfaz de usuario coordina el diseño de la interfaz de usuario. Los diseñadores de interfaz de usuario también están involucrados en la recopilación de requerimientos funcionales y en la realización de prototipos de interfaz de usuario diseñados para satisfacer esos requerimientos. El diseñador de interfaz de usuario se centra en la facilidad de uso del sistema.

Rol arquitecto de software

El arquitecto de software es responsable por la arquitectura de software, que incluye las principales decisiones técnicas que limitan el diseño global y la implementación del proyecto.

El arquitecto de software tiene la responsabilidad general de manejar las decisiones técnicas más importantes, expresadas en la arquitectura de software. Por lo general, esto incluye la identificación y documentación de la arquitectura de los aspectos importantes del sistema, incluidos los requisitos, diseño, implementación y despliegue, "vistas" del sistema.

1.8 Artefactos.

Prototipos de interfaz de usuario

Un prototipo de interfaz de usuario es un ejemplo de la interfaz de usuario que se construye con el fin de explorar y / o validar el diseño de la interfaz de usuario.

Los prototipos de interfaz de usuario se pueden utilizar para explorar un factible y adecuado diseño de la interfaz de usuario que cumpla los requisitos exigidos, ayudando a cerrar la brecha entre lo que se requiere (expresada a través de obtención de los requisitos), y lo que es factible.

Clases del diseño

Una clase es una descripción de un conjunto de objetos que comparten las mismas responsabilidades, las relaciones, las operaciones, atributos, y la semántica. Las clases del diseño son una parte fundamental de un diseño orientado a objetos.

Una clase de diseño es una construcción similar en la implementación del sistema:

- El lenguaje utilizado para especificar una clase del diseño es lo mismo que el lenguaje de programación.
- Las relaciones de aquellas clases del diseño implicadas por otras clases, a menudo tienen un significado directo cuando la clase es implementada.
- Los métodos de una clase del diseño tienen correspondencia directa con el correspondiente método en la implementación de las clases.
- Una clase de diseño puede posponer el manejo de algunos requisitos para las siguientes actividades de implementación, indicándolos como requisitos de implementación de la clase.

- Una clase de diseño puede proporcionar interfaces si tiene sentido hacerlo en el lenguaje de programación.

Realización de los casos de uso del diseño.

- Diagrama de clases del diseño

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema. Principalmente, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas. Los diagramas de clases también son la base para un par de diagramas relacionados: los diagramas de componentes y los diagramas de despliegue. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

- Diagramas de interacción.

Los diagramas de secuencia y los diagramas de colaboración (ambos llamados diagramas de interacción) son dos de los cinco tipos de diagramas de UML que se utilizan para modelar los aspectos dinámicos de los sistemas. Un diagrama de interacción muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes; un diagrama de colaboración es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes.

Subsistemas de diseño.

Un subsistema de diseño es la parte de un sistema que encapsula comportamiento, expone un conjunto de interfaces, paquetes y otros elementos del modelo de diseño.

Desde el exterior, un subsistema es un único elemento del modelo de diseño que colabora con otros elementos del modelo para cumplir con sus responsabilidades.

Son una forma de organizar los artefactos del modelo de diseño en piezas más manejables. Pueden estar compuestos por clases del diseño, realizaciones de casos de uso, interfaces y otros subsistemas. Pueden proporcionar interfaces que representan la funcionalidad que exportan en términos de operaciones.

Los subsistemas pueden representar una separación de aspectos del diseño.

Los subsistemas pueden representar componentes de grano grueso en la implementación del sistema, es decir componentes que proporcionan varias interfaces compuestas a partir de otros varios componentes de grano más fino, como los que especifican clases de implementación individuales.

Paquetes de diseño

Un paquete de diseño es un conjunto de clases, las relaciones, las realizaciones de casos de uso, diagramas y otros paquetes. Se utiliza para estructurar el modelo de diseño dividiéndolo en partes más pequeñas.

Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes software, objetos, procesos (caso particular de un objeto). En general un nodo será una unidad de computación de algún tipo, desde un sensor a un mainframe. Las instancias de componentes software pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz). [8]

Mapa de navegación

El mapa de navegación expresa la estructura de los elementos de la interfaz de usuario en el sistema, junto con sus posibles vías de navegación. Estas son las principales vías a través de las pantallas del sistema y no necesariamente de todos los posibles caminos. Puede ser pensado como una hoja de ruta de la interfaz de usuario del sistema.

Se realiza un mapa de navegación por sistema.

1.9 Visual Paradigm 6.1: herramienta de modelado.

Visual Paradigm es una herramienta que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Este lenguaje es usado para especificar, visualizar, construir y documentar artefactos de un sistema de

software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

Visual Paradigm es una herramienta colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto y permite control de versiones. [9]

Esta herramienta de modelado permite dibujar todos los tipos de diagramas de clases. Tiene capacidades de ingeniería directa e inversa, genera código desde diagramas y genera documentación. Permite además disponibilidad en múltiples plataformas.

1.10 Lenguaje de programación: Personal Home Page (PHP).

PHP es un lenguaje script para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es de Open Source (código abierto), es el más popular y extendido en la web. [10]

PHP 5 es una renovación total de este lenguaje que se ha extendido a numerosos aspectos, como la interacción con otras tecnologías, la nueva sintaxis de orientación a objetos, y un sinfín de novedades más. [11]

El principal objetivo de PHP5 ha sido mejorar los mecanismos de la programación orientada a objetos 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. [12]

1.11 Framework a utilizar: Symfony 1.0.11

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.

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.

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. [13] A continuación se muestran algunas de sus características.

Características de Symfony [13]

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows)
- Independiente del sistema gestor de bases de datos
- Sencillo de usar en la mayoría de los 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 sólo 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 adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros

1.12 Patrones de Diseño

¿Qué es un patrón?

Pareja de **problema / solución** con un nombre, que codifica (estandariza) buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades.

Cada patrón describe un problema que ocurre una y otra vez, y describe la solución a ese problema, de tal manera que dicha solución pueda ser usada un millón de veces más, sin hacerlo necesariamente dos veces del mismo modo.

Patrón de diseño.

Un patrón de diseño proporciona un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe estructuras comunes de la comunicación de los componentes que soluciona un problema de diseño dentro de un contexto particular.

Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.

En la realización del presente trabajo se tuvieron en cuenta algunos patrones que utiliza el framework de desarrollo elegido.

Symfony hace uso del patrón “Decorator”, al usar el archivo layout.php, el cual también se denomina plantilla global. Este archivo 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” y se muestra en la figura 2.

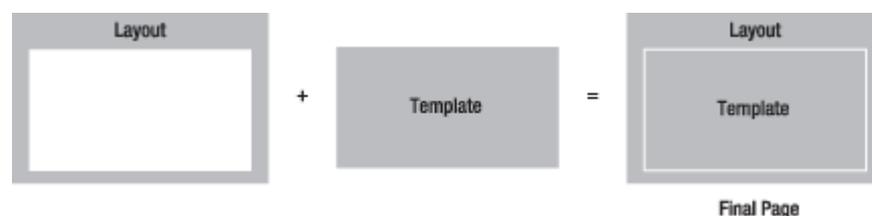


Fig. 2 Plantilla decorada con un layout

Decorator [14]

El patrón Decorator permite añadir responsabilidades a objetos concretos de forma dinámica. Los decoradores ofrecen una alternativa más flexible que la herencia para extender las funcionalidades.

Aplicación

- Para añadir responsabilidades a objetos concretos de forma dinámica y transparente, es decir, sin afectar a otros objetos.
- Para responsabilidades que se pueden añadir o quitar.

- Cuando no se puede utilizar la herencia porque daría lugar a la aparición de multitud de subclases para poder soportar todas las combinaciones posibles.

Ventajas

- Es más flexible que la herencia estática.
- Evita que las clases altas de la jerarquía estén demasiado cargadas de funcionalidad. Un componente y su decorador no son el mismo objeto.

Symfony usa también el patrón clásico de diseño web conocido como “Modelo Vista Controlador” del cual se brinda una explicación de su funcionamiento en el epígrafe 2.2.

Se tuvieron en cuenta además algunos de los patrones GRASP (patrones de asignación de responsabilidades).

Bajo Acoplamiento: Debe haber pocas dependencias entre las clases. Si todas las clases dependen de todas ¿cuanto software podemos extraer de un modo independiente y reutilizarlo en otro proyecto?

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras. [15]

Alta Cohesión: Cada elemento de nuestro diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable.

En la perspectiva del diseño orientado a objetos, la cohesión (o, más exactamente, la cohesión funcional) es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Las clases con baja cohesión a menudo representan un alto grado de abstracción o han asumido responsabilidades que deberían haber delegado a otros objetos. [15]

Creador: Se asigna la responsabilidad de que una clase B cree un Objeto de la clase A solamente cuando

- B contiene a A
- B es una agregación (o composición) de A
- B almacena a A
- B tiene los datos de inicialización de A (datos que requiere su constructor)

- B usa a A.

La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien, el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización. [16]

Controlador: Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. [16]

1.12.1 Ventajas de los patrones de diseño.

- Los patrones de diseño proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifican y describen formas de solucionar problemas que ocurren de forma frecuente en el desarrollo.
- Por tanto, están basados en la recopilación del conocimiento de los expertos en desarrollo de software.
- Es una experiencia real, probada y que funciona. Es historia y nos ayuda a no cometer los mismos errores.

1.13 Conclusiones

El desarrollo de un LIMS es realmente de gran ayuda para el manejo de datos en un laboratorio y con esto lograr el funcionamiento eficiente de éste.

La realización del diseño será llevada a cabo según lo planteado por el proceso de desarrollo RUP y basada en el framework Symfony.

Se desempeñarán los roles de diseñador, diseñador de interfaz de usuario y arquitecto de software definidos por el RUP y se obtendrán artefactos como: Vista Lógica, Diagrama de Despliegue, Diagramas de clases del Diseño, Diagramas de Interacción, Mapa de Navegación, Prototipos de Interfaz de Usuario.

Capítulo 2. Diseño del Sistema.

2.1 Introducción

Este capítulo está dedicado al diseño del sistema, haciendo referencia además a la arquitectura del mismo mediante la vista lógica y la vista de despliegue. Se obtienen varios artefactos de la parte de Diseño del flujo de trabajo Análisis y Diseño definido por el RUP. En este capítulo se modela el sistema para que soporte todos los requisitos.

2.2 Referencia a la arquitectura del Sistema.

La arquitectura del sistema estará guiada por la arquitectura que sigue el framework Symfony que ha sido el escogido para el desarrollo de la aplicación.

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC (Model-View-Controller), que está formado por tres niveles: [17]

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.

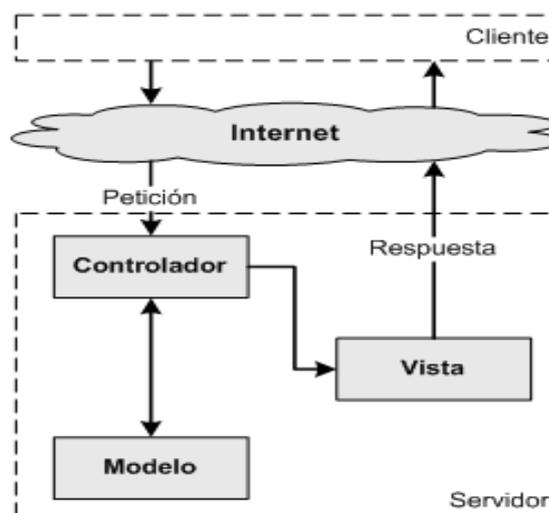


Fig. 3 Funcionamiento del patrón MVC.

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.

2.2.1 Vista Lógica.

La vista lógica muestra los componentes principales del diseño y sus relaciones de forma independiente de los detalles técnicos y de cómo la funcionalidad será implementada en la plataforma de ejecución. Los arquitectos crean modelos de diseño de la aplicación, los cuales son vistas lógicas del modelo funcional y que describen la solución. Se describe la solución en términos de paquetes y clases del diseño. Siguiendo el supuesto de que se está usando el Proceso Unificado, dentro de esta vista se describe la realización de los casos de uso, subsistemas, paquetes y clases de los casos de uso más significativos arquitectónicamente. [18]

Ver **Anexo #2** Vista Lógica.

En la realización de la vista lógica de este módulo no se identificaron subsistemas funcionales. Para estructurar el sistema se organizó por paquetes del diseño a partir de los casos de uso arquitectónicamente significativos, que constituyen la totalidad de los casos de uso identificados para este módulo.

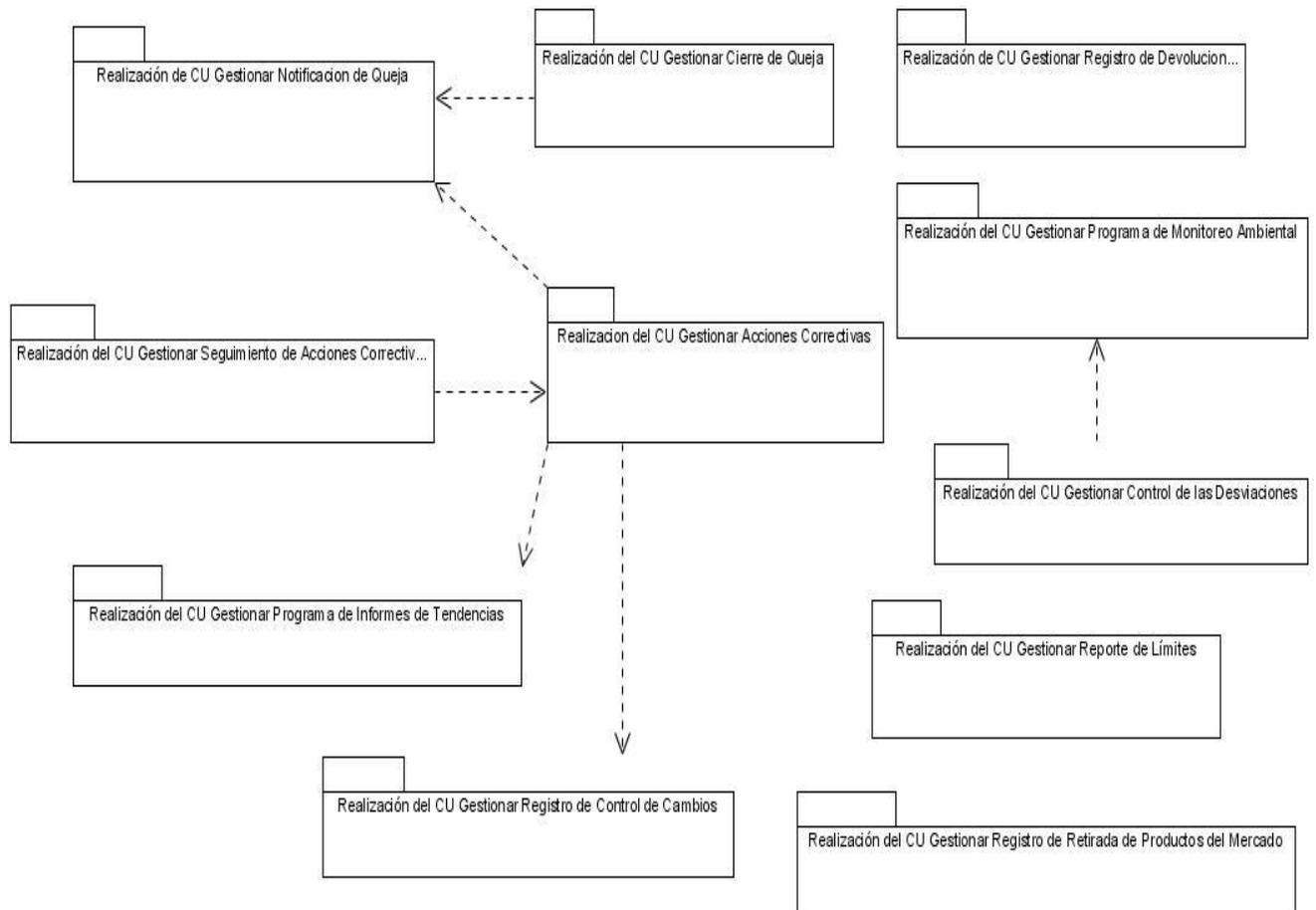
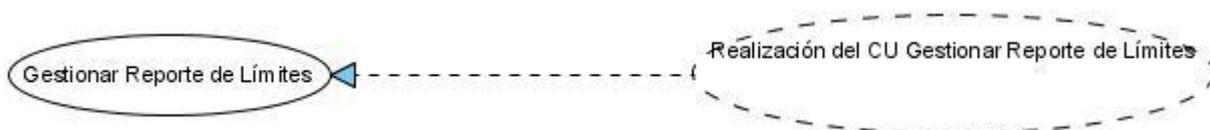
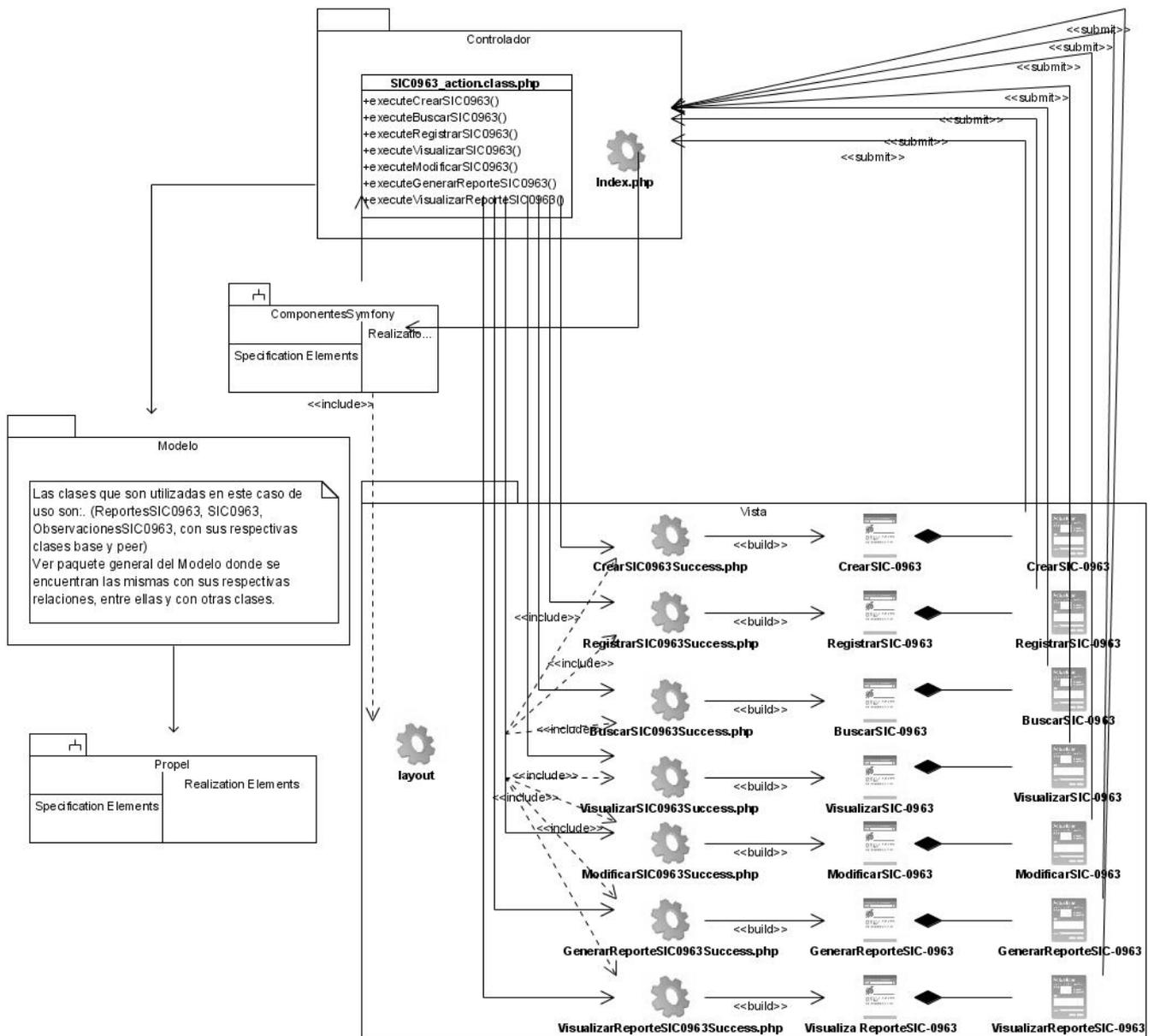


Fig. 4 Vista Lógica.

Cada paquete contiene la realización de un caso de uso. Por ejemplo el paquete relacionado con el caso de uso Gestionar Reporte de Límites, contiene la realización del mismo.



La realización del caso de uso contiene el diagrama de clases del diseño de éste, y los diagramas de secuencia por escenario. En los diagramas de clases del diseño se puede apreciar claramente el estilo arquitectónico Modelo Vista Controlador, seguidamente se muestra el diagrama de clases del diseño del caso de uso Gestionar Reporte de Límites donde se expresa lo anterior.



2.2.2 Vista de Despliegue.

La vista de despliegue de un sistema contiene los nodos que forman la topología hardware sobre la que se ejecuta el sistema. Se preocupa principalmente de la distribución, entrega e instalación de las partes que constituyen el sistema. Los aspectos estáticos de esta vista se representan mediante los diagramas de despliegue.

El sistema estará distribuido de la siguiente forma: un servidor de aplicaciones al cual se van a conectar las PCs clientes y un servidor de base de datos donde se almacenará toda la información. En las PCs clientes existirán conectadas impresoras para la impresión de planillas y otros documentos importantes y escáneres para la digitalización de información importante que se encuentra en formato duro.

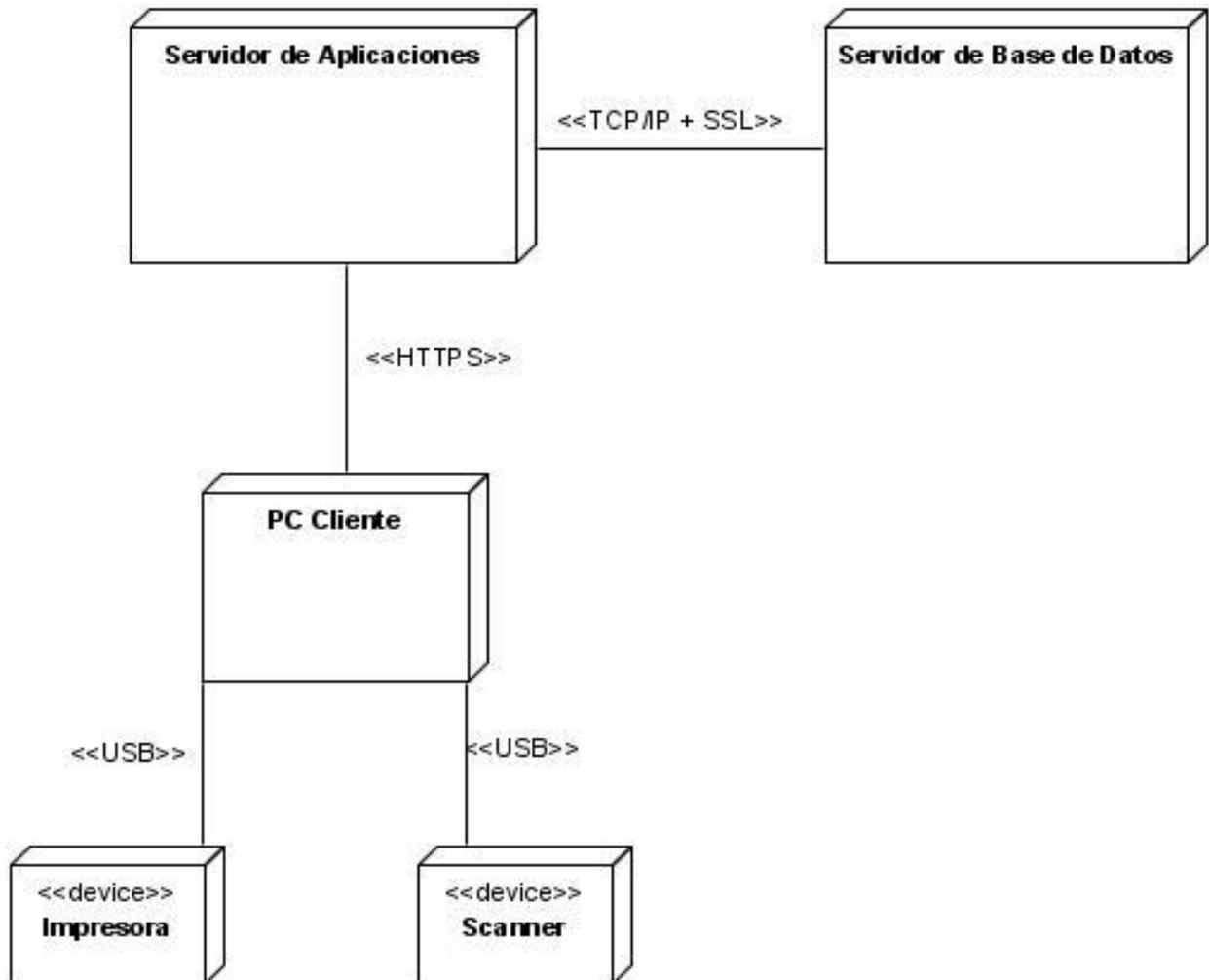


Fig. 5 Diagrama de Despliegue

2.3 Diseño del sistema.

2.3.1 Mapa de Navegación.

El mapa de navegación expresa la estructura de los elementos de la interfaz de usuario en el sistema, junto con sus posibles vías de navegación.

