

# **Universidad de las Ciencias Informáticas**

## **Facultad 6**



### **TÍTULO: LIMS DE CALIDAD DEL CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA: DISEÑO DEL GRUPO DE LIBERACIÓN ANALÍTICA.**

Trabajo de Diploma para optar por el título de Ingeniero Informático

**Autoras:** Karina Mazaira Behar

Yadira Cruz Novas

**Tutor:** Ing. Niusvel Acosta Mendoza

**Co-tutora:** Lic. Miulkenia Navarro Reyes

**Junio, 2008**

**“Año 50 de la Revolución”**

## DECLARACIÓN DE AUTORÍA

Declaramos ser autoras del presente Trabajo de Diploma y autorizamos a la Universidad de las Ciencias Informáticas y al Grupo de Liberación Analítica del área de Calidad del Centro de Ingeniería Genética y Biotecnología a hacer uso del mismo en su beneficio.

Para que así conste se firma la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Yadira Cruz Novas

\_\_\_\_\_

Firma de la autora

Karina M. Mazaira Behar

\_\_\_\_\_

Firma de la autora

Niusvel Acosta Mendoza

\_\_\_\_\_

Firma del tutor

Miulkenia Navarro Reyes

\_\_\_\_\_

Firma de la Co-tutora

## **DATOS DE CONTACTO**

### **Tutor:**

Ing. Niusvel Acosta Mendoza

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: nacosta@uci.cu

### **Co-Tutora:**

Lic. Miulkenia Navarro Reyes

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: mnavarror@uci.

## **DEDICATORIA**

*A mi mamá Tania, mis abuelos Amparo y Jacobo y a mi padrastro Ariel, por su ejemplo y guía, su cariño e incondicionalidad.*

*A mi futuro esposo William por su amor y comprensión, gracias por regalarme los momentos más bellos de mi vida.*

*A las “santis” Diana, Emily, Yadira y Yanet por apoyarme y estar presente en todo momento, por ser las hermanas que nunca tuve.*

*A Marcia y Franger por todo su amor y cariño, son como una familia para mí.*

*A mis compañeros de grupo y apartamento por su amistad y solidaridad.*

*A mi gran amigo Isaac por su cariño, apoyo y confiar siempre en mí.*

*Karina*

*A quienes debo lo que soy, mis padres Antonia y Jorge, por su confianza y cariño, y estar en los momentos más importantes de mi vida.*

*A mis familiares, que han estado cuando más los he necesitado.*

*A las “santis”, a las que forman y han formado parte de este grupo tan especial de amigas que ha marcado mi vida, y estarán en mi corazón por siempre: Yanet, Martica, Karina, Emily y Diana.*

*A mi novio Abel, por su comprensión y amor, gracias por hacerme tan feliz.*

*A mi suegra Alina por ser tan especial conmigo.*

*A todas mis amistades de la universidad, por compartir estos cinco años, llenos de momentos inolvidables, en especial a Adriel y René.*

*Yadira*

## **AGRADECIMIENTOS**

*A la Revolución Cubana y a nuestro Comandante en Jefe Fidel Castro por brindarnos la oportunidad de formarnos en una universidad de excelencia como profesionales dignos de esta patria.*

*A los compañeros del CIGB por su paciencia y atención, especialmente Gerardo, Indira y María de los Ángeles.*

*A nuestros compañeros de proyecto, por el apoyo y la solidaridad brindados.*

*A nuestro tutor Niusvel Acosta Mendoza por su ayuda y atención.*

*A Jose Ángel Franco Navarro por su ayuda desinteresada y su preocupación.*

*A nuestra familia por su cariño y apoyo incondicional.*

## **RESUMEN**

En la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología (CIGB) se manipula un elevado volumen de información referente a las muestras que allí se analizan, para llevar a cabo el control de la calidad de las mismas, y que cuando se comercialicen cumplan con las normas de calidad requeridas. Con el objetivo de mejorar y controlar la gestión de la información en dicha dirección, se comenzó a desarrollar desde años anteriores el Sistema para el Manejo de Información de Laboratorios (LIMS).

Uno de los grupos que conforma la Dirección de Calidad del CIGB es el Grupo de Liberación Analítica, la automatización de los procesos que en él se realizan constituye un módulo del LIMS que se está desarrollando. En investigaciones anteriores se realizó el análisis y el diseño de dicho módulo pero el avance del proyecto y la migración del mismo hacia nuevas tecnologías, como la utilización del framework Symfony, propiciaron la elaboración de éste trabajo donde se traducen los requerimientos anteriormente identificados, en elementos que puedan ser implementados. Por tal situación la investigación está orientada al estudio del proceso de desarrollo de los sistemas de gestión de la información.

**Palabras claves:** LIMS, gestión de la información.

## TABLA DE CONTENIDOS

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	4
Introducción.....	4
1.1 Gestión de la Información .....	4
1.1.1 Sistemas de Gestión de la Información.....	5
1.1.2 La Web como sistema de información .....	5
1.1.3 Sistemas de Gestión de la Información de Laboratorios.....	6
1.2 Metodología y herramientas seleccionadas .....	7
1.2.1 Proceso de desarrollo de Software a utilizar .....	7
1.2.2 Lenguaje de Modelado a utilizar.....	8
1.2.3 Herramienta CASE a utilizar: Visual Paradigm .....	8
1.3 Framework a utilizar: Symfony .....	9
1.3.1 Arquitectura: Symfony .....	10
1.4 Diseño .....	12
1.4.1 Roles y artefactos.....	12
1.5 Patrones de Diseño.....	14
1.5.1 Objetivos de los patrones de diseño .....	14
1.5.2 Patrones de diseño a utilizar .....	15
Conclusiones.....	16
CAPÍTULO 2: DISEÑO .....	17
Introducción.....	17
2.1 Breve referencia a la arquitectura del sistema .....	17
2.1.1 Vista Lógica.....	17
2.1.2 Vista de Despliegue .....	22
2.2 Prototipos no funcionales de interfaz de usuario .....	23
2.4 Diagramas de clases del diseño .....	29
2.5 Diagramas de secuencia.....	55
2.6 Evaluación de la calidad del diseño .....	81
Conclusiones.....	82
CONCLUSIONES.....	83

RECOMENDACIONES .....	84
REFERENCIAS BIBLIOGRÁFICAS .....	85
BIBLIOGRAFÍA .....	86
ANEXOS .....	87
GLOSARIO.....	94

## INTRODUCCIÓN

En la actualidad el uso de las Tecnologías de la Información y de las Comunicaciones (TICs) está contribuyendo al cambio de cultura corporativa e institucional, dando lugar a una nueva vía para fortalecer la economía.

En Cuba se está llevando a cabo un proceso de informatización, en el cual las TICs juegan un papel fundamental para el desarrollo de la economía y la sociedad y brindan una amplia gama de servicios, aplicaciones y tecnologías que utilizan diversos tipos de equipos y programas informáticos.

Como parte del proceso de informatización que se está desarrollando en nuestro país, se ha llevado a cabo un intenso proceso inversionista y de formación de personal, que permite disponer en la actualidad de un complejo e integrado sistema de investigación-producción en la esfera de la Biotecnología aplicada a diferentes ramas de la sociedad. El Centro de Ingeniería Genética y Biotecnología (CIGB) de Ciudad de La Habana es un ejemplo de este proceso, es uno de los centros investigativos que se ha destacado por los resultados obtenidos no sólo en cuanto a la investigación, sino también en el desarrollo, producción y comercialización de productos biológicos obtenidos a través de los métodos de la biotecnología moderna. [1]

Paralelo al proceso de desarrollo de los productos biológicos en el CIGB, se lleva a cabo un control de la calidad de los mismos, para que cuando estos se comercialicen cumplan con las normas de calidad requeridas. De este control se encarga la **Dirección de Calidad** del centro, que se divide en dos departamentos: **Control de la Calidad y Aseguramiento de la Calidad** (Ver Anexo 1).

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 los 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]

- Grupo de Documentación
- Grupo de Metrología
- Sección de Mejoramiento e Ingeniería de la Calidad
- Sección de Inspección, Auditoría y Liberación de lotes

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 el Centro. Para el desempeño de las mismas, cuenta con la ayuda de dos grupos de trabajo y dos secciones: [3]

- Grupo de Desarrollo
- Liberación analítica
- Sección biológica
- Sección físico-química

El grupo de Liberación Analítica tiene la función principal de recibir y distribuir las muestras que llegan al departamento para ser analizadas en los diferentes laboratorios y a su vez recibir los resultados emitidos por los laboratorios y confeccionar el expediente analítico de cada lote de producto. Además son responsables de la recepción, conservación, control y destrucción de las muestras testigos de los diferentes productos terminados. [4]

El grupo de Liberación Analítica manipula y genera grandes volúmenes de información referente a la recepción y distribución de las muestras en los laboratorios que realizarán los procesos de Control de la Calidad, así como la recogida de los resultados emitidos por los laboratorios. El almacenamiento de los datos es en formato duro, por ejemplo: libros de reportes, planillas y expedientes para lotes de productos; lo que significa un cúmulo de información considerable, y dificulta la búsqueda y actualización de la misma, por lo que se comenzó el proceso de informatización de la gestión de la información de dicho grupo, constituyendo éste un módulo del Sistema de Gestión de la Información del Laboratorio (LIMS por sus siglas en inglés) que se está desarrollando para la Dirección de Calidad del CIGB.

En tesis precedentes se trabajó en los flujos de trabajo de Modelamiento del Negocio, Requerimientos y Análisis y Diseño de este módulo, pero para el desarrollo del diseño no se utilizó el symfony como framework. Por todo lo antes planteado se define como **problema científico**: ¿Cómo traducir los requerimientos identificados para el módulo de Liberación Analítica del LIMS de Calidad del CIGB en elementos que puedan ser implementados utilizando symfony como framework?

El **objeto de estudio** identificado para el problema científico planteado es: El proceso de desarrollo de los sistemas de gestión de la información.

Del objeto de estudio analizado, se define el **campo de acción** como: El proceso de diseño de los sistemas de gestión de la información.

Para lograr la solución del problema identificado se plantea el siguiente **objetivo**: Diseñar el módulo de Liberación Analítica para el sistema de gestión de la información de los laboratorios de la Dirección de Calidad del CIGB.

Se desarrollarán las siguientes **Tareas** para dar cumplimiento al objetivo trazado:

- Estudio del Modelamiento del Negocio
- Estudio del Modelo del Sistema
- Obtención de las clases del diseño
- Realización de los casos de uso del diseño: diagramas de clases y diagramas de interacción
- Obtención de los prototipos no funcionales
- Realización del mapa de navegación

La estructura del trabajo es la siguiente: introducción, dos capítulos, conclusiones, recomendaciones, bibliografías y anexos.

Capítulo 1. Fundamentación Teórica: se hace un estudio sobre los Sistemas de Gestión de la Información, abordando algunos aspectos de los Sistemas de Gestión de la Información de los Laboratorios y la Web como Sistema de Información. Se definen los roles a desempeñar, los artefactos a obtener en el flujo de trabajo de diseño, la metodología y las distintas herramientas a utilizar, algunos aspectos del framework seleccionado y los patrones de diseño aplicados para el desarrollo de este trabajo.

Capítulo 2. Diseño del Sistema: se hace una breve referencia a la arquitectura del sistema a través de la vista lógica y la vista de despliegue, se obtienen los artefactos correspondientes al flujo de trabajo de diseño, además de la evaluación de la calidad de las clases diseñadas.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### Introducción

En el presente capítulo se estudian los Sistemas de Gestión de la Información, específicamente algunos aspectos significativos de la web como Sistema de Información y los Sistemas de Gestión de la Información de los Laboratorios. Se explican los artefactos y herramientas enmarcados en los roles de diseñador, arquitecto y diseñador de interfaz de usuario definidos por el Proceso Unificado de Desarrollo, la metodología y herramientas que se van a utilizar en el desarrollo del LIMS para la Dirección de Calidad del CIGB y algunos aspectos significativos del framework seleccionado y los patrones de diseño que se utilizan en el trabajo.

### 1.1 Gestión de la Información

El siglo XXI ha sido denominado como el momento de la Tecnología en la Información, alcanzando niveles de desarrollo impresionantes. Se habla constantemente sobre la sociedad de la información, esta nueva sociedad, con organizaciones basadas en el aprendizaje, cuyo capital máspreciado es el ser humano, se sustenta en un desarrollo tecnológico sin precedentes.

La gestión de información es el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, así como para mejorar los procesos, productos y servicios de la organización.

Los beneficios que aporta una buena gestión de información empresarial y que garantizan el cambio son: [5]

- Mayor control de la información
- Rápida localización de la documentación requerida
- Homogenización de la documentación corporativa
- Acceso a los productos y servicios de información de valor agregado que puedan apoyar la toma de decisiones
- Acceso a la información generada por cada una de las unidades organizativas de la empresa
- Control de versiones de los archivos generados dentro de la organización
- Ahorro de los recursos utilizados para el almacenaje de archivos electrónicos
- Soluciona los problemas de localización de las fuentes internas y externas

- Conocimiento por parte de todos los usuarios de todos los recursos disponibles
- Permite la consulta simultánea e inmediata por miembros de la organización
- Mejora la eficiencia en los procesos de la organización
- Aumento de la eficiencia y del nivel de excelencia operativa

Como resultado del crecimiento de la capacidad de generación y almacenamiento de la información, y la creciente dificultad de su análisis a través de métodos tradicionales, surgen nuevas técnicas y herramientas que favorecen y posibilitan un análisis más eficiente de estos volúmenes de datos y el descubrimiento de patrones de comportamiento que pueden elevar la eficiencia de la entidad en la toma de decisiones organizacionales.

### **1.1.1 Sistemas de Gestión de la Información**

Los Sistemas de Gestión de Información (SGI) para la gestión del conocimiento constituyen hoy una alternativa de imprescindible presencia en cada organización. Al permitir operar casi todos los activos tangibles e intangibles de la institución y llegar a convertirse en la herramienta integral de gerencia más cotizada y necesaria para alcanzar con éxito los resultados propuestos por la organización. [6]

Los SGI permiten:

- Comprender la marcha de las organizaciones desde un enfoque analítico, evaluador y creativo
- 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: [6]

- 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

### **1.1.2 La Web como sistema de información**

La evolución de Internet como red de comunicación global y el surgimiento y desarrollo de la Web como servicio imprescindible para compartir información, creó un excelente espacio para la interacción

del hombre con la información hipertextual, a la vez que sentó las bases para el desarrollo de una herramienta integradora de los servicios existentes en Internet.

Los sitios Web, como expresión de sistemas de información, deben poseer los siguientes componentes:

- Usuarios
- Mecanismos de entrada y salida de la información
- Almacenes de datos, información y conocimiento
- Mecanismos de recuperación de información

En la actualidad, los sistemas de información se encuentran al alcance de las grandes masas de usuarios por medio de Internet; así se crean las bases de un nuevo modelo, en el que los usuarios interactúan directamente con los sistemas de información para satisfacer sus necesidades de información.

### **1.1.3 Sistemas de Gestión de la Información de Laboratorios**

Con la finalidad de aplicar en el entorno del laboratorio los beneficios y mejoras de los SGI, surgen los Sistemas de Gestión de la Información de Laboratorios o LIMS.

Un LIMS 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 y trazabilidad y minimizando los errores debidos a la transferencia de información. [7]

Por los innumerables beneficios que tiene la utilización de un LIMS, entre los que están: mejoras en el procesamiento e integridad de la información, en la productividad y el aseguramiento de la calidad; numerosas empresas se han dedicado a la fabricación de los mismos. Una de las compañías más reconocidas en el desarrollo de los LIMS es STARLIMS. STARLIMS Technologies Ltd. (NASDAQ: LIMS) es un proveedor global de LIMS, con más de 20 años de experiencia. El software STARLIMS® consolida procesos de laboratorio en una única y compatible plataforma con informes comprensibles, y con capacidades de supervisión sobre la red de trabajo. El resultado principal es resaltar el manejo de datos y la facilidad de compartirlos dentro del laboratorio y a través de la empresa. El software de STARLIMS es diseñado para una amplia variedad de laboratorios. La compañía sirve sobre 500 clientes en 40 países, incluyendo organizaciones gubernamentales, industriales y científicas. [8]

## 1.2 Metodología y herramientas seleccionadas

El Sistema de Gestión de la Información del Laboratorio para el CIGB se está desarrollando hace dos años, por lo que la metodología y las herramientas a utilizar para su desarrollo están definidas desde entonces.

### 1.2.1 Proceso de desarrollo de Software a utilizar

El Proceso Unificado de desarrollo (RUP por sus siglas en inglés), junto con el Lenguaje Unificado de Modelado (UML por sus siglas en inglés), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Es aconsejable utilizar la metodología RUP para proyectos grandes, porque requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios.

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al culminar cada uno de ellos, éstos a la vez se dividen en fases que finalizan con un hito donde se debe tomar una decisión importante.

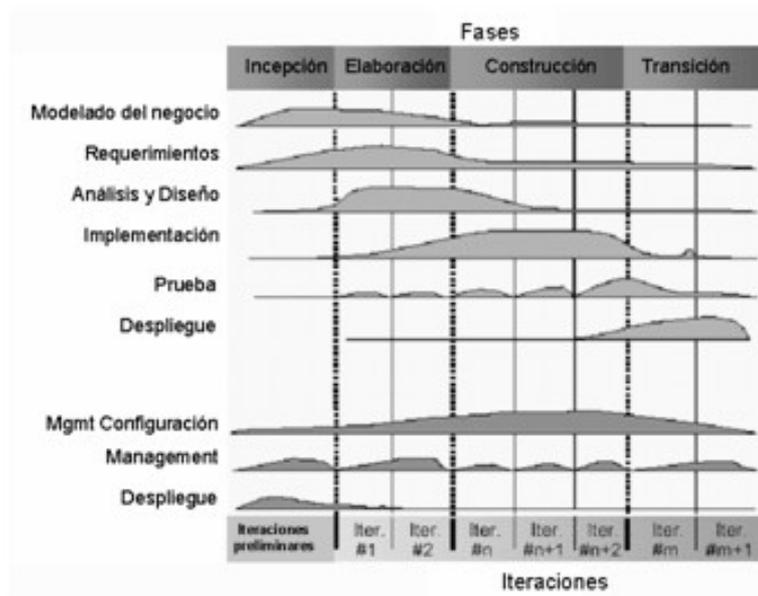


Fig. 1 Fases y flujos de trabajo en RUP

El ciclo de vida de RUP se caracteriza por: [9]

- *Dirigido por casos de uso:* Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los

requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo porque los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo)

- *Centrado en la arquitectura:* La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura
- *Iterativo e Incremental:* RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración

De acuerdo al proceso de desarrollo de software (RUP) en años anteriores se trabajó en los flujos de trabajo de Modelamiento del negocio, Requerimientos y Análisis y Diseño, sin tener en cuenta el uso de ningún framework; en este trabajo se va a realizar del flujo de trabajo de Análisis y Diseño la parte de Diseño, usando symfony como framework.

### **1.2.2 Lenguaje de Modelado a utilizar**

El Lenguaje Unificado de Modelado (UML), es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. [9]

Es un lenguaje para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como RUP), pero no especifica en sí mismo qué metodología o proceso usar.

### **1.2.3 Herramienta CASE a utilizar: Visual Paradigm (Versión 6.1)**

El Visual Paradigm para UML soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue, ayuda a una más rápida construcción de aplicaciones de calidad. Permite dibujar todos los tipos de diagramas de clases, código inverso,

generar código desde diagramas y generar documentación. La herramienta UML CASE proporciona además tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Visual Paradigm ofrece: [10]

- Entorno de creación de diagramas para UML 2.0
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación
- Capacidades de ingeniería directa e inversa
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo
- Disponibilidad de múltiples versiones, para cada necesidad
- Disponibilidad de integrarse en los principales IDEs
- Disponibilidad en múltiples plataformas

### 1.3 Framework a utilizar: **Symfony (Versión: 1.0.11)**

El Framework Symfony está diseñado para optimizar el desarrollo de las aplicaciones web, separa la lógica de negocio, la lógica de servidor y la presentación. Facilita el trabajo del desarrollador, automatizando las tareas más comunes y posee un conjunto de herramientas y clases con el objetivo de reducir el tiempo de desarrollo de un proyecto complejo.

Symfony está desarrollado completamente con PHP 5, 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.

Características: [11]

- Fácil de instalar y configurar en la mayoría de plataformas
- 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 adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo
- Código fácil de leer que incluye comentarios de PHP Documentor y que permite un mantenimiento muy sencillo
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros

### 1.3.1 Arquitectura: Symfony

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC (Modelo-Vista-Controlador), que está formado por 3 niveles:

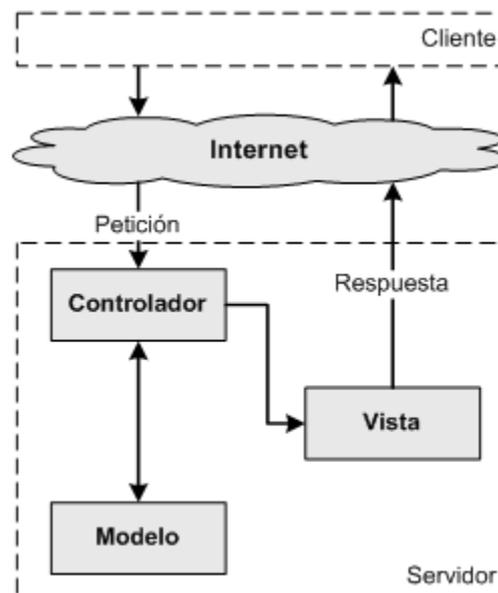


Fig.2 Funcionamiento del patrón MVC

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

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. 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. Symfony toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo. [11]

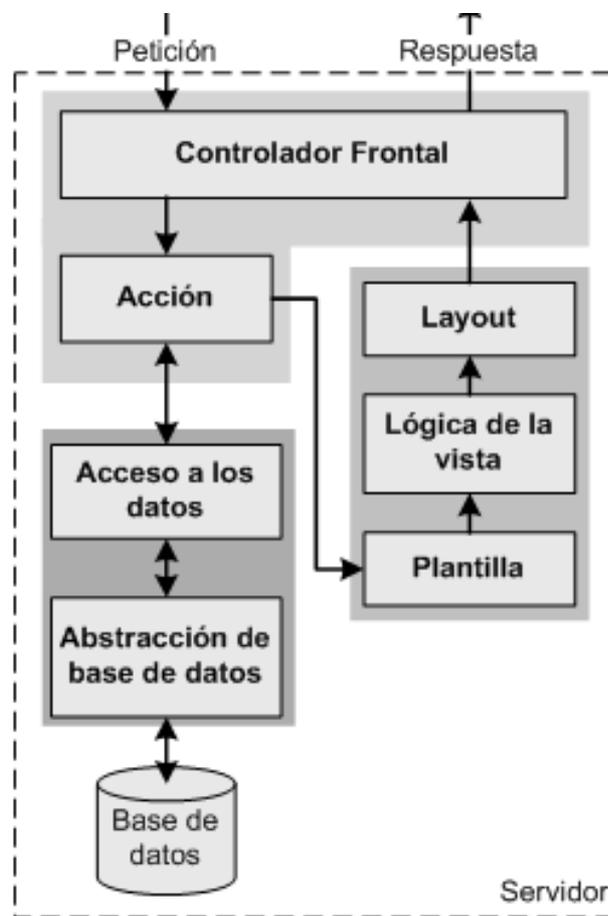


Fig. 3 Flujo de trabajo de Symfony

El MVC hace uso de dos de los patrones básicos de GRASP: bajo acoplamiento y alta cohesión, que constituyen dos de sus principales beneficios:

- Bajo acoplamiento:
  - Desacopla las vistas de los modelos

- Desacopla los modelos de la forma en que se muestran e ingresan los datos
- Alta cohesión:
  - Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo de negocio)

La explicación de los patrones de diseño mencionados anteriormente se encuentra detallada en el epígrafe 1.5.2.

## 1.4 Diseño

El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema. El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos; debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. Cuando la precisión del diseño es muy grande, la implementación puede ser hecha por un generador automático de código.

Propósito:

- Transformar los requerimientos en un diseño de cómo el sistema debe ser
- Desarrollar una robusta arquitectura del sistema
- Adaptar el diseño para que se corresponda con el entorno de implementación, diseñando sus funcionalidades

### 1.4.1 Roles y artefactos

Un rol es una definición abstracta de una serie de actividades y responsabilidades para ser desarrolladas y artefactos que deben ser producidos. Para el desarrollo de este trabajo se van a desempeñar tres de los roles que define RUP para el flujo de trabajo de Análisis y Diseño, y a obtener los artefactos correspondientes.

Diseñador: Este rol es responsable de diseñar una parte del sistema, dentro de las limitaciones de los requisitos, la arquitectura, y el proceso de desarrollo para el proyecto. El mismo identifica y define las responsabilidades, operaciones, atributos y relaciones de elementos del 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.

Artefactos que debe obtener el diseñador:

- *Clases del Diseño:* Una clase es una descripción de un conjunto de objetos que comparten las mismas responsabilidades, relaciones, operaciones, atributos, y la semántica
- *Realización de los casos de uso:* Describe cómo un caso de uso en particular es realizado, en el modelo del diseño en términos de colaboración de objetos. Tiene una descripción de flujos de eventos textual, diagramas de clases que muestra sus clases de diseño participantes y diagramas de interacción que muestran la realización de un flujo o escenarios concretos de un caso de uso en términos de interacción entre objetos 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:* La parte de un sistema que encapsula comportamiento, expone un conjunto de interfaces, paquetes y otros elementos del modelo. 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

- *Paquete 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

Arquitecto de software: Es el responsable de la arquitectura del Software, que incluye las decisiones técnicas claves que limitan el diseño global y la aplicación del proyecto.

Artefacto que debe obtener el arquitecto de software:

- *Modelo de despliegue*: El modelo de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los links de comunicación entre ellos, y las instancias de los componentes y objetos que residen en ellos

Diseñador de interfaz de usuario: Coordina el diseño de la interfaz de usuario. Está también involucrado en la recopilación de los requisitos y el diseño de los prototipos candidatos de la interfaz de usuario para satisfacer esas necesidades.

Artefacto que debe obtener el diseñador de interfaz de usuario:

- *Mapa de navegación*: Expresa la estructura de los elementos de la interfaz de usuario en el sistema, junto con sus caminos de navegación potenciales
- *Prototipos de interfaz de usuario no funcionales*: Es un ejemplo de la interfaz de usuario que es construida con el fin de explorar y/o validar el diseño de la interfaz de usuario

## 1.5 Patrones de Diseño

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

### 1.5.1 Objetivos de los patrones de diseño

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente
- Formalizar un vocabulario común entre diseñadores

- Estandarizar el modo en que se realiza el diseño
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente

No es obligatorio utilizar los patrones siempre, solo en el caso de tener el mismo problema o similar que soluciona el patrón, siempre teniendo en cuenta que en un caso particular puede no ser aplicable. Abusar o forzar el uso de los patrones puede ser un error.