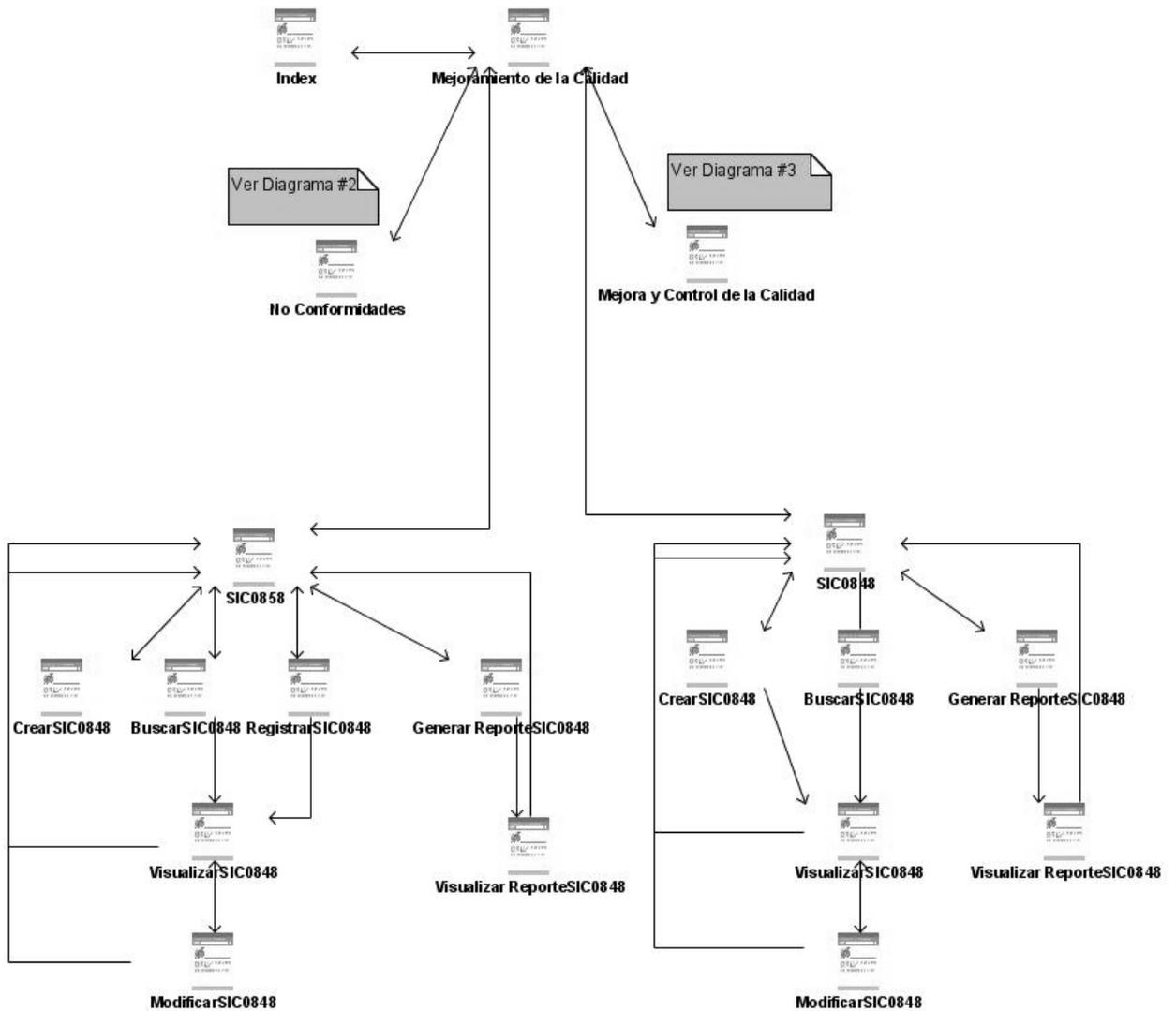


Fig. 6 Mapa de navegación

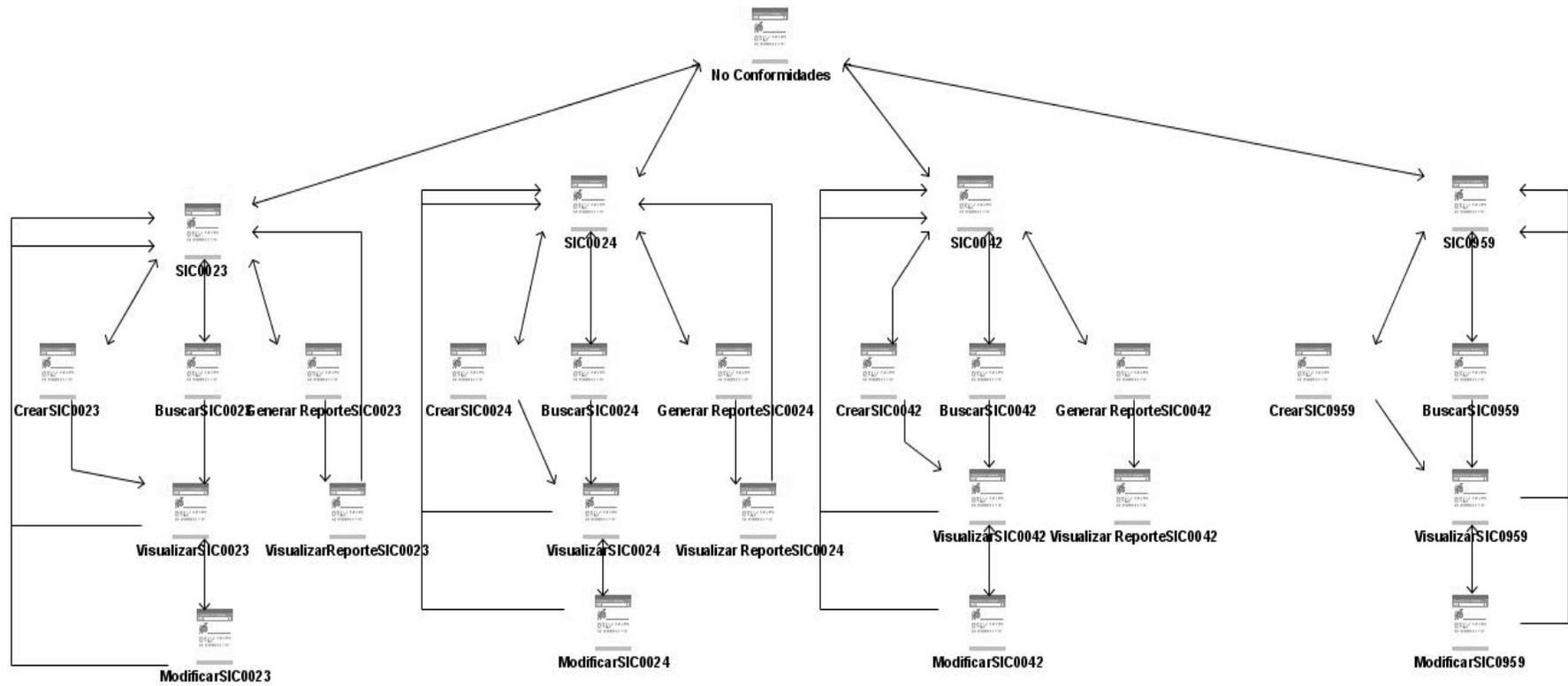


Fig. 7 Mapa de navegación: Diagrama #2

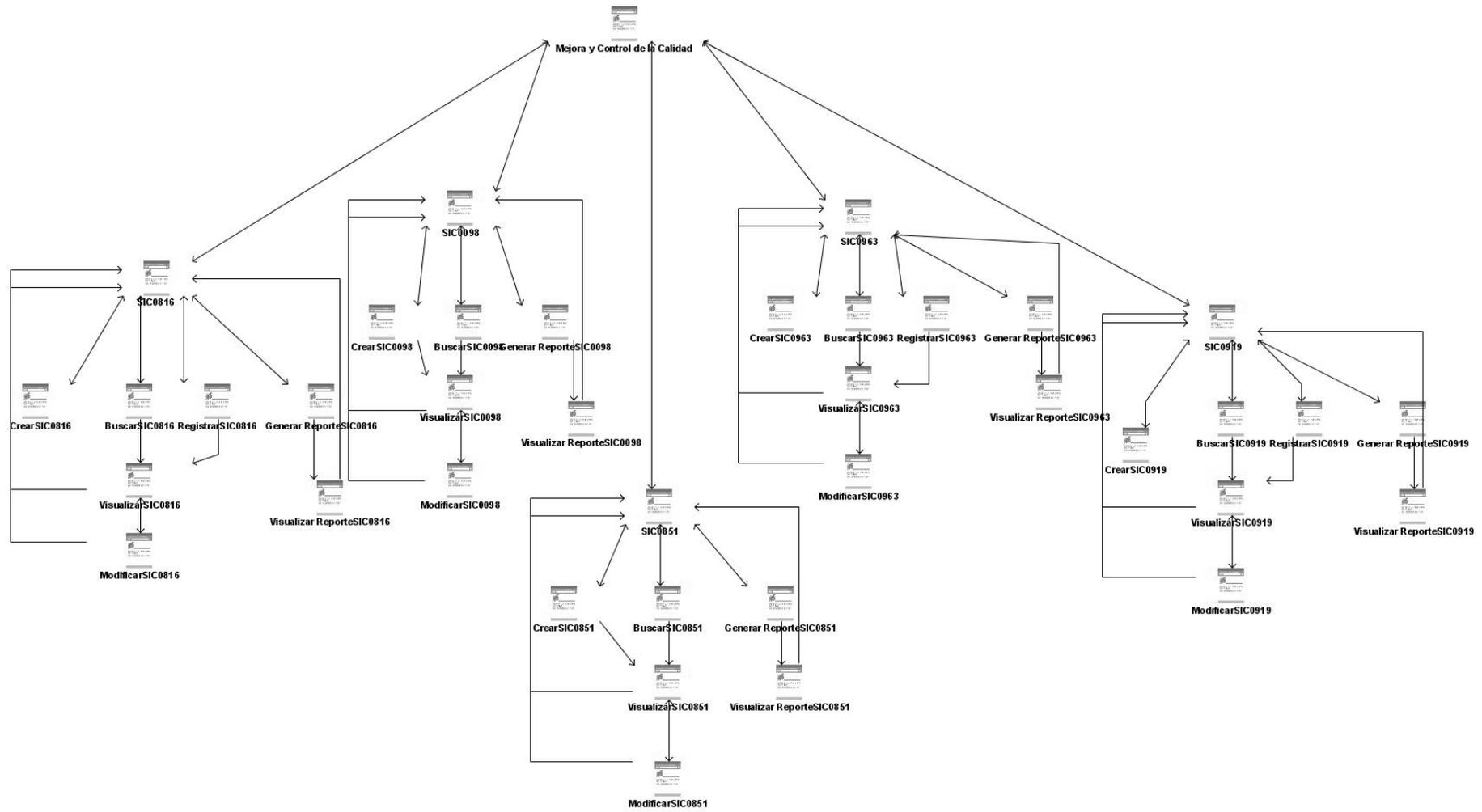


Fig. 8 Mapa de navegación: Diagrama #3

2.3.2 Prototipos de interfaz de usuario por caso de uso del sistema.

Seguidamente se muestra una representación de los prototipos de interfaz de usuario del caso de uso Gestionar Registro de Control de Cambios. Ver **Anexo# 3** donde se muestran los prototipos referentes a los casos de uso Gestionar Programa de Monitoreo Ambiental y Gestionar Programa de Informes de Tendencias. En el Expediente del Proyecto se encuentran los prototipos referentes a los demás casos de uso.

CU Gestionar Registro de Control de Cambios (SIC-0816)

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA DIRECCIÓN DE CALIDAD	SIC-0816 Edición 03	PPO 4.10.096.98
Registro de Control del Cambio		Folio: <input style="width: 50px;" type="text"/>

Fig. 9 Prototipo de Crear Registro de Control de Cambios

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA DIRECCIÓN DE CALIDAD Búsqueda del Registro de Control de Cambios		
Código/Nombre del Cambio: <input style="width: 80px;" type="text" value="—Seleccione—"/>	Clasificación del Cambio:	<input type="radio"/> Mayor <input type="radio"/> Menor

Código/Nombre del Cambio	Fecha de Recepción	Decisión Final	En Reg. y Reg.
			Visualizar

Fig. 10 Prototipo de Buscar Registro de Control de Cambios

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA						SIC-0816	PPO 4.10.096.98	
						Edición 03		
DIRECCIÓN DE CALIDAD						Folio: -----		
Registro de Control del Cambio								
Código/Nombre del Cambio	Fecha de Recepción	Clasificación		En Revisión	Pendiente de documentación	En Reg. y Reg.	Decisión Final	Fecha de conclusión
		Me.	Ma.					
-----	-	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-	-
-----	-	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	-	-
-----	-	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-	-
Observaciones: ----- ----- -----								

Modificar Imprimir Enviar

Fig. 11 Prototipo de Visualizar Registro de Control de Cambios

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA						SIC-0816	PPO 4.10.096.98	
						Edición 03		
DIRECCIÓN DE CALIDAD						Folio: _____		
Registro de Control del Cambio								
Código/Nombre del Cambio	Fecha de Recepción	Clasificación		En Revisión	Pendiente de documentación	En Reg. y Reg.	Decisión Final	Fecha de conclusión
		Me.	Ma.					
---	---	<input type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	---	---
---	---	<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	---	---
---	---	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	---	---
Observaciones: ----- ----- -----								

Modificar Cancelar

Fig. 12 Prototipo de Modificar Registro de Control de Cambios

Registrar datos en el Registro de Control de Cambios

Folio:

Código/Nombre del Cambio	Fecha de Recepción	Clasificación		En Revisión	Pendiente de documentación	En Reg. y Reg.	Decisión Final	Fecha de conclusión
		Me.	Ma.					
<input type="text"/>	<input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/>
Insertar un nuevo cambio								
Observaciones:								
<div style="border: 1px solid black; height: 60px;"></div>								

Fig. 13 Prototipo de Registrar Registro de Control de Cambios

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA
DIRECCIÓN DE CALIDAD
Registro de Control de Cambios

Generar reporte del Registro de Control de Cambios

Parámetros para la generación del reporte

Clasificación:	<input type="radio"/> Me	Fecha de Reg. y Reg.	Fecha Inicio: <input type="text"/>	Decisión Final:	Fecha Inicio: <input type="text"/>
	<input type="radio"/> Ma		Fecha Final: <input type="text"/>		Fecha Final: <input type="text"/>

Parámetros a mostrar en el reporte del Registro de Control de Cambios

<input type="checkbox"/> Todos	<input type="checkbox"/> Fecha de Conclusión.
<input type="checkbox"/> Código/Nombre del cambio.	<input type="checkbox"/> Decisión Final.

Fig. 14 Prototipo de Generar Reporte del Registro de Control de Cambios

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA				
DIRECCIÓN DE CALIDAD				
Registro de Control de Cambios				
Reporte del Registro de Control de Cambios				
-----	-----	-----	-----	-----
-----	-----	-----	-----	-----

Fig. 15 Prototipo de Visualizar Reporte del Registro de Control de Cambios

2.3.3 Diagramas de clases del diseño por caso de uso del sistema.

Los diagramas de clases del diseño realizados están basados en el funcionamiento del framework utilizado. Ver **Anexo #4** El flujo de trabajo de Symfony.

Estos diagramas están conformados por tres paquetes fundamentales Modelo, Vista y Controlador.

El paquete Controlador contiene el controlador frontal y la clase de la acción. Todas las peticiones web son manejadas por un solo controlador frontal, que es el punto de entrada único 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 es un componente que sólo tiene código relativo al MVC, por lo que no es necesario crear uno, ya que Symfony lo genera de forma automática, el archivo index.php. La clase que representa las acciones de un módulo (cada caso de uso es un módulo diferente) se encuentra en el archivo actions.class.php. Las acciones son métodos con el nombre executeNombreAcción, estas son el corazón de la aplicación, puesto que contienen toda la lógica de la misma. Las acciones utilizan el modelo y definen variables para la vista.

El subsistema ComponentesSymfony, contiene todos los componentes (ficheros, archivos yml, etc.) que nos brinda Symfony para su uso.

El paquete Vista contiene las páginas clientes con sus respectivos formularios, las plantillas (que son la presentación de los datos de la acción que se está ejecutando) y el layout (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.) La vista se encarga de producir las páginas que se muestran como resultado de las acciones.

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.

El paquete Modelo contiene todas las clases del modelo y sus relaciones. Por cada tabla del modelo entidad relación de la base de datos se generan 4 clases. Para crear el modelo de objetos de datos que utiliza Symfony, se debe traducir el modelo relacional de la base de datos a un modelo de objetos de datos. Para realizar ese mapeo o traducción, el ORM (object relational mapping) necesita una descripción del modelo relacional, que se llama esquema. En el esquema se definen las tablas, sus relaciones y las características de sus columnas. Las clases con nombre Base son las que se generan directamente a partir del esquema. Nunca 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 heredan de las clases con nombre Base. Estas clases no se modifican cuando se genera el modelo, por lo que son las clases en las que se añaden los métodos propios. Las clases de tipo "peer" son clases que tienen métodos estáticos para trabajar con las tablas de la base de datos. Proporcionan los medios necesarios para obtener los registros de las tablas. Sus métodos devuelven normalmente un objeto o una colección de objetos de la clase objeto relacionada.

El componente que se encarga por defecto de gestionar el modelo en Symfony es una capa de tipo ORM realizada mediante el proyecto Propel.

Propel, que también es un proyecto de software libre, es una de las mejores capas de abstracción de objetos/relacional disponibles en PHP 5. Propel está completamente integrado en Symfony. Las clases de la capa del modelo se generan automáticamente, en función de la estructura de datos de la aplicación. La librería Propel se encarga de esta generación automática, ya que crea el esqueleto o estructura básica de las clases y genera automáticamente el código necesario.

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

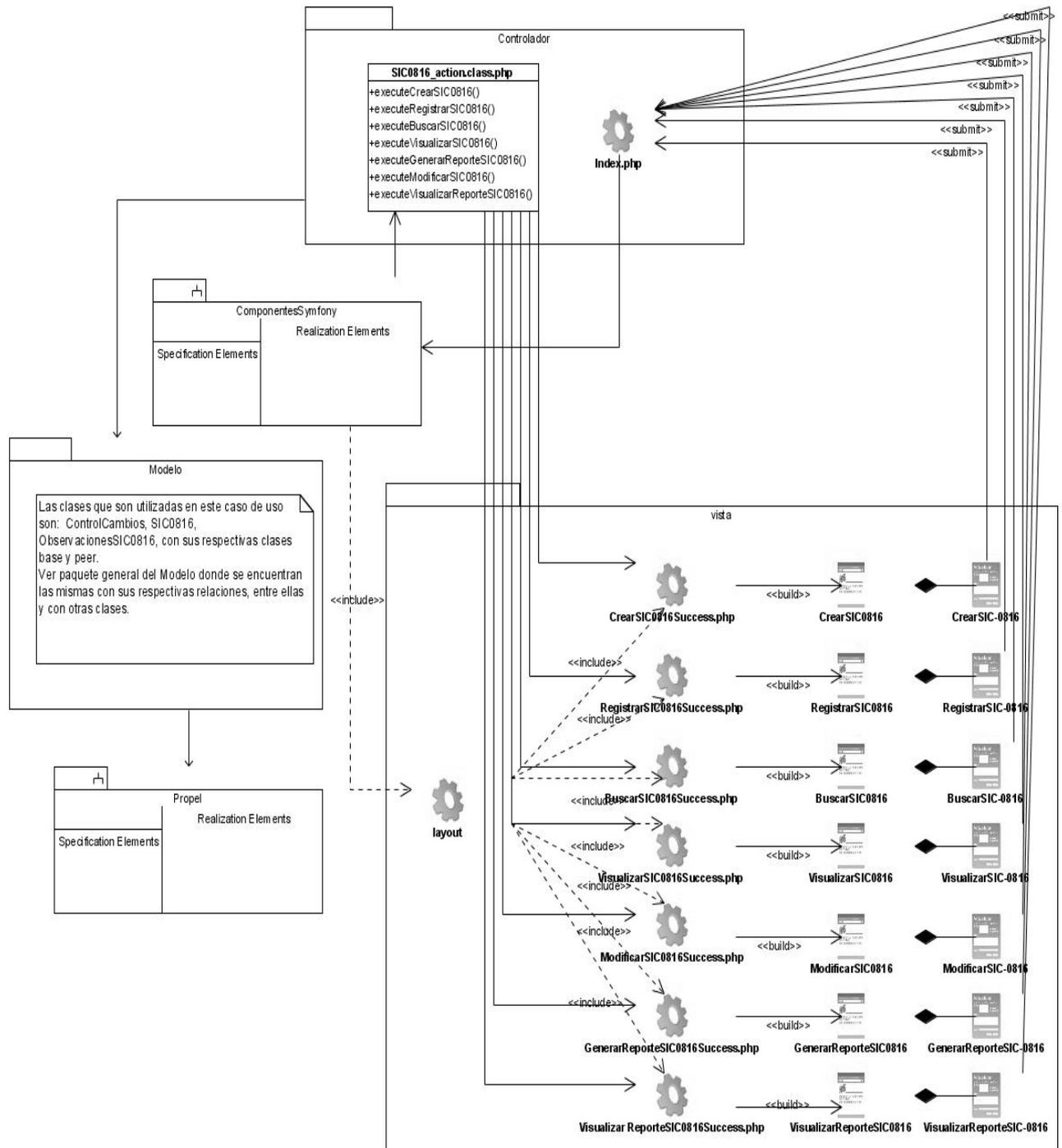


Fig. 16 CU Gestionar Registro de Control de Cambios (SIC-0816).

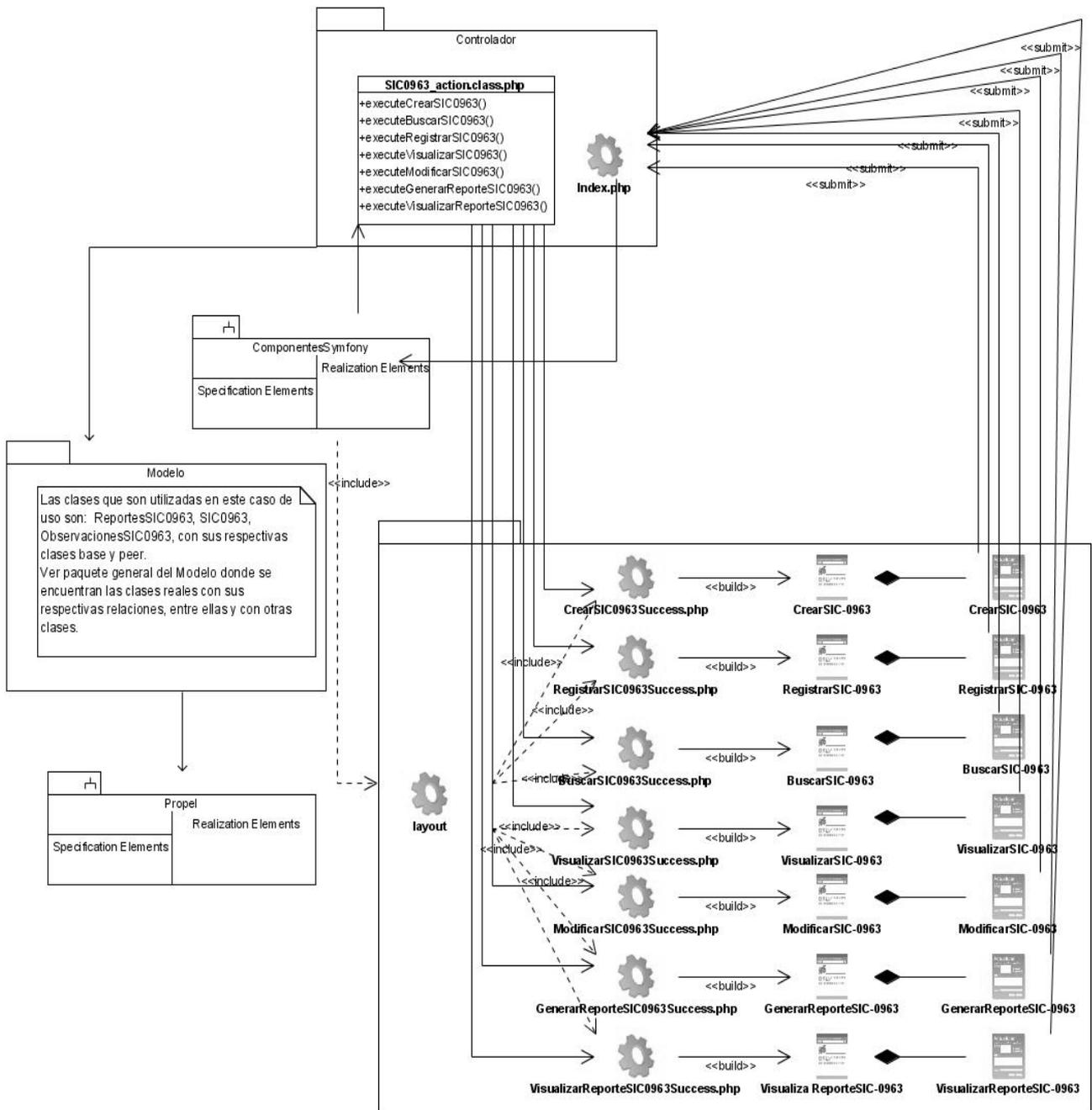


Fig. 17 CU Gestionar Reporte de Límites (SIC-0963).

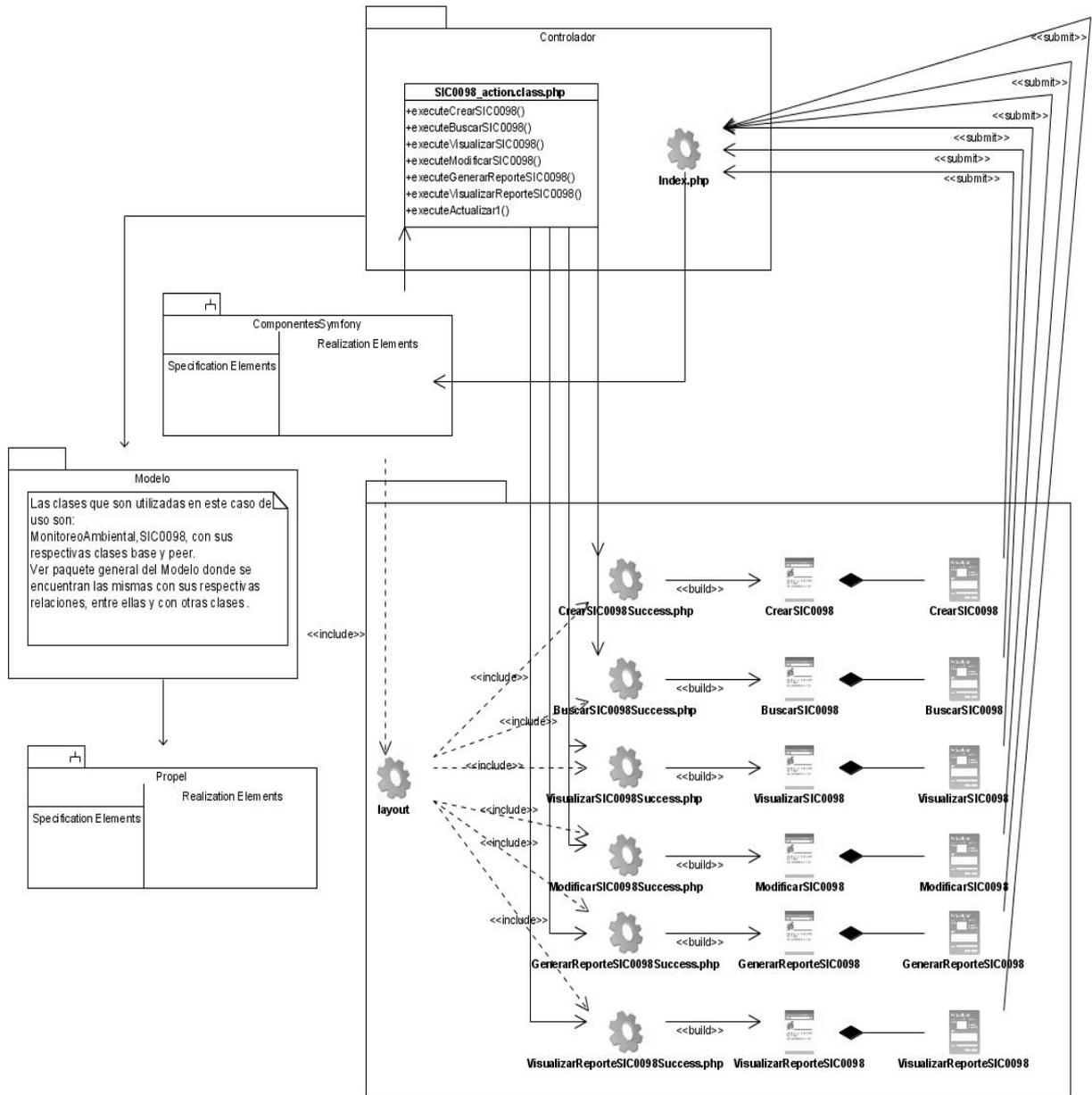


Fig. 18 CU Gestionar Programa de Monitoreo Ambiental (SIC-0098).

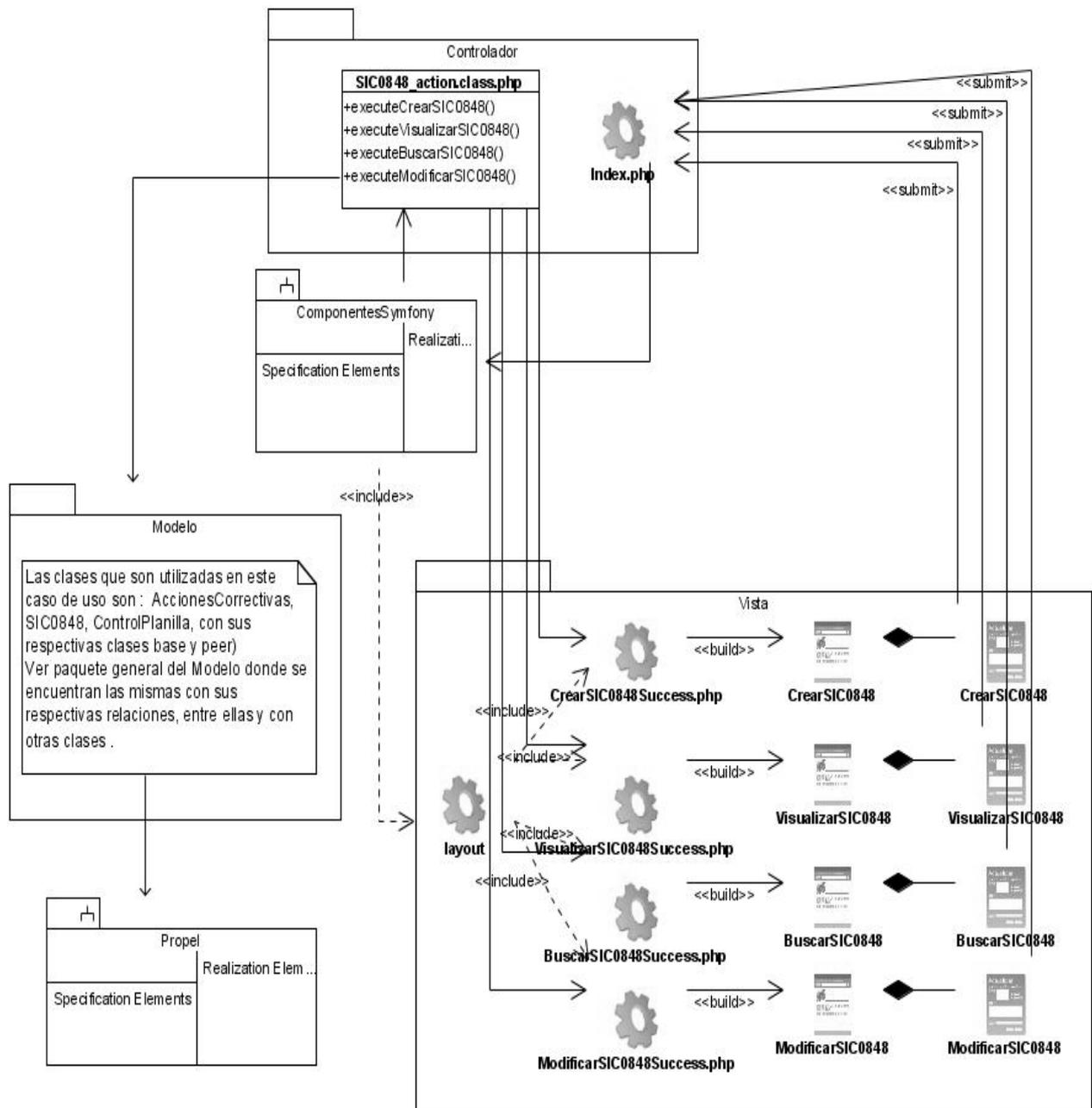


Fig. 19 CU Gestionar Acciones Correctivas (SIC-0848).

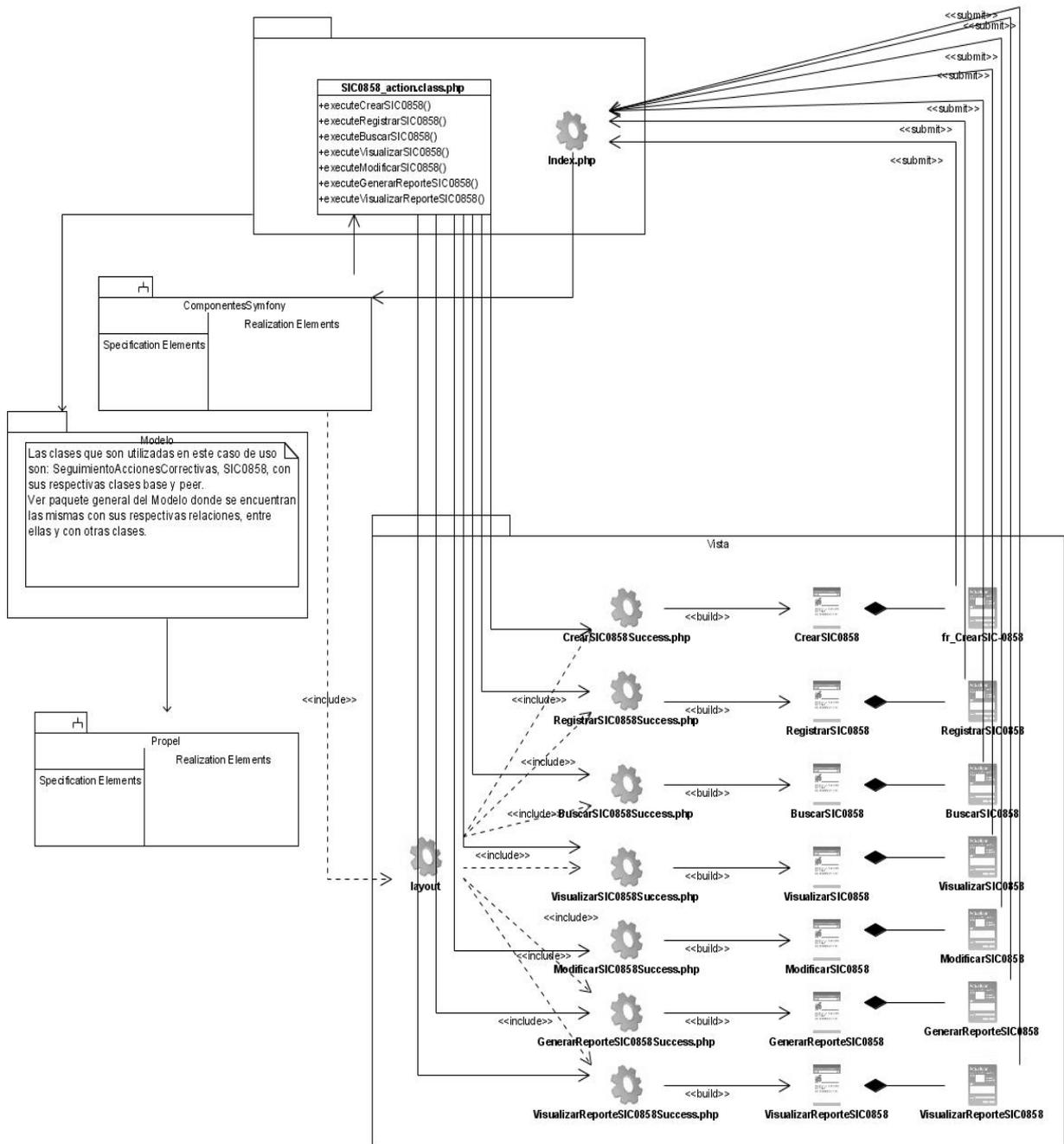


Fig. 20 CU Gestionar Seguimiento de Acciones Correctivas (SIC-0858).

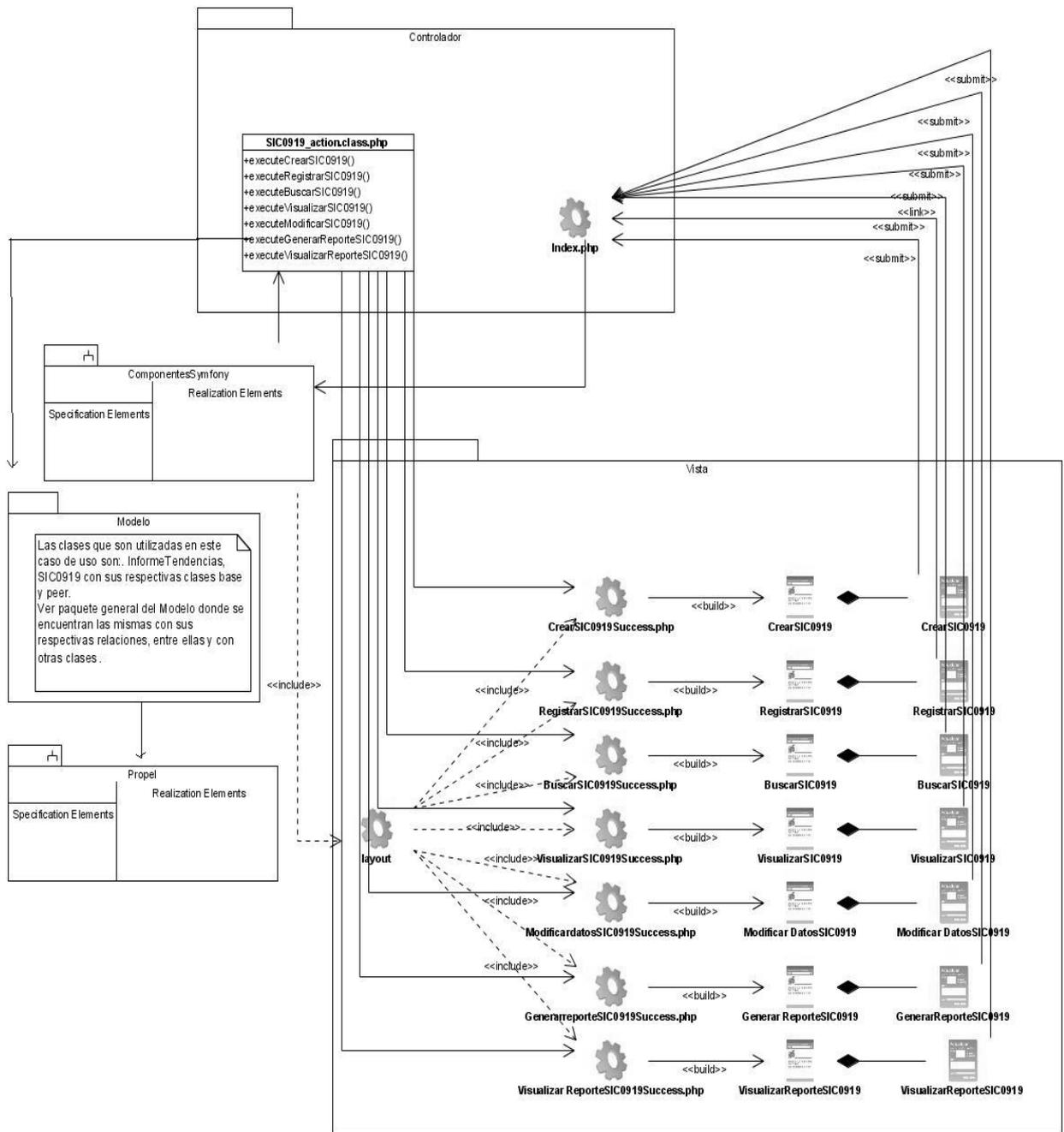


Fig. 21 CU Gestionar Programa de Informes de Tendencias (SIC-0919).

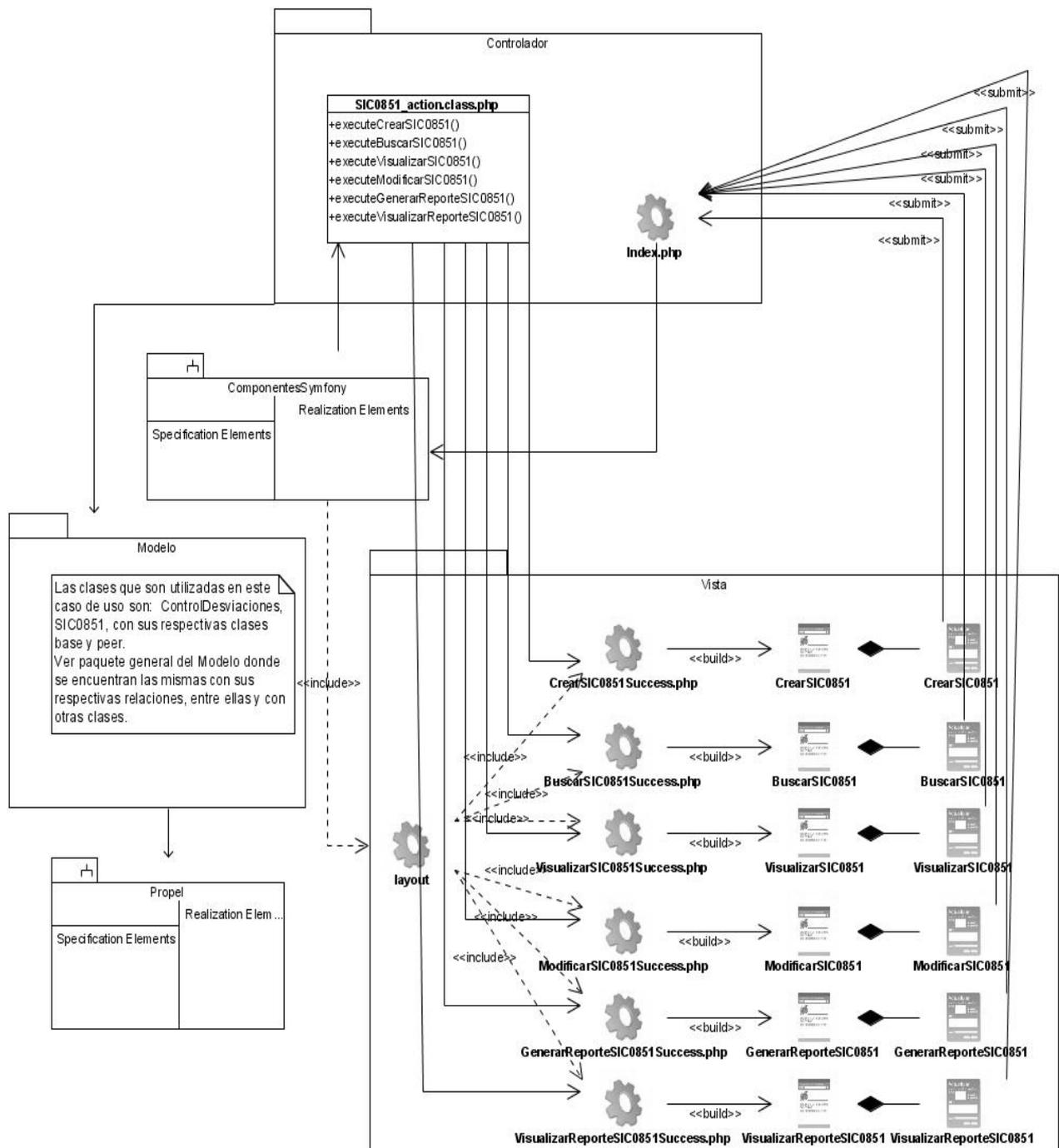


Fig. 22 CU Gestionar Registro de Control de las Desviaciones (SIC-0851)

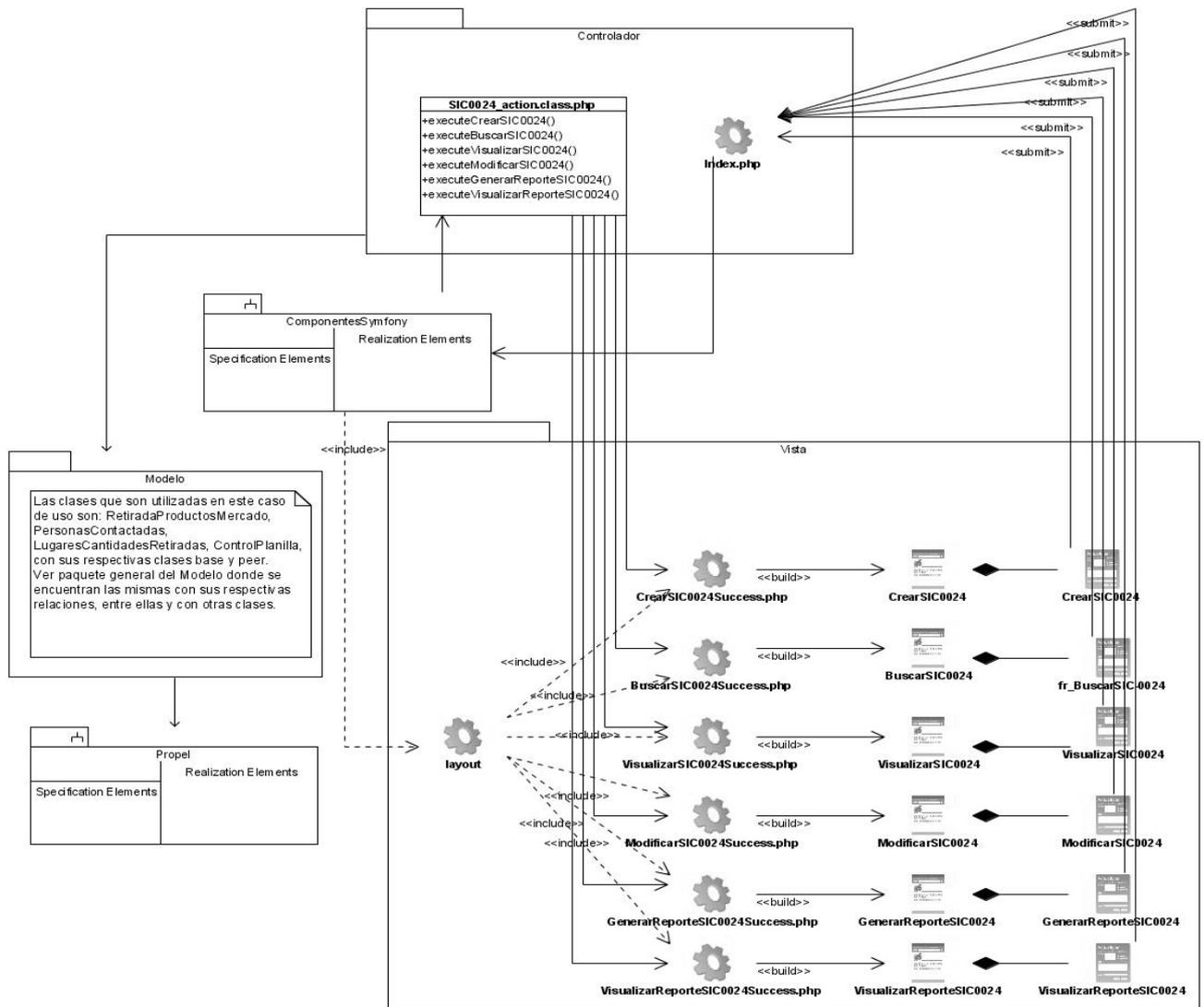


Fig. 24 CU Gestionar Registro de Retirada de Productos del Mercado (SIC-0024)

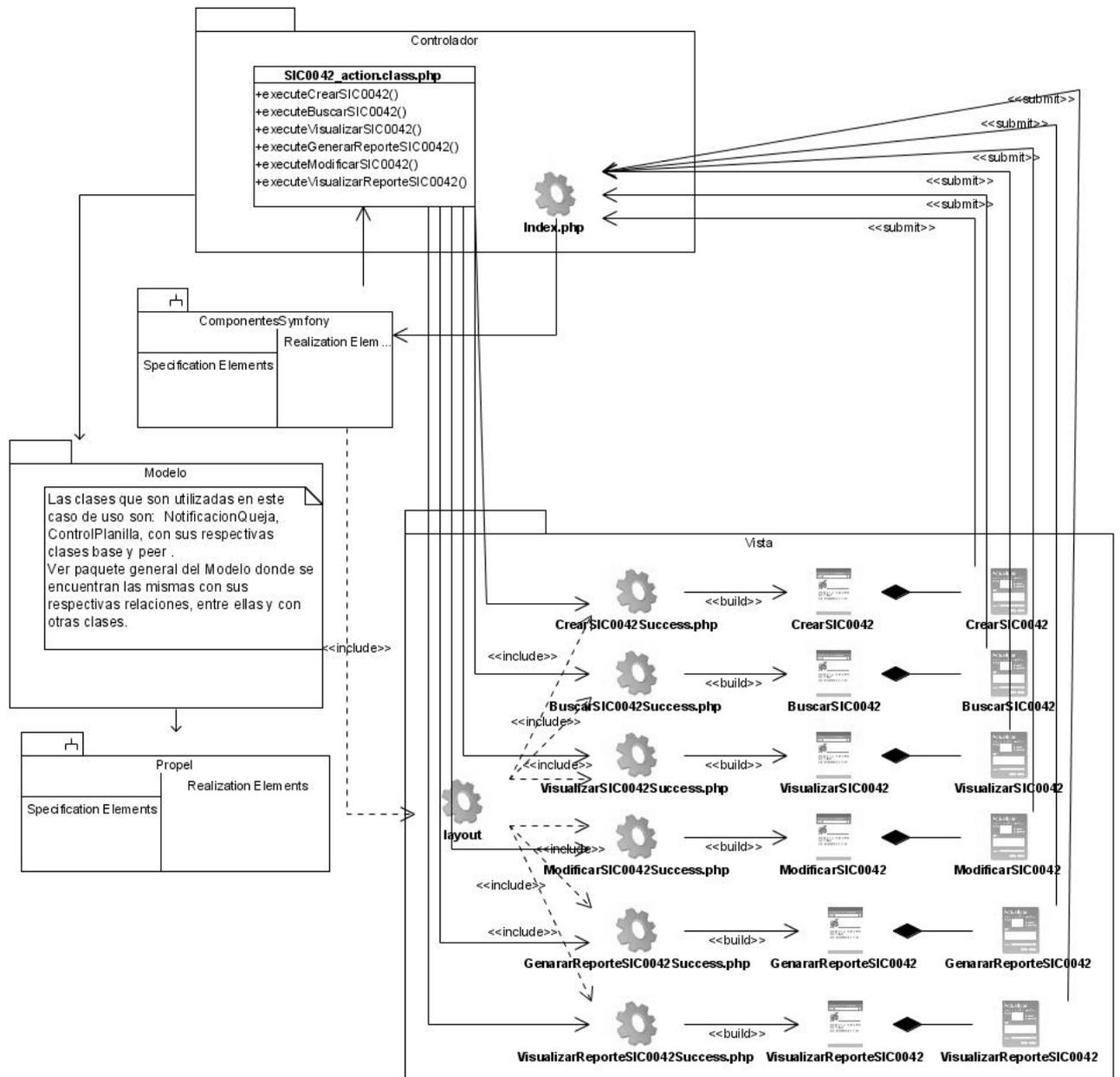


Fig. 25 CU Gestionar Notificación de Queja (SIC-0042)

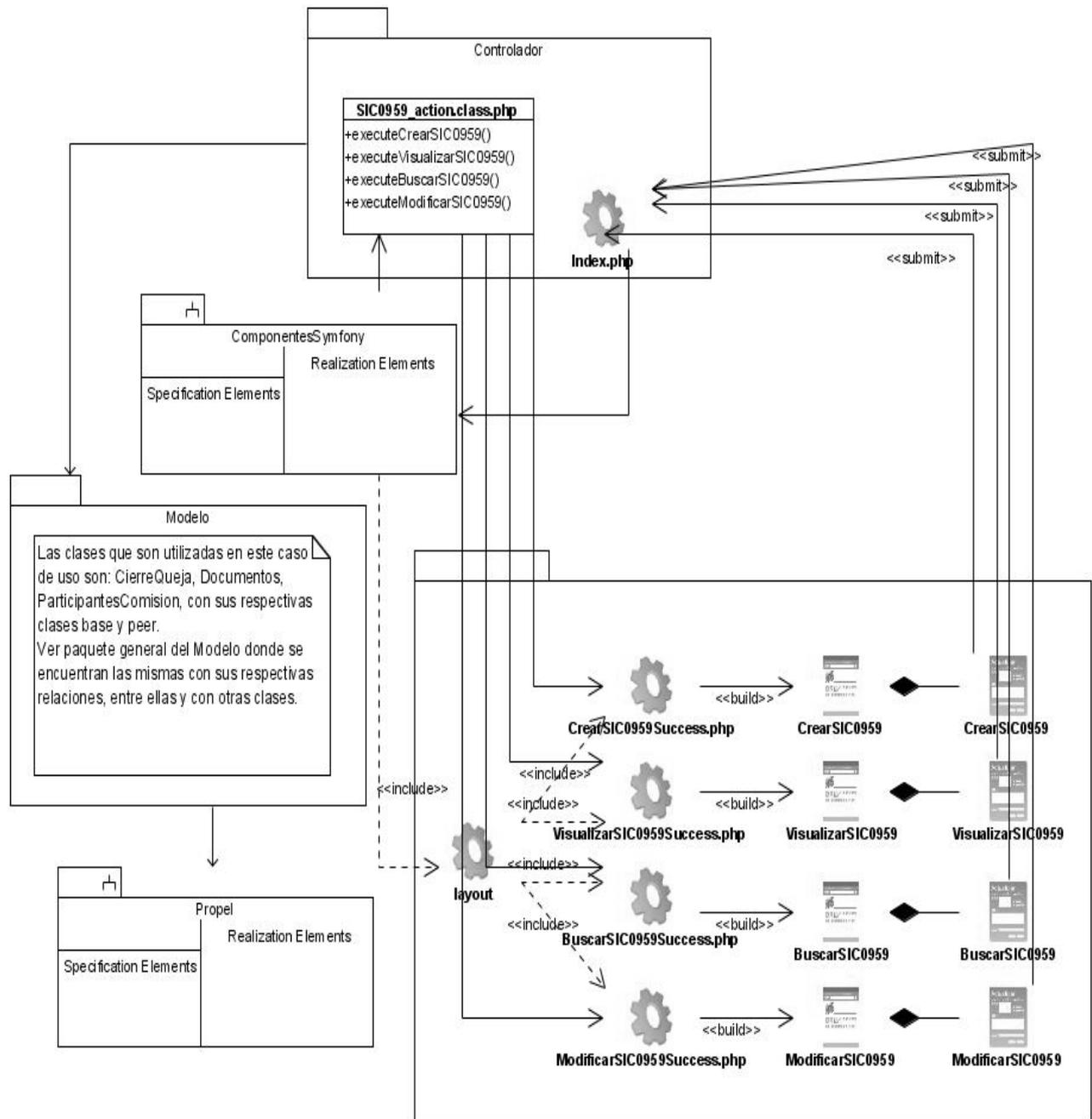


Fig. 26 CU Gestionar Cierre de Queja (SIC-0959)

Paquete Modelo.

A continuación se muestra una representación del paquete Modelo donde se encuentran las clases y sus relaciones. En el Expediente del Proyecto se muestran todas las clases del paquete Modelo.

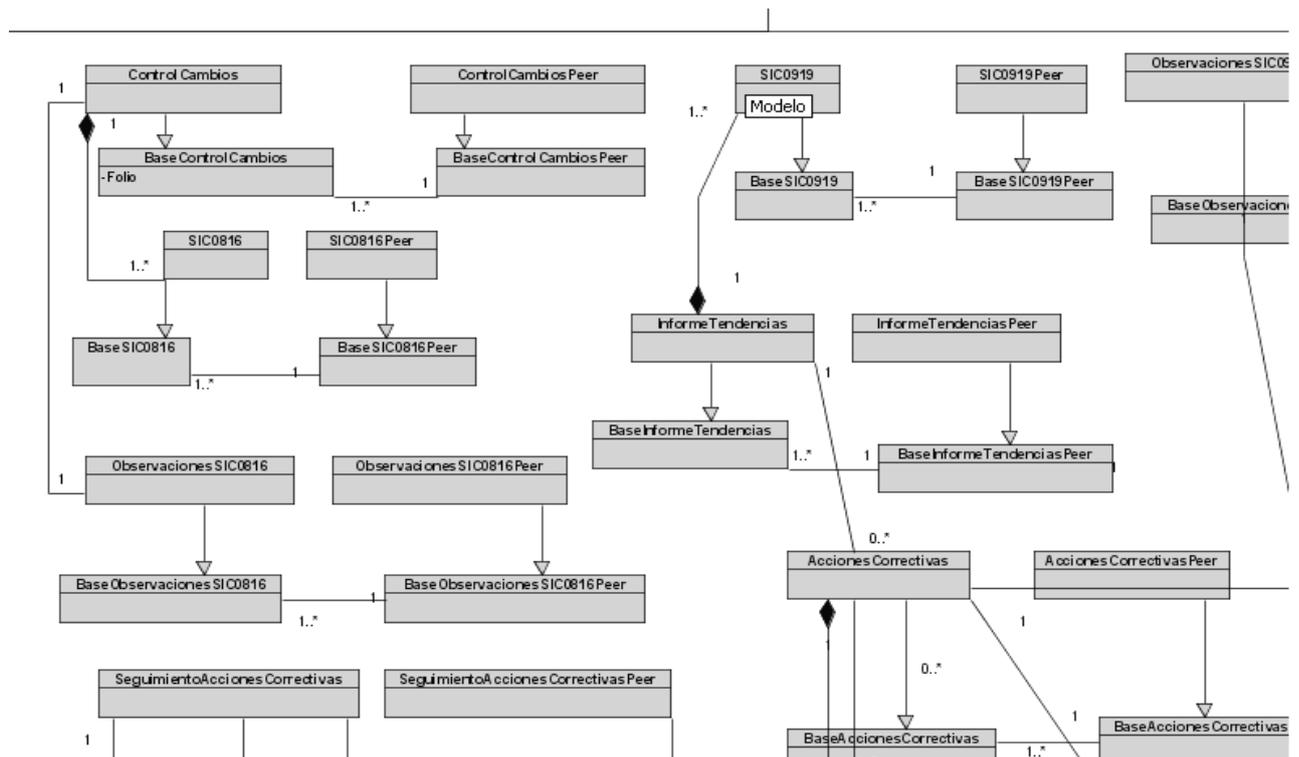


Fig. 27 Paquete Modelo.