### 1.5.2 Patrones de diseño a utilizar

#### ➤ Decorator:

Responde a la necesidad de añadir dinámicamente funcionalidad a un objeto. No es necesario crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera. (Ver estructura del patrón Anexo 2)

El decorator se debe usar para: [12]

- Adicionar responsabilidades a objetos individuales dinámicamente sin afectar otros objetos
- Para agregar responsabilidades que pueden ser retiradas
- Cuando no es práctico adicionar responsabilidades por medio de la herencia

Ventajas del patrón de diseño decorator: [12]

- Más flexible que la herencia: responsabilidades pueden añadirse y eliminarse en tiempo de ejecución
- Diferentes decoradores pueden ser conectados a un mismo objeto
- Reduce el número de propiedades en las clases de la parte alta de la jerarquía
- Es simple añadir nuevos decoradores de forma independiente a las clases que extienden
- Un objeto decorador tiene diferente OID (identificador único que proporciona el sistema) que el objeto que decora
- Sistemas con muchos y pequeños objetos

El decorator en Symfony es implementado específicamente por el layout, archivo denominado también plantilla global que 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 otro punto de vista, el layout decora la plantilla. (Ver Anexo 3)

➤ Patrón Bajo Acoplamiento

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, contrario a las clases con fuerte acoplamiento, estas últimas pueden presentar varios problemas, entre los que están que los cambios de las clases de las que depende ocasionan cambios locales y son más difíciles de reutilizar, porque se requiere la presencia de otras clases de las que depende.

➤ Patrón Alta Cohesión

La cohesión 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. Una clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo, este tipo de clase puede presentar varios problemas; son difíciles de comprender, reutilizar y conservar.

## Conclusiones

Como consecuencia del crecimiento de la capacidad de generación y almacenamiento de la información, se hace necesario tener en cuenta mecanismos para manipular la misma de forma eficiente, por lo que se hace importante el manejo y gestión de la información a través de los Sistemas de Gestión de la Información, dos ejemplos ilustrativos de la importancia de estos sistemas son los LIMS y la web como Sistema de Información. Para el desarrollo del LIMS, específicamente el diseño del módulo de Liberación Analítica se seleccionaron como herramientas a utilizar el Visual Paradigm como herramienta CASE, el Symfony como framework, RUP como Proceso de Desarrollo y UML como Lenguaje de Modelado.

## CAPÍTULO 2: DISEÑO

### Introducción

En este Capítulo se hace una breve referencia a la arquitectura del sistema a través de las vistas lógica y de despliegue. Se obtienen además algunos de los artefactos del flujo de trabajo de Análisis y Diseño, tales como prototipos de interfaz de usuario, mapa de navegación, diagramas de clases del diseño y diagramas de interacción (específicamente diagramas de secuencia). Finalmente se hace una evaluación de la calidad de las clases diseñadas.

### 2.1 Breve referencia a la arquitectura del sistema

A continuación se hace referencia a la arquitectura que se va a usar para desarrollar el LIMS de Calidad del CIGB a través de la vista lógica y la vista de despliegue.

*Vista Lógica:* Describe el diseño más importante de las clases y su organización en paquetes y subsistemas, y la organización de éstos en capas. También contiene algunas realizaciones de casos de uso. Esta muestra cómo la funcionalidad es diseñada en el interior del sistema, en términos de la estructura estática y comportamiento dinámico del sistema.

*Vista de despliegue:* Suministra una base para la comprensión de la distribución física de un sistema a través de nodos. Suele utilizarse cuando el sistema está distribuido, esto incluye la asignación de tareas provenientes de la vista de procesos en los nodos.

#### 2.1.1 Vista Lógica

Esta sección tratará la vista lógica de la arquitectura, para lo que se tendrán en cuenta los elementos más significativos del flujo de trabajo de diseño. Para estructurar el sistema, no se identificaron en este módulo subsistemas funcionales, se organizó por paquetes a partir de los casos de uso arquitectónicamente significativos, que constituyen la totalidad de los casos de uso identificados para el módulo de Liberación Analítica.

Pueden existir paquetes formados por otros paquetes y éstos tendrán como todos, su realización de casos de uso. Tal es el caso de los paquetes: Gestión de Análisis y Cálculos y planillas relacionadas, Gestión de muestras testigos y planillas relacionadas y Gestión de resultados de pruebas y planillas relacionadas. En la vista lógica que se muestra a continuación, éstos aparecen representados al igual que los otros de forma independiente y posteriormente se muestra el contenido de cada uno de ellos.

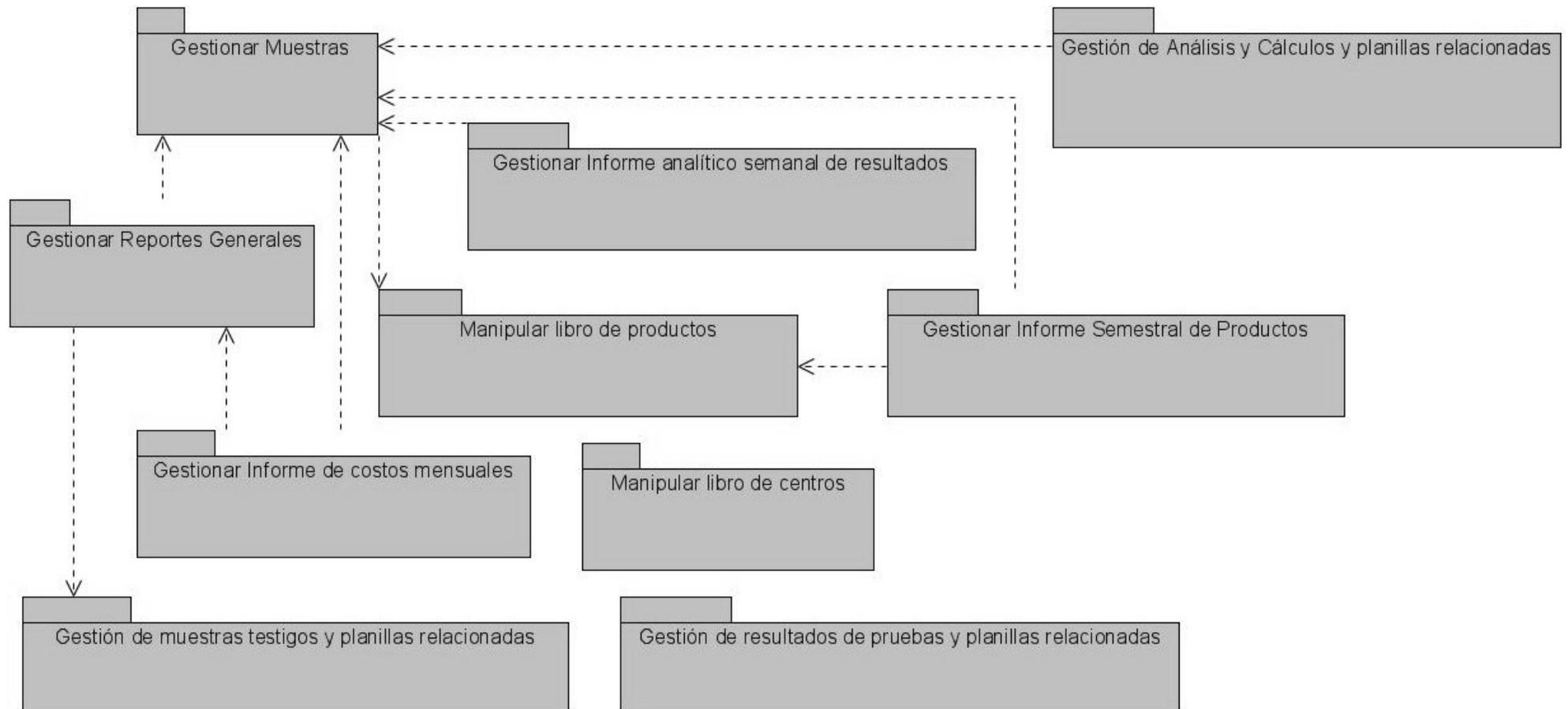
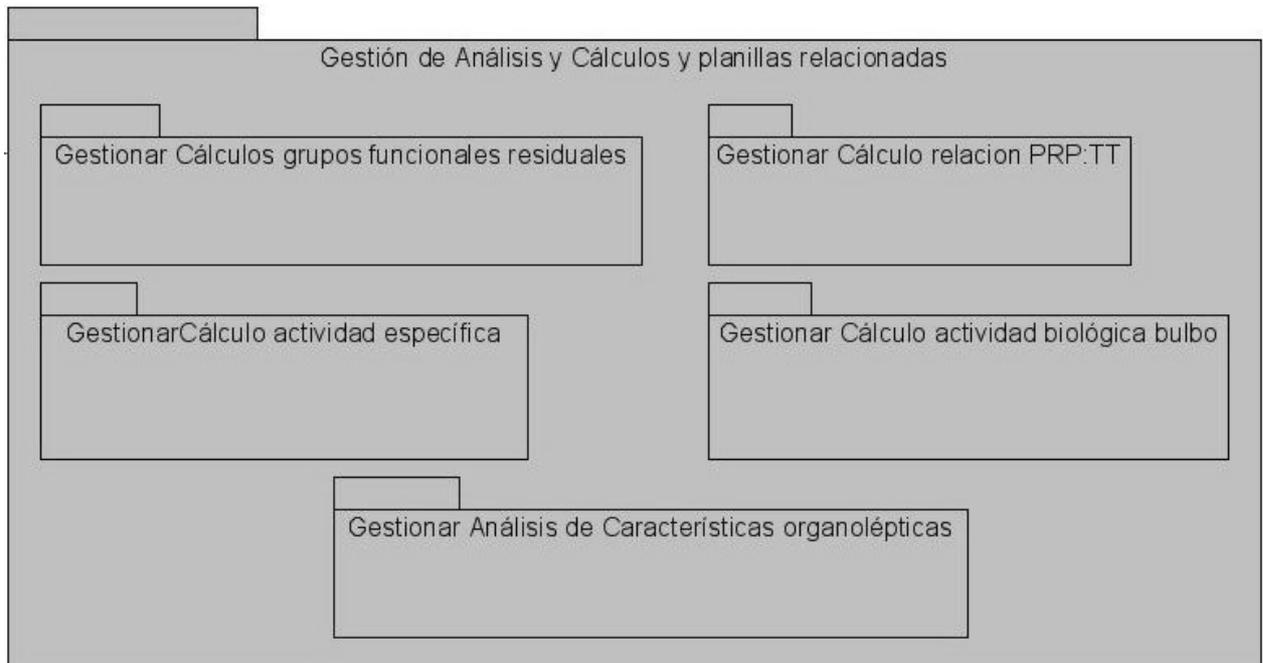
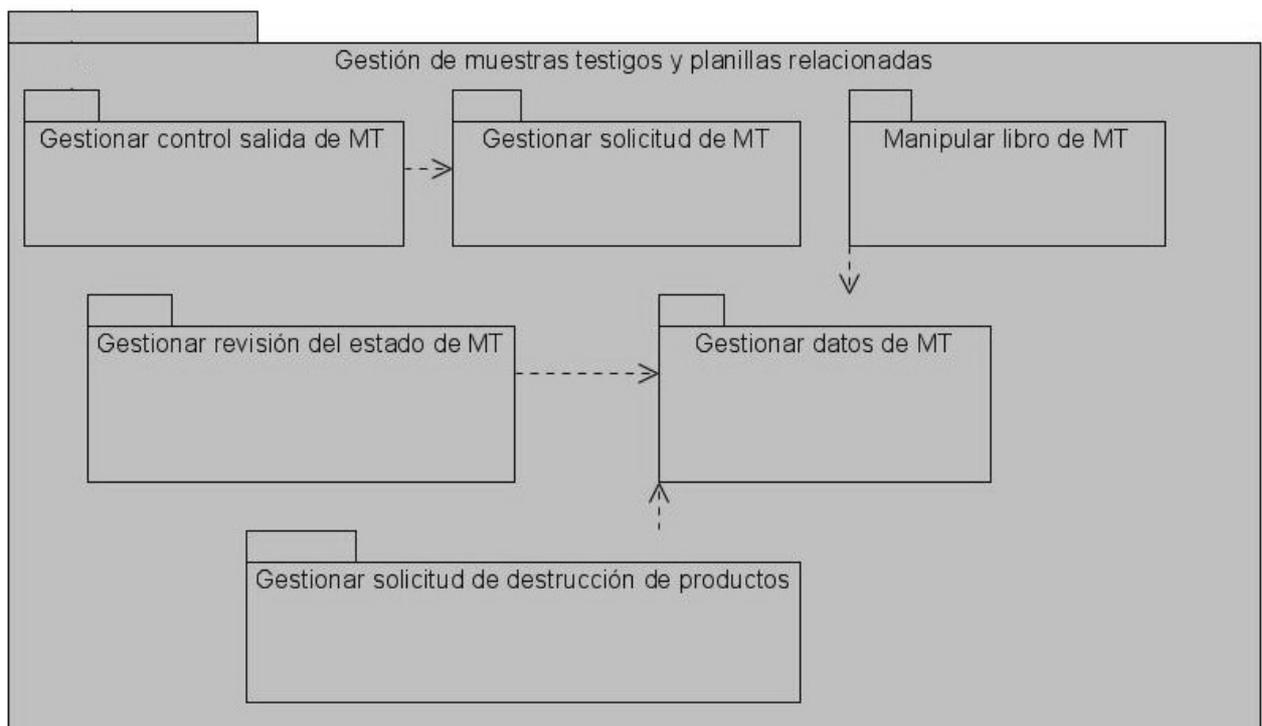


Fig. 4 Vista Lógica



**Fig. 5 Paquete Gestión de Análisis y Cálculos y planillas relacionadas**



**Fig. 6 Paquete Gestión de muestras testigos y planillas relacionadas**

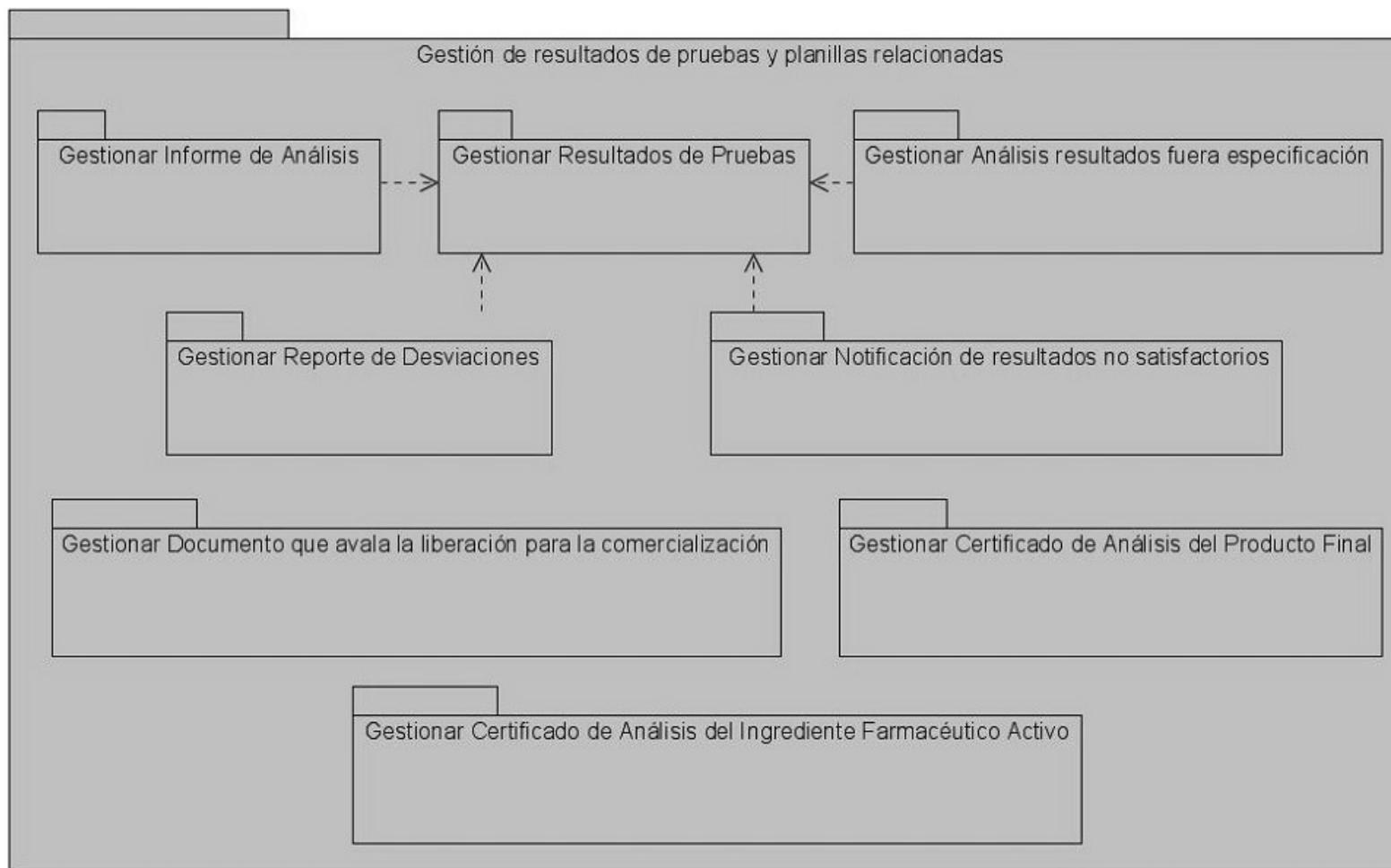


Fig. 7 Gestión de resultados de pruebas y planillas relacionadas

Cada paquete que conforma la vista lógica tiene su realización de CU, es decir, su diagrama de clases del diseño y de interacción. A continuación se muestra el diagrama de clases del diseño del paquete Manipular libro de productos, en el que se evidencia el patrón arquitectónico definido (Modelo-Vista-Controlador) a partir del uso del Symfony como framework.

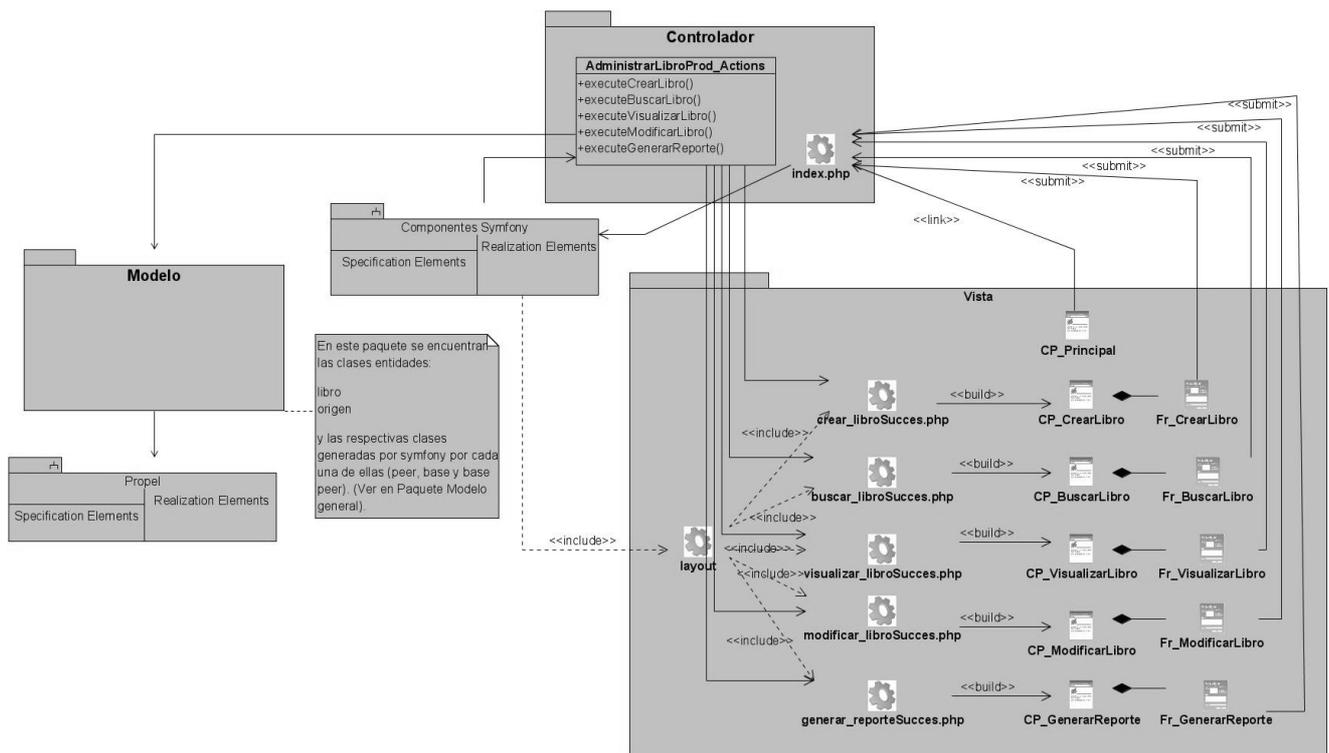


Fig. 8 Diagrama de clases del diseño CU Manipular libro de productos

### 2.1.2 Vista de Despliegue

La vista de despliegue se va a representar mediante un diagrama de despliegue en el Visual Paradigm, el propósito del mismo es representar la configuración de los elementos de procesamiento del sistema y las conexiones entre ellos, además de los componentes y objetos. Éste consiste en uno o varios nodos (elementos de procesamiento con al menos un procesador, memoria y posiblemente otros dispositivos), dispositivos (nodos sin capacidad de procesamiento) y conectores, entre los nodos y entre éstos y los dispositivos.

La aplicación estará distribuida de la siguiente forma:

El servidor de aplicaciones contiene la aplicación Web para que los usuarios transcriban diferentes datos, se relacionará mediante una conexión ADO con el servidor de Base de Datos donde van a estar centralizados los datos recopilados. Los usuarios, desde sus estaciones de trabajo podrán acceder mediante el protocolo HTTPS al Servidor de Aplicaciones; contando cada estación de trabajo con una impresora para la impresión de planillas y otros documentos de importancia, y un Scanner para la digitalización de información importante proveniente de otros centros, que se encuentra en formato duro.

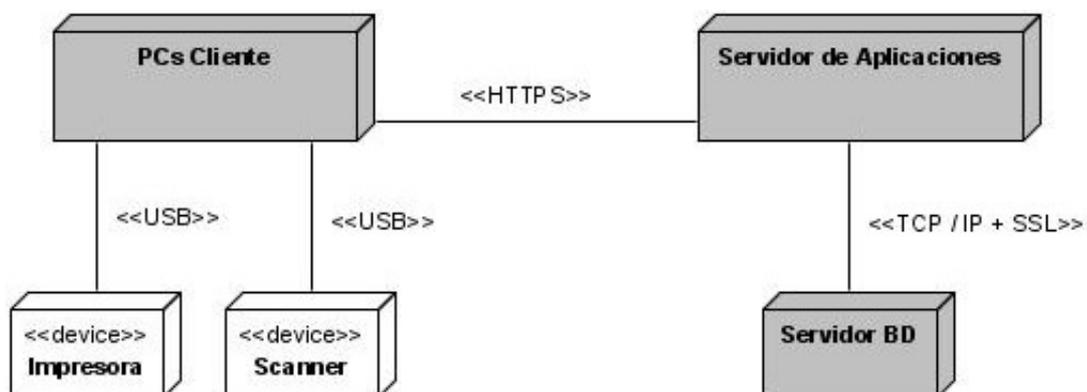


Fig. 9 Vista de Despliegue

## 2.2 Prototipos no funcionales de interfaz de usuario

Los prototipos no funcionales de interfaz de usuario son una presentación de la interfaz del producto que representa la funcionalidad contenida en los casos de uso (CU); de manera que permita que el usuario verifique que el sistema va a satisfacer sus necesidades.

Para el módulo de Liberación Analítica en investigaciones anteriores se identificaron veinticinco CU (Ver Expediente de Proyecto). De ellos, los CU Manipular libro de productos y Gestionar Muestras tienen gran importancia para el funcionamiento del módulo de Liberación Analítica, ya que a partir de ellos se derivan los procesos que se llevan a cabo en dicho módulo. A continuación se muestran los prototipos del CU Manipular libro de productos y los correspondientes al CU Gestionar Muestras se pueden observar en los Anexos. (Ver Anexo 4)

*Prototipos no funcionales del CU Manipular libro de productos*

*Crear nuevo libro por etapa*

<b>CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA</b>		
<b>DIRECCION DE CALIDAD</b>		
<b>LIBRO DE PRODUCTOS</b>		
<b>Nombre del producto:</b>	<b>Etapas:</b>	<b>Fuente:</b>
<input type="text"/>	<input type="text"/>	<input type="text"/>
		<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>

**Fig. 10 Prototipo Crear Libro por Etapa**

Buscar y visualizar libro

<b>CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA</b>			
<b>DIRECCION DE CALIDAD</b>			
<b>LIBRO DE PRODUCTOS</b>			
<b>Nombre del producto:</b>	<b>Etapa:</b>	<b>Fuente:</b>	
<input type="text"/>	<input type="text"/>	▼	
<input type="button" value="Buscar"/> <input type="button" value="Cancelar"/>			
<b>Nombre del producto:</b>	<b>Etapa:</b>	<b>Fuente:</b>	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Visualizar"/>

Fig. 11 Prototipo Buscar Libro de Productos

<b>CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA</b>			
<b>DIRECCION DE CALIDAD</b>			
<b>LIBRO DE PRODUCTOS</b>			
<b>Nombre del producto:</b>	<b>Etapa:</b>	<b>Fuente:</b>	
<input type="text"/>	<input type="text"/>	▼	
<input type="button" value="Modificar"/> <input type="button" value="Nueva Búsqueda"/> <input type="button" value="Imprimir"/> <input type="button" value="Cancelar"/>			

Fig. 12 Prototipo Visualizar Libro de Productos

Modificar datos del libro

<b>CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA</b>		
<b>DIRECCION DE CALIDAD</b>		
<b>LIBRO DE PRODUCTOS</b>		
<b>Nombre del producto:</b>	<b>Etapas:</b>	<b>Fuente:</b>
<input type="text"/>	<input type="text"/>	<input type="text"/>
		<input type="button" value="Actualizar"/> <input type="button" value="Cancelar"/>

Fig. 13 Prototipo Modificar Libro de Productos

Generar reportes

<b>CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA</b>		
<b>DIRECCION DE CALIDAD</b>		
<b>LIBRO DE PRODUCTOS</b>		
<b>Nombre del producto</b>	<b>Etapa</b>	<b>Fuente</b>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Generar Reporte"/> <input type="button" value="Cancelar"/>		
<b>Nombre del producto:</b>	<b>Etapa:</b>	<b>Fuente:</b>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Imprimir"/> <input type="button" value="Generar otro Reporte"/> <input type="button" value="Cancelar"/>		

Fig. 14 Prototipo Generar Reportes Libro de Productos

## 2.3 Mapa de Navegación

Los mapas de navegación se basan en los casos de uso y muestran las más importantes rutas de navegación. Una ruta de navegación es una secuencia de pantallas (ventanas, páginas web) atravesada por el usuario.

UML posee una extensión para el modelado de aplicaciones Web, que es usada para el diseño de las clases y usa varios estereotipos: <<Server Page>>, <<Client Page>> y <<Form>>. Esta notación de UML fue utilizada para la construcción del mapa de navegación, el estereotipo de <<Client Page>> se usa para ilustrar las diferentes páginas web por las que se puede navegar y el <<link >> para las rutas de navegación entre las páginas.

La navegabilidad se estructuró de la siguiente manera: a partir de la página principal (CP\_Principal) se podrá acceder a las páginas de los laboratorios y grupos de la Dirección de Calidad, entre ellos el grupo de Liberación Analítica, en el mismo estarán disponibles las páginas que permiten la gestión de las muestras, los libros y planillas, estas tienen acceso a las distintas páginas en dependencia de la acción que se necesita efectuar. Como parte de la navegabilidad se podrá ir de las diferentes páginas de gestión antes mencionadas, al grupo de Liberación Analítica, y de esta última a la principal.

En este documento se muestra un fragmento del mapa de navegación confeccionado, específicamente para el caso de la Gestión de los documentos: Solicitud de destrucción de productos, Análisis de las características organolépticas y Resultado de pruebas, para los restantes documentos que se manipulan en el módulo de Liberación Analítica el proceso ocurre de forma similar. (Ver Mapa de Navegación completo en Expediente de Proyecto)

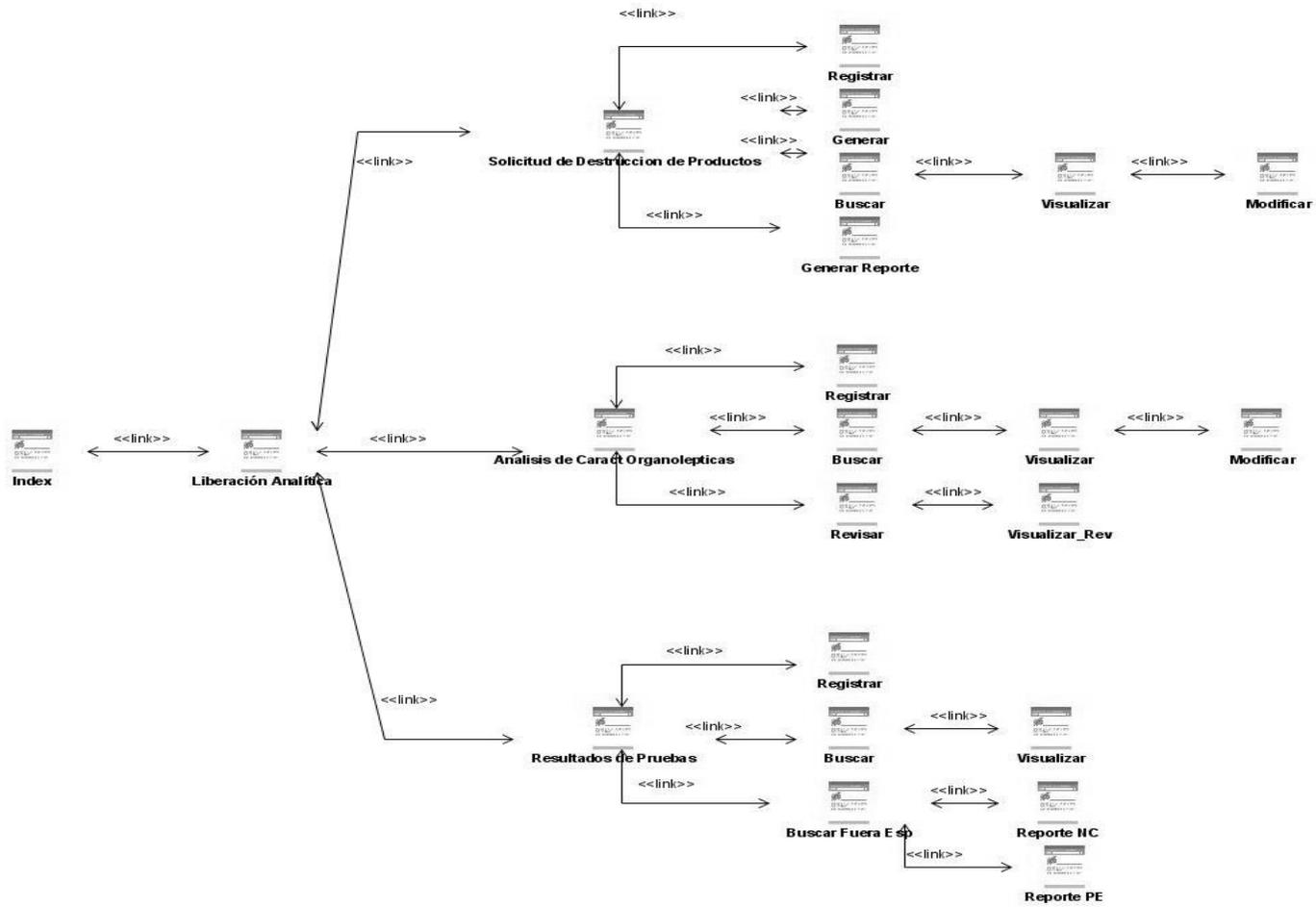


Fig. 15 Fragmento del Mapa de Navegación

## 2.4 Diagramas de clases del diseño

Para mayor claridad en los diagramas de clases del diseño, se agruparon las clases en paquetes, a continuación se muestran los paquetes y su composición por cada diagrama de clases:

- Vista:
  - Páginas clientes
  - Formularios
  - Páginas servidoras
  - Layout
- Controlador:
  - Controlador frontal: Es el único punto de entrada a la aplicación, carga la configuración y determina la acción a ejecutarse
  - Acciones: Contienen la lógica de la aplicación, verifican la integridad de las peticiones y preparan los datos requeridos por la capa de presentación
- Modelo: Se encuentra vacío en cada uno de los diagramas, pues las clases implicadas en el mismo se encuentran en un paquete Modelo general, en el que van a estar todas las entidades persistentes, con sus respectivas relaciones y las clases generadas por Symfony (Base, Peer y Base Peer correspondientes a cada una de ellas), de las cuales las clases Objeto y Peer tienen una relación de asociación con la acción correspondiente y las clases Base utilizan la librería Propel para el mapeo de objetos a base de datos. En cada diagrama una nota específica cuáles clases del Paquete Modelo general se van a usar
  - Clases Base: Son las que se generan directamente a partir del esquema, no se deben modificar esas clases, porque cada vez que se genera el modelo, estas se borran
  - Clases Objeto: Heredan de las clases con nombre Base, estas no se modifican cuando se genera el modelo, por lo que en las mismas se añaden los métodos propios
  - Clases Peer: tienen métodos estáticos para trabajar con las tablas de la base de datos, proporcionan los medios necesarios para obtener los registros de las tablas y sus métodos devuelven normalmente un objeto o una colección de objetos de la clase objeto relacionada

A continuación se muestra un fragmento del paquete Modelo al que hacen referencia todos los diagramas de clases del diseño: (Ver Modelo completo en Expediente de Proyecto)

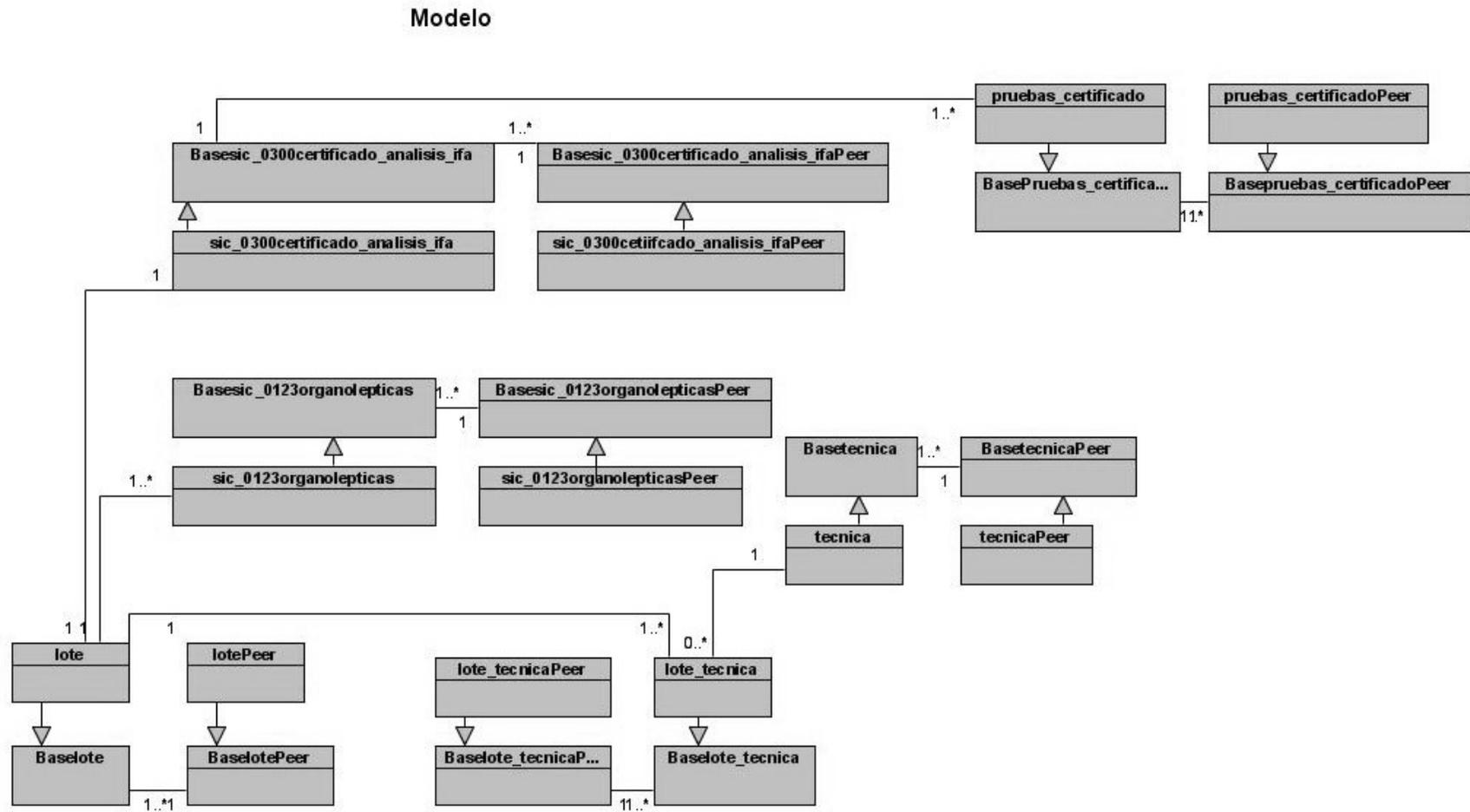


Fig. 16 Fragmento del Paquete Modelo

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

Caso de Uso (CU) Gestionar\_ Muestras

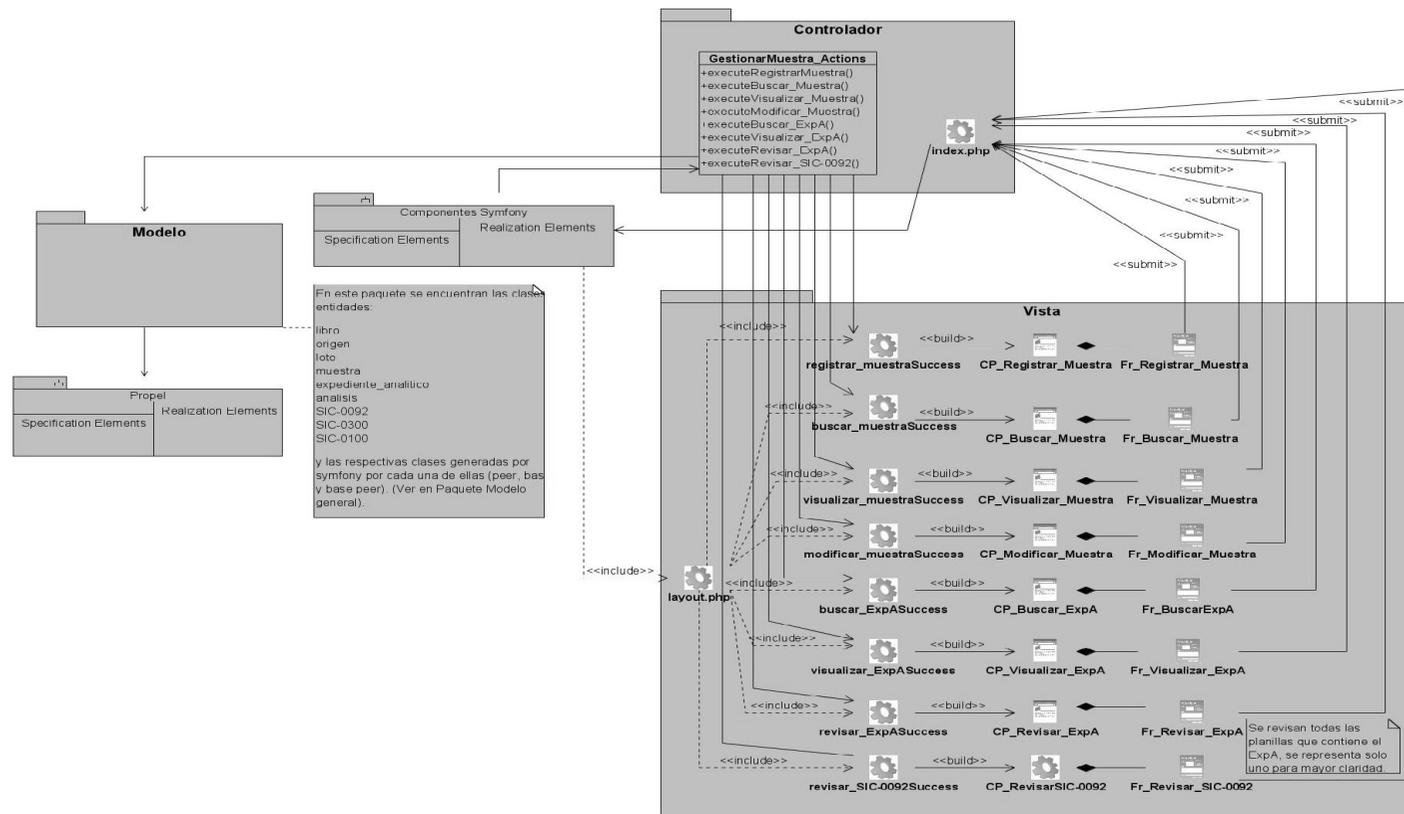


Fig. 17 Diagrama de clases del diseño CU Gestionar Muestras

CU Administrar libro de registros por centros

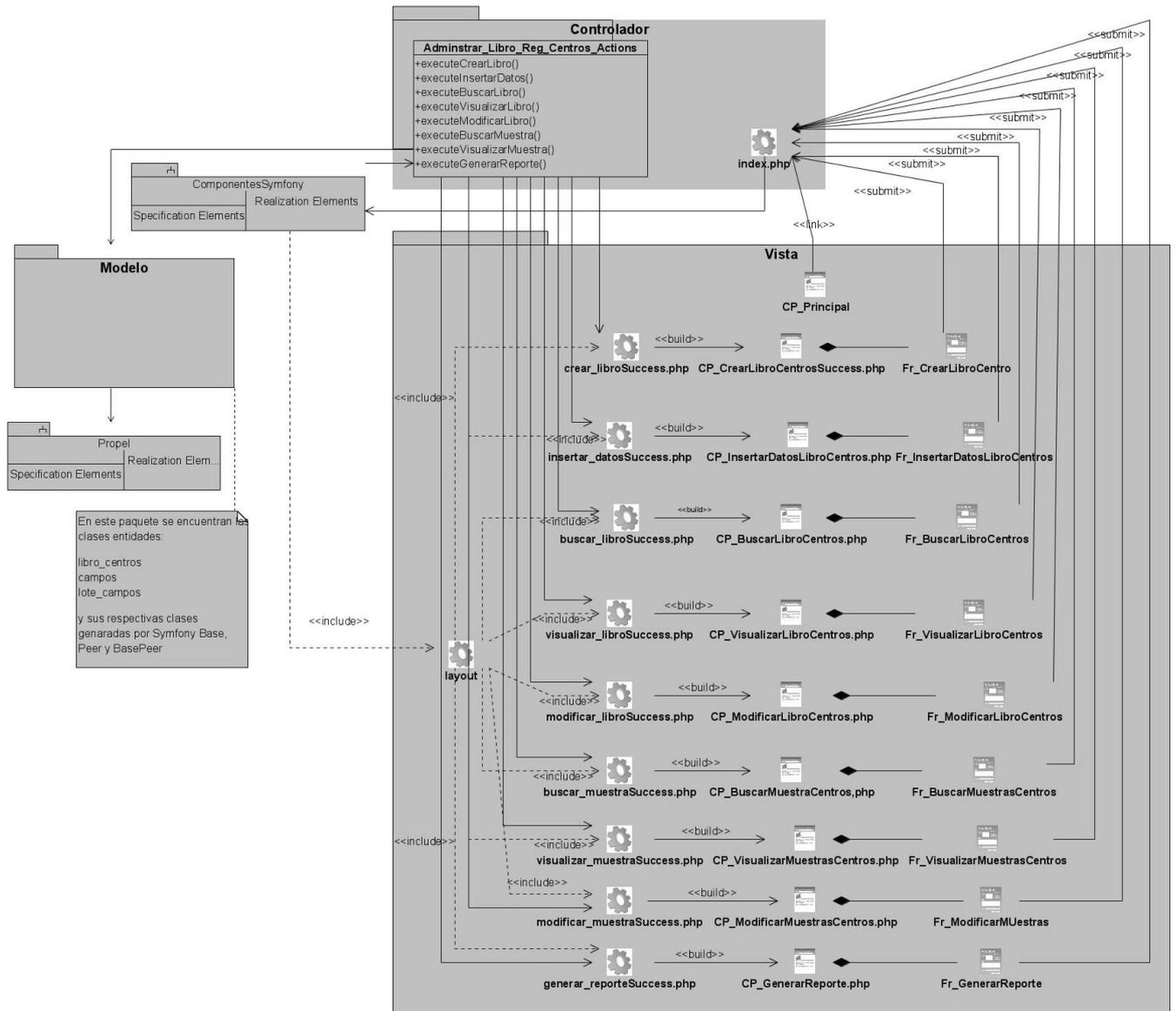


Fig. 18 Diagrama de clases del diseño CU Administrar Libro de registro por centros

CU Gestionar datos de muestras testigo

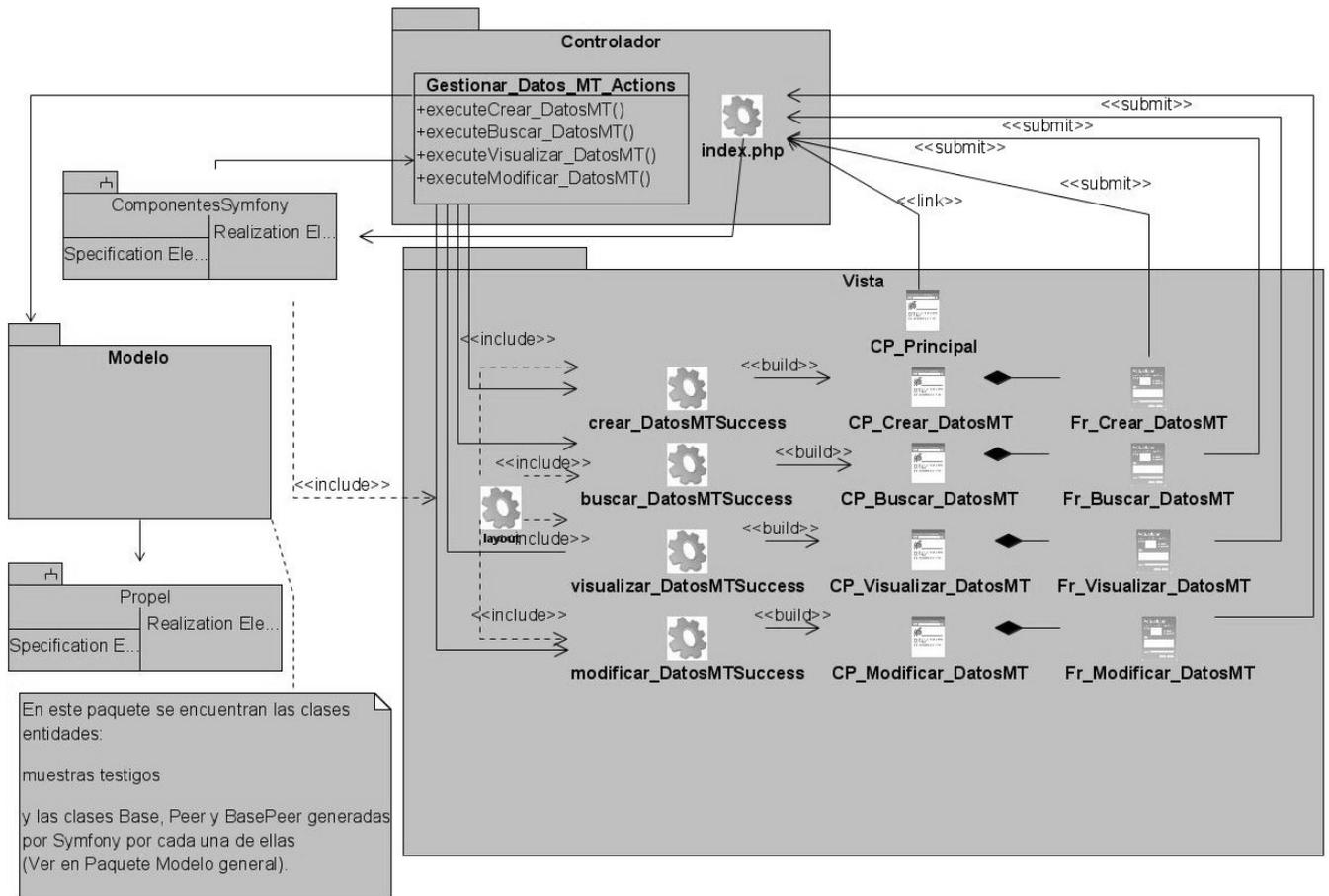


Fig.19 Diagrama de clases del diseño CU Gestionar datos de muestras testigo

CU Manipular libro de registros de muestras testigos por producto

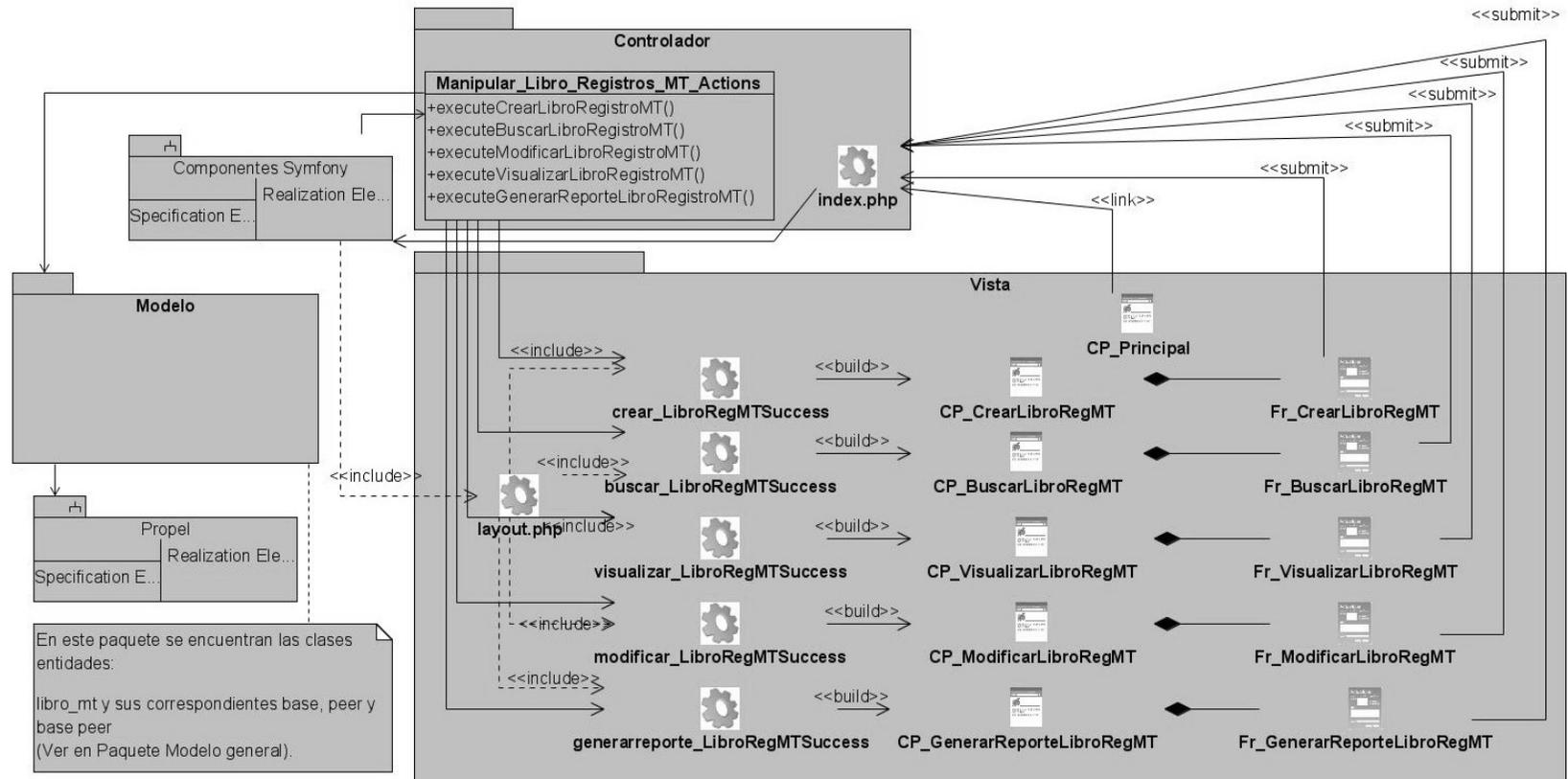


Fig.20 Diagrama de clases del diseño CU Manipular libro de registros de muestras testigos por producto