2.3.4 Diagramas de interacción por escenario de caso de uso del sistema.

Se muestra una representación de los diagramas de secuencia por escenario de los casos de uso. En el Expediente del Proyecto se encuentran los restantes diagramas de secuencia.

A continuación se describe el flujo general de los diagramas:

El actor luego de haberse autenticado selecciona en la página principal la opción que desea realizar a una planilla (crear planilla, registrar planilla, buscar planilla, modificar planilla, generar reporte de planilla), esta página hace un link al index.php (controlador frontal) indicándole el nombre del módulo y

la acción escogida. Luego el controlador frontal pasa estos datos a la sfcontroller, que es la clase que se encarga de decodificar la petición y transferirla a la acción correspondiente de la clase de la acción, a continuación se verifica la seguridad y se validan los formularios. Posteriormente se ejecuta la acción haciendo llamados a su vez a funciones que se encuentran en las clases 'peer'. Finalmente con la unión de los datos requeridos, la plantilla success y el layout se construyen las páginas clientes por las que se visualizan los datos de acuerdo a la solicitud del actor.

Es importante prestar atención al esfuerzo invertido en la modelación, tratando de hacerla lo más eficientemente posible, y de manera que el resultado obtenido (los modelos), sean comprensibles y permitan una comunicación efectiva.

Un elemento que puede ser favorable en este sentido son los Mecanismos de Diseño, artefacto propuesto y descrito en RUP 2003. Los mismos reportan beneficios para diversos propósitos, entre ellos se encuentra reutilizar la documentación, lo cual se pone de manifiesto en los diagramas de secuencia del presente trabajo. Debido a que los mensajes enviados al controlador frontal y del controlador frontal con la sfcontroller, los mensajes de validar formularios y verificar seguridad van a ser siempre los mismos, lo cual puede afirmarse que resulta redundante y que carga el diagrama con objetos y mensajes que siempre serán los mismos, pues corresponden al mismo Mecanismo de Diseño, se decidió omitir el controlador frontal y la sfcontroller de los diagramas, apareciendo únicamente en los escenarios del primer caso de uso.

Seguidamente se muestran los diagramas de secuencia referentes a los escenarios de los casos de uso Gestionar Programa de Monitoreo Ambiental, Gestionar Registro de Control de Cambios, Gestionar Programa de Informes de Tendencias, Gestionar Registro de Control de las Desviaciones, y Gestionar Reporte de Límites.

CU Gestionar Programa de Monitoreo Ambiental

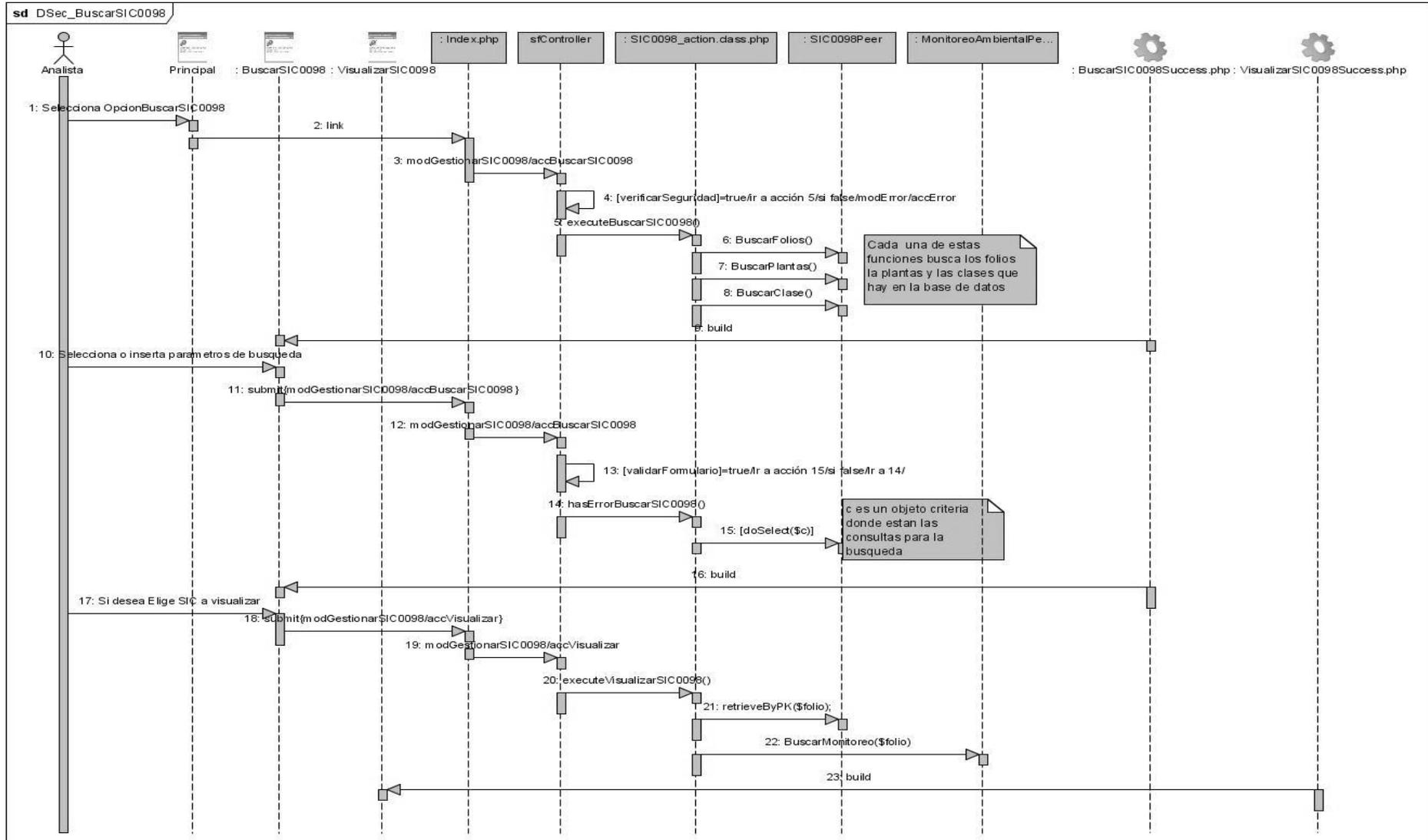


Fig. 28 Escenario Buscar Programa de Monitoreo Ambiental

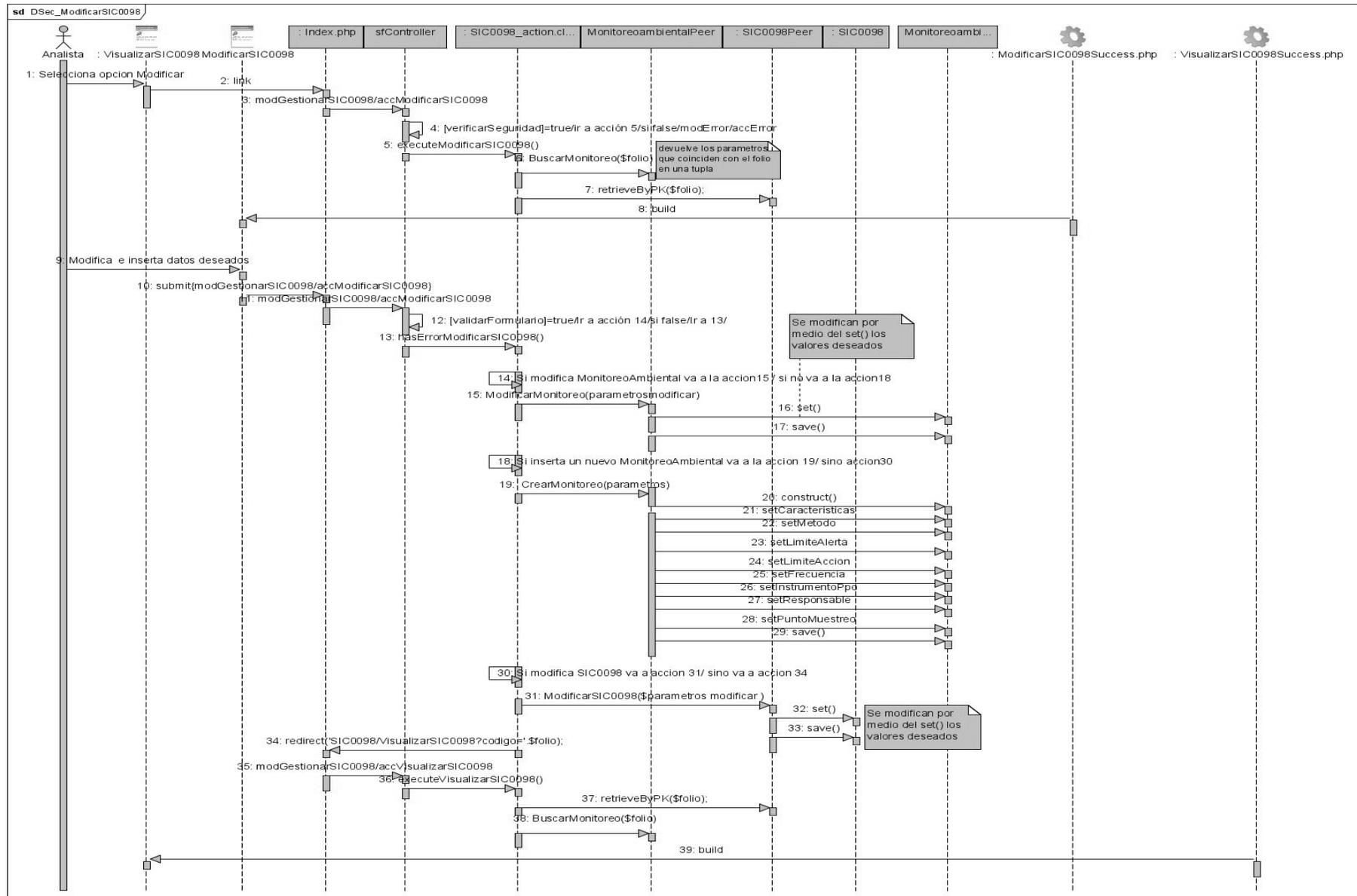


Fig. 29 Escenario Modificar Programa de Monitoreo Ambiental

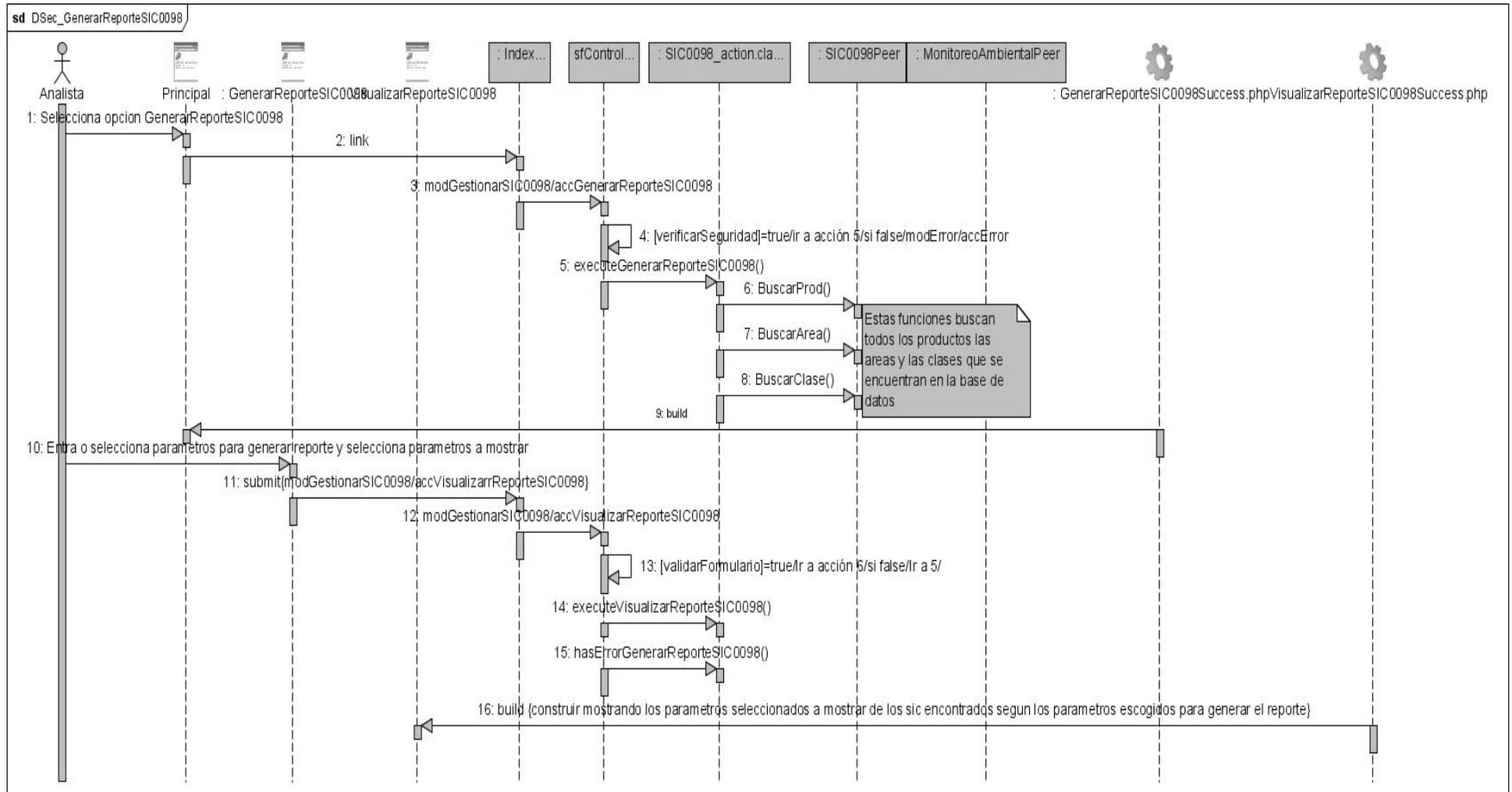


Fig. 30 Escenario Generar Reporte del Programa de Monitoreo Ambiental

CU Gestionar Registro de Control de cambios

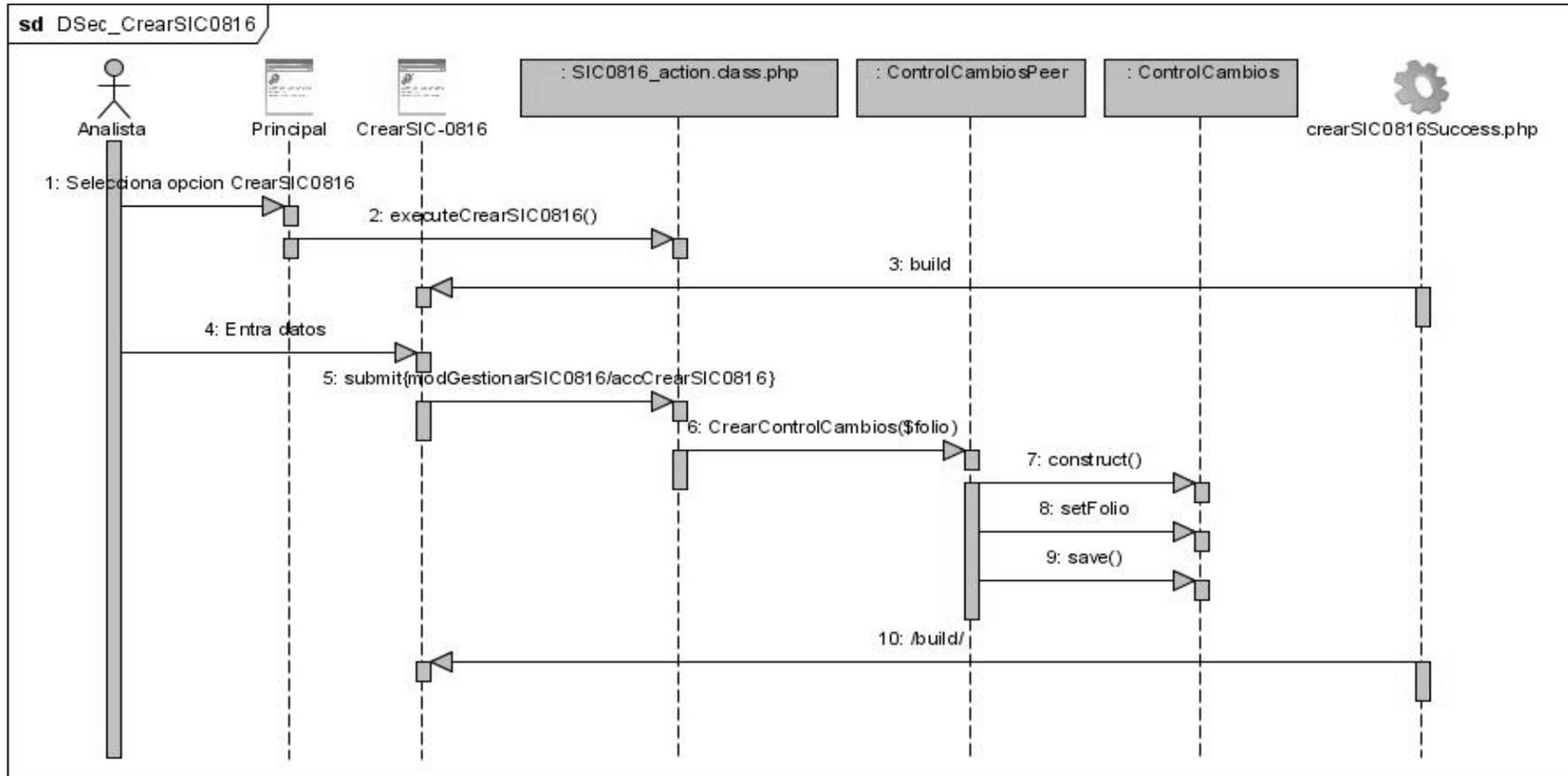


Fig. 31 Escenario Crear Registro de Control de Cambios

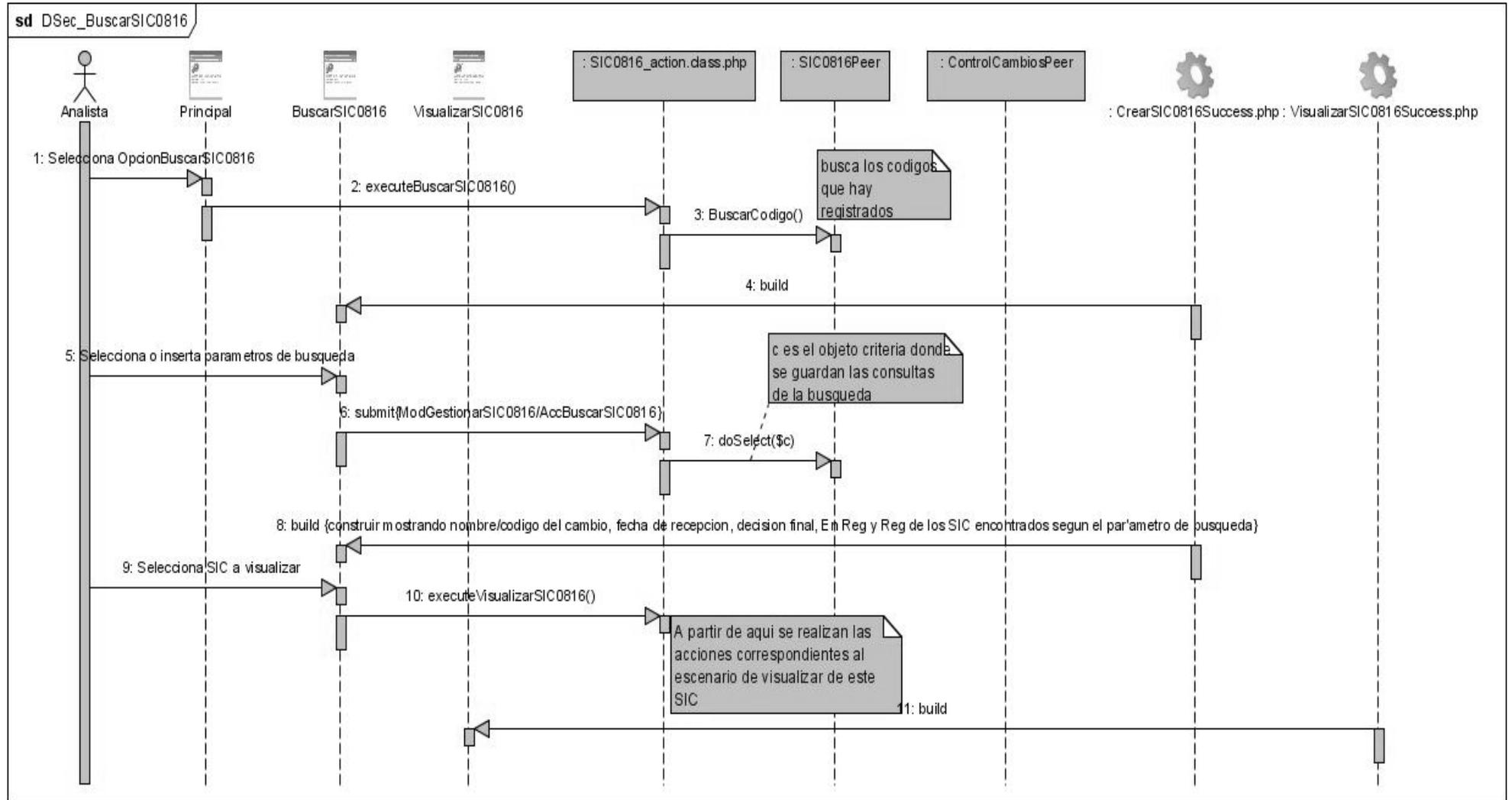


Fig. 32 Escenario Buscar Registro de Control de Cambios

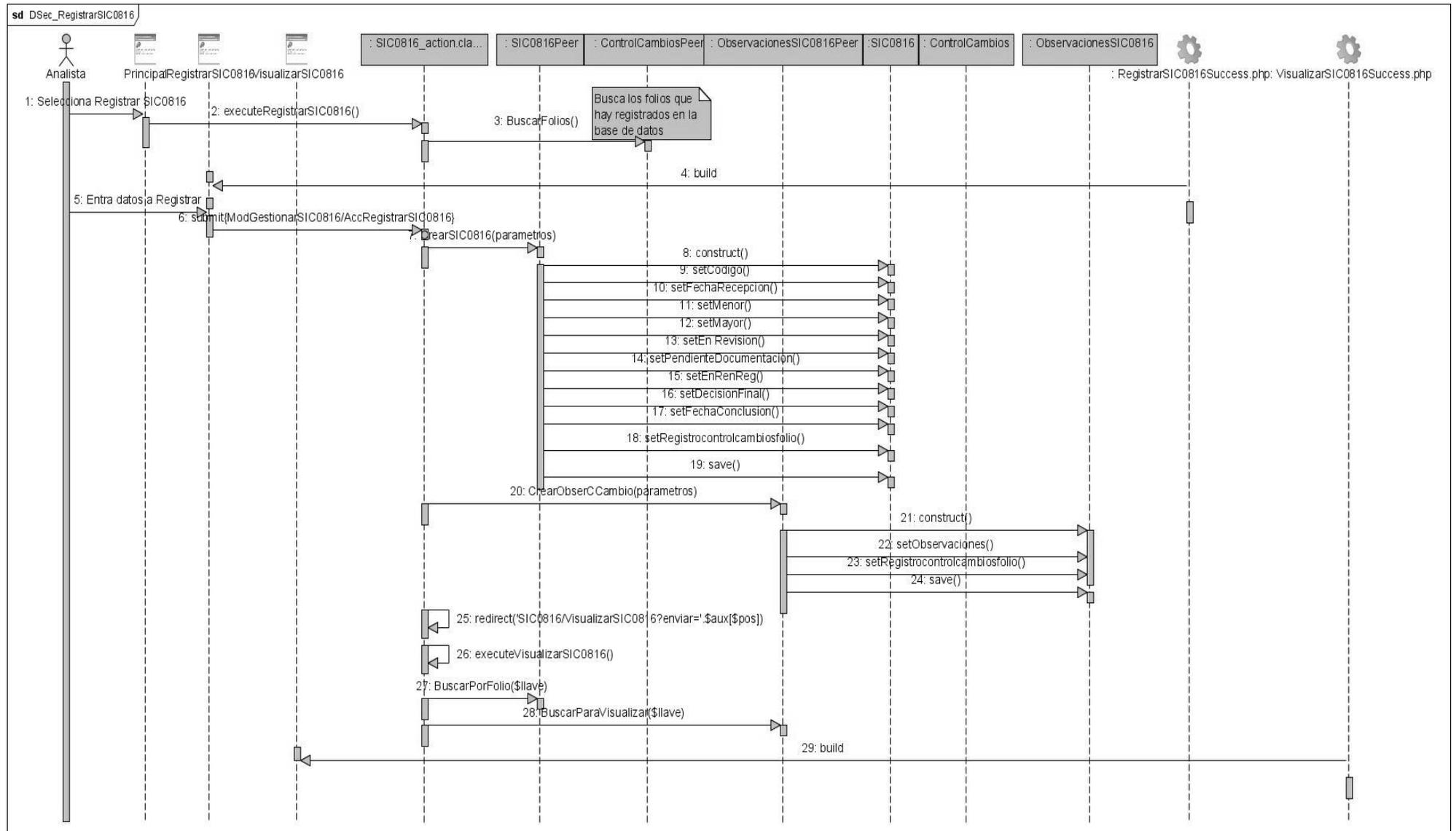


Fig. 33 Escenario Registrar Registro de Control de Cambios

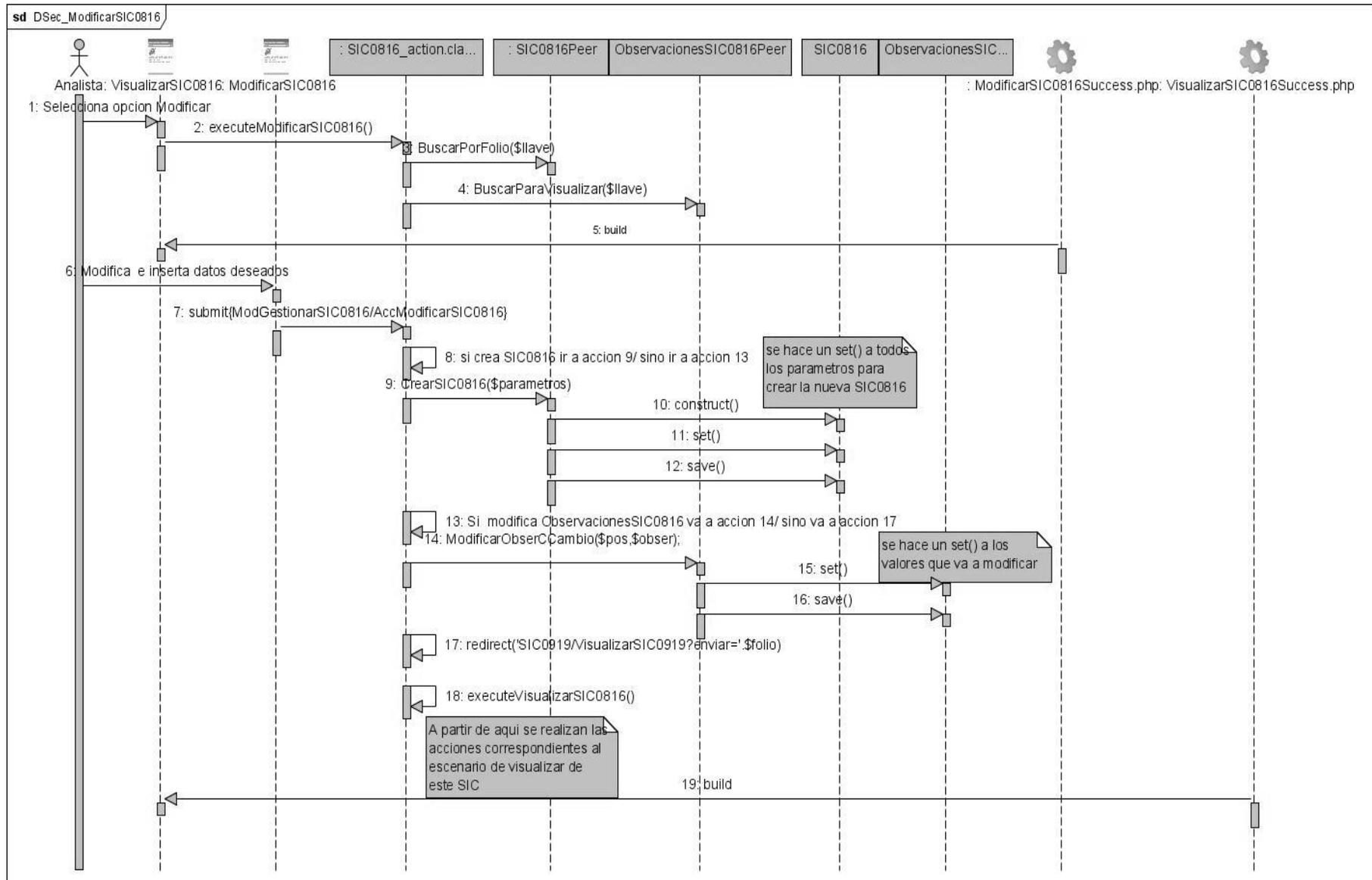


Fig. 34 Escenario Modificar Registro de Control de Cambios

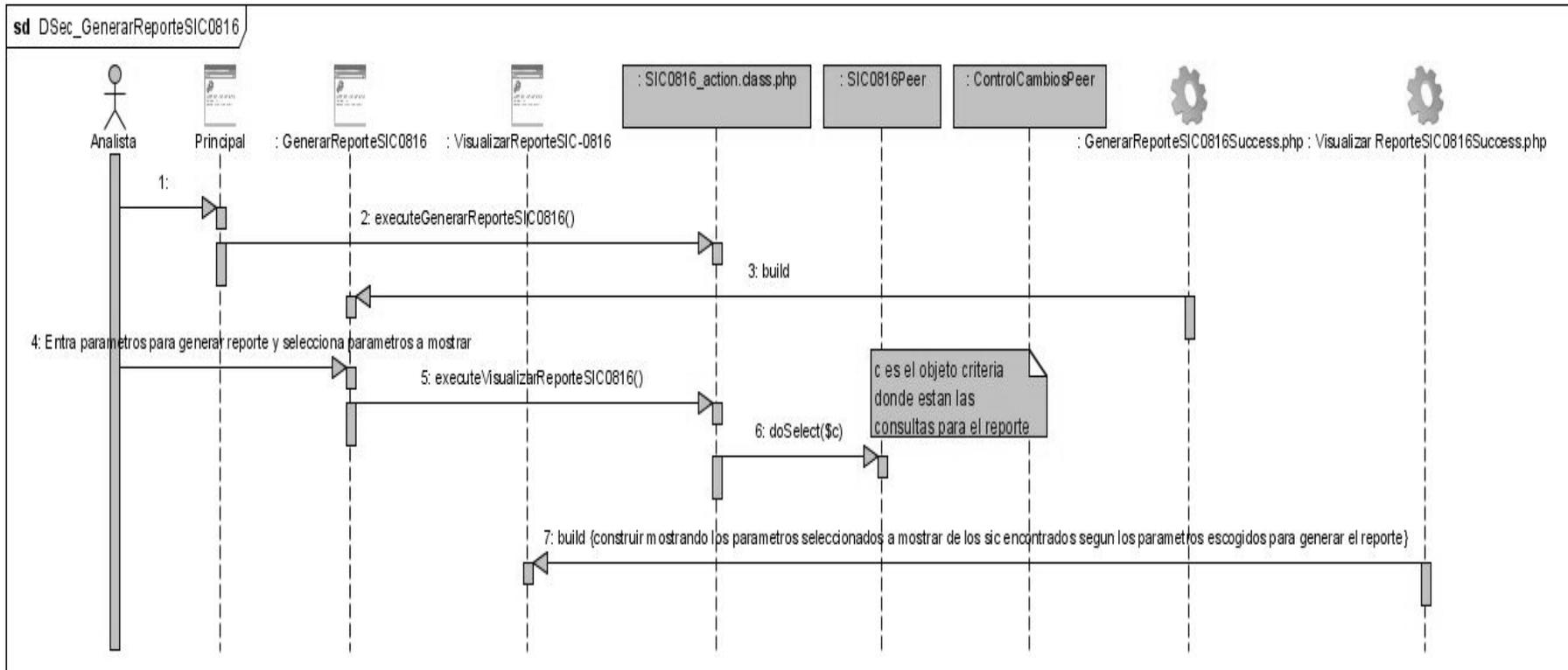


Fig. 35 Escenario Generar Reporte del Registro de Control de Cambios

CU Gestionar Programa de informe de Tendencias

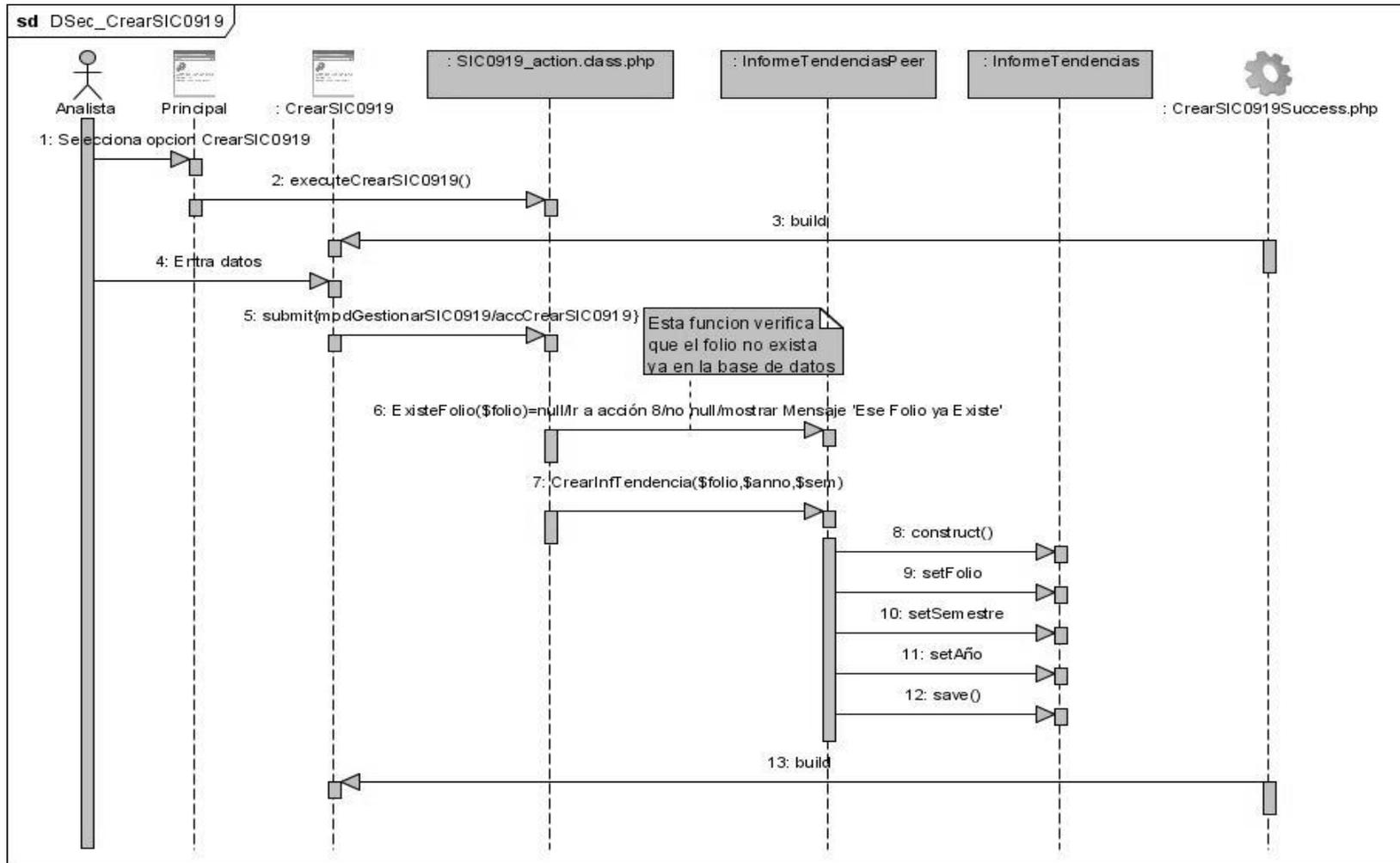


Fig. 36 Escenario Crear Programa de Informe de Tendencias

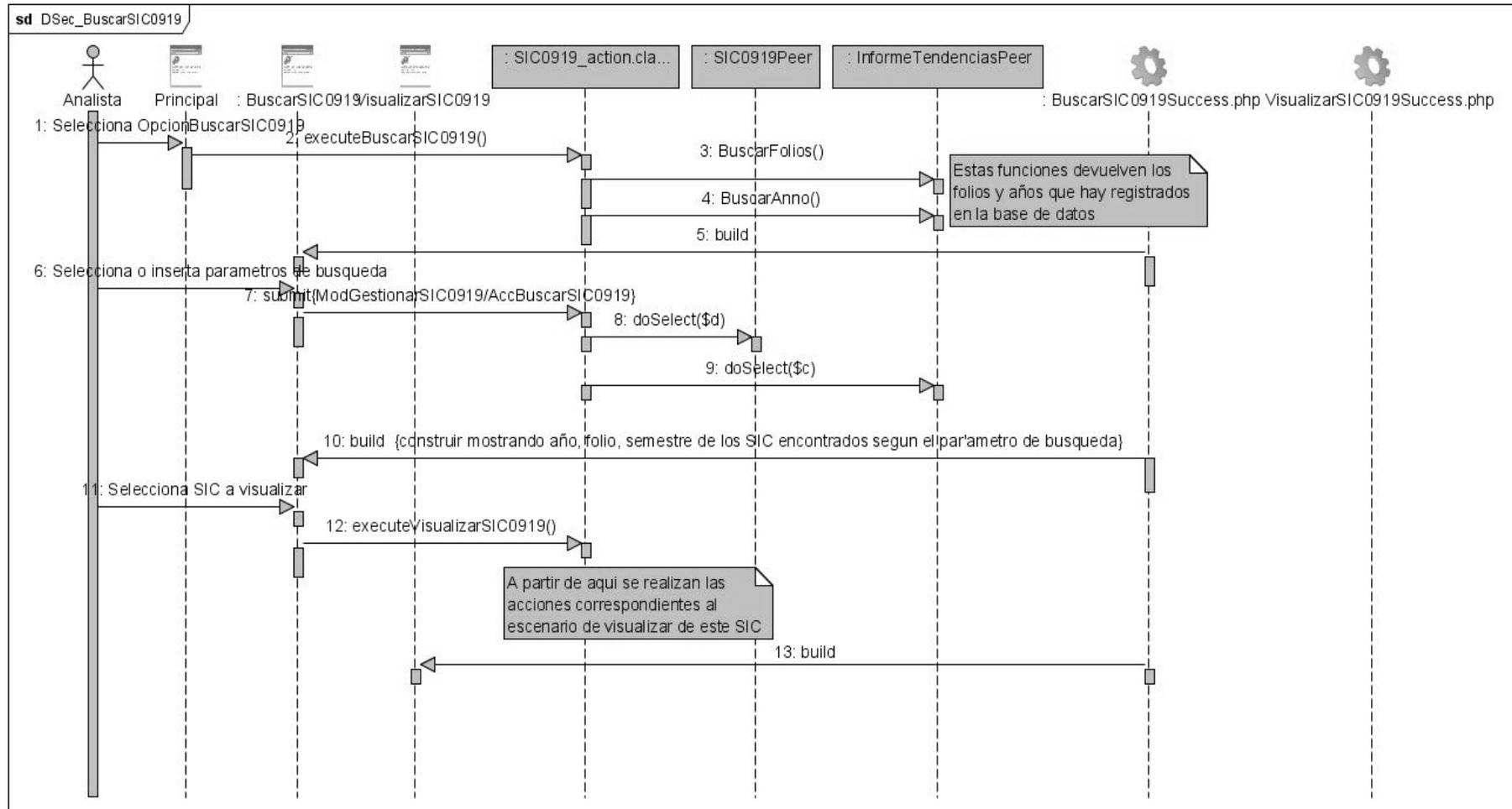


Fig. 37 Escenario Buscar Programa de Informe de Tendencias

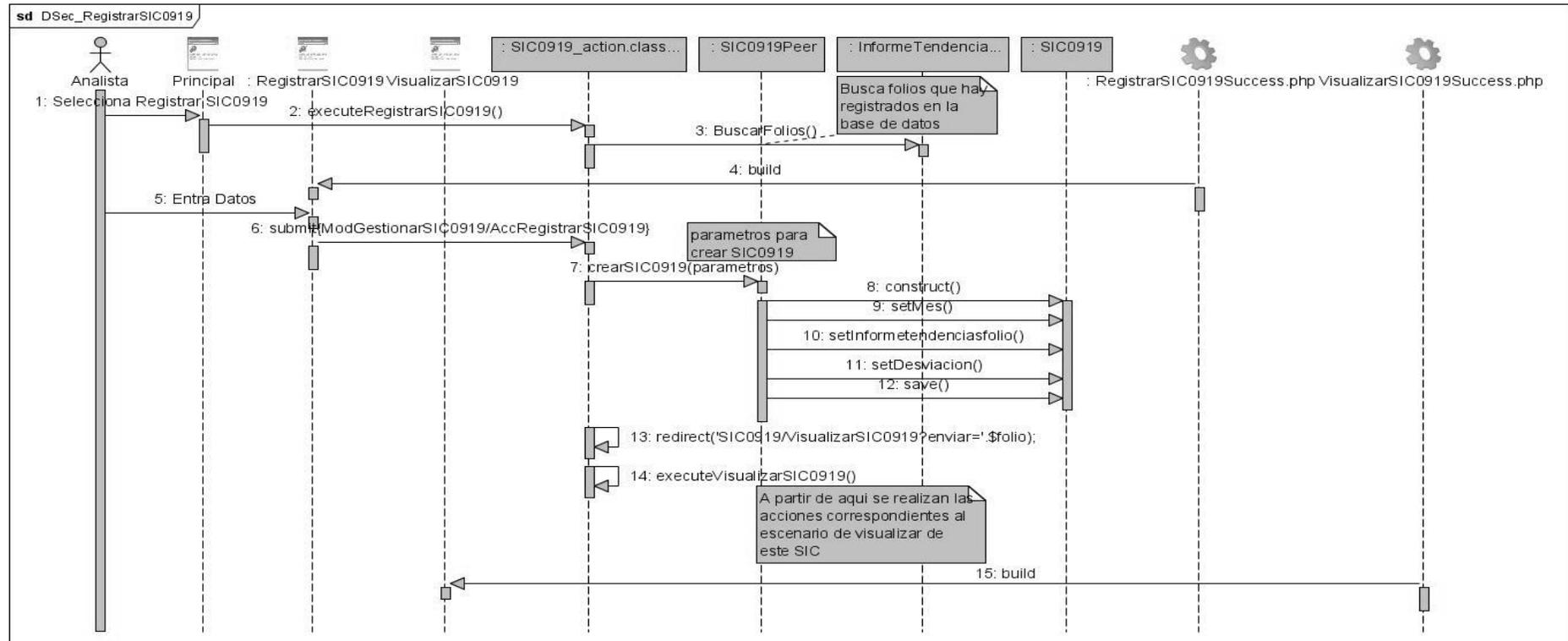


Fig. 38 Escenario Registrar Programa de Informe de Tendencias

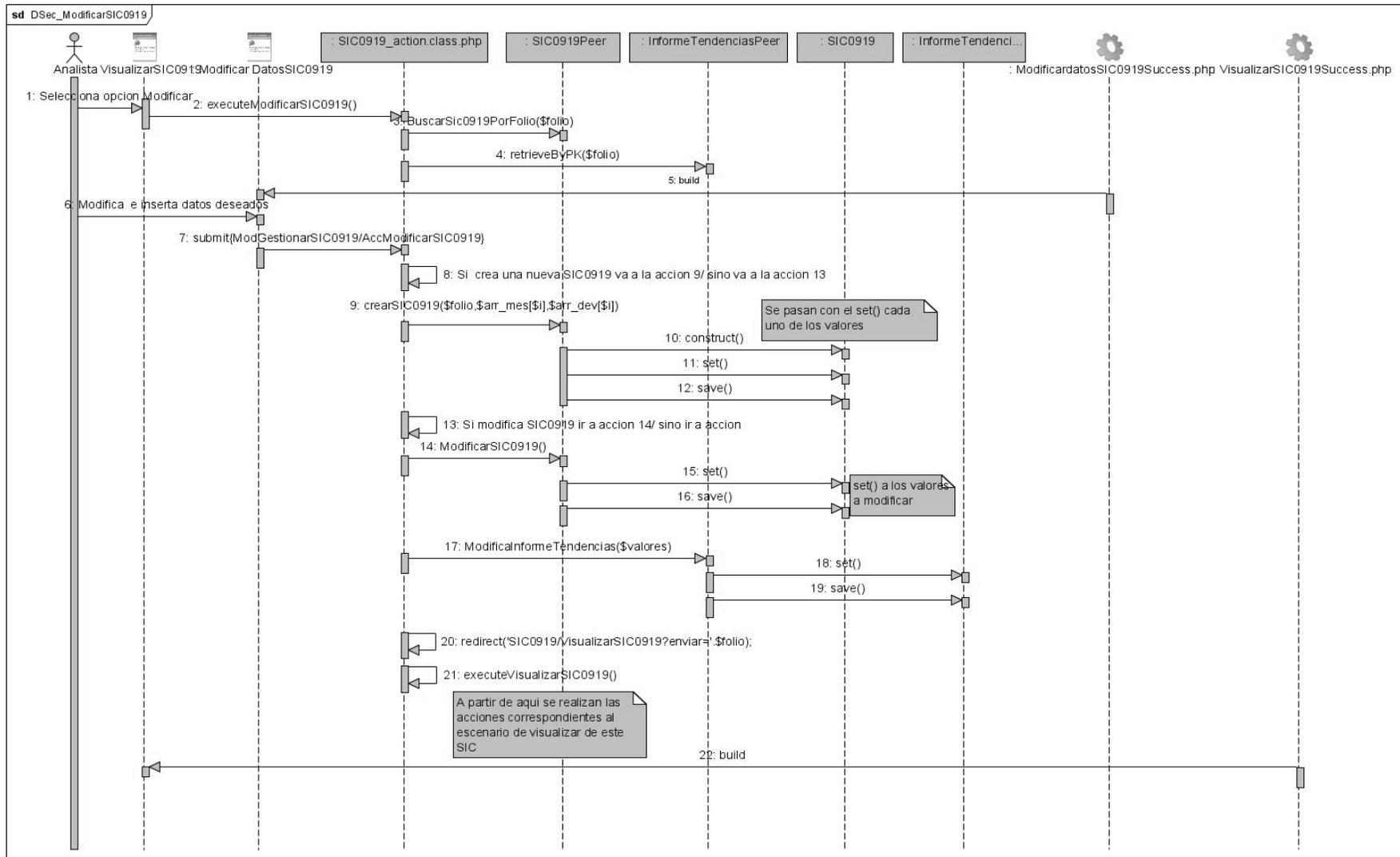
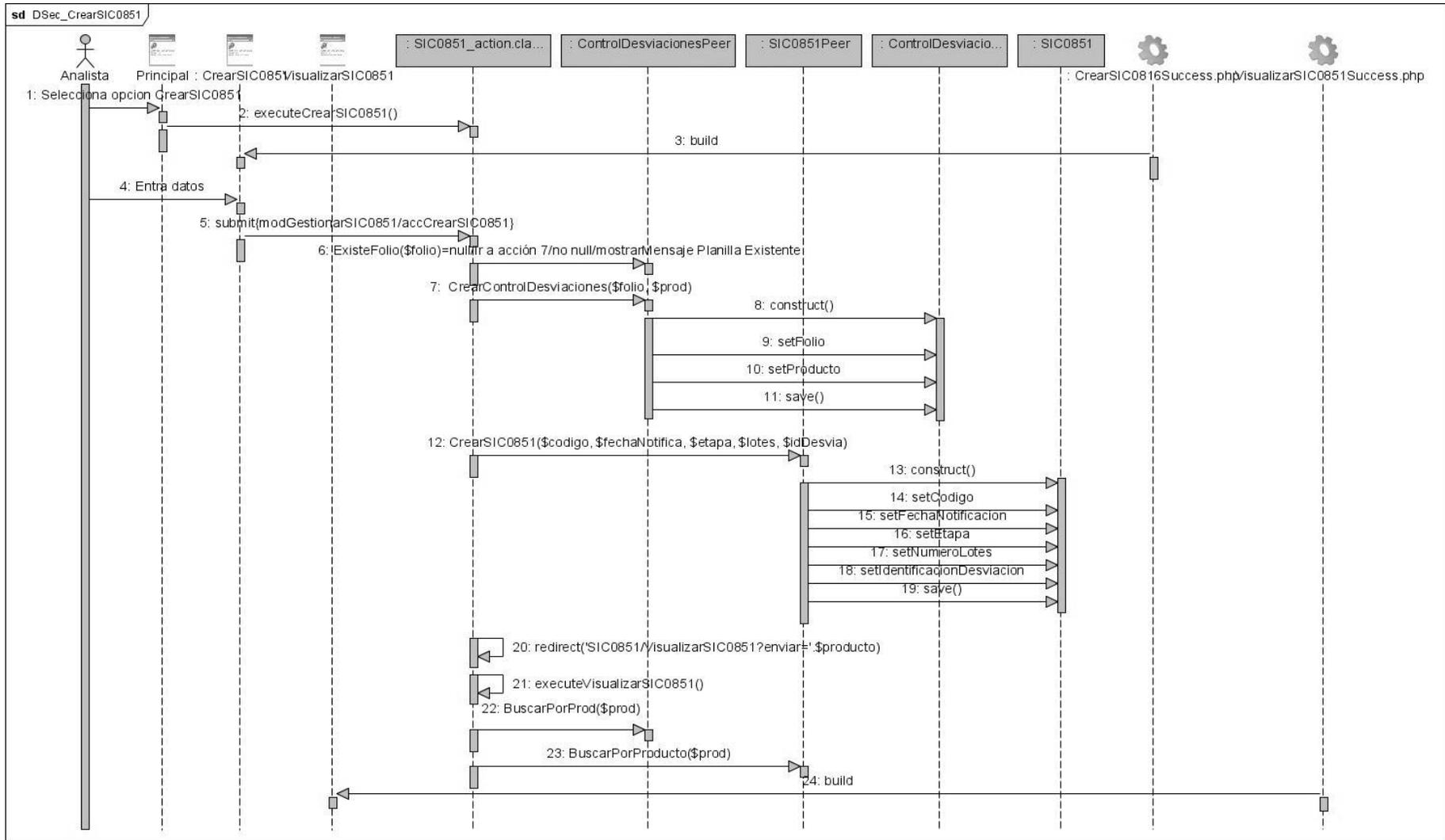