CU Gestionar Solicitud de muestras testigo

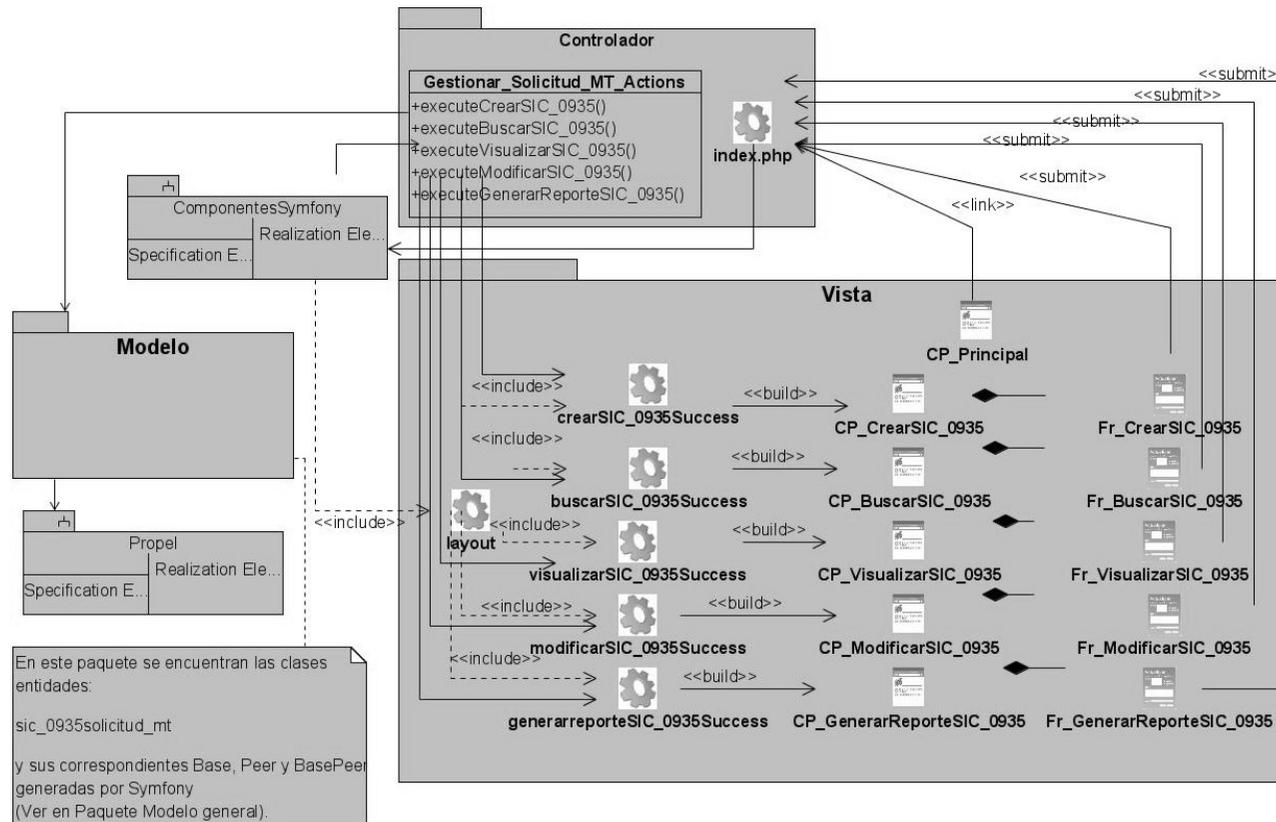


Fig. 21 Diagrama de clases del diseño CU Gestionar Solicitud de muestras testigo

CU Gestionar Control de Salida de muestras testigo

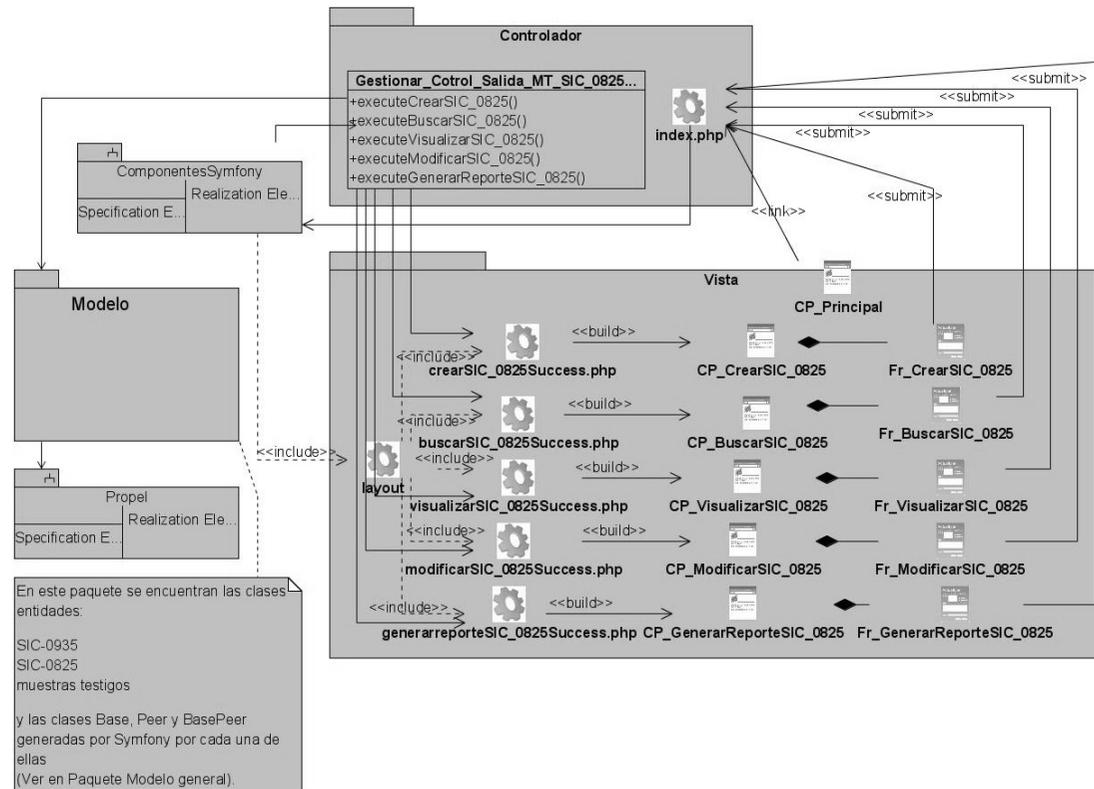


Fig. 22 Diagrama de clases del diseño CU Gestionar Control de Salida de muestras testigo

CU Gestionar Revisión del Estado de muestras testigos

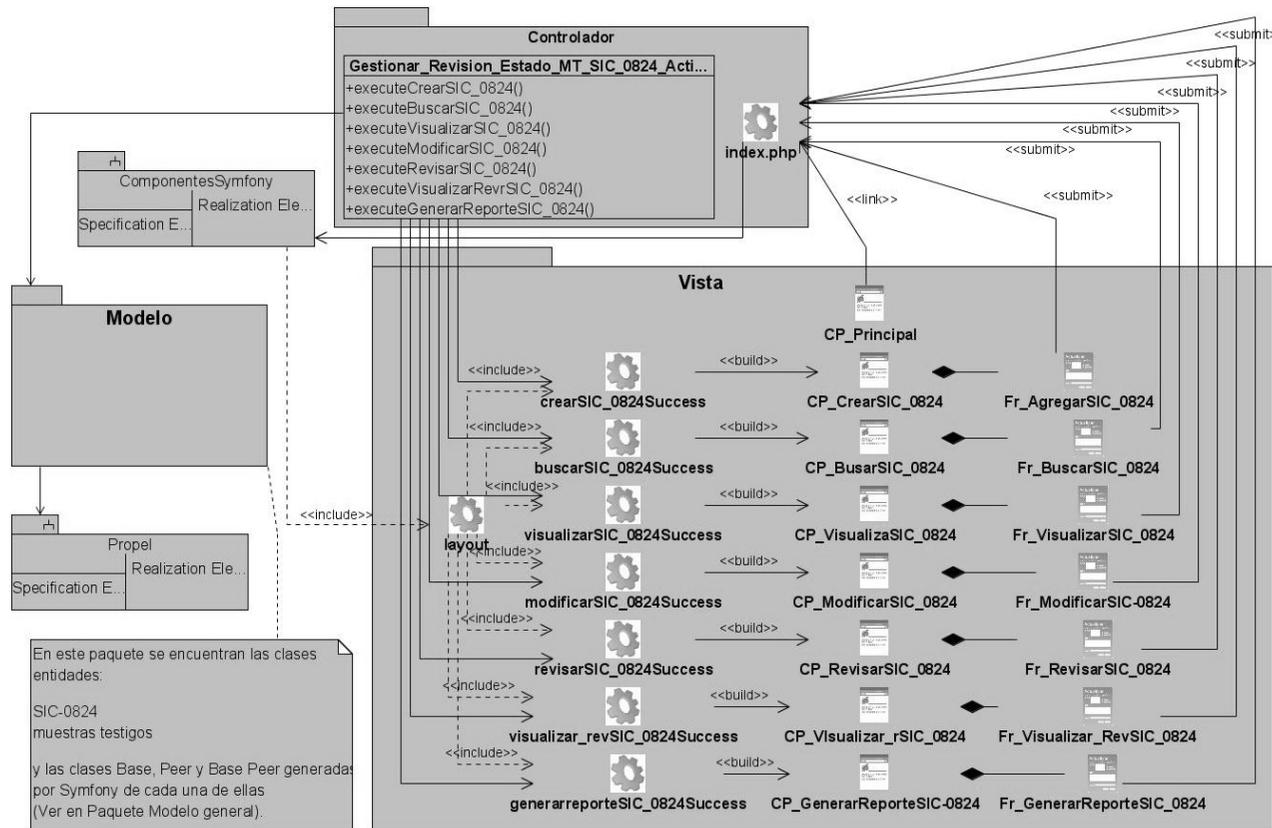


Fig. 23 Diagrama de clases del diseño CU Gestionar Revisión del estado de muestras testigo

CU Gestionar Solicitud de destrucción de productos

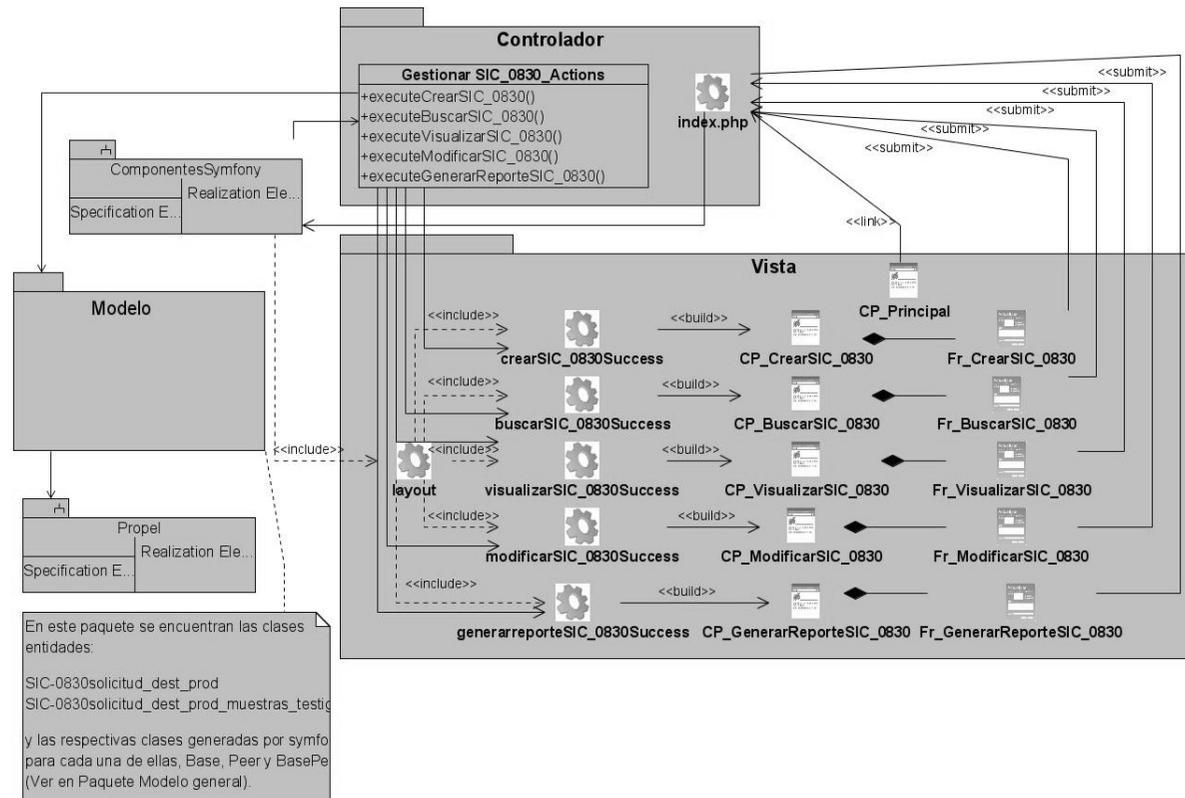


Fig. 24 Diagrama de clases del diseño CU Gestionar Solicitud de destrucción de productos

CU Gestionar Análisis de Características Organolépticas

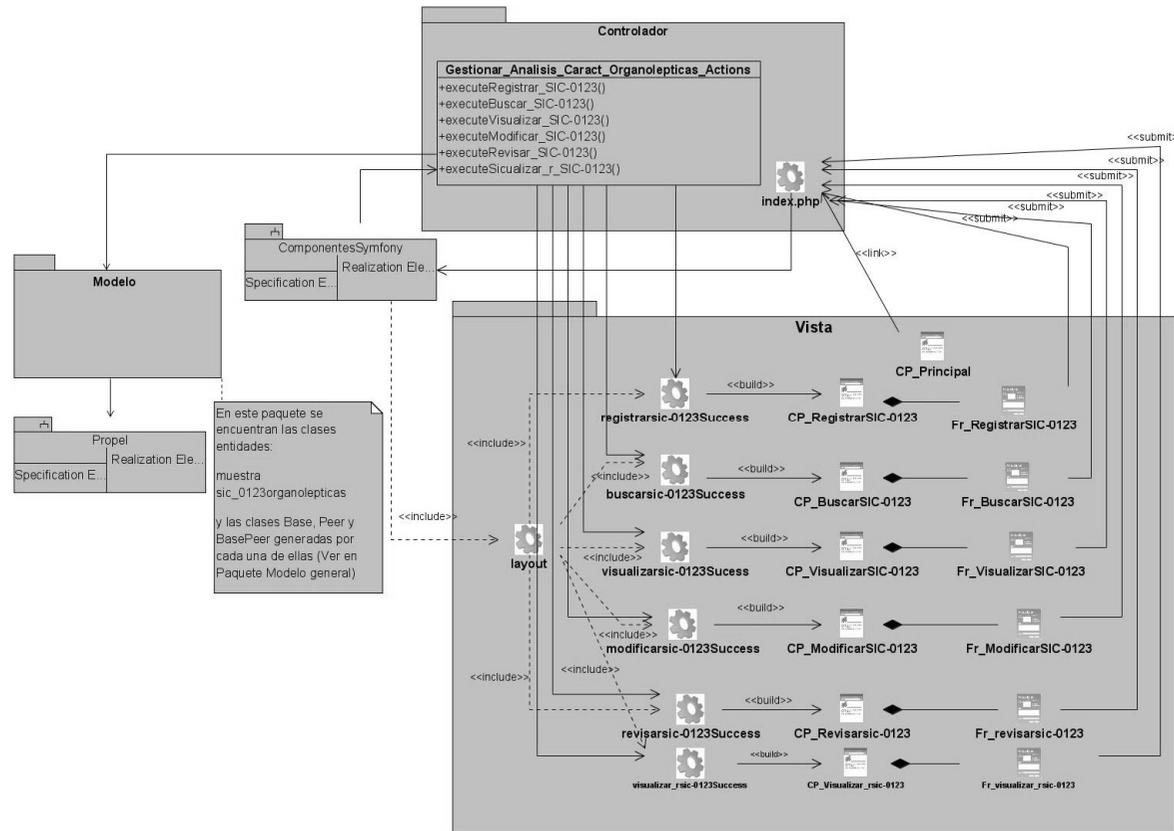


Fig. 25 Diagrama de clases del diseño CU Gestionar Análisis de características organolépticas

CU Gestionar resultados de las pruebas

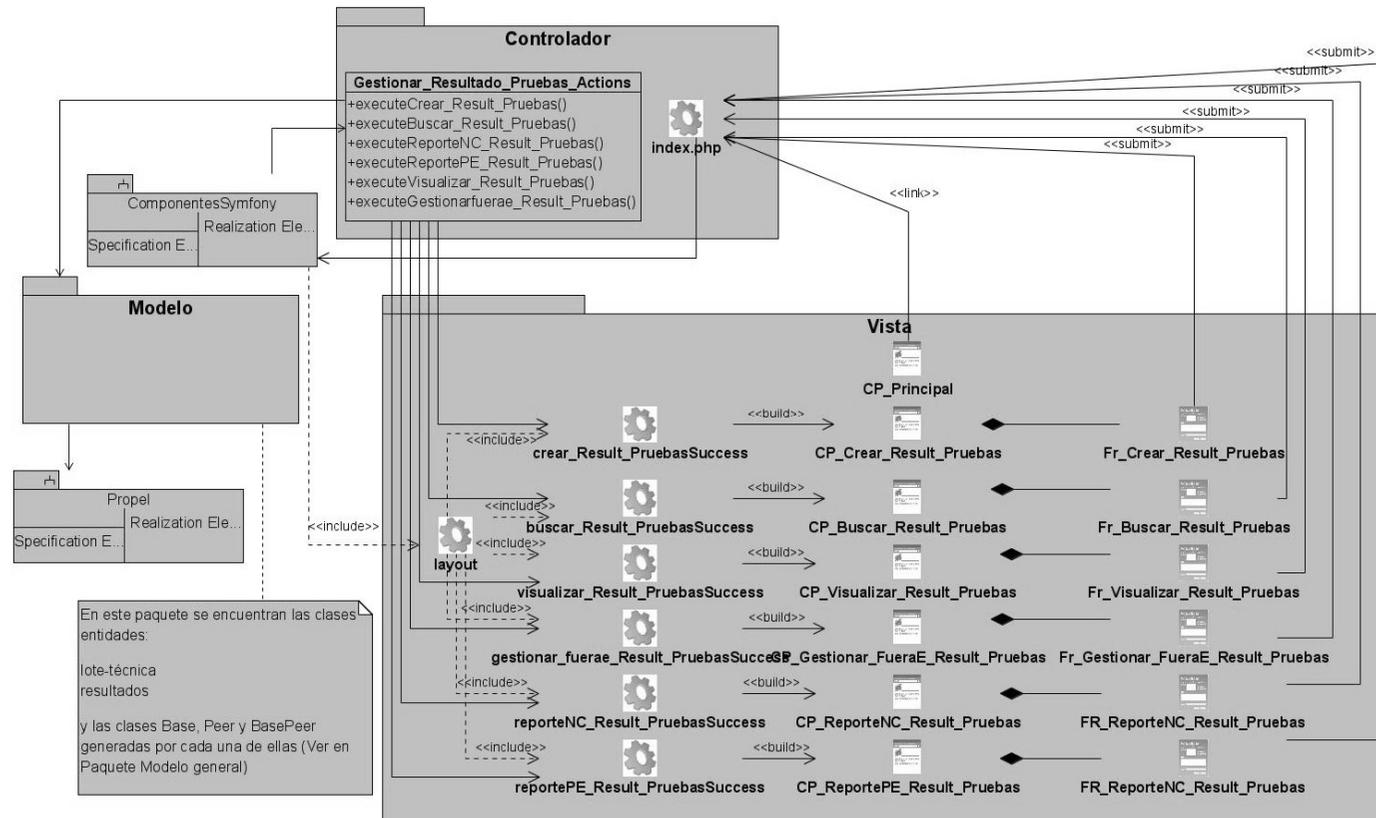


Fig. 26 Diagrama de clases del diseño CU Gestionar resultados de las pruebas

CU Gestionar Análisis de resultados fuera de especificación

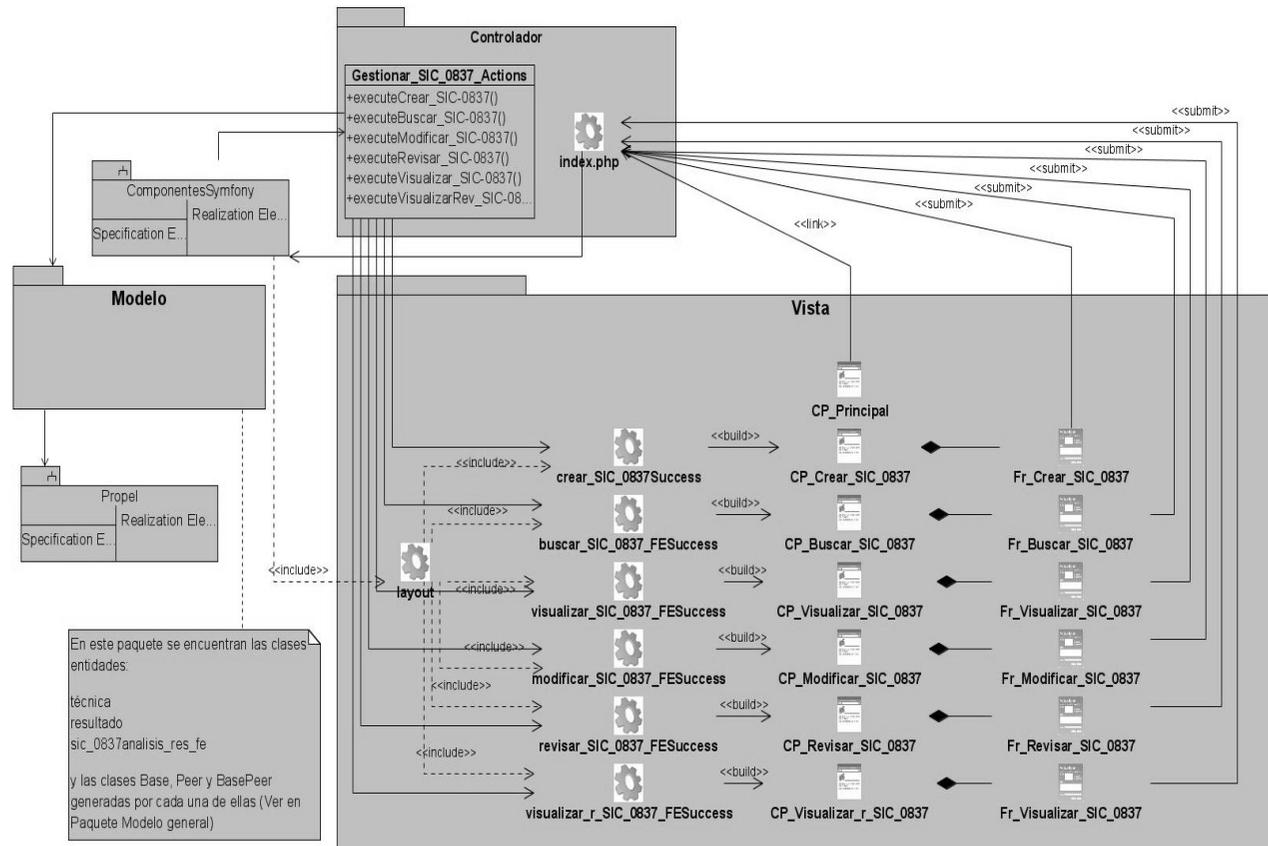


Fig. 27 Diagrama de clases del diseño CU Gestionar Análisis de resultados fuera de especificación