Fig. 39 Escenario Modificar Programa de Informe de Tendencias



CU

Fig. 40 Escenario Generar Reporte del Programa de Informe de Tendencias

Gestionar Registro de Control de las Desviaciones

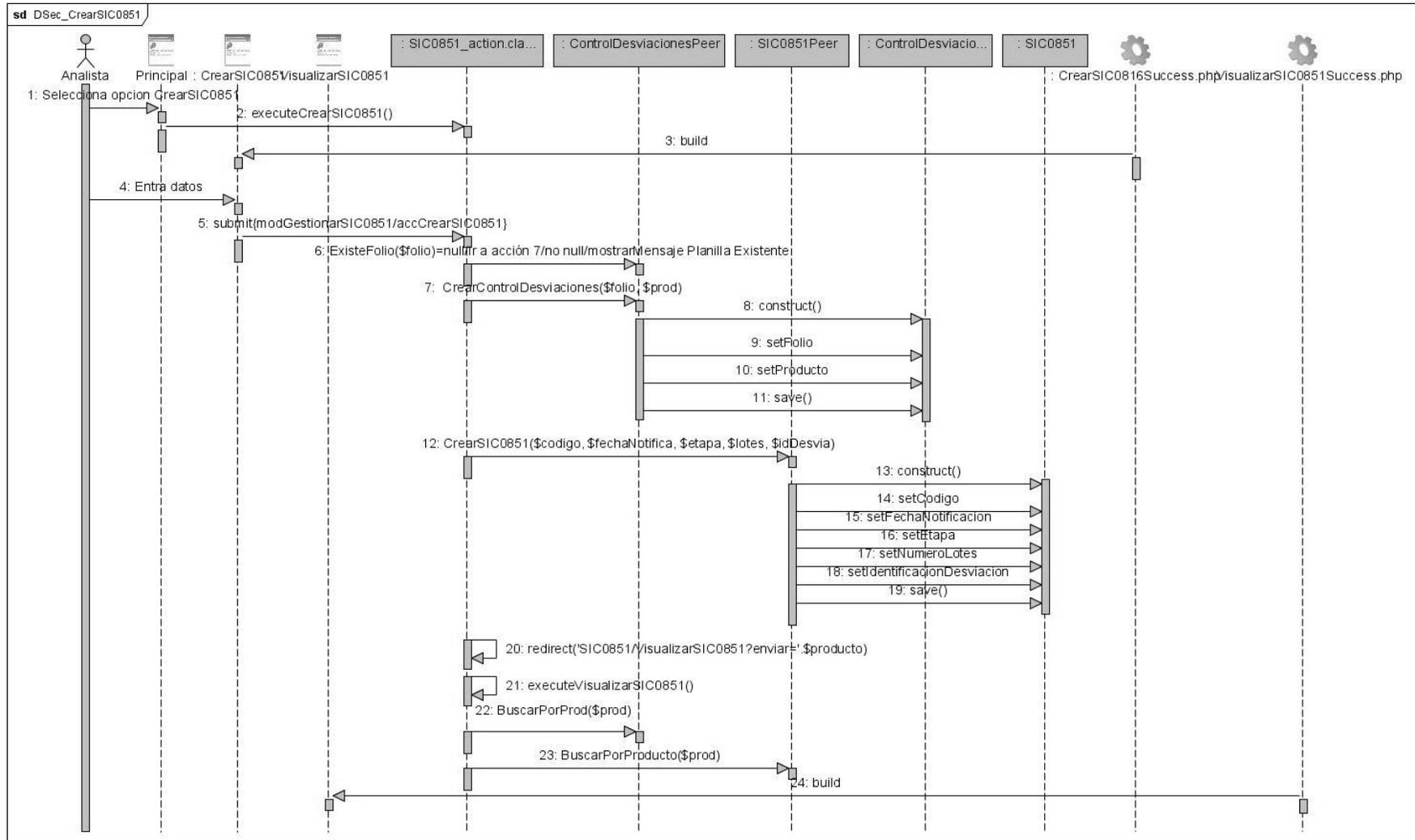


Fig. 41 Escenario Crear Registro de Control de las Desviaciones

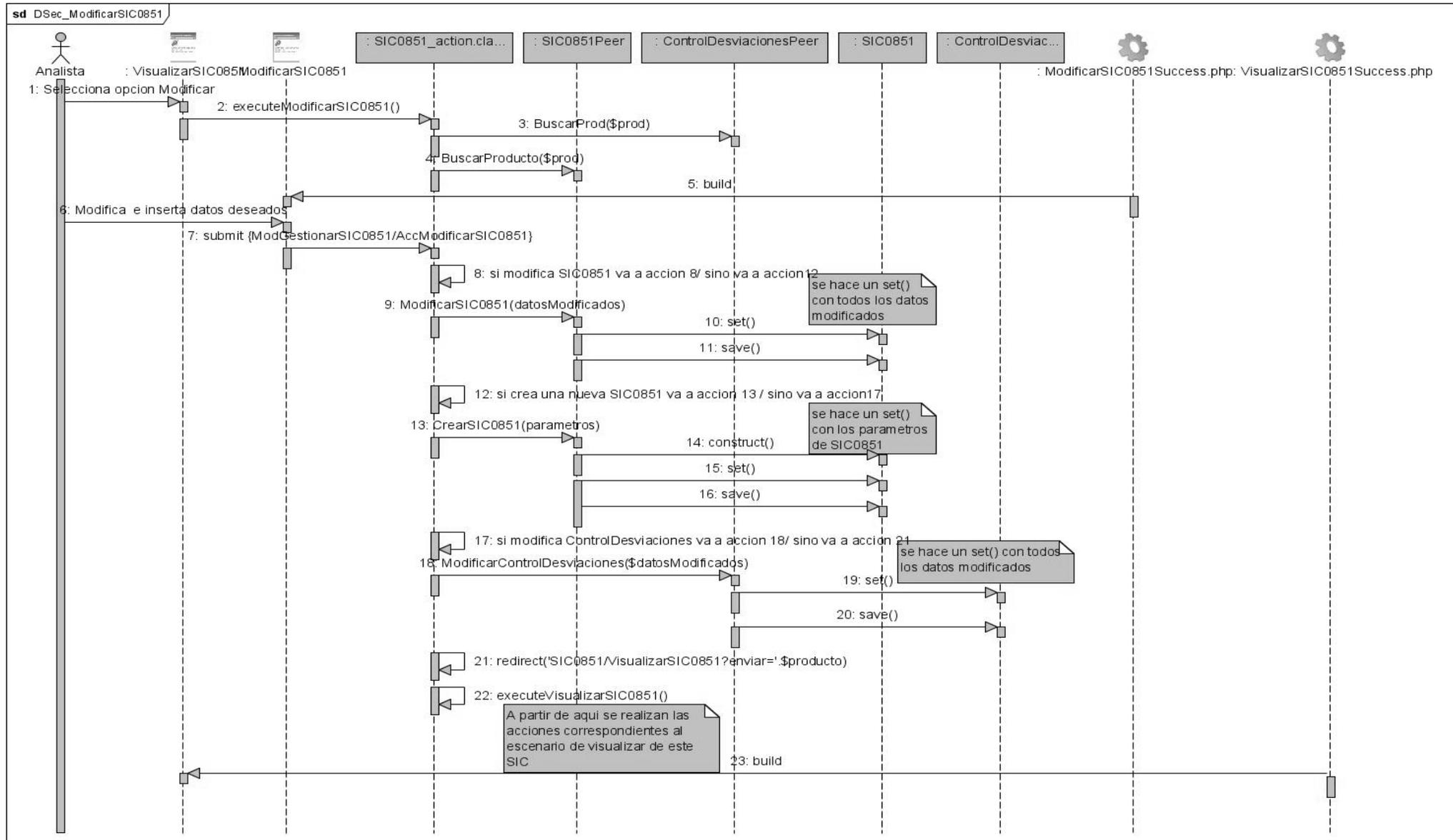


Fig. 42 Escenario Modificar Registro de Control de las Desviaciones

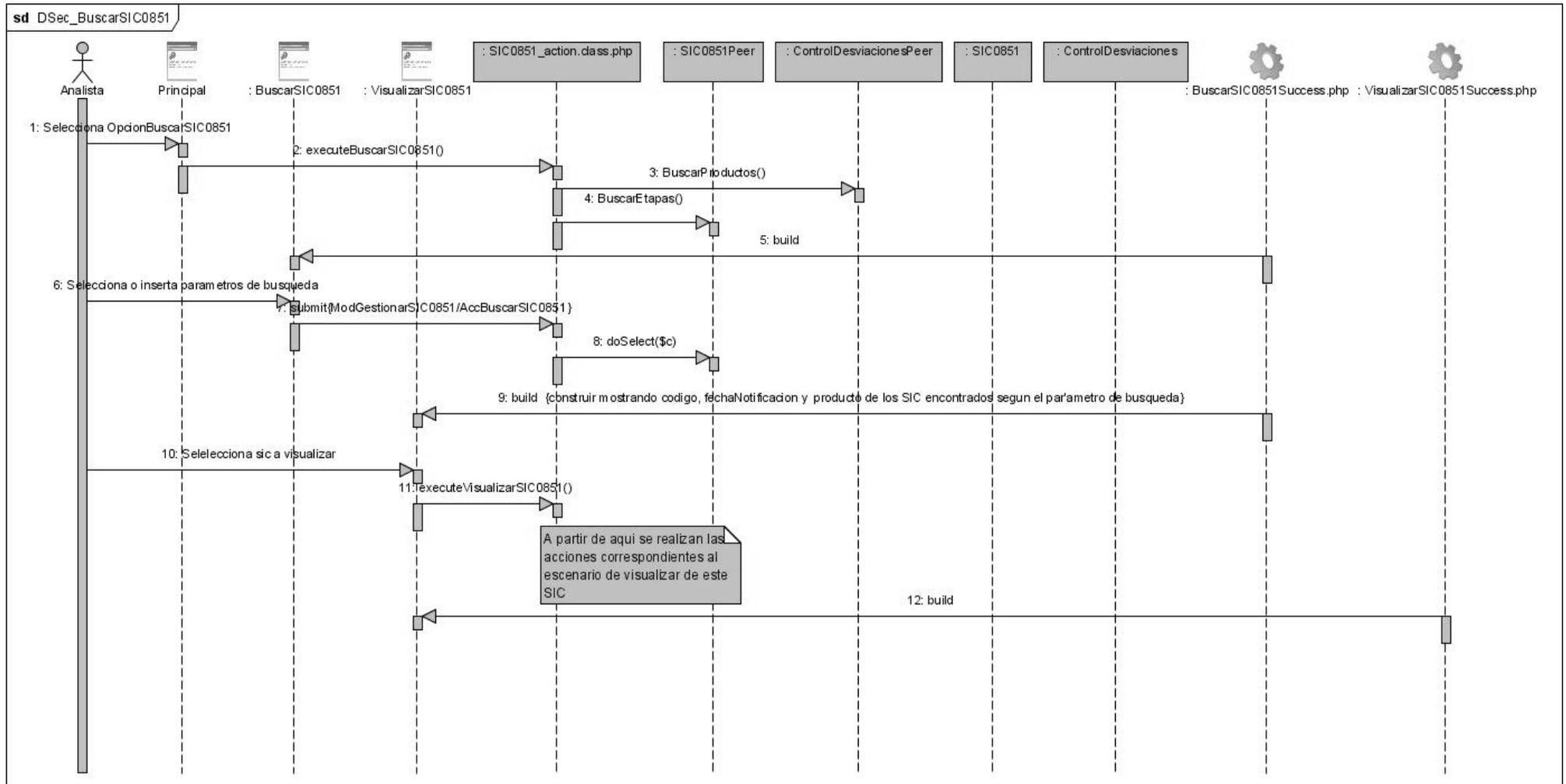


Fig. 43 Escenario Buscar Registro de Control de las Desviaciones

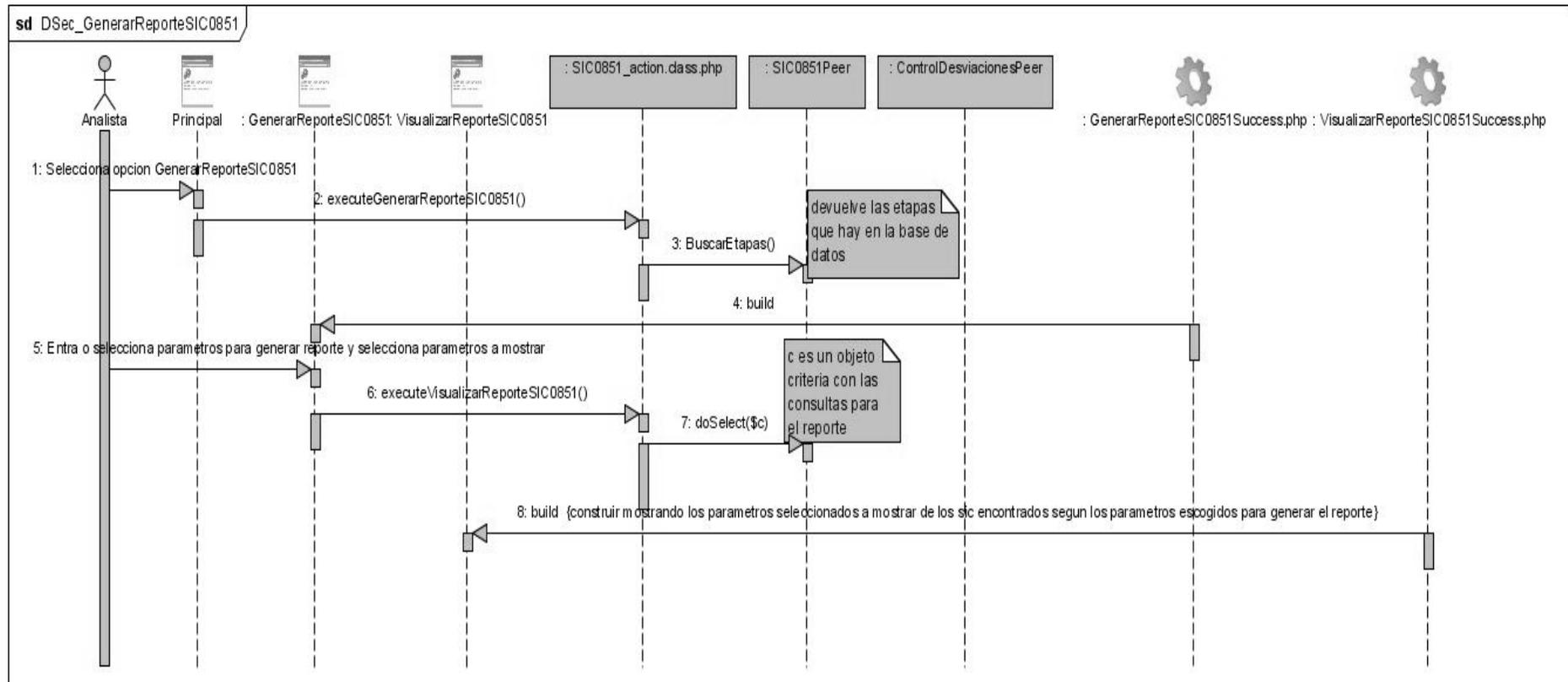


Fig. 44 Escenario Generar Reporte del Registro de Control de las Desviaciones

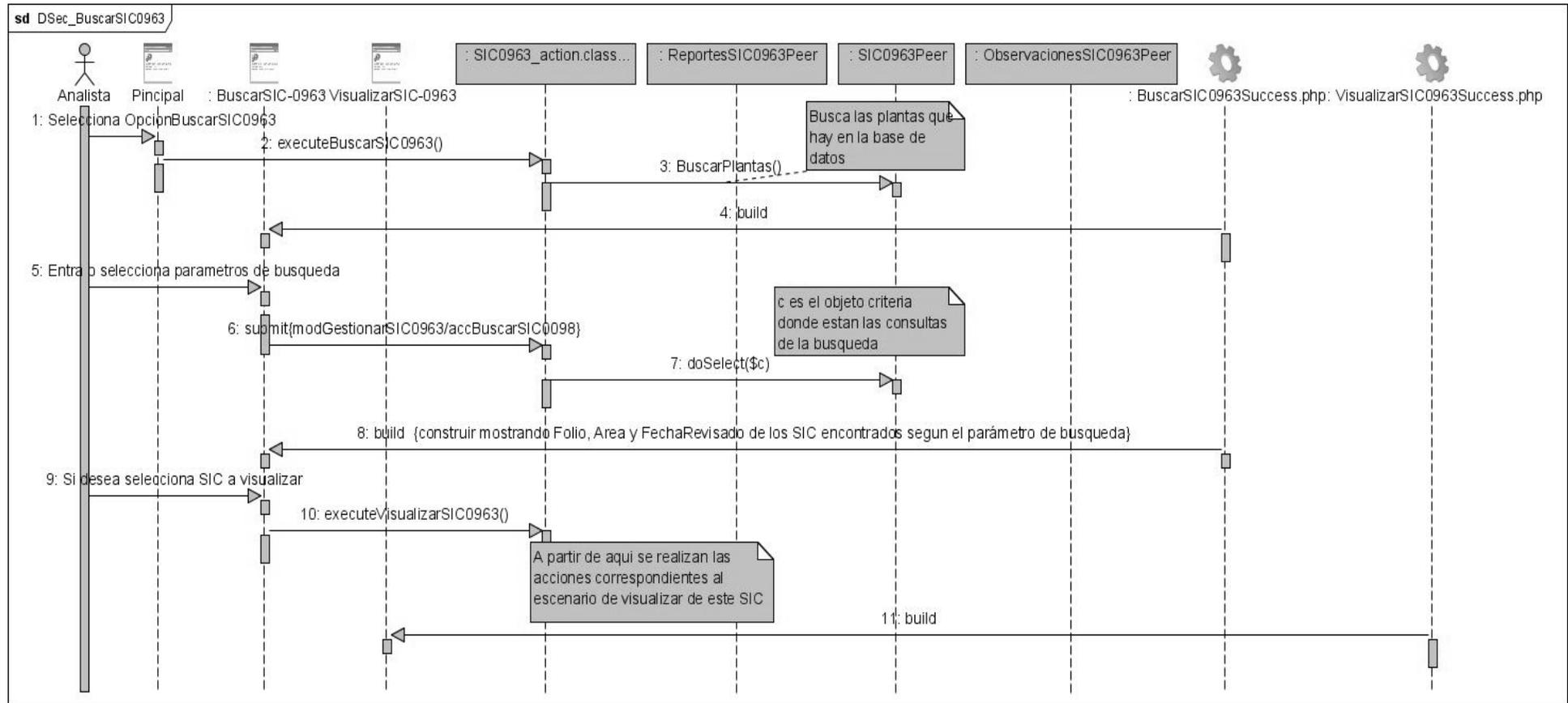


Fig. 46 Escenario Buscar Reporte de Límites

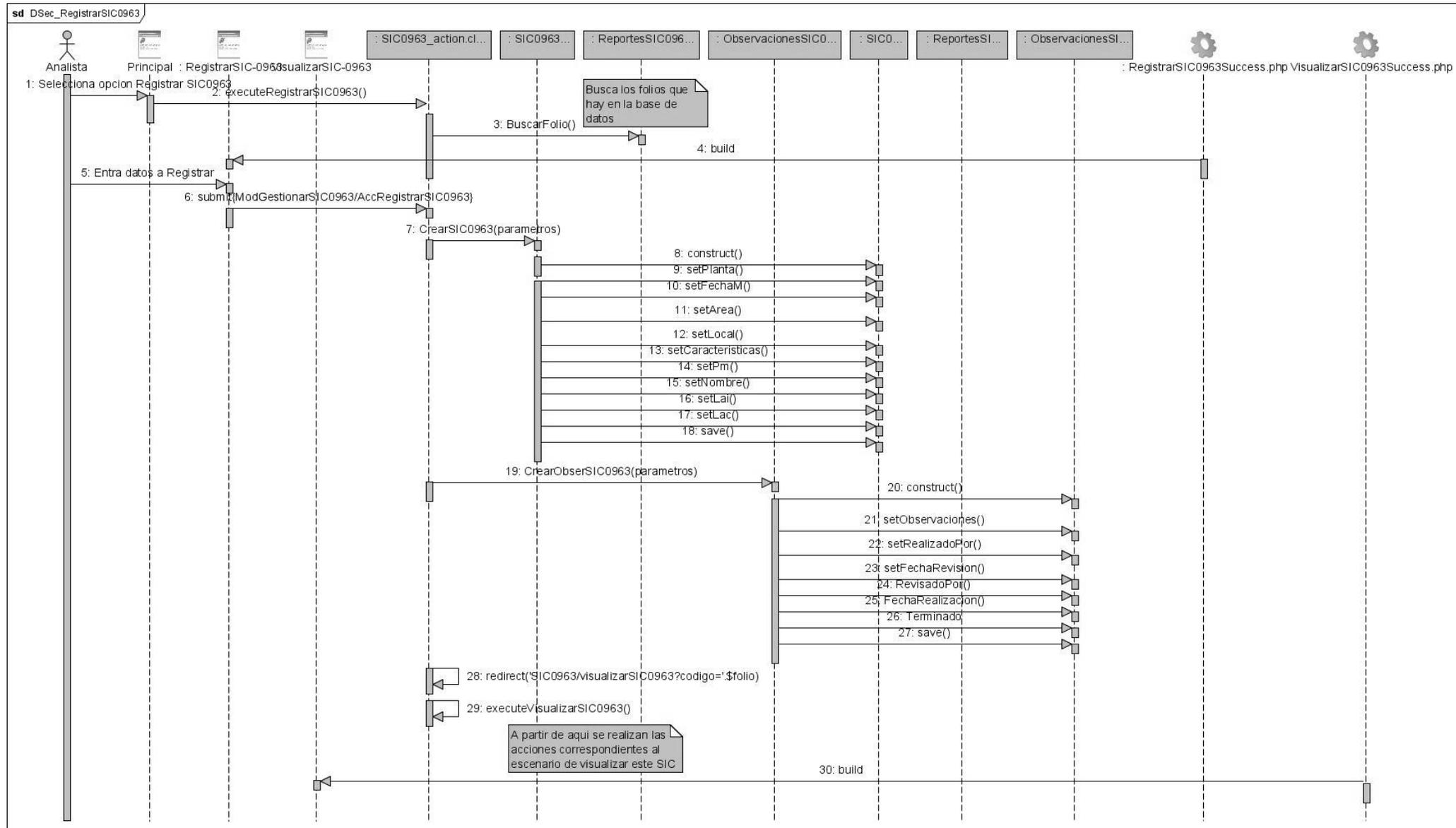


Fig. 47 Escenario Registrar Reporte de Límites

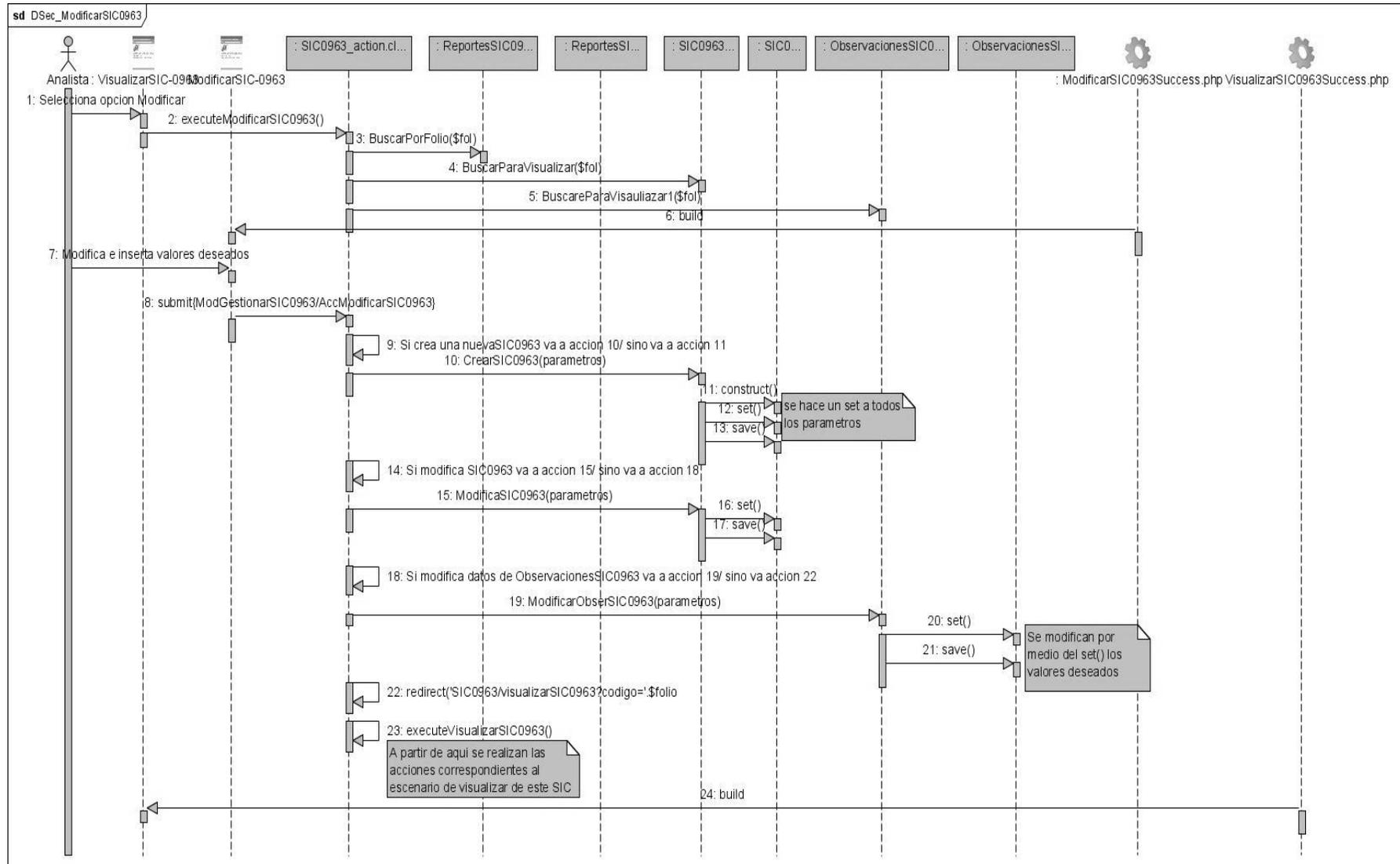


Fig. 48 Escenario Modificar Reporte de Límites

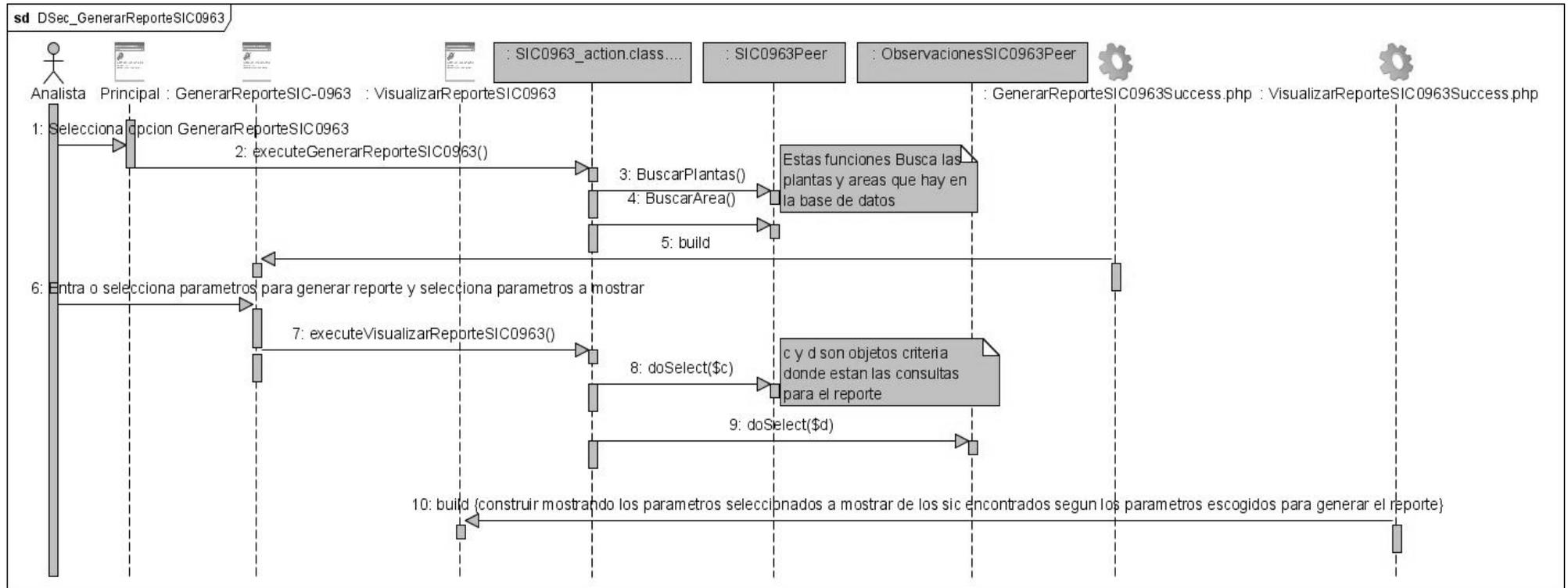


Fig. 49 Escenario Generar Reporte de Reporte de Límites

2.4 Validación del diseño realizado.

Es muy importante que el diseño que se realice sirva como guía para la programación de la aplicación. Con un buen diseño se logra que los programadores trabajen con mayor facilidad y organización. ¿Que mayor evidencia de la eficacia de un diseño, que su utilización en la implementación de la aplicación?

Para validar el diseño realizado se muestran los diagramas de secuencia de los distintos escenarios de un caso de uso que se toma como ejemplo y se muestra el código de lo programado basado en ese diseño.

Seguidamente se dará una explicación del flujo del primer diagrama de secuencia referido al escenario Crear Notificación de Quejas y se mostrará la correspondencia que presenta el código programado con el diseño. Para representar esta correspondencia se enumera la línea de código con el número que se corresponde con la acción del diagrama de secuencia.

El flujo comienza cuando el actor selecciona la opción CrearSIC0042 esta petición es recibida por el controlador frontal Index.php y cuando la clase sfController decodifica la petición se ejecuta la acción correspondiente de la clase action.class.php (**acción 2**), entonces se construye la página CrearSIC0042 (**acción 3**). A continuación el actor inserta los datos requeridos y la página hace un submit al controlador frontal y después de validar los datos se ejecuta la función ExisteCodigo(\$codigo) (**acción 6**), esta función verifica que no exista ya este código en la base de datos, si existe muestra el mensaje 'Ese Código Propuesto ya Existe', sino, se ejecuta la función CrearSIC0042(parametros) (**acción 7**), esta función se encuentra en la clase NotificaciónquejasPeer y se llama desde la clase action.class.php pasándole los parámetros requeridos, allí se construye el objeto, se ejecuta la función set() con cada uno de los parámetros y después save(), seguidamente se repite la misma operación con la función ControlPlanilla(parametros) (**acción 27**), que se encuentra en la clase ControlPlanillaPeer. Posteriormente se ejecuta la función redirect() (**acción 36**), esta función redirecciona a la acción VisualizarSIC0042 pasando los parámetros requeridos para mostrar la planilla creada. Se ejecuta la acción executeVisualizarSIC0042() (**acción 37**), entonces se hace un llamado a la función retrieveByPK(\$llave) en la clase SIC0042Peer (**acción 38**) que es una función de symfony que conociendo la llave primaria, devuelve el objeto relacionado, después se llama la función BuscarPlanilla(\$id) (**acción 39**) que se encuentra en la clase ControlPlanillaPeer, que pasándole el identificador devuelve el objeto relacionado. Finalmente se construye la página VisualizarSIC0042, con los datos encontrados.

A continuación se presentan los diagramas de secuencia y parte del código de la aplicación Web referente a estos.

Para mayor entendimiento se presentan cada uno de los escenarios de los diagramas de secuencia y debajo de cada cual, el código referente a la clase de la acción y al final se muestra el código de las clases 'peer' implicadas en este caso de uso.