CU Gestionar Cálculo relación PRP\_ TT

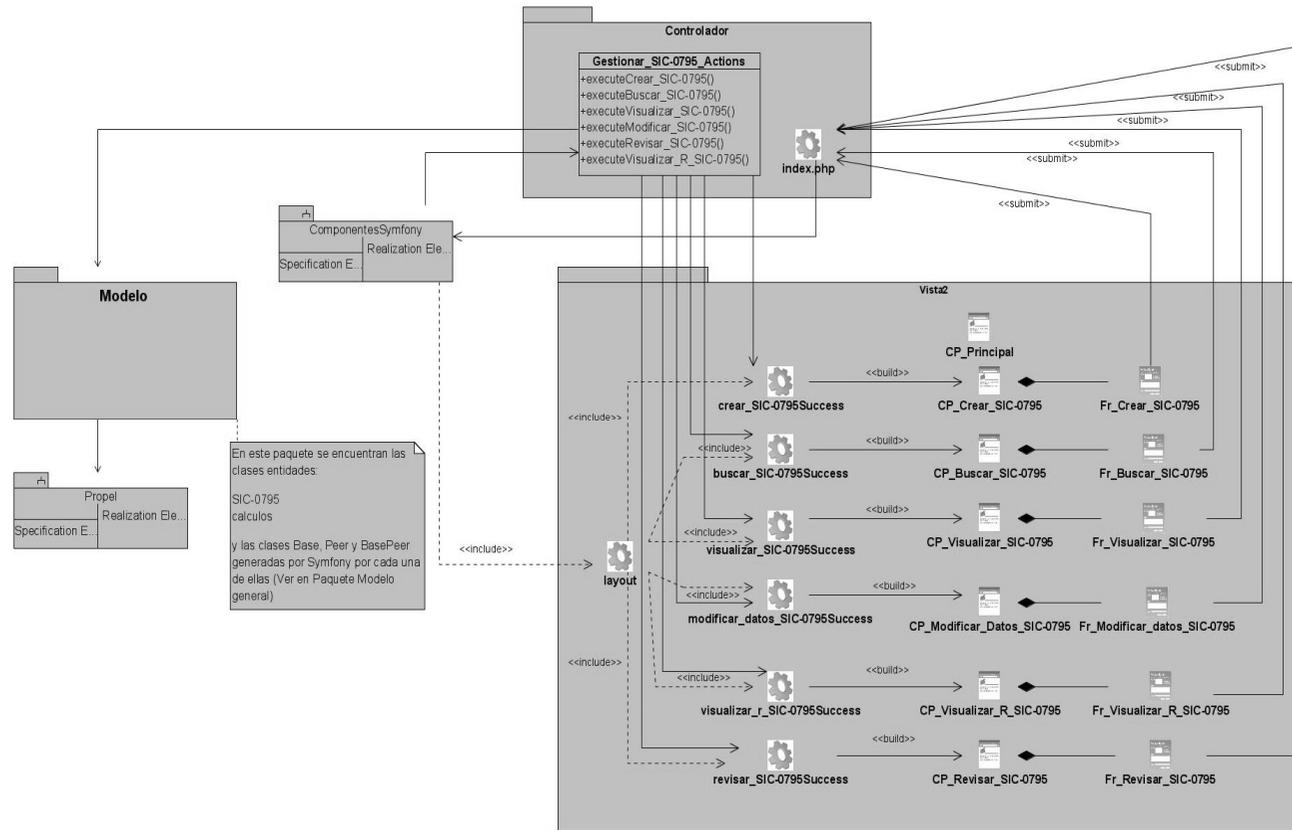


Fig. 29 Diagrama de clases del diseño CU Gestionar Cálculo de relación PRP: TT

CU Gestionar Cálculo grupos funcionales residuales

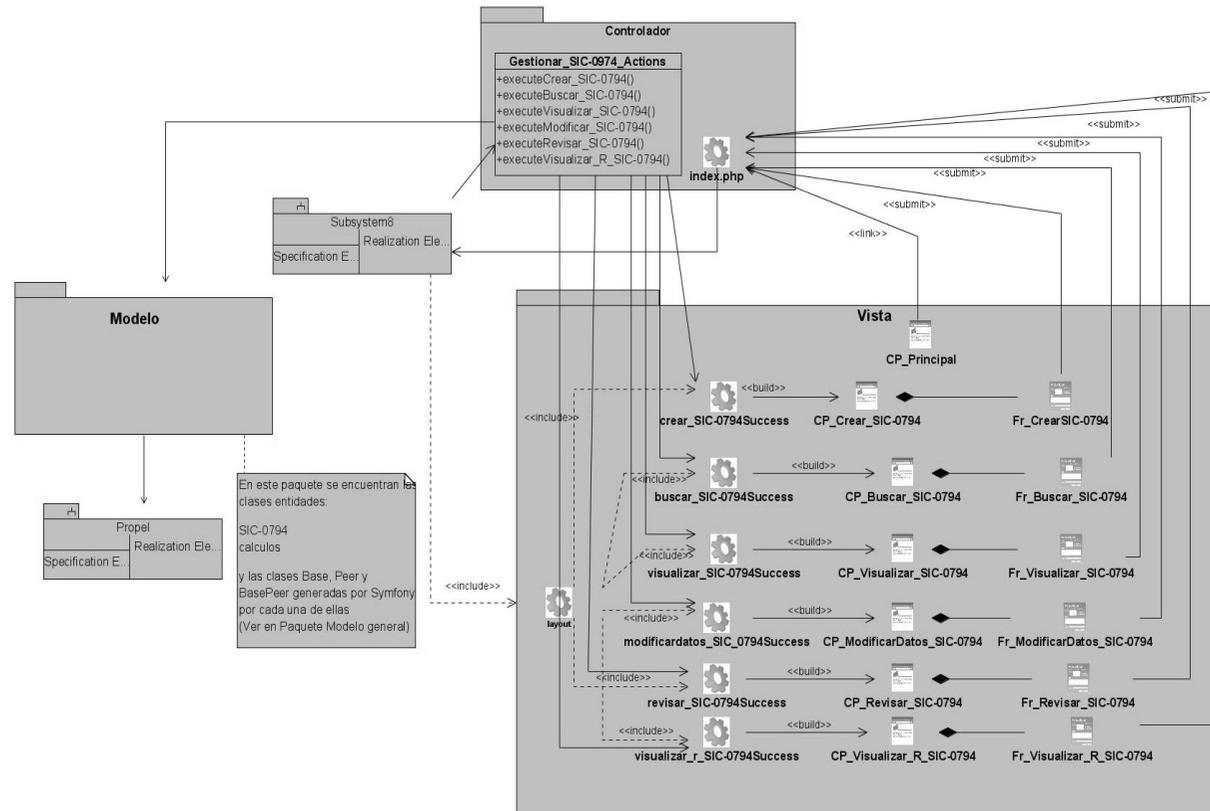


Fig. 30 Diagrama de clases del diseño CU Gestionar Cálculo de grupos funcionales residuales

CU Gestionar Cálculo de la actividad biológica por bulbo

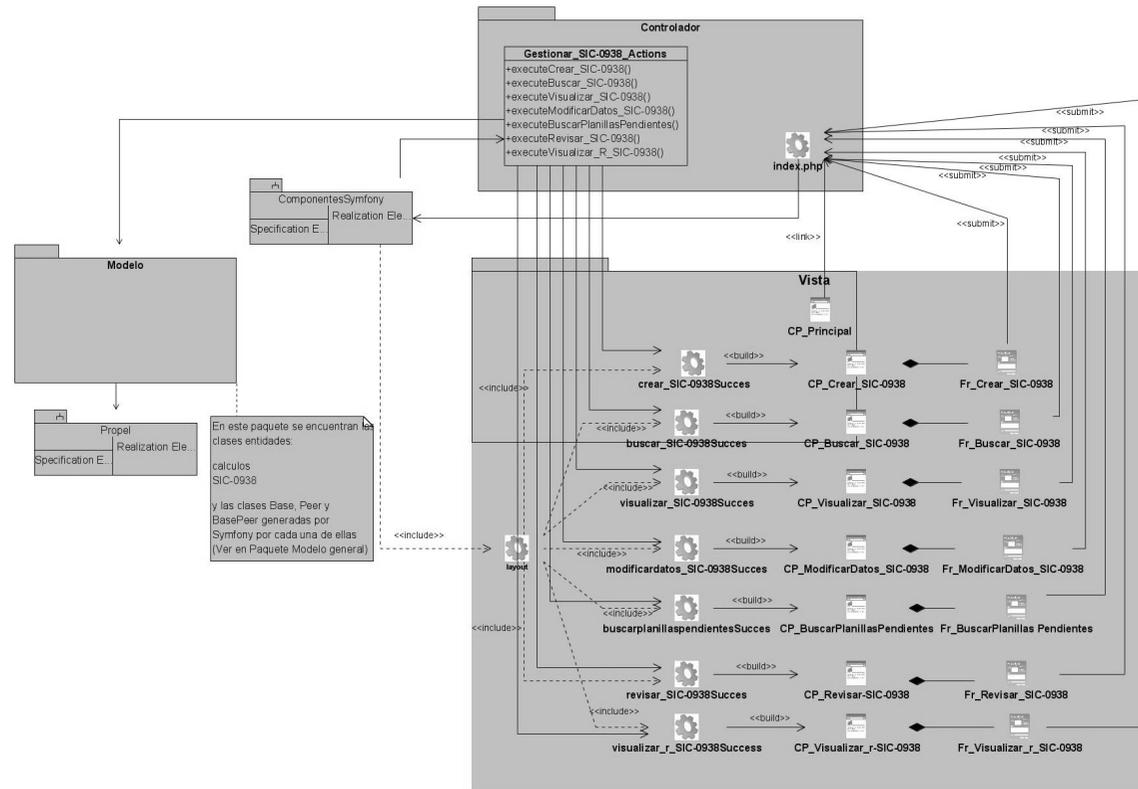


Fig. 31 Diagrama de clases del diseño CU Gestionar Cálculo de la actividad biológica por bulbo

CU Gestionar Notificación de resultados no satisfactorios

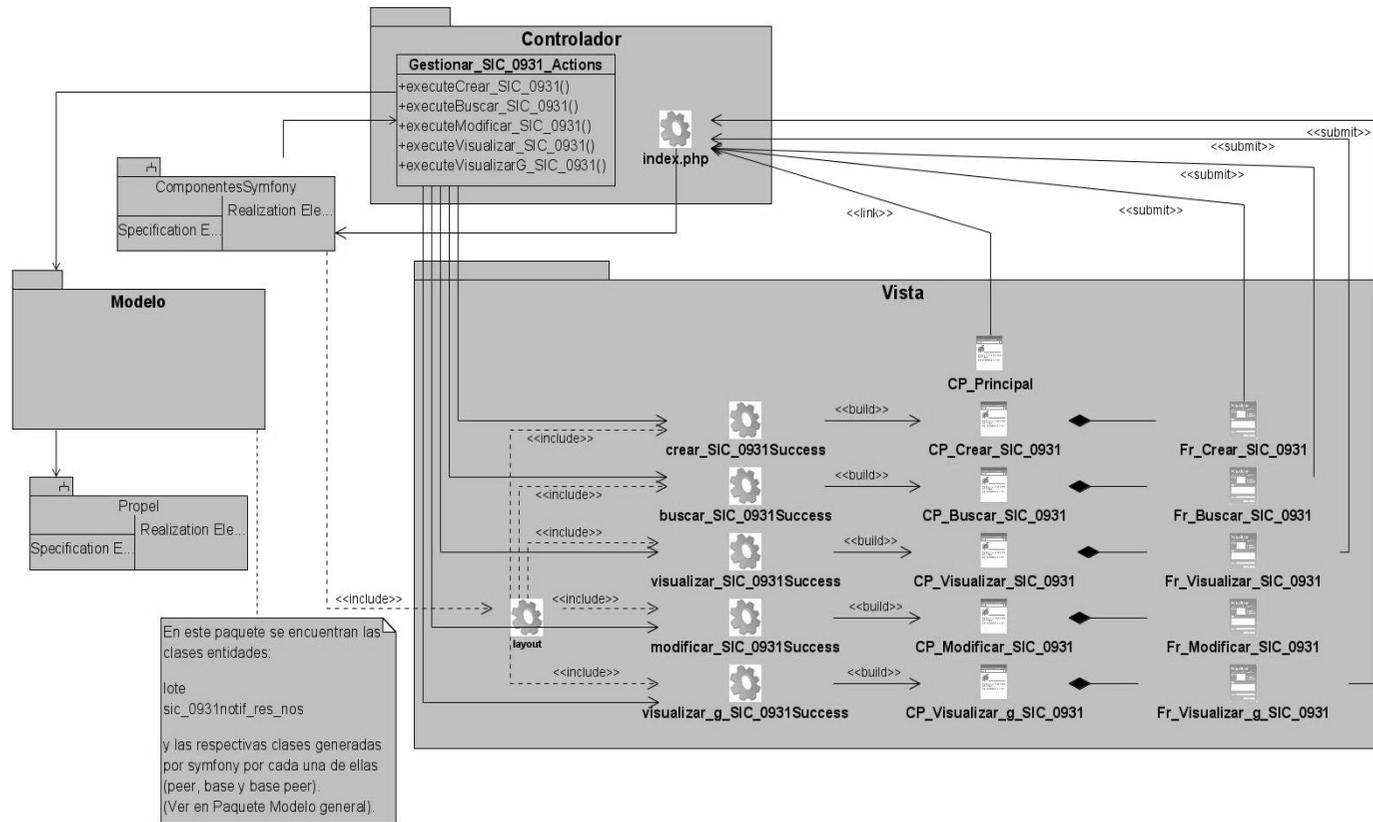


Fig. 32 Diagrama de clases del diseño CU Gestionar Notificación de resultados no satisfactorios

CU Gestionar Informe de Análisis

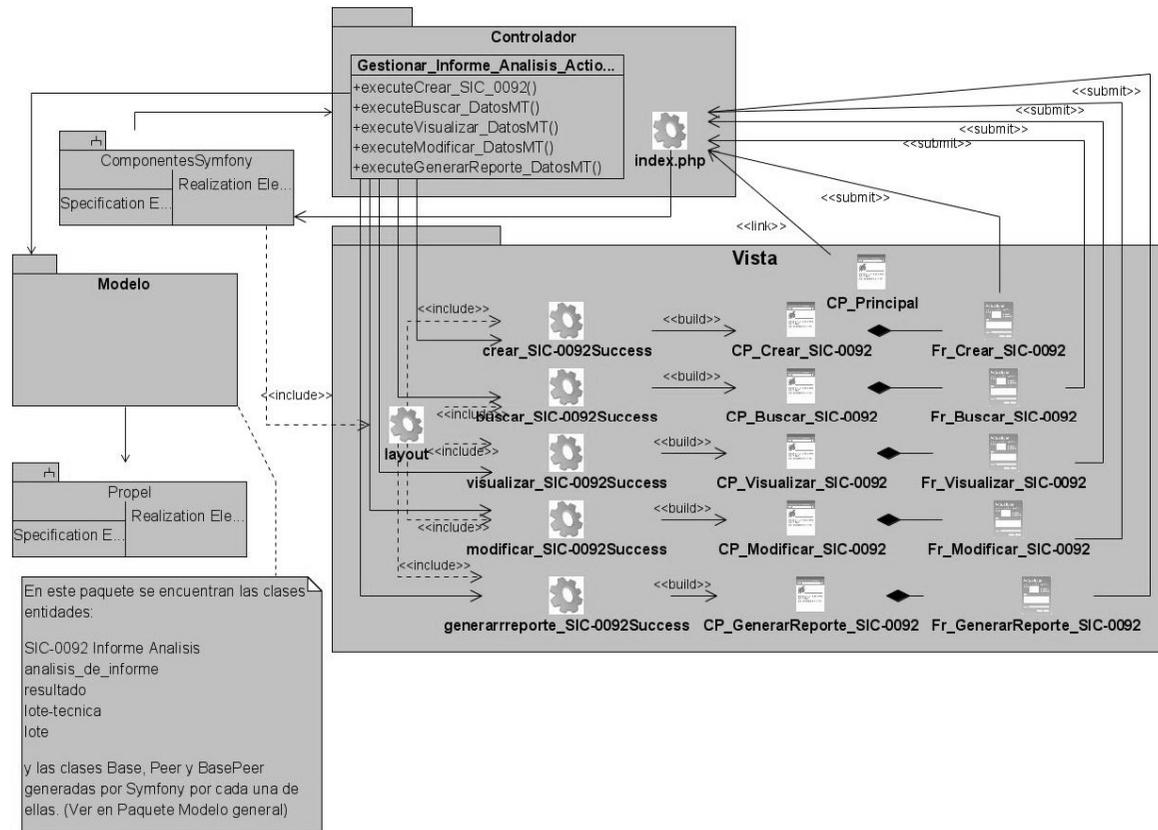


Fig. 33 Diagrama de clases del diseño CU Gestionar Informe de Análisis

CU Gestionar Certificado de Análisis del Ingrediente Farmacéutico

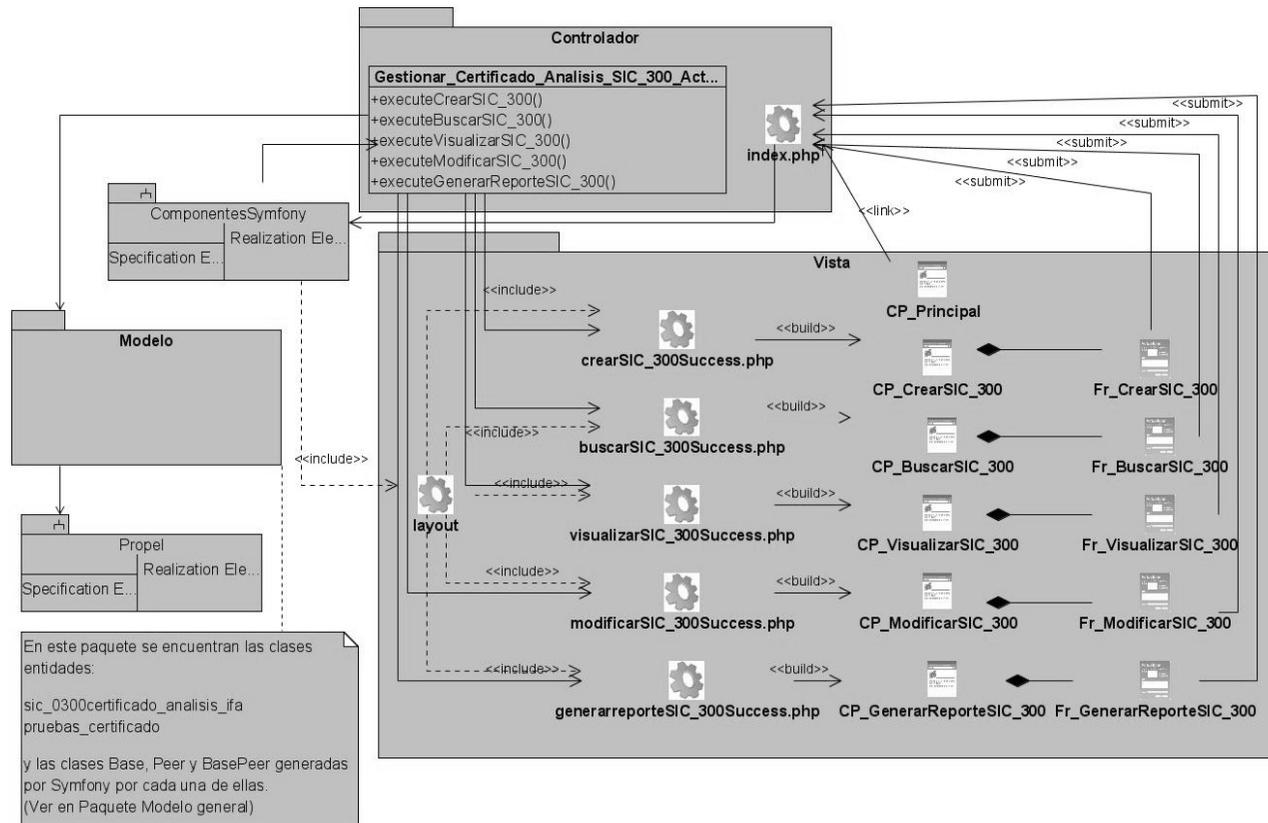


Fig. 34 Diagrama de clases del diseño CU Gestionar certificado de análisis del ingrediente farmacéutico

CU Gestionar Certificado de Análisis del Producto Final

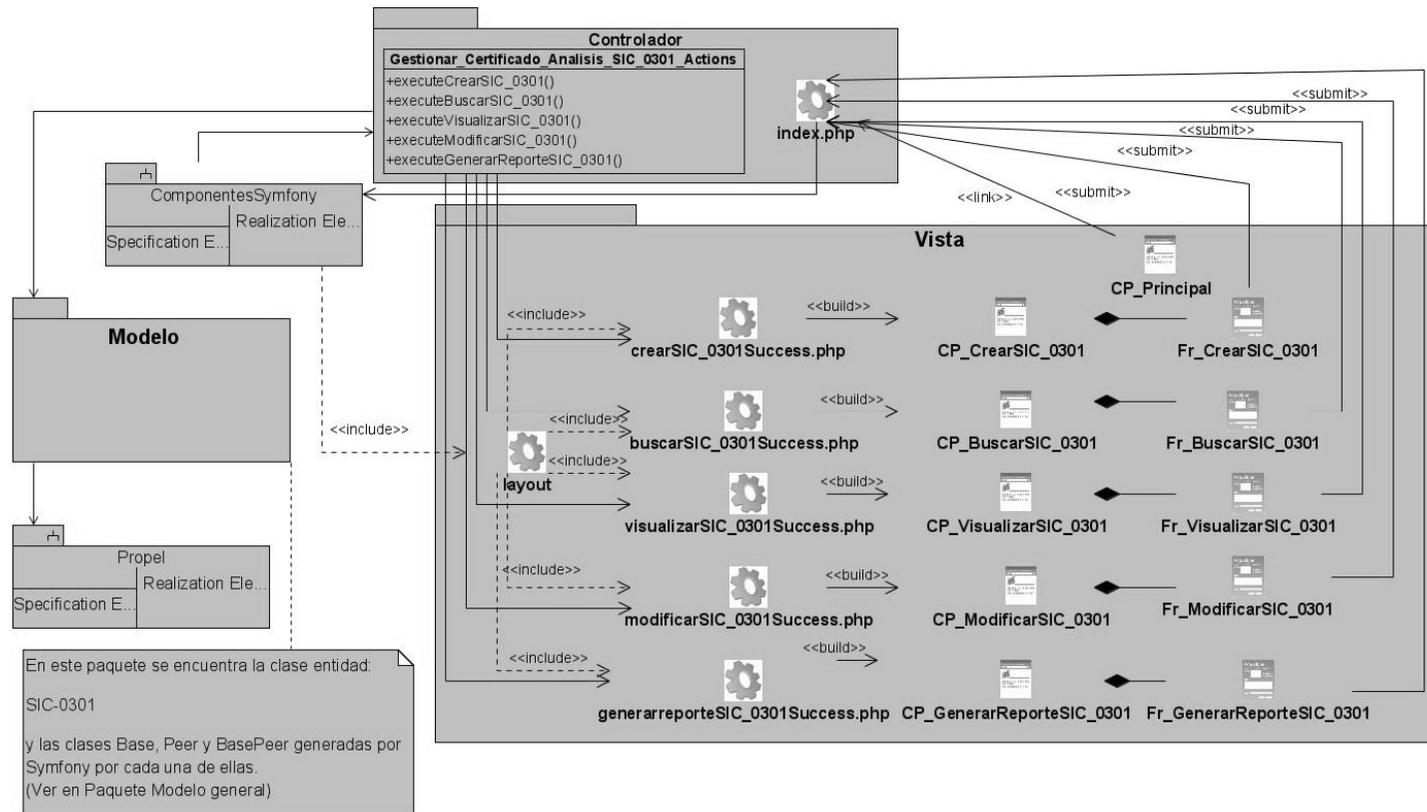


Fig.35 Diagrama de clases del diseño CU Gestionar certificado de análisis del producto final

CU Gestionar Documento que avala la liberación para la comercialización

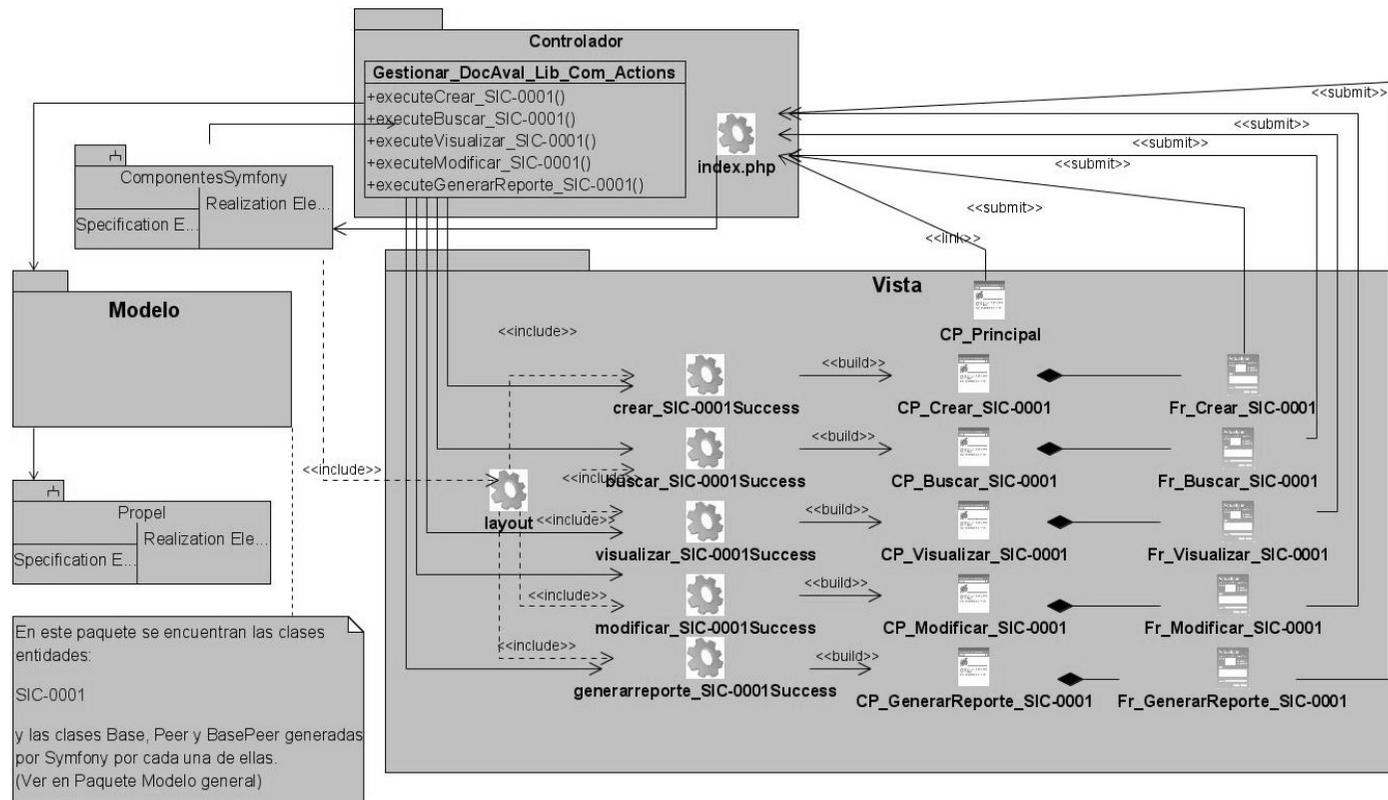


Fig. 36 Diagrama de clases del diseño CU Gestionar documento que avala la liberación para la comercialización

CU Gestionar Informe analítico semanal de resultados

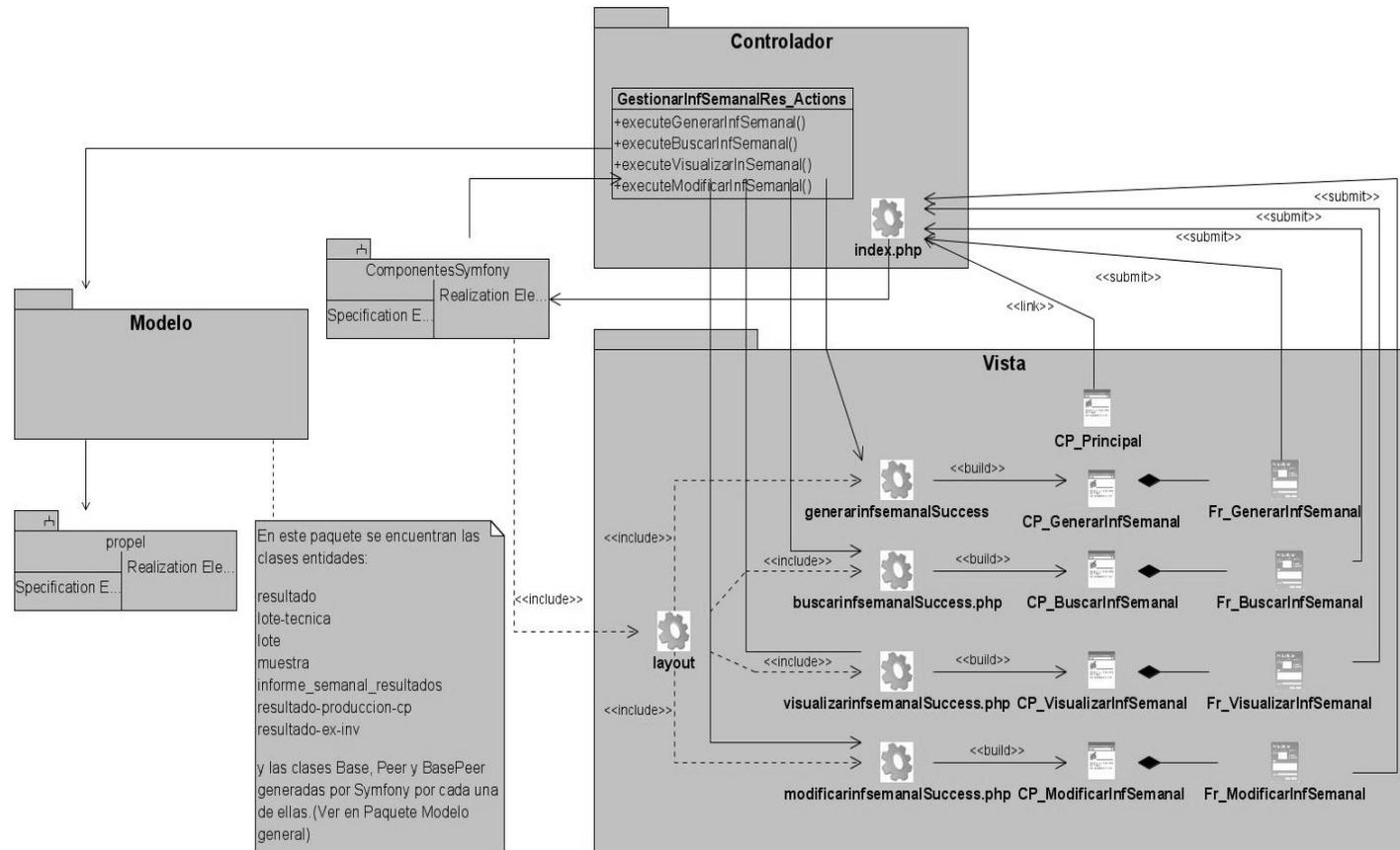


Fig. 37 Diagrama de clases del diseño CU Gestionar informe analítico semanal de resultados

CU Generar reportes generales

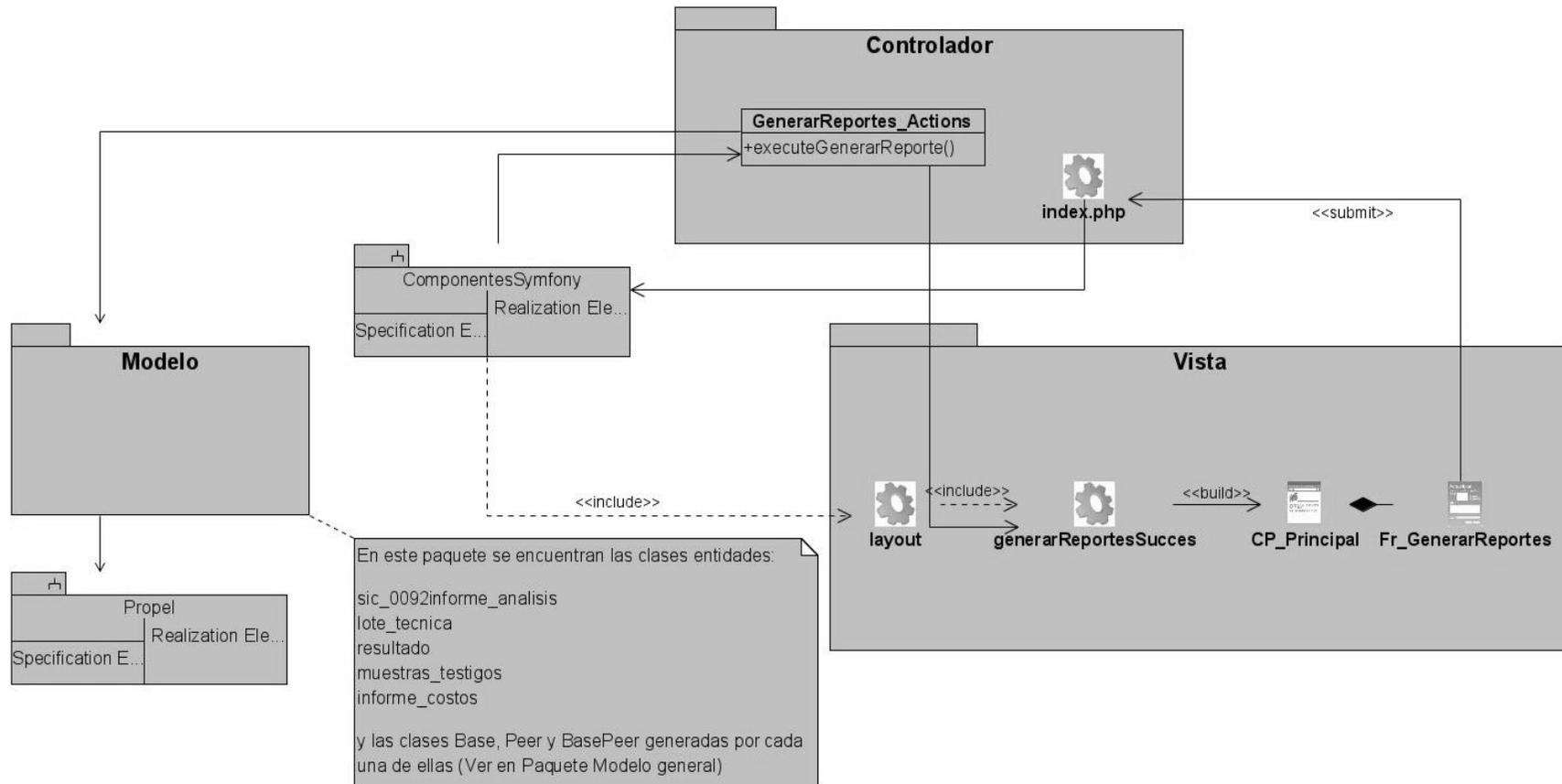


Fig. 38 Diagrama de clases del diseño CU Generar reportes generales

CU Gestionar informe de costos mensuales

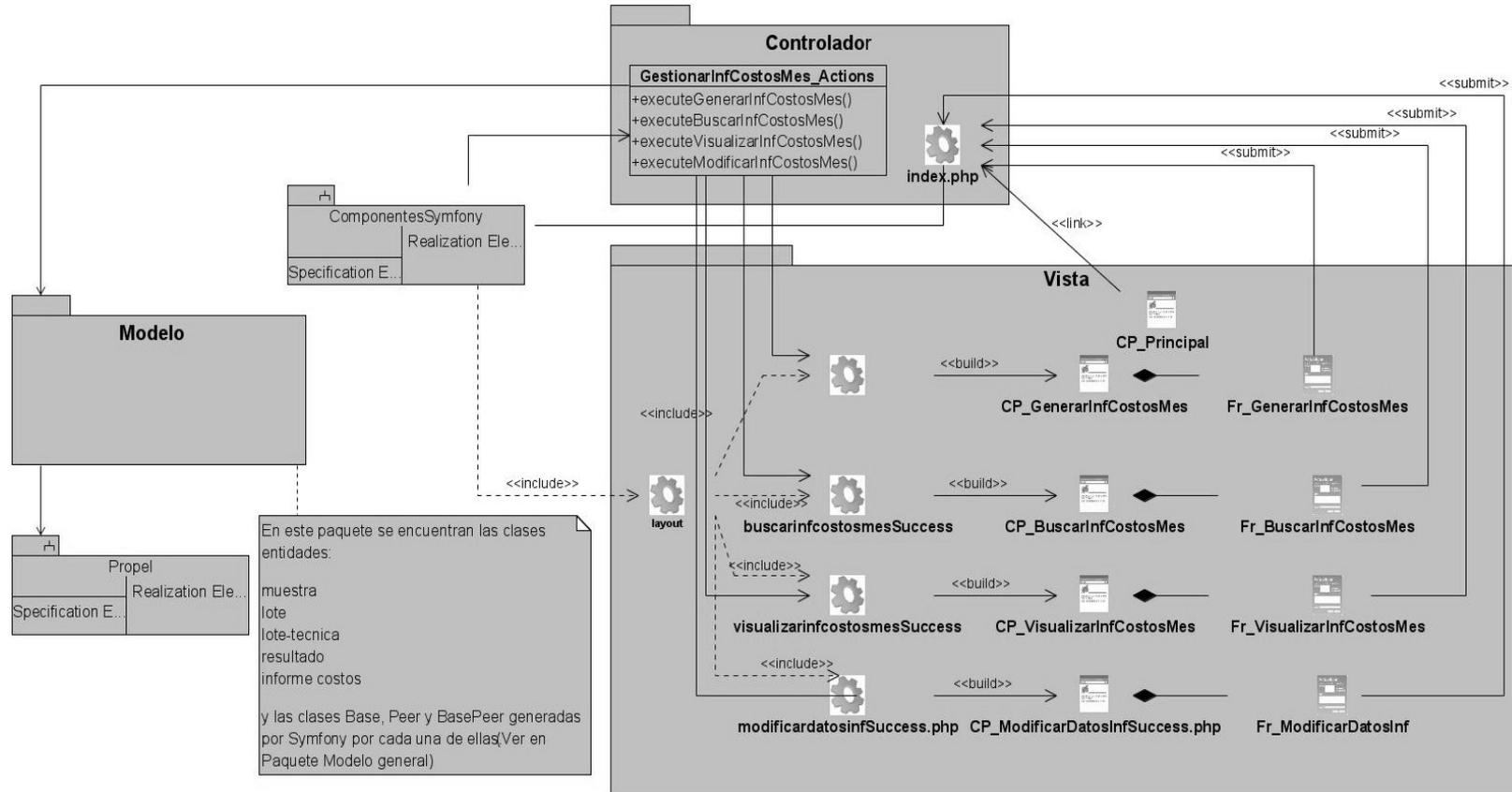


Fig. 39 Diagrama de clases del diseño CU Gestionar informe de costos mensuales

CU Gestionar resumen semestral de productos

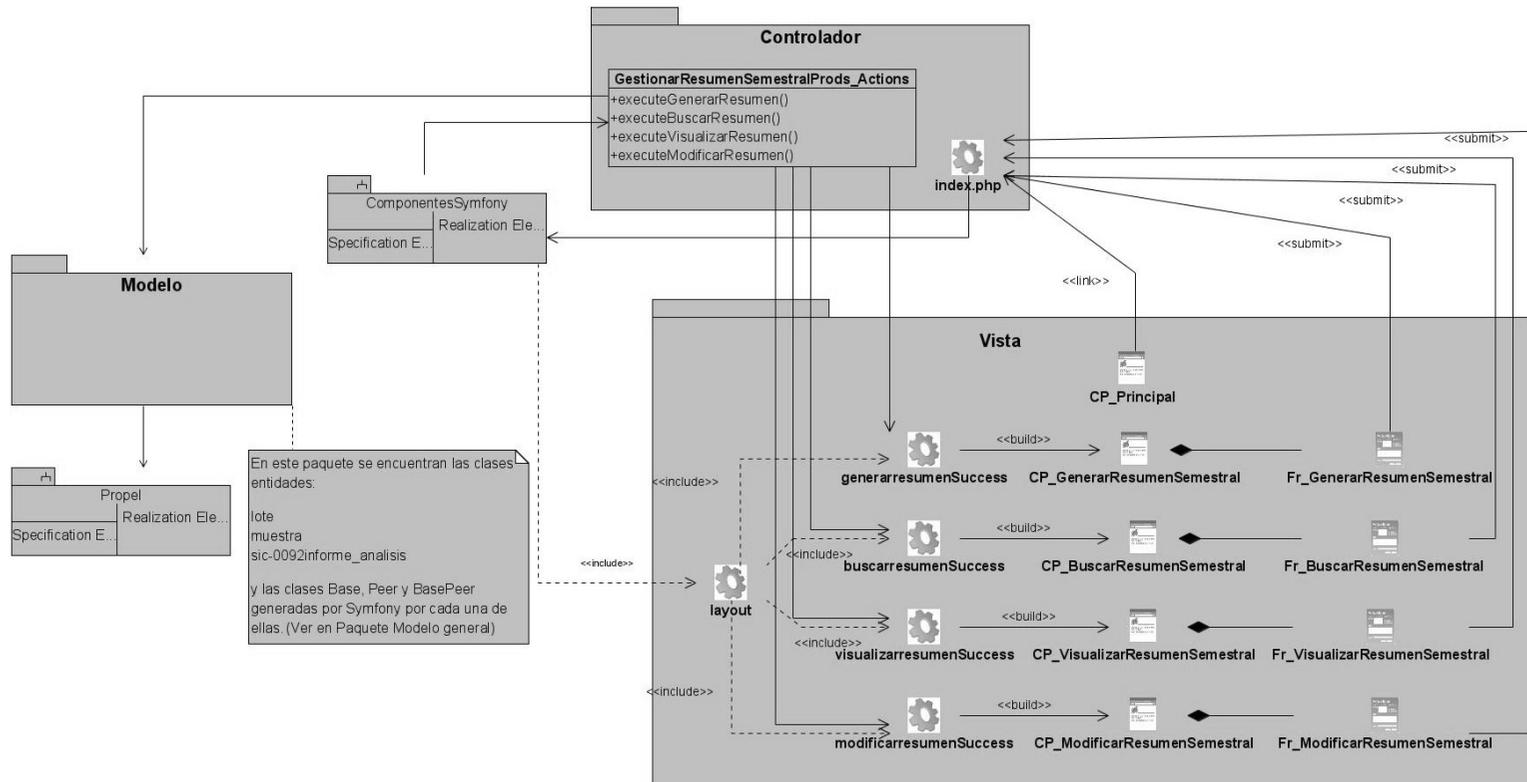


Fig. 40 Diagrama de clases del diseño CU Gestionar resumen semestral de productos

## 2.5 Diagramas de secuencia

Para lograr hacer una modelación de forma efectiva y que el resultado de la misma sea comprensible, son muy útiles los Mecanismos de Diseño, artefacto propuesto y descrito en RUP 2003, entre sus beneficios están:

- Mantener la homogeneidad en el diseño
- Reutilizar soluciones anteriormente probadas
- Reutilizar documentación

En este trabajo se ve evidenciada la ventaja de reutilizar documentación. A continuación se muestra y explica un diagrama de secuencia con el objetivo de documentar de manera global el proyecto.

La línea es la siguiente:

El actor solicita en la página principal (CP\_Principal), la primera que se muestra luego de haberse autenticado, crear libros de producto, dicha página como resultado hace un link al Index.php (controlador frontal) enviándole el nombre del módulo (Manipular Libro Producto) y la acción (CrearLibro()). Posteriormente Index.php le envía esos datos al componente de Symfony (sfController), que es el encargado de verificar la seguridad del sistema y validar los formularios e indicarle a la clase Actions cual es la acción que debe ejecutar. Dicha clase, de acuerdo a la acción indicada, hace llamadas a los métodos en las correspondientes clases peer (origenPeer y libroPeer) que consultan o almacenan datos en la Base de Datos. Posteriormente la Actions envía a la Success (crear\_libroSuccess) los datos requeridos para con ellos construir la Client Page (CP\_CrearLibro), la que muestra a través del formulario los datos necesarios de acuerdo a la solicitud del actor.

Los mensajes que se envían al controlador frontal (Index.php), y los que son enviados por este al componente de Symfony (sfController), así como las validaciones que se efectúan en el mismo, para posteriormente enviar a la clase Actions la acción que debe ejecutar; van a ser los mismos en todos los diagramas. Debido a la gran cantidad de objetos y clases involucradas en cada diagrama, se decide omitir este fragmento, para no sobrecargarlos de información redundante (mensajes y objetos que siempre serán los mismos).

CU Manipular libro producto escenario Crear Libro

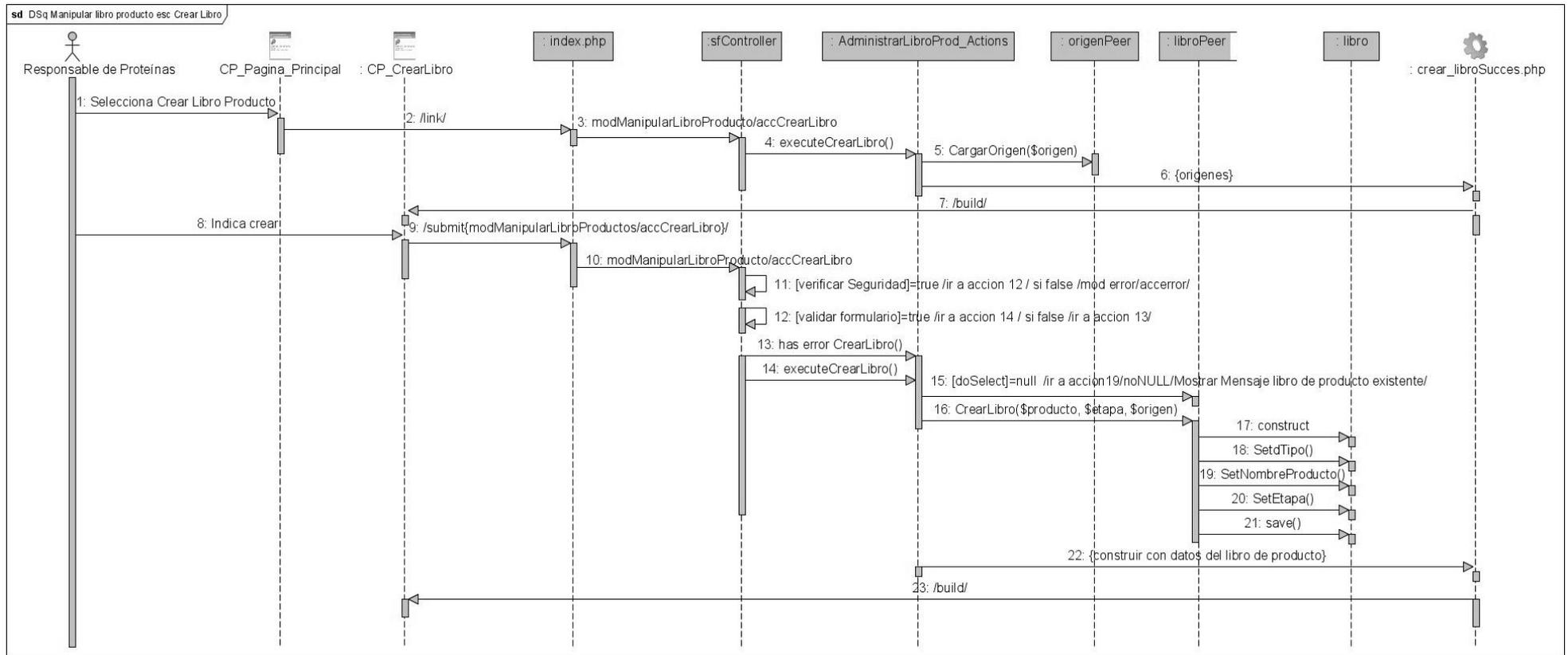


Fig. 41 Diagrama de secuencia CU Manipular libro de productos escenario Crear Libro

El diagrama correspondiente al caso de uso Manipular libro de productos escenario crear libro, quedaría de la siguiente forma:

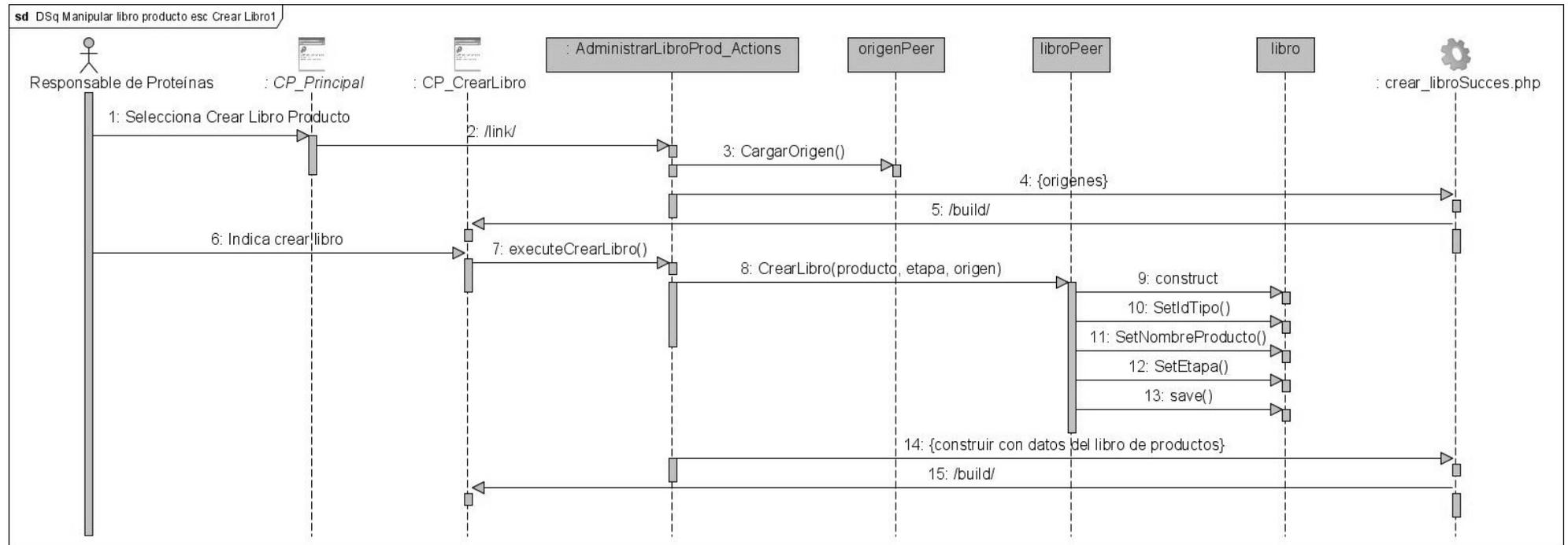


Fig. 42 Diagrama de secuencia resumido CU Manipular libro de productos escenario Crear Libro

Diagramas de Secuencia CU Manipular libro de productos

CU Manipular libro producto escenario Buscar-Visualizar Libro

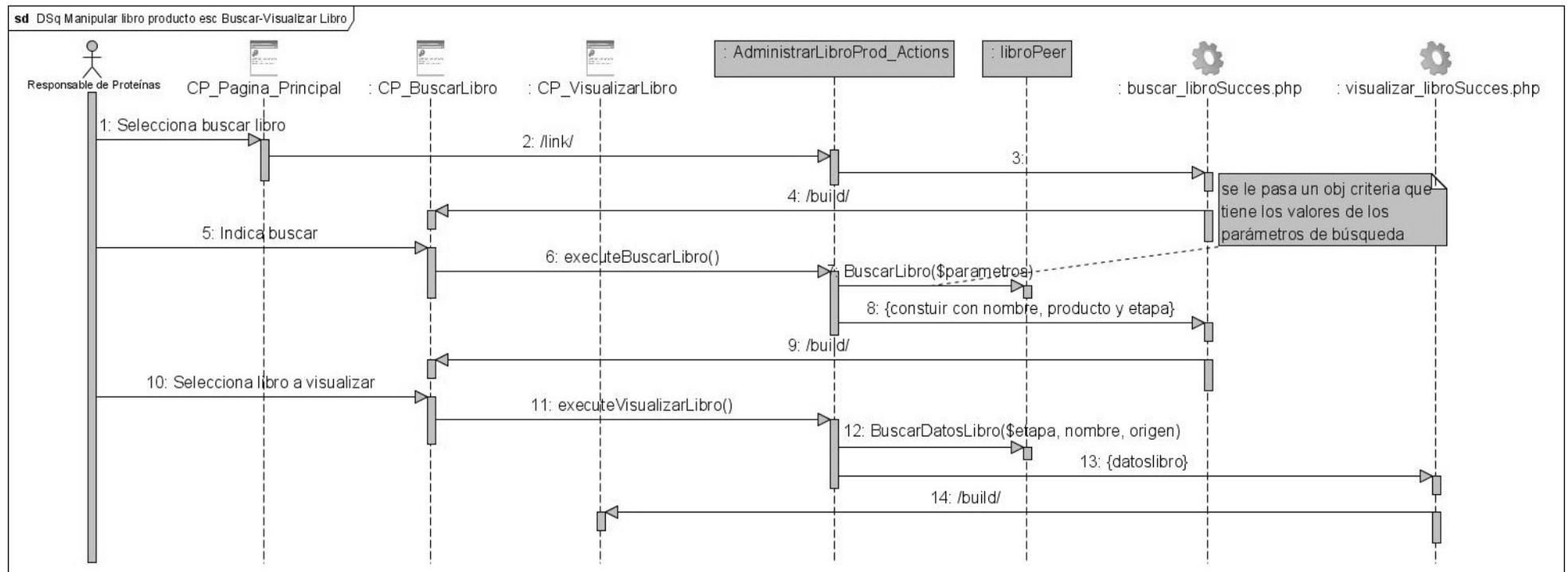


Fig. 43 Diagrama de secuencia CU Manipular libro de productos escenario Buscar-Visualizar Libro

CU Manipular libro producto escenario Modificar Libro

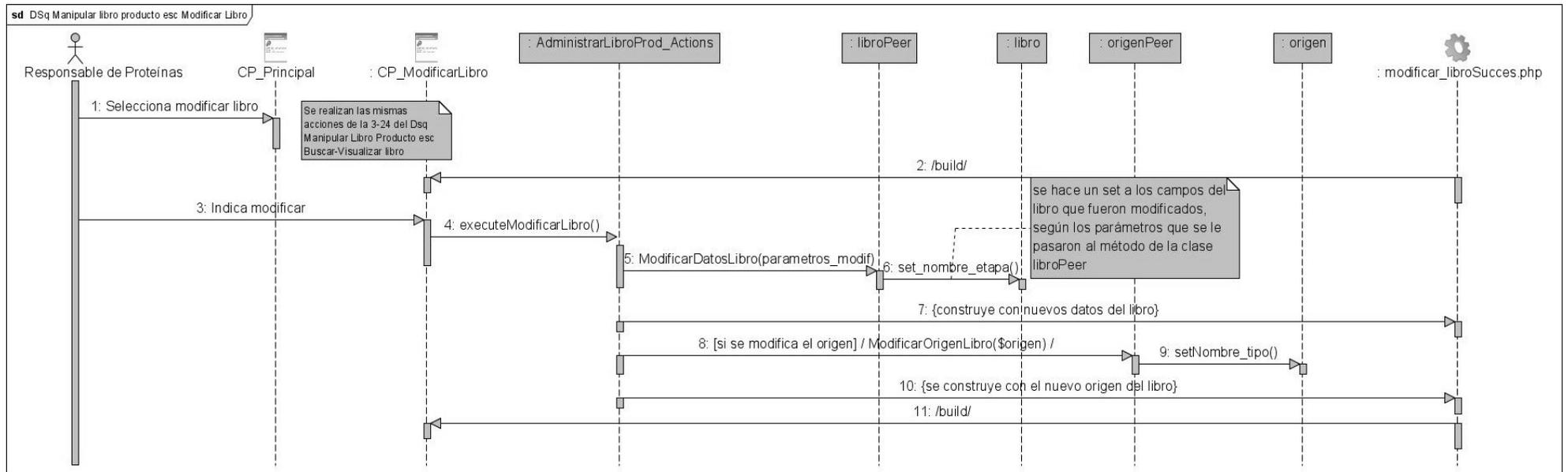


Fig. 44 Diagrama de secuencia CU Manipular libro de productos escenario Modificar Libro