CU Notificación de Quejas

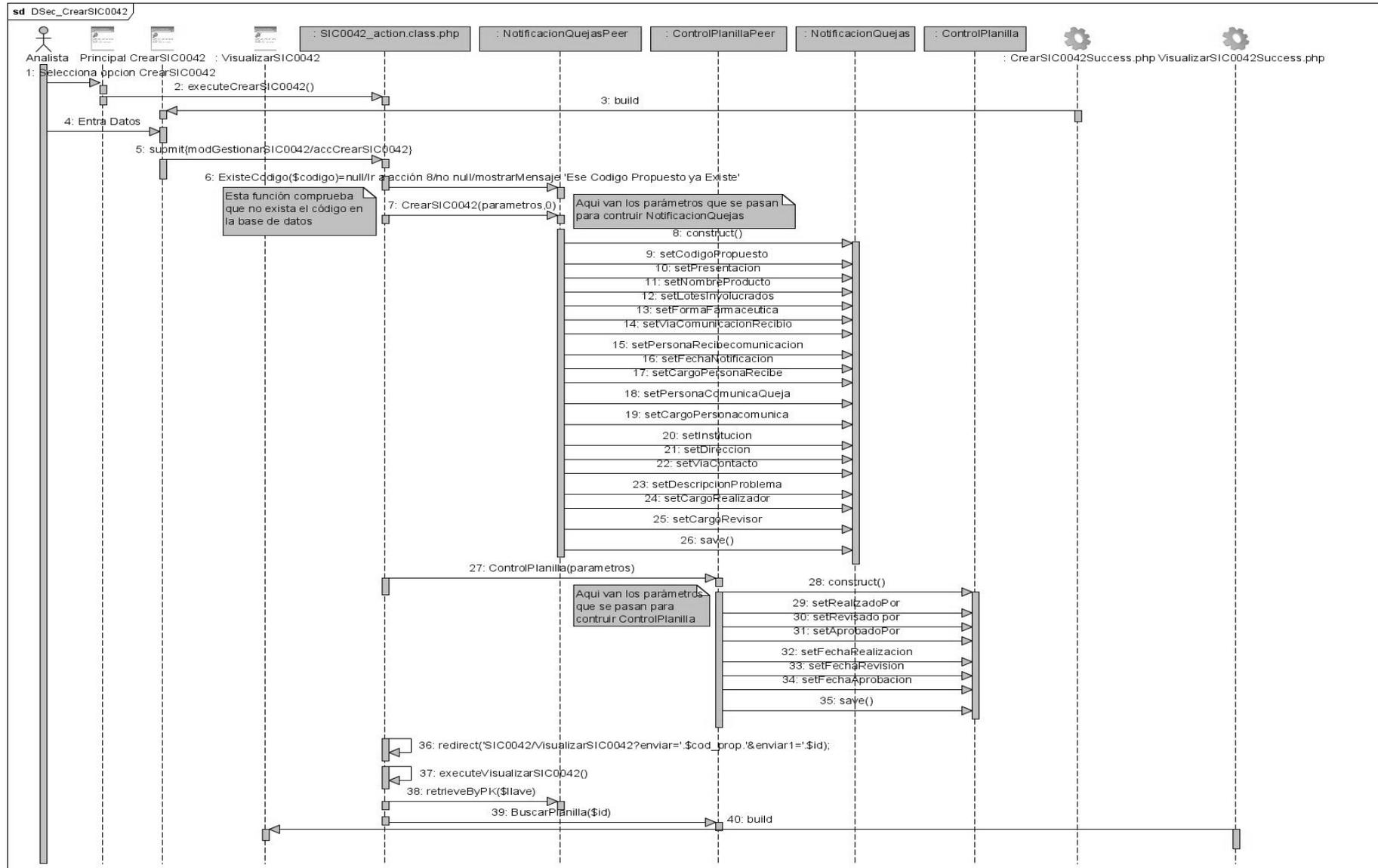


Fig. 50 Escenario Crear Notificación de quejas

Código de la clase de la acción del Módulo SIC0042 referente a la acción de CrearSIC0042 y VisualizarSIC0042.

```
(acción 2) public function executeCrearSIC0042()
{
    if ($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $this->mensaje='';
        (acción 3) return sfView::SUCCESS;
    }
    else
    {
        $cod_prop=$this->getRequestParameter('cod_prop');
        (acción 6) if(NotificacionquejasPeer::ExisteCodigo($cod_prop))
        {
            $this->mensaje='EseCodigoPropuesto ya Existe';
        }
        else
        {
            $nom_prod=$this->getRequestParameter('nom_prod');
            $for_farm=$this->getRequestParameter('for_farm');
            $f_not=$this->getRequestParameter('f_not');
            $present=$this->getRequestParameter('present');
            $lote_inv=$this->getRequestParameter('lote_inv');
            $via_com=$this->getRequestParameter('via_com');
            $pers_not=$this->getRequestParameter('pers_not');
            $car_per=$this->getRequestParameter('car_per');
            $persona_not=$this->getRequestParameter('persona_not');
            $car_per1=$this->getRequestParameter('car_per1');
            $inst=$this->getRequestParameter('inst');
            $direc=$this->getRequestParameter('direc');
            $vias_cont=$this->getRequestParameter('vias_cont');
            $desc_prob=$this->getRequestParameter('desc_prob');

            $real_por=$this->getRequestParameter('real_por');
            $carg_real=$this->getRequestParameter('carg_real');
            $f_real=$this->getRequestParameter('f_real');
            $revi_por=$this->getRequestParameter('revi_por');
            $carg_rev=$this->getRequestParameter('carg_rev');
            $f_rev=$this->getRequestParameter('f_rev');
            $terminado=$this->getRequestParameter('terminado');
```

```
(acción7) $llave=NotificacionquejasPeer::CrearSIC0042($cod_prop,$nom_prod,$for_farm,$f_not,$present,$lote_inv,$via_com,$pers_not,$car_per,$persona_not,$car_per1,$inst,$direc,$vias_cont,$desc_prob,$carg_rev,$carg_real,$id,0);
(acción27) $id=ControlplanillaPeer::ControlPlanilla('','',$real_por,$f_real,'',$revi_por,$f_rev);

        $this->ver=$llave;
        $this-
(acción36) >redirect('SIC0042/VisualizarSIC0042?enviar='.$cod_prop.'&enviar1=
'.$id);    }
        }
    }

(acción 37) public function executeVisualizarSIC0042()
{
    $llave=$this->getRequestParameter('enviar');
    $id=$this->getRequestParameter('enviar1');

(acción 38)     $dev1=NotificacionquejasPeer::retrieveByPK($llave);
                $this->ver1=NotificacionquejasPeer::ConvArrObj($dev1);

(acción 39) $dev=ControlplanillaPeer::BuscarPlanilla($id);
                $this->ver=ControlplanillaPeer::ConvArrayObj($dev);

}
}
```

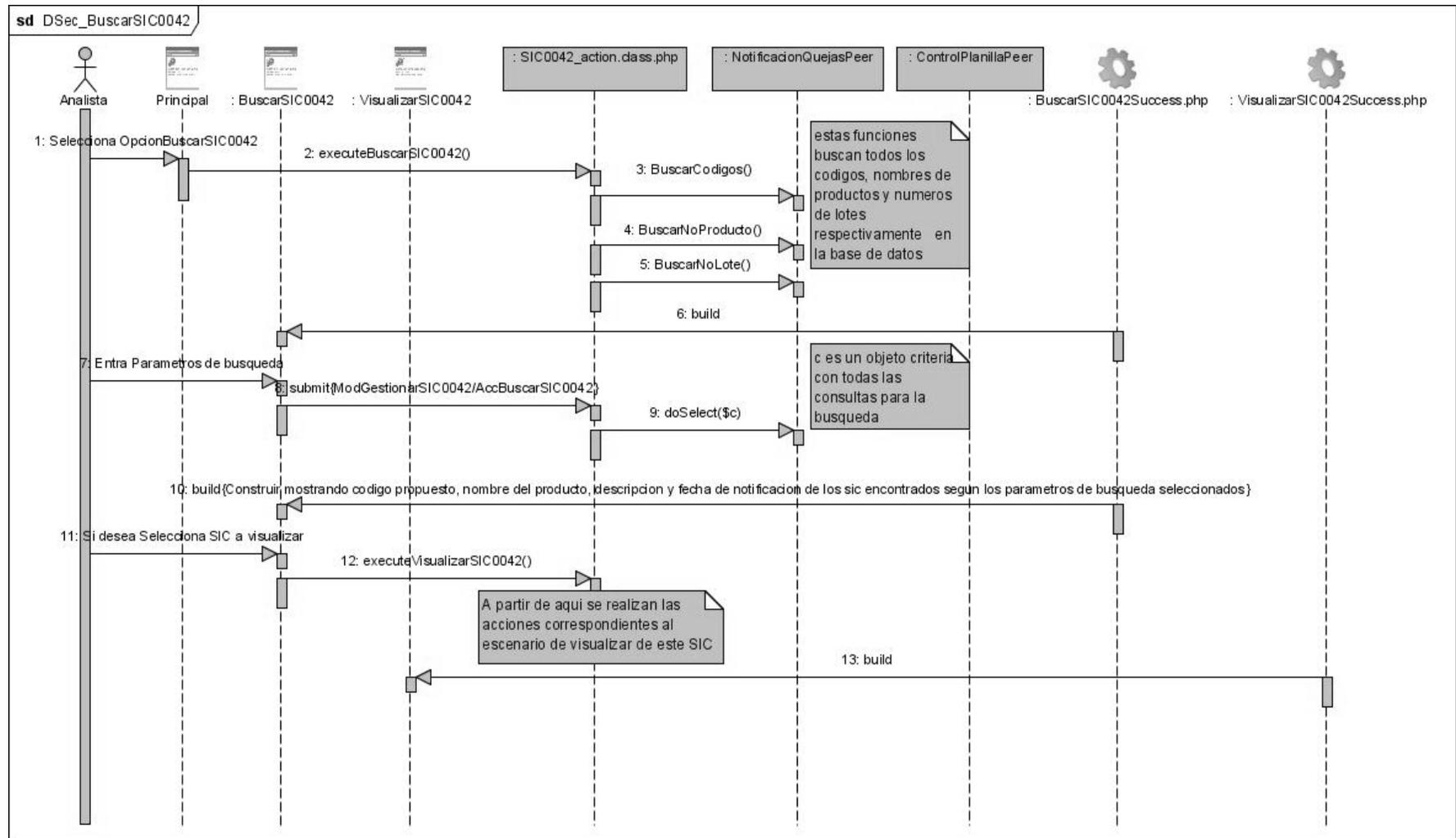


Fig. 51 Escenario Buscar Notificación de quejas

Código de la clase de la acción del Módulo SIC0042 referente a la acción de BuscarSIC0042.

```
public function executeBuscarSIC0042 ()
{
    if ($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $this->lis_cod=NotificacionquejasPeer::BuscarCodigos();
        $this->lis_nomprod=NotificacionquejasPeer::BuscarNoProducto();
        $this->lis_lote=NotificacionquejasPeer::BuscarNoLote();
        $this->ver='';
        $this->mensaje='';
        return sfView::SUCCESS;
    }
    else
    {
        $arr_aux=array();

        $codigo=NotificacionquejasPeer::BuscarCodigos();
        $this->lis_cod=$codigo;
        $nom=NotificacionquejasPeer::BuscarNoProducto();
        $this->lis_nomprod=$nom;
        $lote=NotificacionquejasPeer::BuscarNoLote();
        $this->lis_lote=$lote;

        $pos_cod=$this->getRequestParameter('select_cod');
        $pos_lot=$this->getRequestParameter('select_lot');
        $pos_pro=$this->getRequestParameter('select_prod');

        $f_i=$this->getRequestParameter('f_ini');
        $f_f=$this->getRequestParameter('f_fin');
        if( $pos_cod==0 && $pos_lot==0 && $pos_pro==0 && ($f_f==' ' ||
        $f_i==' '))
        {
            $this->ver='';
            $this->mensaje='Entre Criterio de Busqueda';
            return sfView::SUCCESS;
        }
        else
        {
            $this->mensaje='';
            $c= new Criteria();
            if($pos_cod>0)
            $c->add(NotificacionquejasPeer::CODIGOPROP,$codigo[$pos_cod]);
            if($pos_lot>0)
            $c->add(NotificacionquejasPeer::LOTESINVOLUCRADOS,$lote[$pos_lot]);
        }
    }
}
```

```
        if($pos_pro>0)
        $c->add(NotificacionquejasPeer::NOMBREPRODUCTO,$nom[$pos_pro]);

        $sic=NotificacionquejasPeer::doSelect($c);
        if($f_f!='' || $f_i!='')
        {
            $fecha_i=str_replace('-', '', $f_i);
            $fecha_f=str_replace('-', '', $f_f);
for($i=0;$i<count($sic);$i++)
        {
            $fecha=$sic[$i]->getFechanotificacion();
            if($fecha>=$fecha_i && $fecha<=$fecha_f)
            array_push($arr_aux,$sic[$i]);
        }
        else
            $arr_aux=$sic;

        }

        $this->ver=NotificacionquejasPeer::ConvArrObj1($arr_aux);
    }
}
```

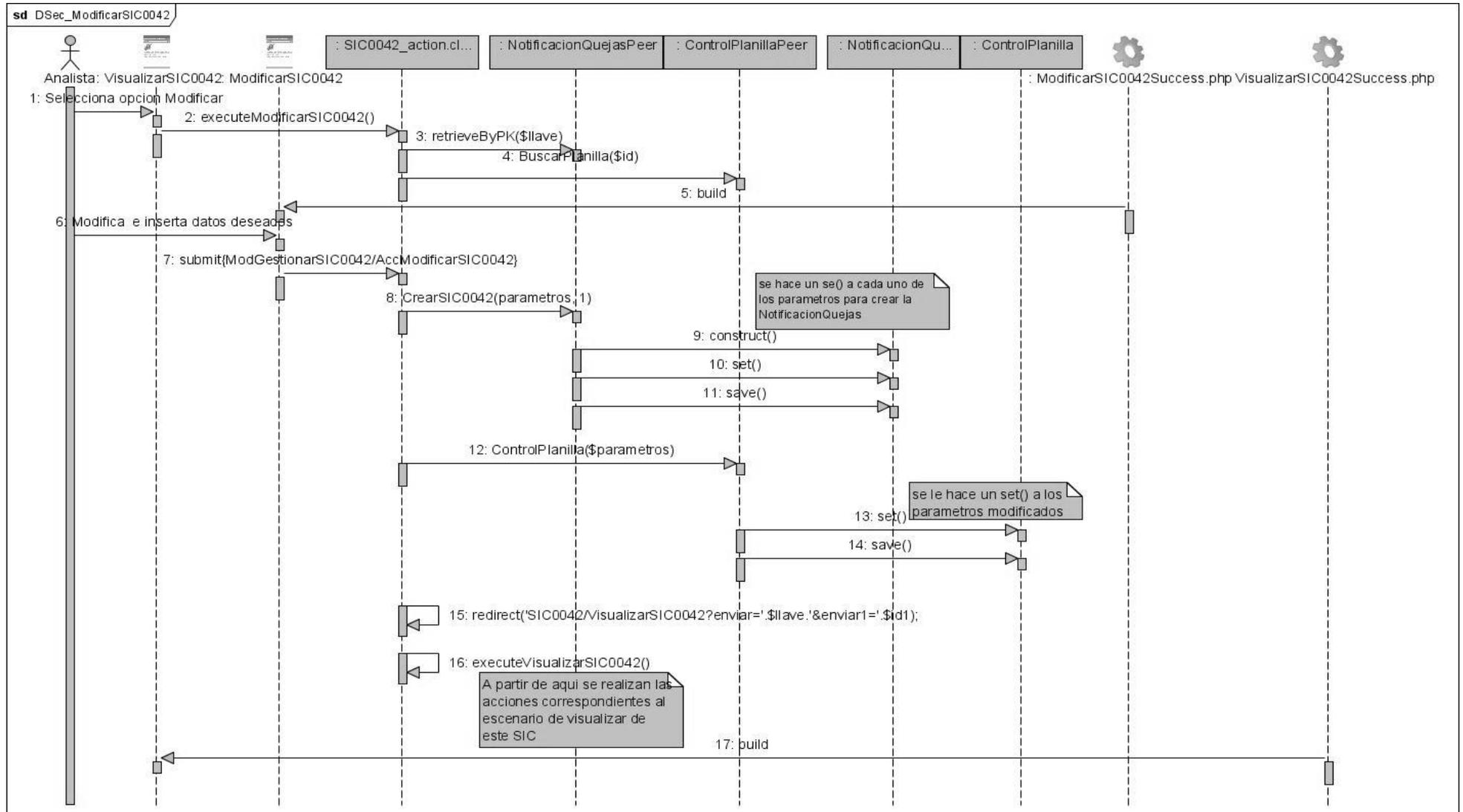


Fig. 52 Escenario Modificar Notificación de quejas

Código de la clase de la acción del Modulo SIC0042 referente a la acción de ModificarSIC0042.

```
public function executeModificarSIC0042()
{
    if ($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $llave=$this->getRequestParameter('enviar');
        $id=$this->getRequestParameter('enviar1');

        $dev=ControlplanillaPeer::BuscarPlanilla($id);
        $this->ver=ControlplanillaPeer::ConvArrayObj($dev);

        $dev1=NotificacionquejasPeer::retrieveByPK($llave);
        $this->ver1=NotificacionquejasPeer::ConvArrObj($dev1);
        $this->mensaje='';

        return sfView::SUCCESS;

    }
    else
    {
        $llave=$this->getRequestParameter('enviar');
        $id1=$this->getRequestParameter('enviar1');

        $dev=ControlplanillaPeer::BuscarPlanilla($id1);
        $this->ver=ControlplanillaPeer::ConvArrayObj($dev);

        $dev1=NotificacionquejasPeer::retrieveByPK($llave);
        $this->ver1=NotificacionquejasPeer::ConvArrObj($dev1);

        $nom_prod=$this->getRequestParameter('nom_prod');
        $for_farm=$this->getRequestParameter('for_farm');
        $f_not=$this->getRequestParameter('f_not');
        $present=$this->getRequestParameter('present');
        $lote_inv=$this->getRequestParameter('lote_inv');
        $via_com=$this->getRequestParameter('via_com');
        $pers_not=$this->getRequestParameter('pers_not');
        $car_per=$this->getRequestParameter('car_per');
        $persona_not=$this->getRequestParameter('persona_not');
        $car_per1=$this->getRequestParameter('car_per1');
        $inst=$this->getRequestParameter('inst');
        $direc=$this->getRequestParameter('direc');
        $vias_cont=$this->getRequestParameter('vias_cont');
        $desc_prob=$this->getRequestParameter('desc_prob');

        $real_por=$this->getRequestParameter('real_por');
        $carg_real=$this->getRequestParameter('carg_real');
        $f_real=$this->getRequestParameter('f_real');
```

```
        $revi_por=$this->getRequestParameter('revi_por');

        $carg_rev=$this->getRequestParameter('carg_rev');

    $f_rev=$this->getRequestParameter('f_rev');
        $terminado=$this->getRequestParameter('terminado');
        $this->ver3=$llave;
        $this->ver4=$id1;

        $id=ControlplanillaPeer::ControlPlanilla($id1,'',$real_por,$f_real,'',
    $revi_por,$f_rev);

        $llavep=NotificacionquejasPeer::CrearSIC0042($llave,$nom_prod,$for_far
    m,$f_not,$present,$lote_inv,$via_com,$pers_not,$car_per,$persona_not,$car_pe
    r1,$inst,$direc,$vias_cont,$desc_prob,$carg_rev,$carg_real,$id1,1);

        $this-
    >redirect('SIC0042/VisualizarSIC0042?enviar=.'.$llave.'&enviar1=.'.$id1);

        }
    }
```

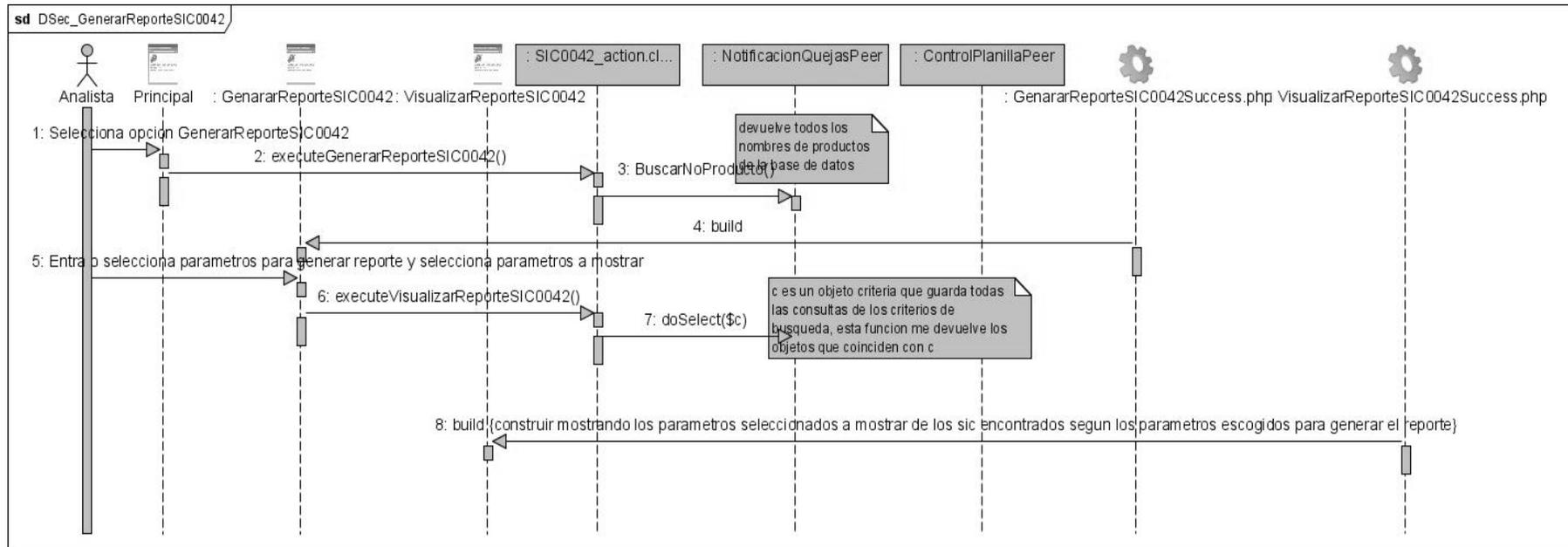


Fig. 53 Escenario Generar Reporte de la Notificación de quejas

Código de la clase de la acción del Módulo SIC0042 referente a la acción de GenerarReporteSIC0042.

```
public function executeGenerarReporteSIC0042 ()
{
    $this->mensaje=$this->getRequestParameter('mensaje');
    $this->list_prod=NotificacionquejasPeer::BuscarNoProducto();
}
public function executeVisualizarReporteSIC0042 ()
{
    $arr_aux=array();
    $arry_dev=array();
    $nom=NotificacionquejasPeer::BuscarNoProducto();
    $nomprod=$this->getRequestParameter('select_prod');

    $f_i=$this->getRequestParameter('f_ini');
    $f_f=$this->getRequestParameter('f_fin');

    if( $nomprod==0 &&( $f_f==' ' || $f_i==' '))
    {
        $this->redirect('SIC0042/GenerarReporteSIC0042?mensaje=Entre Criterio
Para Generar Reporte');
    }
    else
    {
        $c=new Criteria();
        if($nomprod>0)
        $c->add(NotificacionquejasPeer::NOMBREPRODUCTO,$nom[$nomprod]);

        $sic=NotificacionquejasPeer::doSelect($c);
        if($f_f!=' ' || $f_i!=' ')
        {
            $fecha_i=str_replace('-', '', $f_i);
            $fecha_f=str_replace('-', '', $f_f);

            for($i=0;$i<count($sic);$i++)
            {
                $fecha=$sic[$i]->getFechanotificacion();
                if($fecha>=$fecha_i && $fecha<=$fecha_f)
                array_push($arr_aux,$sic[$i]);
            }
        }
        else
        $arr_aux=$sic;
    }

    $ver2=NotificacionquejasPeer::ConvArrObj1($arr_aux);
}
```

```
// $this->ver=$ver2;

$todos=$this->getRequestParameter('todos');
$forma=$this->getRequestParameter('forma');
$cod=$this->getRequestParameter('cod');
$via_com=$this->getRequestParameter('via_com');

$f_not=$this->getRequestParameter('f_not');
$n_prod=$this->getRequestParameter('n_prod');

if($todos==1 || $forma==1 || $cod==1 || $via_com==1 || $f_not==1 ||
$n_prod==1)
{
    if($todos==1)
    {
        $forma=1;
        $cod=1;
        $via_com=1;
        $f_not=1;
        $n_prod=1;
    }
    if($forma==1)
    {
$aux=array('Forma Farmaceutica');
        for($i=0;$i<count($ver2);$i++)
        {
            $aux[]=$ver2[$i][1];
        }
        $arry_dev[]=$aux;
    }
    if($cod==1)
    {
        $aux=array('Codigo Propuesto');
        for($i=0;$i<count($ver2);$i++)
        {
            $aux[]=$ver2[$i][0];
        }
        $arry_dev[]=$aux;
    }

    if($via_com==1)
    {
        $aux=array('Via de Comunicacion');
        for($i=0;$i<count($ver2);$i++)
        {
            $aux[]=$ver2[$i][6];
        }
    }
}
```

```
    $arry_dev[]=$aux;
}
if($f_not==1)
{
    $aux=array('Forma Farmaceutica');
    for($i=0;$i<count($ver2);$i++)
    {
        $anno= substr($ver2[$i][3], 0, 4);
        $mes=substr($ver2[$i][3], 4, 2);
        $dia=substr($ver2[$i][3], 6);
        $frd=$anno.'-'. $mes.'-'. $dia;
        $aux[]=$frd;
    }
    $arry_dev[]=$aux;
}
if($n_prod==1)
{
    $aux=array('Nom. del Producto');
    for($i=0;$i<count($ver2);$i++)
    {
        $aux[]=$ver2[$i][2];
    }
    $arry_dev[]=$aux;
}
else
{
    $this->redirect('SIC0042/GenerarReporteSIC0042?mensaje=Marca Campo para
Mostrar el Reporte');
}
    $this->ver=$arry_dev;
}
}
```

Código de las clases *NotificaciónQuejasPeer* y *ControlPlanillaPeer*.