CU Manipular libro producto escenario Generar Reporte

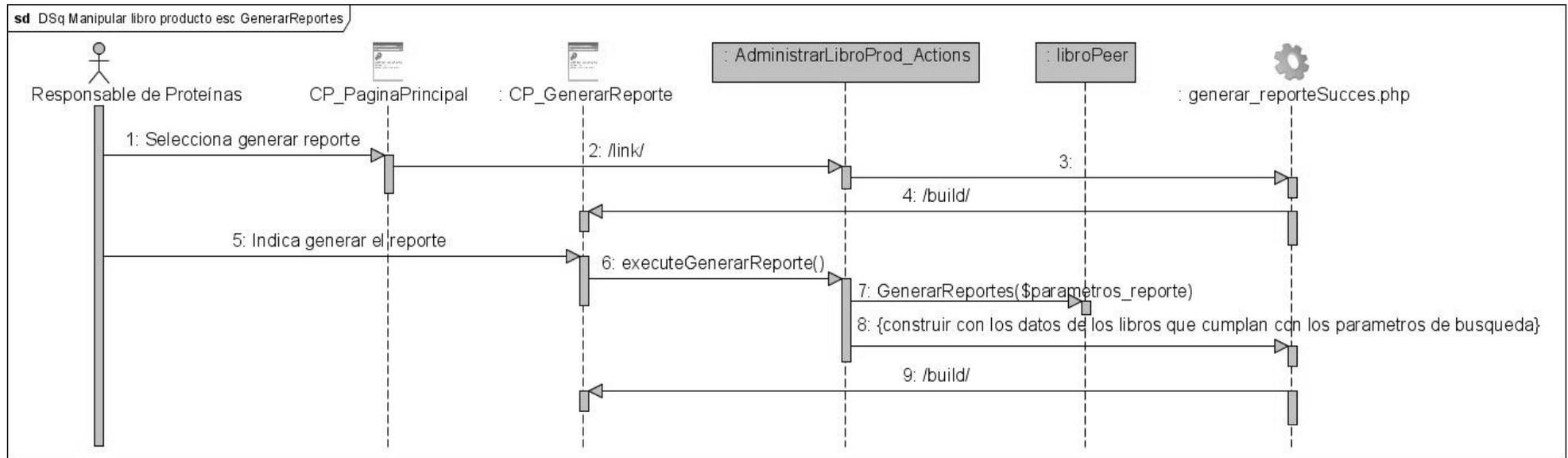


Fig. 45 Diagrama de secuencia CU Manipular libro de productos escenario Generar Reporte

Diagramas de Secuencia CU Manipular libro de centros

CU Manipular libro de centros escenario Crear libro

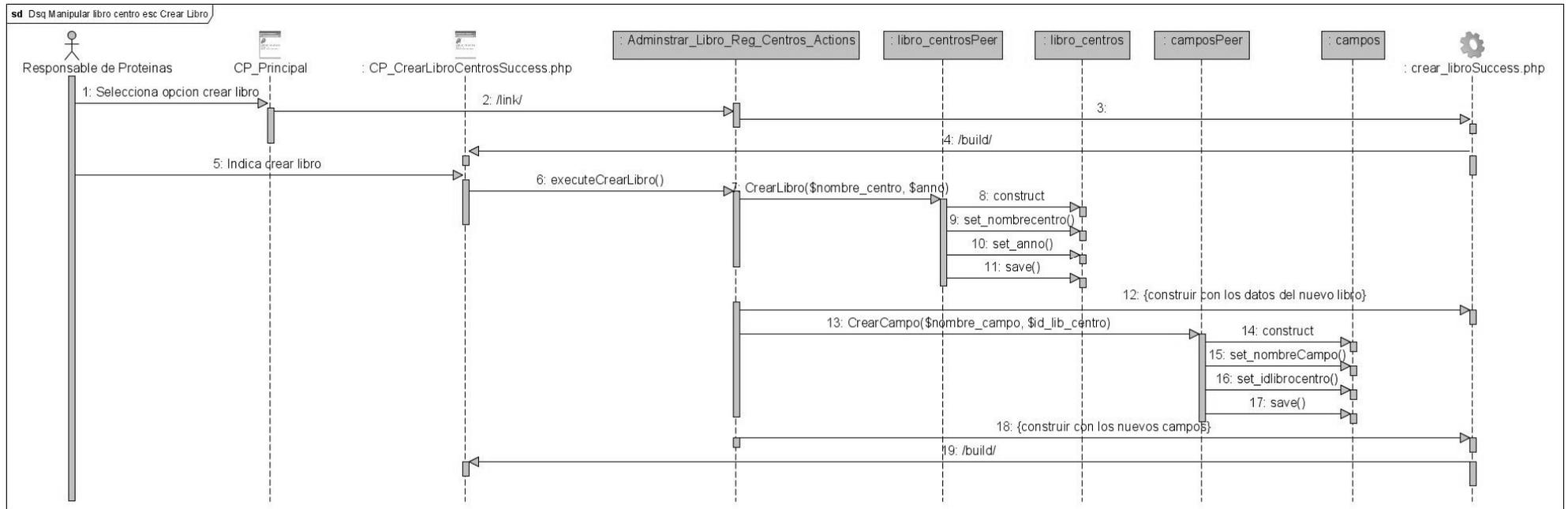


Fig. 46 Diagrama de secuencia CU Manipular libro de centros escenario Crear Libro

CU Manipular libro de centros escenario Insertar Datos

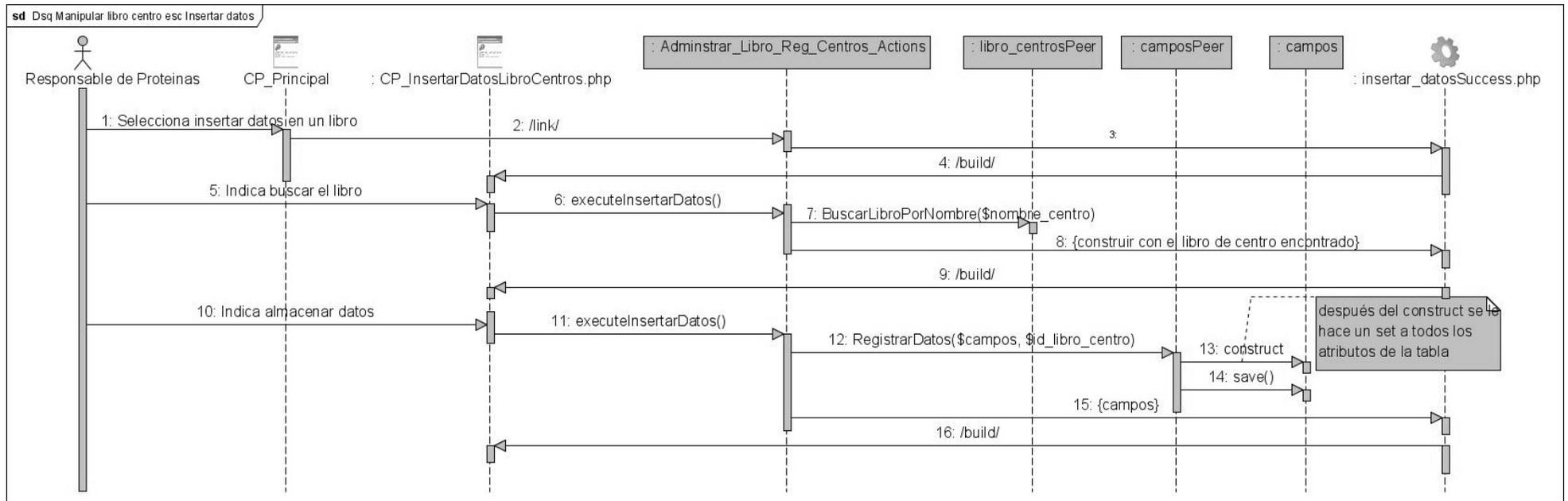


Fig. 47 Diagrama de secuencia CU Manipular libro de centros escenario Insertar Datos

CU Manipular libro de centros escenario Buscar-Visualizar libro

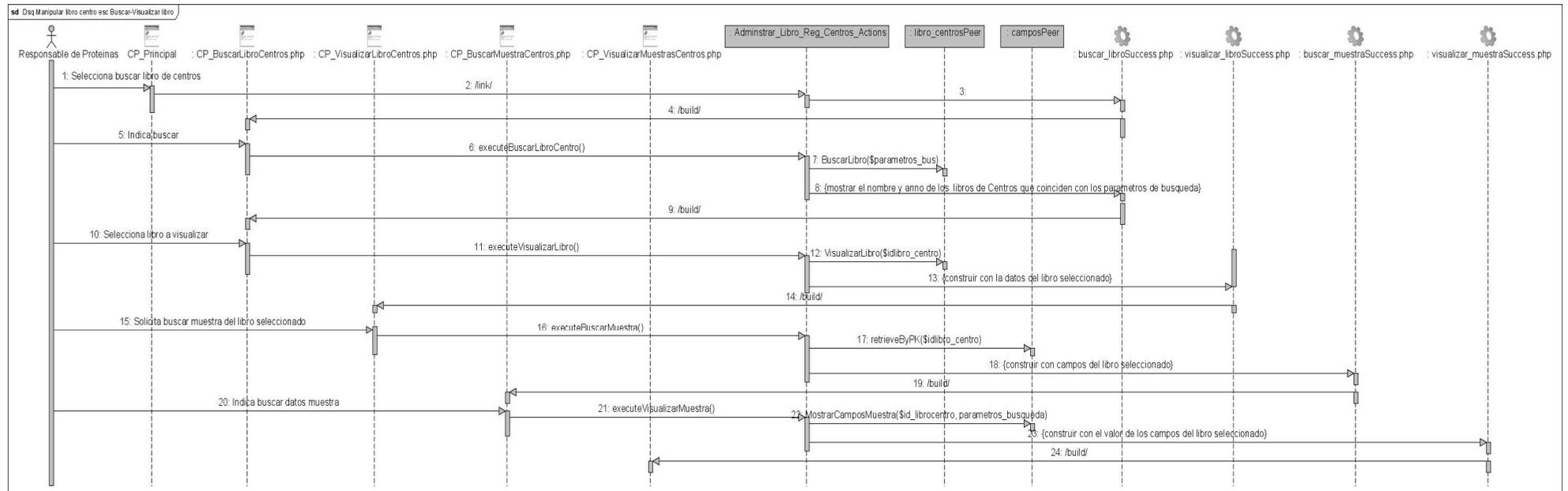


Fig. 48 Diagrama de secuencia CU Manipular libro de centros escenario Buscar-Visualizar Libro

CU Manipular libro de centros escenario Modificar libro

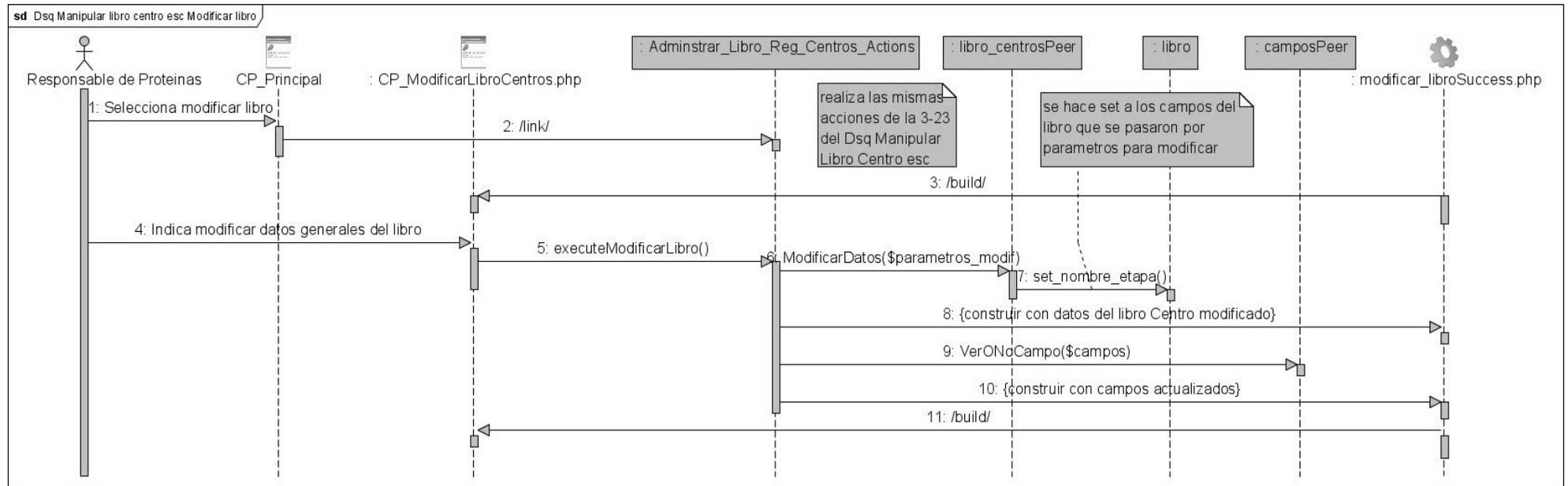


Fig. 49 Diagrama de secuencia CU Manipular libro de centros escenario Modificar Libro

CU Manipular libro de centros escenario Generar Reporte

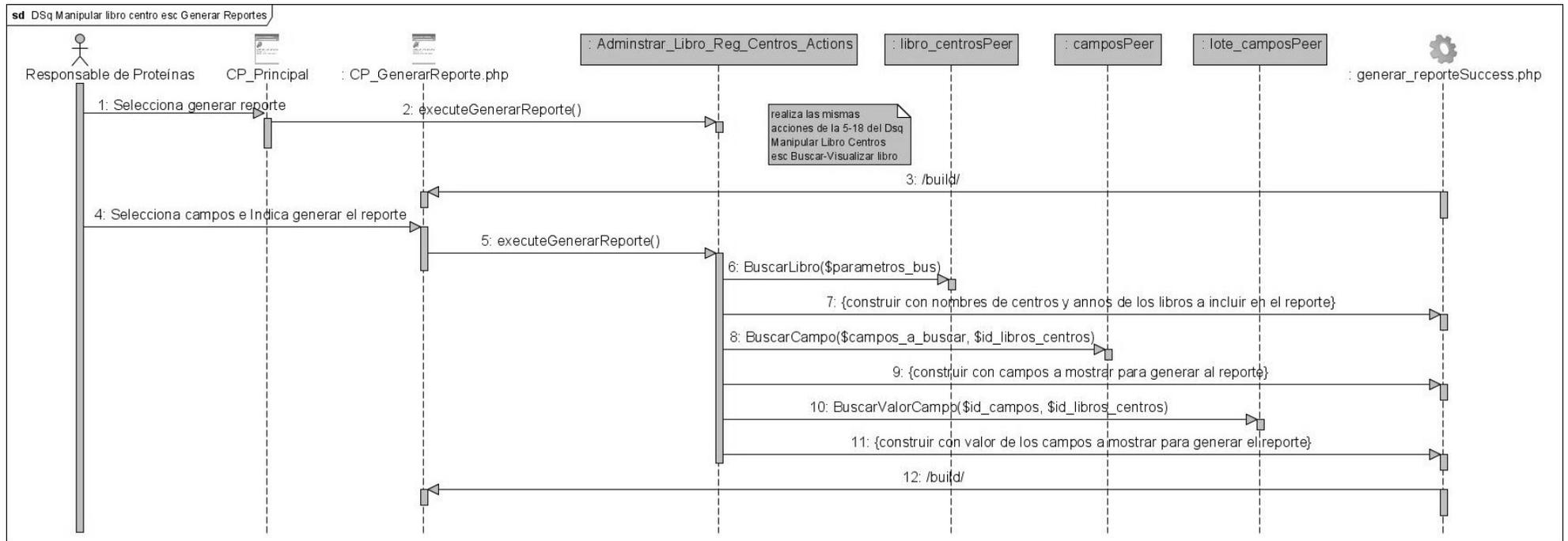


Fig. 50 Diagrama de secuencia CU Manipular libro de centros escenario Generar Reporte

Diagramas de Secuencia del CU Gestionar muestras

Ver en Expediente de Proyecto los diagramas de secuencia correspondientes a los escenarios Registrar Muestra, Buscar-Visualizar muestra y Buscar-Visualizar Expediente Analítico del CU Gestionar Muestras.

CU Gestionar muestras escenario Modificar Muestra

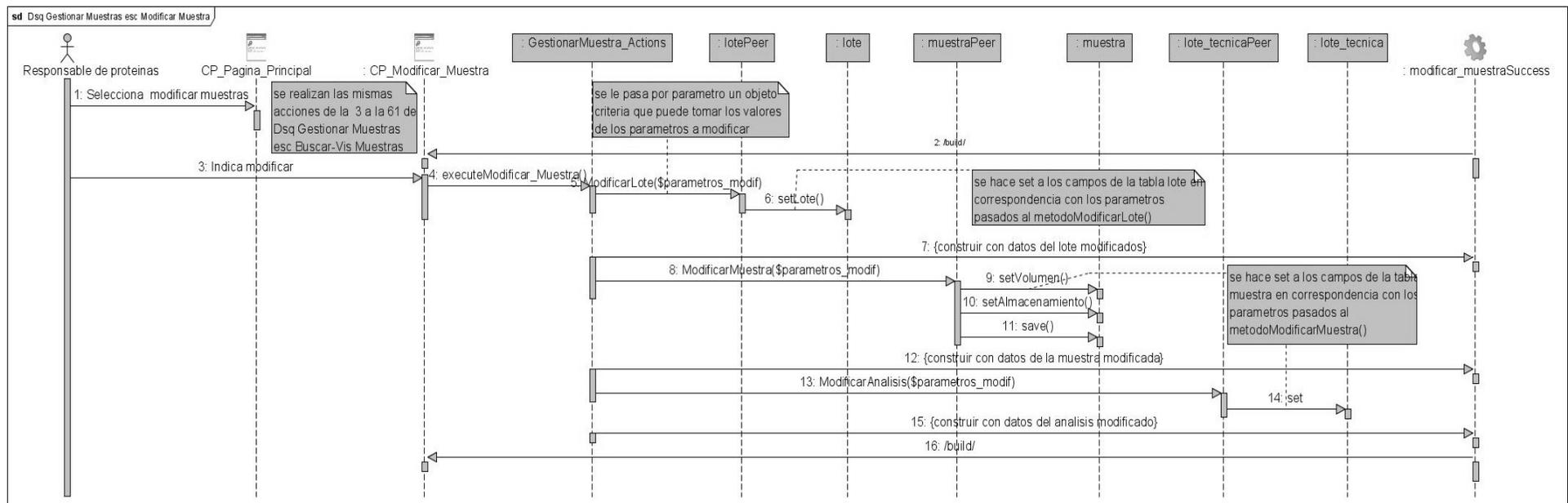


Fig. 51 Diagrama de secuencia CU Gestionar Muestras escenario Modificar Muestra

CU Gestionar muestras escenario Revisar Expediente Analítico

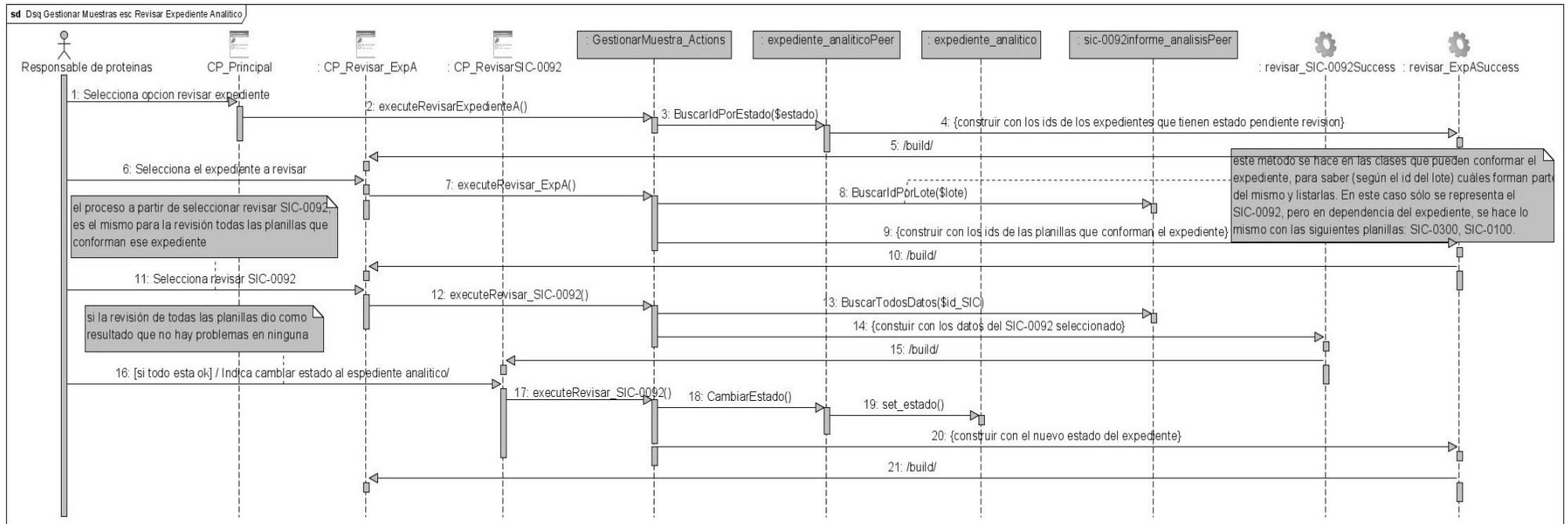


Fig. 52 Diagrama de secuencia CU Gestionar Muestras escenario Revisar Expediente Analítico

Diagramas de Secuencia del CU Gestionar Cálculo de la actividad específica

Ver en Expediente de Proyecto el diagrama de secuencia correspondiente al escenario Crear SIC-0709 del CU Gestionar Cálculo de la actividad específica.