NotificaciónQuejasPeer

```
class NotificacionquejasPeer extends BaseNotificacionquejasPeer
{
    public static function
    CrearSIC0042($cod,$nom_prod,$for_farm,$f_not,$present,$slot_inv,$via_c,$pers_
    res,$carg_pers_res,$pers_com,$carg_pers_com,$inst,$direcc,$via_cont,$desc_pr
    ob,$carg_rev,$carg_reali,$id,$ban)
    {
        if($ban==0)
        {
            $sic= new Notificacionquejas();

            $sic->setCodigoprop($cod);
        }
        else
            $sic=NotificacionquejasPeer::retrieveByPK($cod);

        $sic->setCargopersonacomunica($carg_pers_com);
        $sic->setCargopersonarecibe($carg_pers_res);
        $sic->setCargorevisor($carg_rev);
        $sic->setCargorealizador($carg_reali);
        $sic->setDescripcionproblema($desc_prob);
        $sic->setDireccion($direcc);
        $fec2=str_replace('-', '', $f_not);
        $sic->setFechaNotificacion($fec2);
        $sic->setFormafarmacutica($for_farm);
        $sic->setInstitucion($inst);
        $sic->setLotesinvolucrados($slot_inv);
        $sic->setNombreproducto($nom_prod);
        $sic->setPersonacomunicaqueja($pers_com);
        $sic->setPersonarecibecomunicacion($pers_res);
        $sic->setPresentacion($present);
        $sic->setViacomunicacionrecibio($via_c);
        $sic->setViacontacto($via_cont);
        $sic->setControlplanillaidcontrolplanilla($id);

        $sic->save();

    }
    public static function ExisteCodigo($cod)
    {
        $arr=NotificacionquejasPeer::retrieveByPK($cod);
        if(count($arr)>0)
```

```
        return true;

        return false;
    }
    public static function ConvArrObj($arr)
    {
$aux=array();
        for($i=0;$i<count($arr);$i++)
        {
            array_push($aux,$arr->getCodigoprop());
            array_push($aux,$arr->getFormafarmaceutica());
            array_push($aux,$arr->getNombreproducto());
            array_push($aux,$arr->getFechanotificacion());
            array_push($aux,$arr->getPresentacion());
            array_push($aux,$arr->getLotesinvolucrados());
            array_push($aux,$arr->getViacomunicacionrecibio());
            array_push($aux,$arr->getPersonarecibecomunicacion());
            array_push($aux,$arr->getCargopersonarecibe());
            array_push($aux,$arr->getPersonacomunicaqueja());
            array_push($aux,$arr->getCargopersonacomunica());
            array_push($aux,$arr->getInstitucion());
            array_push($aux,$arr->getDescripcionproblema());
            array_push($aux,$arr->getViacontacto());
            array_push($aux,$arr->getDescripcionproblema());
            array_push($aux,$arr->getCargorealizador());
            array_push($aux,$arr->getCargorevisor());
            array_push($aux,$arr->getControlplanillaidcontrolplanilla());

            return $aux;
        }
    }
    public static function ConvArrObj1($arr)
    {

$aux1=array();
        for($i=0;$i<count($arr);$i++)
        {

$aux=array();
            array_push($aux,$arr[$i]->getCodigoprop());
            array_push($aux,$arr[$i]->getFormafarmaceutica());
            array_push($aux,$arr[$i]->getNombreproducto());
            array_push($aux,$arr[$i]->getFechanotificacion());
            array_push($aux,$arr[$i]->getPresentacion());
            array_push($aux,$arr[$i]->getLotesinvolucrados());
```

```
array_push($aux,$arr[$i]->getViacomunicacionrecibio());
    array_push($aux,$arr[$i]->getPersonarecibecomunicacion());
    array_push($aux,$arr[$i]->getCargopersonarecibe());
    array_push($aux,$arr[$i]->getPersonacomunicaqueja());
    array_push($aux,$arr[$i]->getCargopersonacomunica());
    array_push($aux,$arr[$i]->getInstitucion());
    array_push($aux,$arr[$i]->getDescripcionproblema());
    array_push($aux,$arr[$i]->getViacontacto());
    array_push($aux,$arr[$i]->getDescripcionproblema());
    array_push($aux,$arr[$i]->getCargorealizador());
    array_push($aux,$arr[$i]->getCargorevisor());
    array_push($aux,$arr[$i]-
>getControlplanillaidcontrolplanilla());
    array_push($aux1,$aux);
}
    return $aux1;

}

public static function BuscarCodigos()
{
    $aux=array('Select...');
    $arr=NotificacionquejasPeer::doSelect(new Criteria);
    for($i=0;$i<count($arr);$i++)
    {
        $aux[]=$arr[$i]->getCodigoprop();
    }
    return $aux;
}

public static function BuscarNoProducto()
{
    $arre=array();
    $arrel=array('Select...');
    $c = new Criteria();
    $re=NotificacionquejasPeer::doSelect($c);

    for($i=0;$i<count($re);$i++)
    array_push($arre,$re[$i]->getNombreproducto());

    array_push($arrel,$arre[0]);

    for($i=1;$i<count($arre);$i++)
    {$aux=true;
        for($j=0;$j<count($arrel);$j++)
        {
            $aux=true;
            if($arre[$i]==$arrel[$j])
```

```
break;
    else
    $aux=false;
    }
    if($aux==false)
    array_push($arrel,$arre[$i]);

}
return $arrel;
}
public static function BuscarNoLote()
{
    $arre=array();
    $arrel=array('Select...');
    $c = new Criteria();
    $re=NotificacionquejasPeer::doSelect($c);

    for($i=0;$i<count($re);$i++)
    array_push($arre,$re[$i]->getLotesinvolucrados());

    array_push($arrel,$arre[0]);

    for($i=1;$i<count($arre);$i++)
    {$aux=true;
        for($j=0;$j<count($arrel);$j++)
        {
            $aux=true;
            if($arre[$i]==$arrel[$j])
            break;
            else
            $aux=false;
            }
            if($aux==false)
            array_push($arrel,$arre[$i]);
        }
    }
    return $arrel;
}
public static function BuscarNombre($foli)
{
    $arr= NotificacionquejasPeer::retrieveByPK($foli);

    return $arr->getNombreproducto();
}
}
```

ControlPlanillaPeer

```
class ControlplanillaPeer extends BaseControlplanillaPeer
{
    public static function BuscarPlanilla($id)
    {
        return ControlplanillaPeer::retrieveByPK($id);
    }
    public static function ConvArrayObj($arr_obj)
    {
        $aux= array();

        array_push($aux,$arr_obj->getAprobador());
        array_push($aux,$arr_obj->getFechaaprobacion());
        array_push($aux,$arr_obj->getFecharealizacion());

        array_push($aux,$arr_obj->getRealizador());
        array_push($aux,$arr_obj->getRevisador());
        array_push($aux,$arr_obj->getRevisador());
        array_push($aux,$arr_obj->getFecharevision());
        array_push($aux,$arr_obj->getIdcontrolplanilla());
        return $aux;
    }
    public static function
ControlPlanilla($id,$apro_por,$reali_por,$f_real,$f_apro,$revi_por,$f_revi)
    {
        if($id=='')
            $sic= new Controlplanilla();
        else
            $sic=ControlplanillaPeer::retrieveByPK($id);
            $fec2=str_replace('-', '', $f_real);
            $sic->setFecharealizacion($fec2);
            $fec1=str_replace('-', '', $f_apro);
            $sic->setFechaaprobacion($fec1);
            $sic->setRealizador($reali_por);
            $sic->setAprobador($apro_por);
            $sic->setRevisador($revi_por);
            $fec=str_replace('-', '', $f_revi);
            $sic->setFecharevision($fec);
            $sic->save();

            return $sic->getIdcontrolplanilla();
    }
}
```

Como se pudo apreciar en lo anteriormente expuesto, la mayor parte de los nombres de las variables y los nombres de las funciones, así como el orden de lo programado, coincide en gran medida con el flujo indicado en los diagramas de secuencia. Lo cual demuestra que los programadores han hecho uso de este diseño, logrando así mayor organización y uniformidad en el código programado.

2.5 Conclusiones.

Se realizaron 11 diagramas de clases del diseño y 47 diagramas de secuencia.

El uso del patrón Modelo Vista Controlador implica una clara separación entre interfaz, lógica de negocio y de presentación, con lo que se logra sencillez para crear distintas representaciones de los mismos datos y simplicidad en el mantenimiento de los sistemas.

La validación del diseño a través del código programado tomando este diseño como guía, constituye una vía efectiva para probar la factibilidad del mismo.

Conclusiones

- A partir de los requerimientos identificados anteriormente, se diseñaron todas las clases, y se realizaron todos los diagramas de clases y diagramas de interacción, dejando este módulo listo para la implementación.
- El uso del framework Symfony aunque dificulta un poco la representación del diseño, optimiza el desarrollo de las aplicaciones Web.
- El uso de RUP como metodología de desarrollo es de gran utilidad para lograr una correcta estructura del trabajo de un proyecto en sus diferentes etapas.
- La etapa de diseño de una aplicación es de gran importancia, ya que es donde se modela el sistema, por lo que se debe proceder con precisión y organización para que soporte todos los requisitos.

Recomendaciones

Luego de haber concluido el presente trabajo se recomienda:

- Continuar con el uso de RUP como metodología de desarrollo de software en posteriores etapas del desarrollo de la aplicación.
- Profundizar en el aprendizaje del uso del framework Symfony con el objetivo de lograr una mayor calidad en el desarrollo de la aplicación.
- Mantener una activa vinculación con los directivos de Calidad del CIGB para conservar una correcta actualización de los requerimientos debido al continuo proceso de cambios que se produce en el centro.

Referencias Bibliográficas

1. Departamento de Control de la Calidad. *cigb.edu*. [En línea] 2003. [Citado el: 12 de Octubre de 2007.] <http://www.cigb.edu.cu/pages/ccalidad.htm>.
2. Departamento de Aseguramiento de la Calidad. *cigb.edu*. [En línea] 2003. [Citado el: 12 de Octubre de 2007.] <http://www.cigb.edu.cu/pages/acalidad.htm>.
3. Información para la gestión y gestión de la información. *scn.org*. [En línea] [Citado el: 23 de Noviembre de 2007.] <http://www.scn.org/mpfc/modules/mon-miss.htm>.
4. **Contreras Díaz, Yimian de Lyz y Rivero Amador, Soleydi**. DISEÑO DEL SISTEMA DE GESTIÓN DE INFORMACIÓN DEL CENTRO DE ESTUDIOS DE MEDIO AMBIENTE Y RECURSOS NATURALES (CEMARNA) DE LA UNIVERSIDAD DE PINAR DEL RÍO. *gestiopolis.com*. [En línea] [Citado el: 25 de Noviembre de 2007.] <http://www.gestiopolis.com/administracion-estrategia/sistemas-de-gestion-de-informacion-en-estudio-de-medio-ambiente.htm>.
5. *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones*. **Aja Quiroga., Lourdes**. Ciudad de La Habana : s.n., 10 de Mayo de 2002. [En línea] [Citado el: 25 de Noviembre de 2007.] http://bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm.
6. **Cuevas Melis, Olga**. Onl@bweb. *onlabweb*. [En línea] 6 de Diciembre de 2007. [Citado el: 15 de Junio de 2008.] <http://www.onlabweb.com/lims.htm>.
7. **Olguín Espinoza, José Martín**. Curso de Análisis Orientado a Objetos. *yaqui.mxl.uabc.mx*. [En línea] 21 de Junio de 2004. [Citado el: 10 de Diciembre de 2007.] <http://yaqui.mxl.uabc.mx/~molguin/as/RUP.htm>.
8. **Fernandez Vilas, Ana Fernandez Vilas**. Diagrama de Despliegue. *gris.det.uvigo*. [En línea] [Citado el: 15 de Diciembre de 2007.] <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>.
9. Free Download Manager. *freedownloadmanager.org*. [En línea] [Citado el: 20 de Diciembre de 2007.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(Iglesia_Anglicana\)_%5BMac_OS_X_cuenta_14717_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5BMac_OS_X_cuenta_14717_p/).
10. Ciberaula. *ciberaula*. [En línea] 2006. [Citado el: 15 de Junio de 2008.] http://php.ciberaula.com/articulo/introduccion_php/.
11. AGAPEA. *agapea*. [En línea] [Citado el: 15 de Junio de 2008.] <http://www.agapea.com/PHP-5-n154892i.htm>.
12. **Alvarez, Miguel Angel**. *desarrolloweb.com*. *desarrolloweb*. [En línea] [Citado el: 15 de Junio de 2008.] <http://www.desarrolloweb.com/articulos/1696.php>.

13. librosweb.es. *librosweb.es*. [En línea] [Citado el: 10 de Enero de 2008.]
http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html.
14. **Manuel Dodero, Juan y Fernández Llamas, Camino**. Patrones estructurales: Decorator. *dei.inf.uc3m*. [En línea] [Citado el: 18 de Enero de 2008.]
http://www.dei.inf.uc3m.es/docencia/p_s_ciclo/tdp/curso0203/apuntes/decorator.pdf.
15. **Eugenia Valencia, María**. Patrones para asignar responsabilidades. *eisc.univalle.edu*. [En línea] [Citado el: 25 de Enero de 2008.]
http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/patrones.pdf.
16. **Saavedra Gutierrez, Jorge A**. El Mundo Informático. *jorgesaavedra.wordpress*. [En línea] [Citado el: 25 de Enero de 2008.] <http://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman/>.
17. Symfony, la guía definitiva. *librosweb*. [En línea] [Citado el: 8 de Febrero de 2008.]
http://www.librosweb.es/symfony/capitulo2/el_patron_mvc.html.
18. **Lasso, Adrian**. Arquitectura de Software. *scribd*. [En línea] 2008. [Citado el: 15 de Marzo de 2008.]
<http://www.scribd.com/doc/210452/Arquitectura-de-Software-Adrian-Lasso>.

Bibliografía

1. Ingeniería del Software 2. *teleformacion.uci.cu* [En línea] 2004. [Citado el: 19 de Octubre de 2007.] <http://teleformacion.uci.cu/mod/resource/view.php?id=21363>.
2. Ingeniería del Software 2. *teleformacion.uci*. [En línea] 2004. [Citado el: 19 de Octubre de 2007.] <http://teleformacion.uci.cu/mod/resource/view.php?id=21514>.
3. Ayuda extendida del Rational Rose Enterprise Edition 2003.
4. **Sarmiento Almenares, Alieski y Cutiño Díaz, Elian.** *LIMS DE CALIDAD DEL CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA: ANÁLISIS DEL GRUPO DE RECEPCIÓN DE MUESTRAS Y MANIPULACIÓN DE EXPEDIENTES.* . Ciudad de La Habana : s.n., 2006.
5. **Díaz Laurencio, Elennis y Nieto Cervantes, Liusmila.** *LIMS DE CALIDAD DEL CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA: ANÁLISIS DE LA SECCIÓN DE MEJORAMIENTO DE LA CALIDAD Y DEL GRUPO DE DESARROLLO.* Ciudad de la Habana : s.n., 2007.
6. **Robles Aranda, Yadira y Sardiñas Ramírez, Ana Margarita.** *LIMS DE CALIDAD DEL CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA: ANÁLISIS DE LOS GRUPOS DE DOCUMENTACIÓN Y METROLOGÍA.* Ciudad de la Habana : s.n., 2007.
7. **Potencier, Fabien y Zaninotto, François.** *Symfony, la guía definitiva.*
8. Free Download Manager . *freedownloadmanager.org*. [En línea] [Citado el: 20 de Diciembre de 2007.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).
9. **Saavedra Gutierrez, Jorge A.** El Mundo Informático. *jorgesaavedra.wordpress*. [En línea] [Citado el: 25 de Enero de 2008.] <http://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman/>.
10. **BIOTECNOLOGÍA, CENTRO DE INGENIERÍA GENÉTICA Y. DIRECCIÓN DE CALIDAD.** *cigb.edu*. [En línea] 2003. [Citado el: 26 de Febrero de 2008.] <http://www.cigb.edu.cu/pages/calidad.htm>.
11. **BIOTECNOLOGÍA, CENTRO DE INGENIERÍA GENÉTICA Y.** Departamento de Control de la Calidad . *cigb.edu*. [En línea] 2003. [Citado el: 27 de Febrero de 2008.] <http://www.cigb.edu.cu/pages/ccalidad.htm>.
12. **León Pavón, Eduardo.** Blog de Eduardo León . *slion2000.blogspot*. [En línea] Noviembre de 2005. [Citado el: 28 de Marzo de 2008.] <http://slion2000.blogspot.com/2007/04/visual-paradigm-una-herramienta-de-lo.html>.
13. **Lasso, Adrián.** Scribd. *scribd*. [En línea] [Citado el: 28 de Mayo de 2008.] <http://www.scribd.com/doc/210452/Arquitectura-de-Software-Adrian-Lasso>.

14. **Franco Navarro, Jose Angel.** *UML en acción. Modelando Aplicaciones Web.*

Anexos

Anexo#1

Centro de Ingeniería Genética y Biotecnología

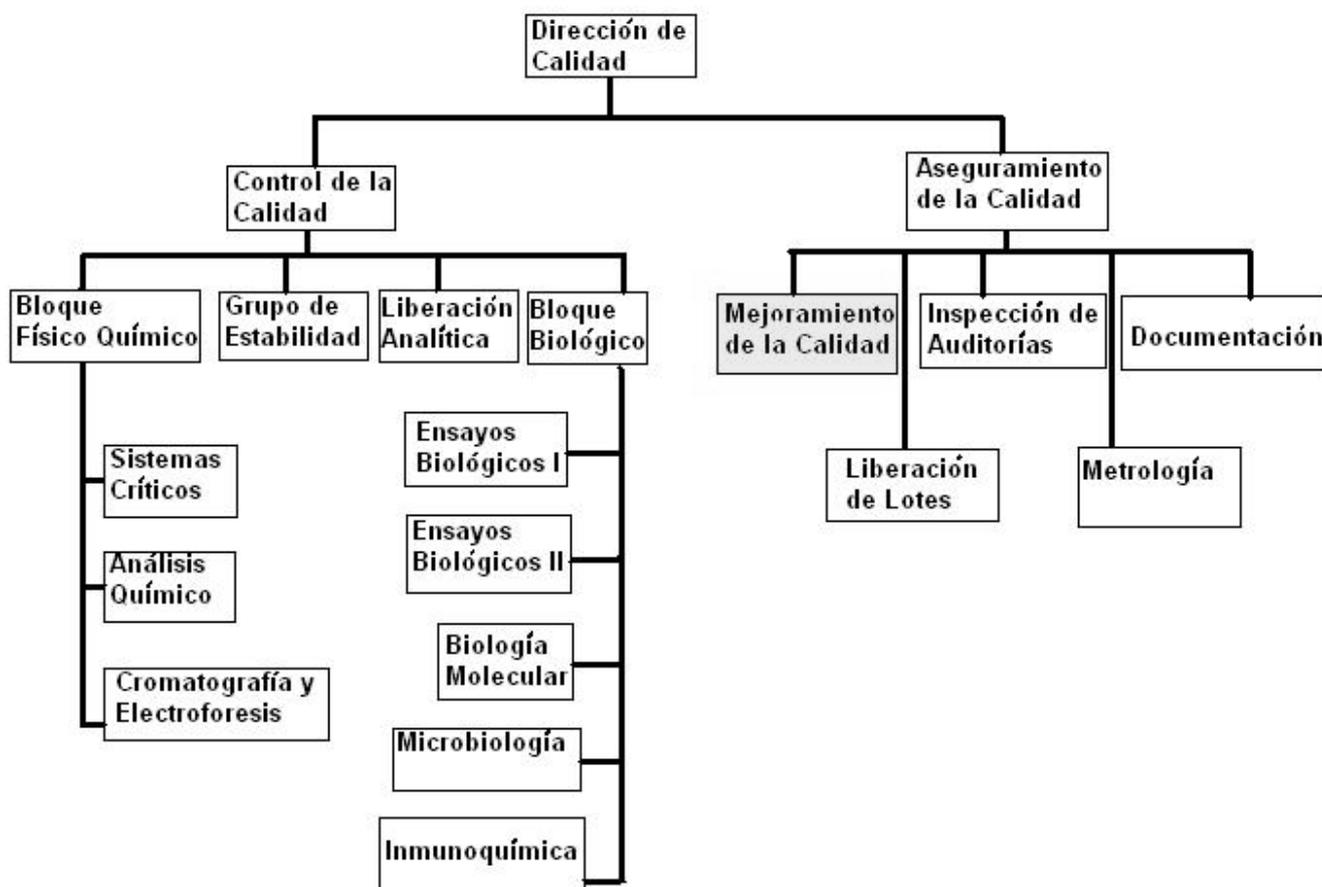


Fig. 54 Representación de la Estructura jerárquica de la Dirección de Calidad del CIGB.

Anexo# 2

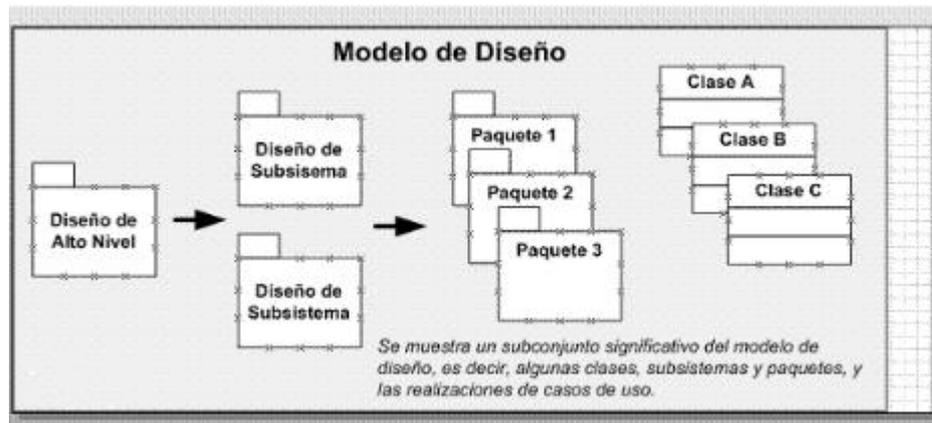


Fig. 55 Representación de la Vista Lógica.

Anexo# 3 Prototipos de interfaz de usuario de los casos de uso Gestionar Programa de Monitoreo Ambiental y Gestionar Programa de Informe de Tendencias

CU Gestionar Programa de Monitoreo Ambiental (SIC-0098).

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA				SIC-0098	PPO 4.10.120.96		
DIRECCIÓN DE CALIDAD				Edición 04	Pág 1 de 1		
PROGRAMA DE MONITOREO AMBIENTAL				Modificaciones 03	Folio:		
Producción:	Área:			NP:			
Local:	Clase:	Zona de trabajo:					
Característica	Método	Límite Alerta	Límite Acción	Frecuencia	Instrumento/PPO	Punto Muestreo (Referencia)	Responsable
Insertar							
Insertar							
	Nombre		Cargo		Firma	Fecha	
Realizado por:							
Revisado por:							
Aprobado por:							

Terminado

Fig. 56 Prototipo de Crear Programa de Monitoreo Ambiental

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA DIRECCIÓN DE CALIDAD Búsqueda del Programa de Monitoreo Ambiental			
Folio:	—Seleccione— ▼	Planta:	—Seleccione— ▼
Zona de Trabajo:		Área:	NP:
<input type="button" value="Buscar"/>			<input type="button" value="Cancelar"/>
Producción	Área	Método	Características
			Visualizar

Fig. 57 Prototipo de Buscar Programa de Monitoreo Ambiental

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA				SIC-0098		PPO 4.10.120.96	
DIRECCIÓN DE CALIDAD				Edición 04		-	
PROGRAMA DE MONITOREO AMBIENTAL				Modificaciones 03		Folio: -----	
Producción: -----		Área: -----				NP: -----	
Local: -----		Clase: -----		Zona de trabajo: -----			
Característica	Método	Límite Alerta	Límite Acción	Frecuencia	Instrumento/PPO	Punto Muestreo (Referencia)	Responsable
-----	-----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----	-----

Modificar Imprimir

Fig. 58 Prototipo de Visualizar Programa de Monitoreo Ambiental

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA				SIC-0098		PPO 4.10.120.96	
DIRECCIÓN DE CALIDAD				Edición 04		Pág 1 de 1	
PROGRAMA DE MONITOREO AMBIENTAL				Modificaciones 03		Folio: _____	
Producción: _____		Área: _____				NP: _____	
Local: _____		Clase: _____		Zona de trabajo: _____			
Característica	Método	Límite Alerta	Límite Acción	Frecuencia	Instrumento/PPO	Punto Muestreo (Referencia)	Responsable
_____	_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____	_____
							Insertar
_____	_____	_____	_____	_____	_____	_____	_____
							Insertar
	Nombre		Cargo		Firma	Fecha	
Realizado por:	_____		_____		_____	_____	
Revisado por:	_____		_____		_____	_____	
Aprobado por:	_____		_____		_____	_____	

Terminado

Cancelar Modificar

Fig. 59 Prototipo de Modificar Programa de Monitoreo Ambiental

CU Gestionar Programa de Informes de Tendencias (SIC-0919).

CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA DIRECCION DE CALIDAD	SIC-0919	PPO 4.10.056.93
	Edición 01	Folio: <input type="text"/>
PROGRAMA DE INFORMES DE TENDENCIAS	Año: <input type="text"/>	Pág 1 de 1
	Semestre: <input type="text"/>	

Crear Cancelar

Fig. 60 Prototipo de Crear Programa de Informes de Tendencias

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA DIRECCIÓN DE CALIDAD Búsqueda del Programa de Informes de Tendencias			
Folio: <input type="text" value="—Seleccione—"/>	Mes: <input type="text" value="—Seleccione—"/>	Año: <input type="text"/>	<input type="text"/>

Buscar Cancelar

Año	Folio	Semestre	Visualizar
<input type="text"/>	<input type="text"/>	<input type="text"/>	

Fig. 61 Prototipo de Buscar Programa de Informes de Tendencias

CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA DIRECCION DE CALIDAD		SIC-0919	PPO 4.10.056.93
		Edición 01	Folio: ----
PROGRAMA DE INFORMES DE TENDENCIAS		Año: ---	Pág 1 de 1
		Semestre: ---	
Desviaciones del Programa			
Mes	Desviación	Firma del Jefe de Grupo	
----	-----	--	
----	-----	--	
----	-----	--	

Fig. 62 Prototipo de Visualizar Programa de Informes de Tendencias

CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA DIRECCION DE CALIDAD		SIC-0919	PPO 4.10.056.93
		Edición 01	Folio: <input type="text" value=""/>
PROGRAMA DE INFORMES DE TENDENCIAS		Año: <input type="text" value=""/>	Pág 1 de 1
		Semestre: <input type="text" value=""/>	
Desviaciones del Programa			
Mes	Desviación	Firma del Jefe de Grupo	
<input type="text" value=""/>	<input type="text" value=""/>		
<input type="text" value=""/>	<input type="text" value=""/>		
<input type="text" value=""/>	<input type="text" value=""/>		

Fig. 63 Prototipo de Modificar Programa de Informes de Tendencias

Registrar Programa de Informes de Tendencias

Folio: Año:

Desviaciones del Programa		
Mes	Desviación	Firma del Jefe de Grupo
<input type="text"/>	<input type="text"/>	<input type="text"/>

[Insertar](#)

Fig. 64 Prototipo de Registrar Programa de Informes de Tendencias

<p>CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA</p> <p>DIRECCIÓN DE CALIDAD</p> <p>Programa de Informes de Tendencias</p>	
<p>Generar reporte del Programa de Informes de Tendencias</p>	
<p>Parámetros para la generación del reporte</p>	
<p>Año: <input type="text" value="-Seleccione-"/></p>	<p>Mes: <input type="text"/></p>
<p>Parámetros a mostrar en el reporte del Programa de Informes de Tendencias.</p>	
<p><input type="checkbox"/> Todos</p> <p><input type="checkbox"/> Desviación.</p>	<p><input type="checkbox"/> Mes.</p> <p><input type="checkbox"/> Año.</p>

Fig. 65 Prototipo de Generar Reporte del Programa de Informes de Tendencias

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA DIRECCIÓN DE CALIDAD Programa de Informes de Tendencias				
Reporte del Programa de Informes de Tendencias				
-----	-----	-----	-----	-----
-----	-----	-----	-----	-----

Fig. 66 Prototipo de Visualizar Reporte del Programa de Informes de Tendencias

Anexo# 4

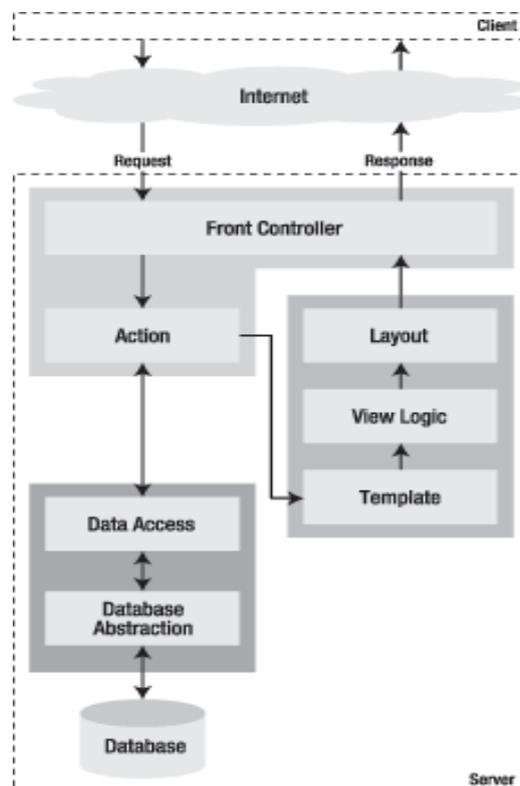


Fig. 67 El flujo de trabajo de Symfony.

Glosario de Términos

Artefacto

Pieza de información utilizada o producida por un proceso de desarrollo de software, como un documento externo o el producto de un trabajo. Un artefacto puede ser un modelo, una descripción o un software.

Aseguramiento de la Calidad

Conjunto de acciones planificadas y sistemáticas que son necesarias para proporcionar la confianza de que un producto o servicio satisface los requisitos de calidad preestablecidos.

Calidad

Grado en que un conjunto de características inherentes cumplen con los requisitos.

Control de la Calidad

Técnicas y actividades de carácter operativo utilizadas para satisfacer los requisitos de calidad.

Framework

Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

Fuentes factográficas

Conjunto de datos.

HTTPS

Protocolo seguro de transferencia de hipertexto.

Interfaz de usuario

Una colección de operaciones que son utilizadas para especificar un servicio de una clase o de un componente.

Lote

Cantidad definida de materia prima, material de envase o producto terminado, elaborado en un solo proceso o una serie de procesos de tal manera que puede esperarse que sea homogéneo. En el caso de un proceso continuo de fabricación, el lote debe corresponder a una fracción definida de la producción, caracterizada por la homogeneidad que se busca en el producto.

Metodología

Un sistema de principios y normas generales de organización y estructuración teórico práctica de actividades.

Número de lote

Combinación característica de números y/o letras que identifica específicamente a un lote.

ORM

Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos. En la práctica esto crea una base de datos orientada a objetos virtual.

SIC

Sistema de Información y Control.

SSL

Protocolo de Compras Seguras o Nivel de Sockets Seguros.

TCP/IP

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

URL

Localizador uniforme de recurso.

USB

El Universal Serial Bus (bus universal en serie) es un puerto que sirve para conectar periféricos a una computadora.