CU Gestionar Cálculo de la actividad específica escenario Buscar-Visualizar SIC-0709

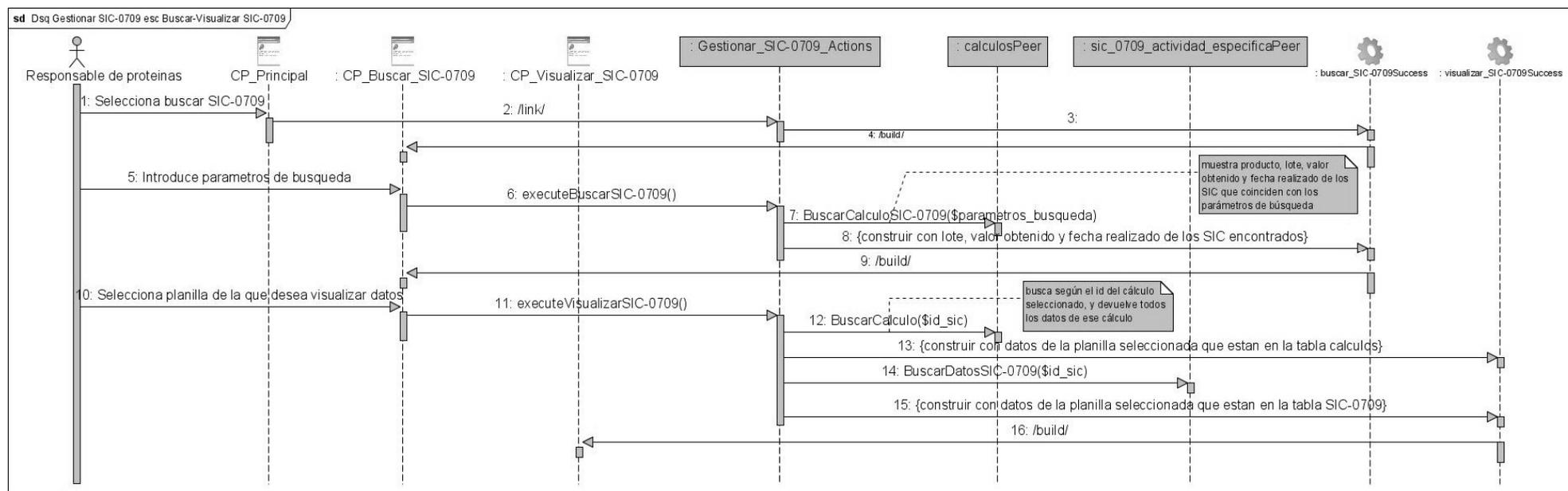


Fig. 53 Diagrama de secuencia CU Gestionar Cálculo de la actividad específica escenario Buscar-Visualizar SIC-0709

CU Gestionar Cálculo de la actividad específica escenario Modificar SIC-0709

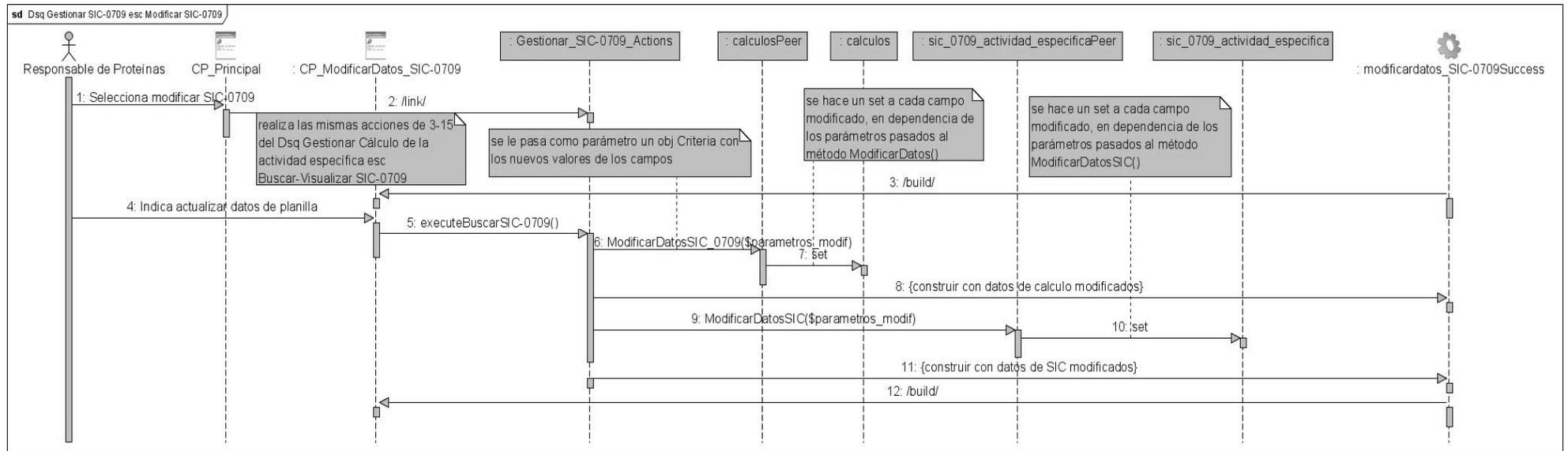


Fig. 54 Diagrama de secuencia CU Gestionar Cálculo de la actividad específica escenario Modificar SIC-0709

CU Gestionar Cálculo de la actividad específica escenario Revisar SIC-0709

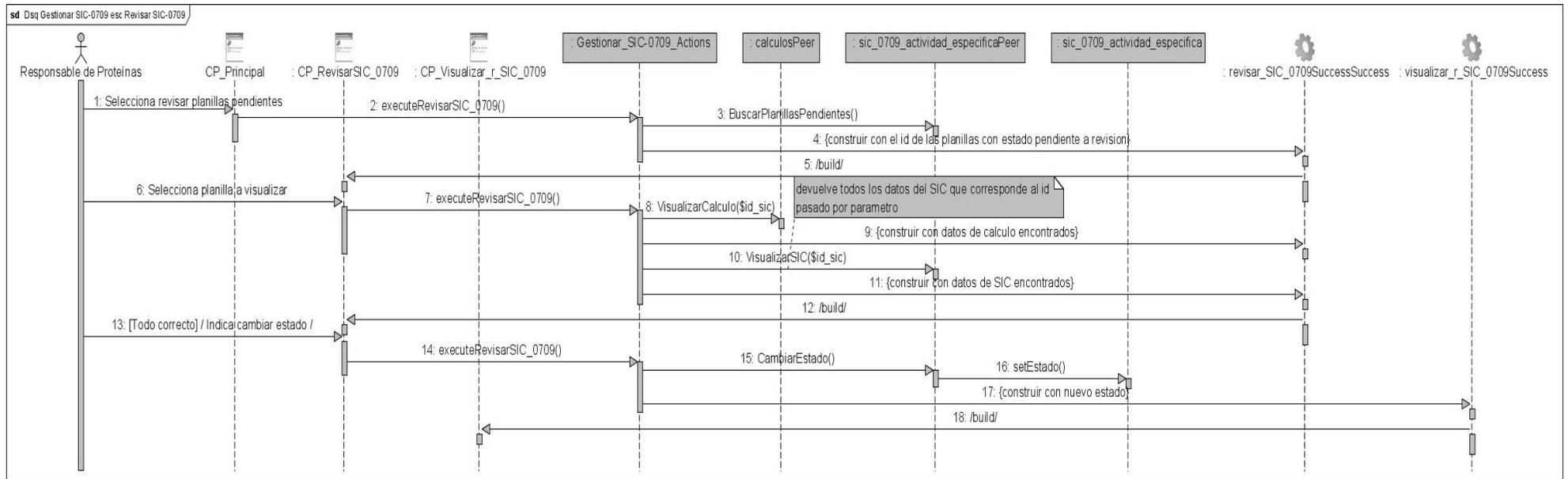


Fig. 55 Diagrama de secuencia CU Gestionar Cálculo de la actividad específica escenario Revisar SIC-0709

Diagramas de Secuencia CU Gestionar Solicitud de muestras testigo

CU Gestionar Solicitud de muestras testigo escenario Crear SIC-0935

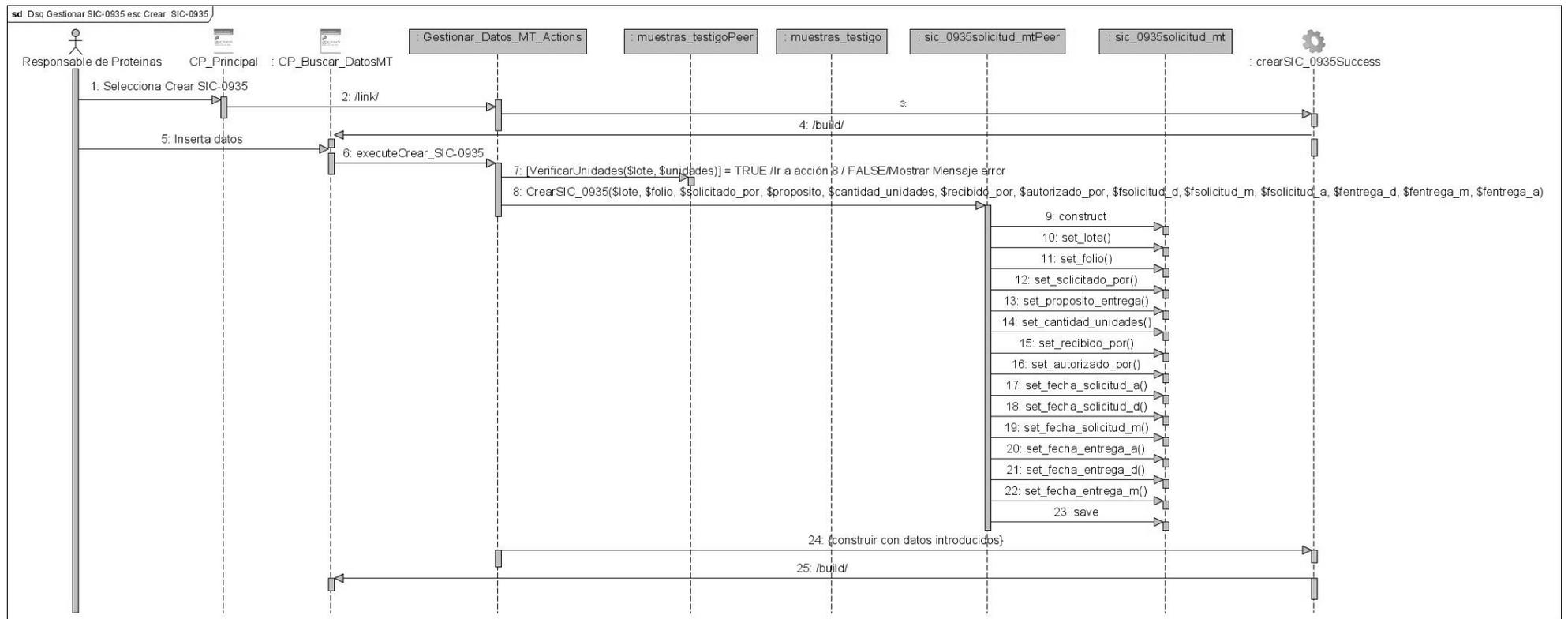


Fig. 56 Diagrama de secuencia CU Gestionar Solicitud de muestras testigo escenario Crear SIC-0935

CU Gestionar Solicitud de muestras testigo escenario Buscar-Visualizar SIC-0935

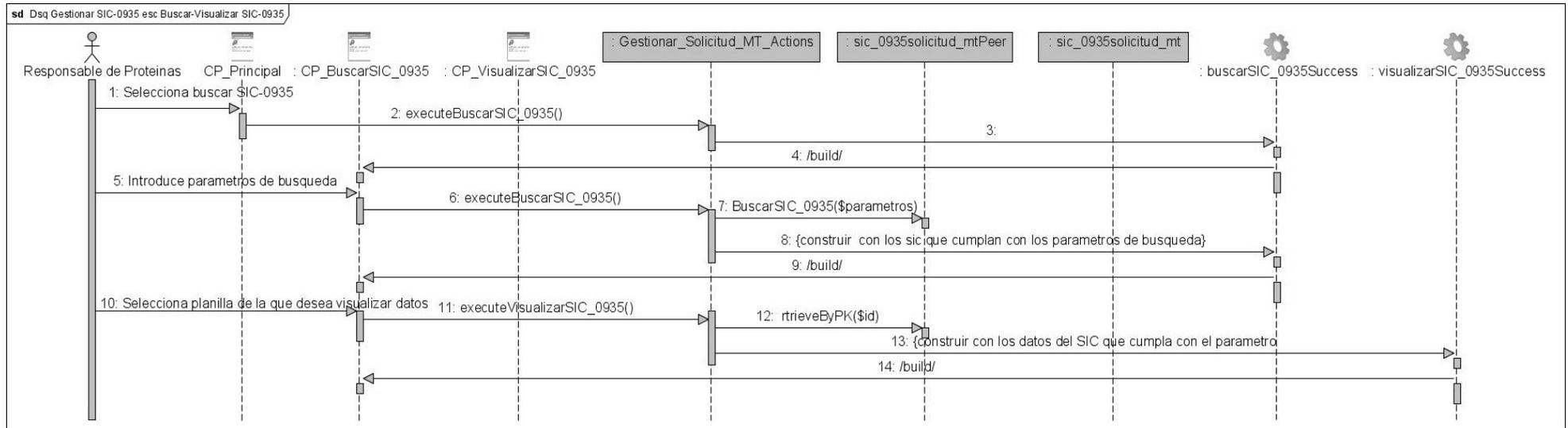


Fig. 57 Diagrama de secuencia CU Gestionar Solicitud de muestras testigo escenario Buscar-Visualizar SIC-0935

CU Gestionar Solicitud de muestras testigo escenario Modificar SIC-0935

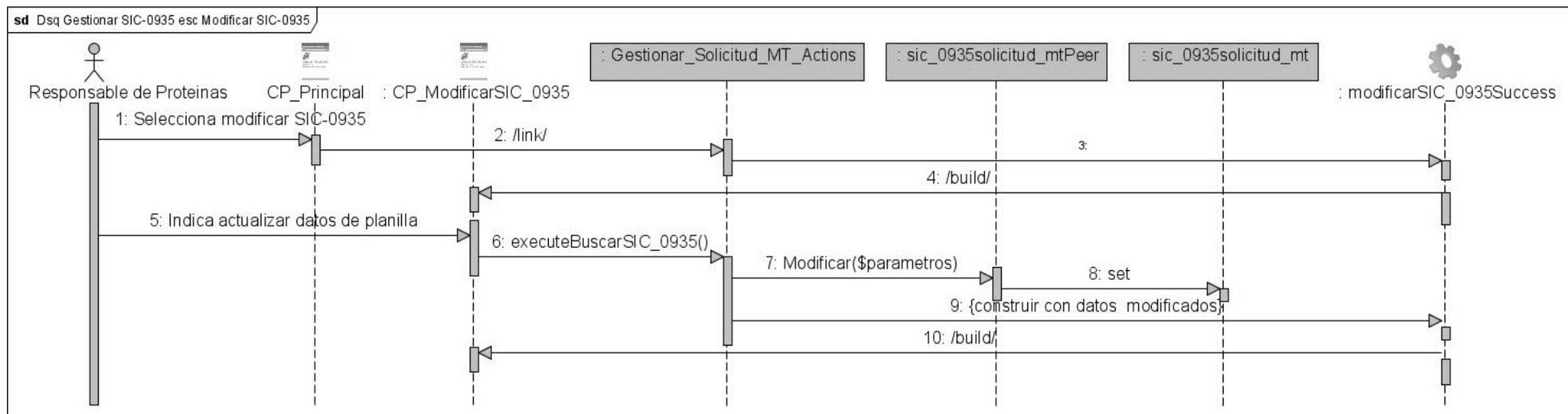


Fig. 58 Diagrama de secuencia CU Gestionar Solicitud de muestras testigo escenario Buscar-Visualizar SIC-0935

CU Gestionar Solicitud de muestras testigo escenario Generar Reporte SIC-0935

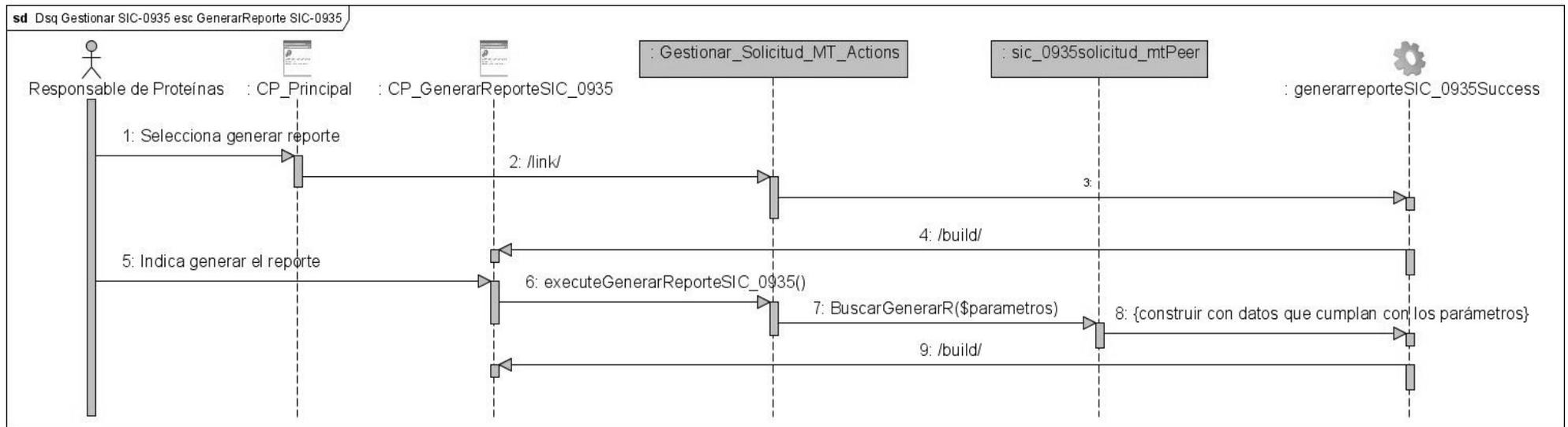


Fig. 59 Diagrama de secuencia CU Gestionar Solicitud de muestras testigo escenario Buscar-Visualizar SIC-0935

Diagramas de Secuencia CU Manipular libro registros muestras testigo por producto

CU Manipular libro registros muestras testigo por producto escenario Crear

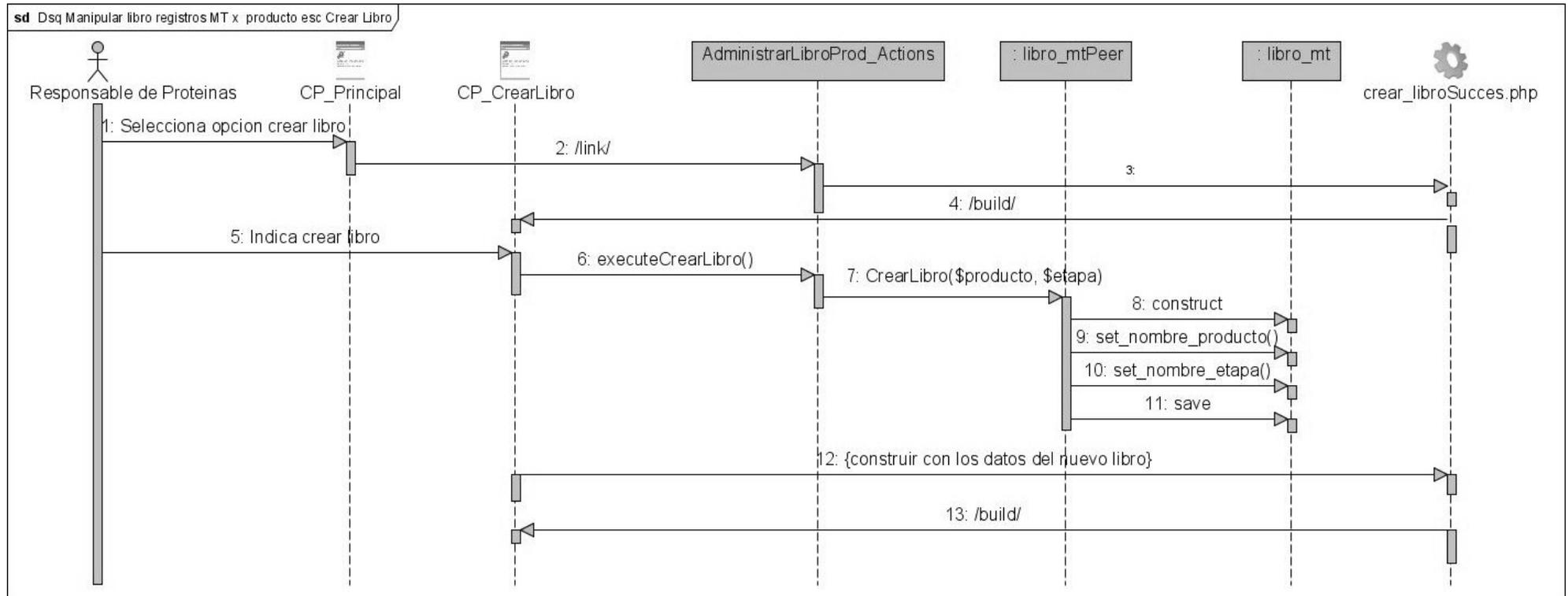


Fig. 60 Diagrama de secuencia CU Manipular libro registros muestras testigo por producto escenario Crear

CU Manipular libro registros muestras testigo por producto escenario Buscar-Visualizar

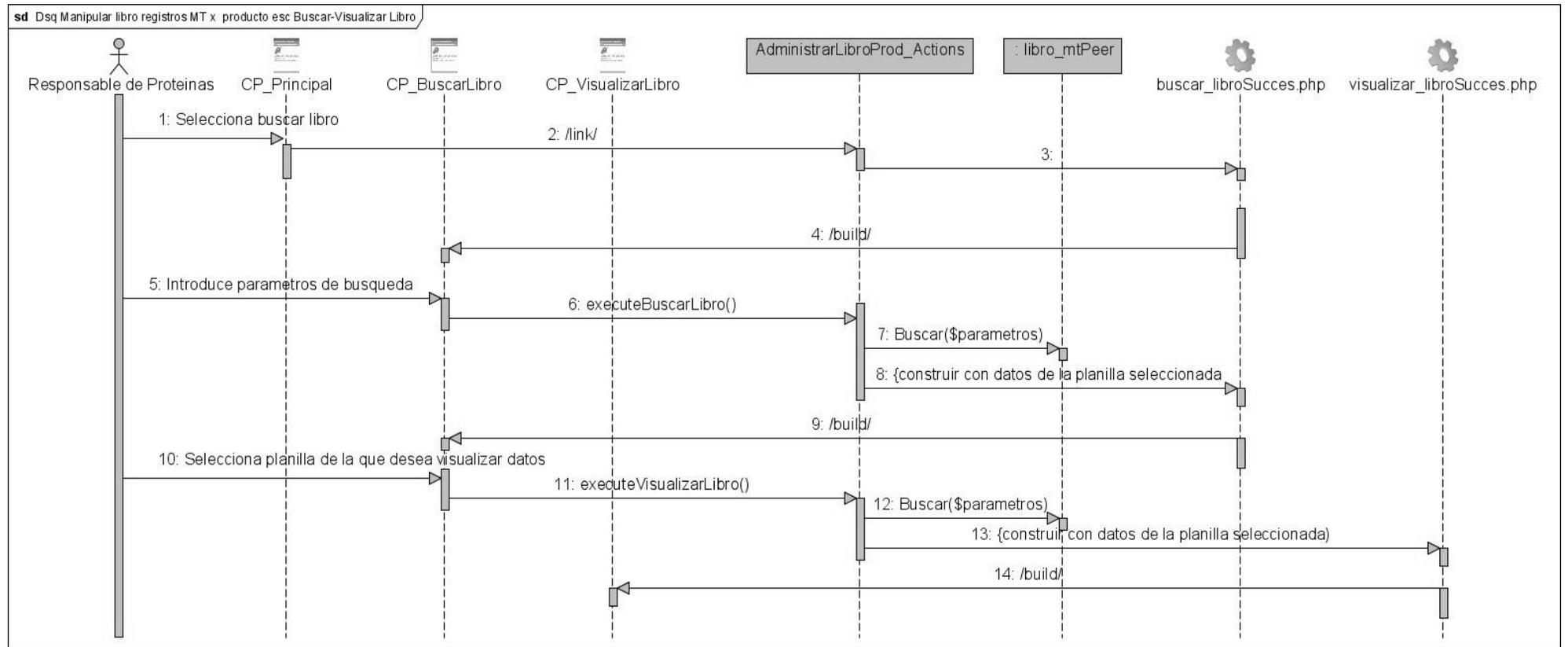


Fig. 61 Diagrama de secuencia CU Manipular libro registros muestras testigo por producto escenario Buscar-Visualizar

CU Manipular libro registros muestras testigo por producto escenario Modificar

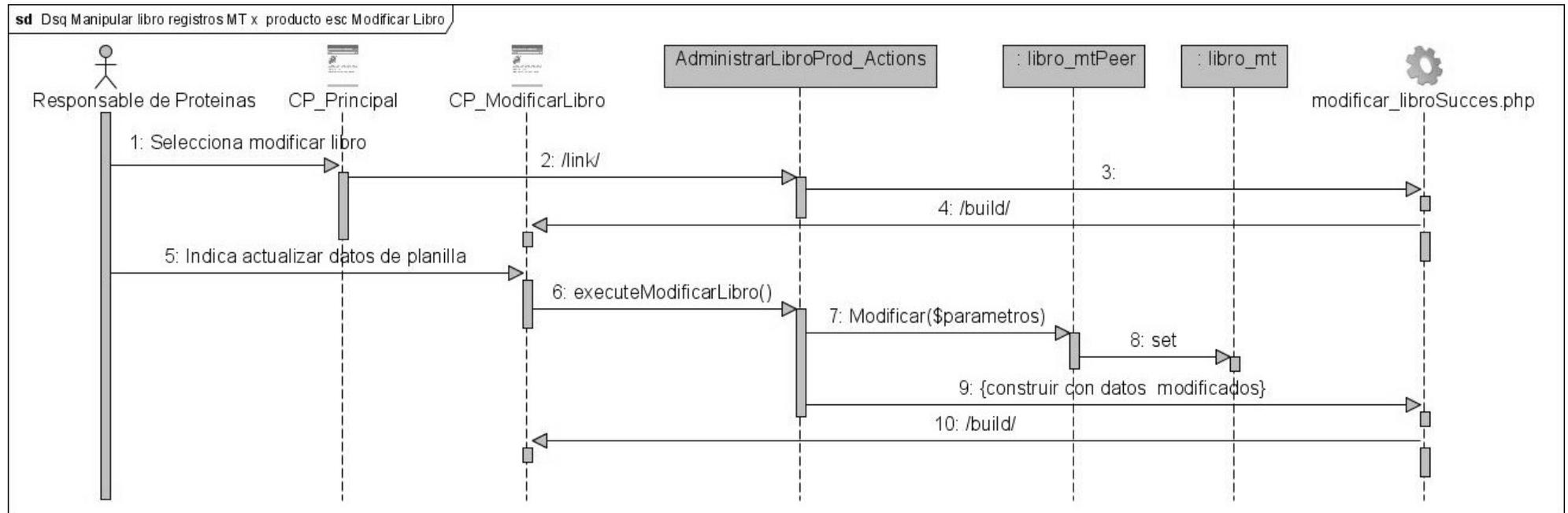


Fig. 62 Diagrama de secuencia CU Manipular libro registros muestras testigo por producto escenario Modificar

Diagramas de Secuencia CU Gestionar Certificado Análisis Producto Final

Ver en Expediente de Proyecto el diagrama de secuencia correspondiente al escenario Crear SIC-0301 del CU Gestionar Certificado Análisis Producto Final.

CU Gestionar Certificado Análisis Producto Final escenario Buscar-Visualizar SIC-0301

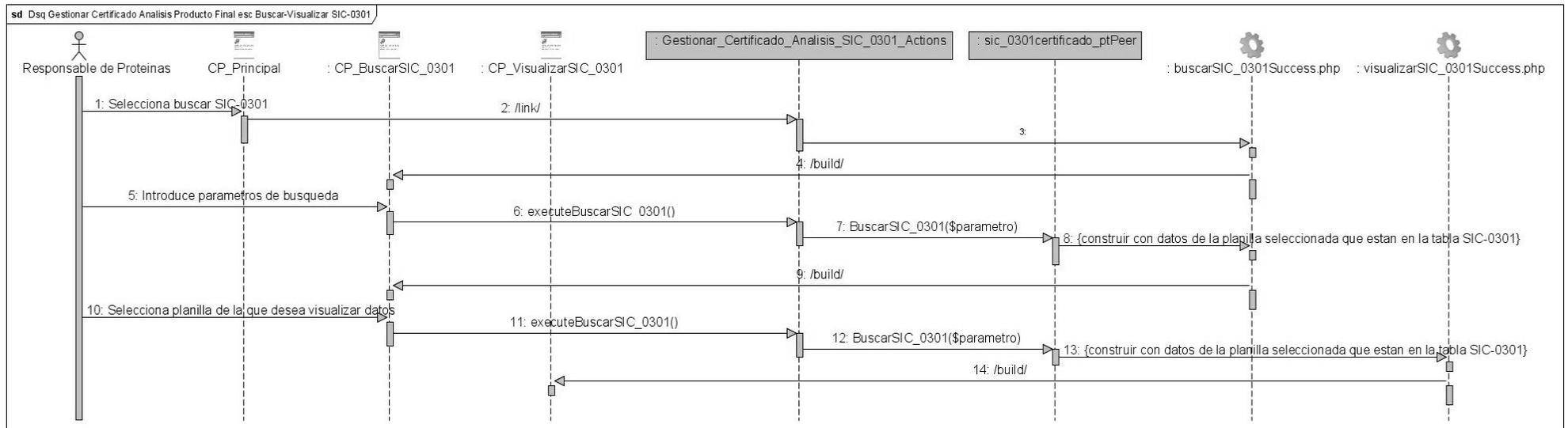


Fig. 63 Diagrama de secuencia CU Gestionar Certificado Análisis Producto Final escenario Buscar-Visualizar SIC-0301

CU Gestionar Certificado Análisis Producto Final escenario Modificar SIC-0301

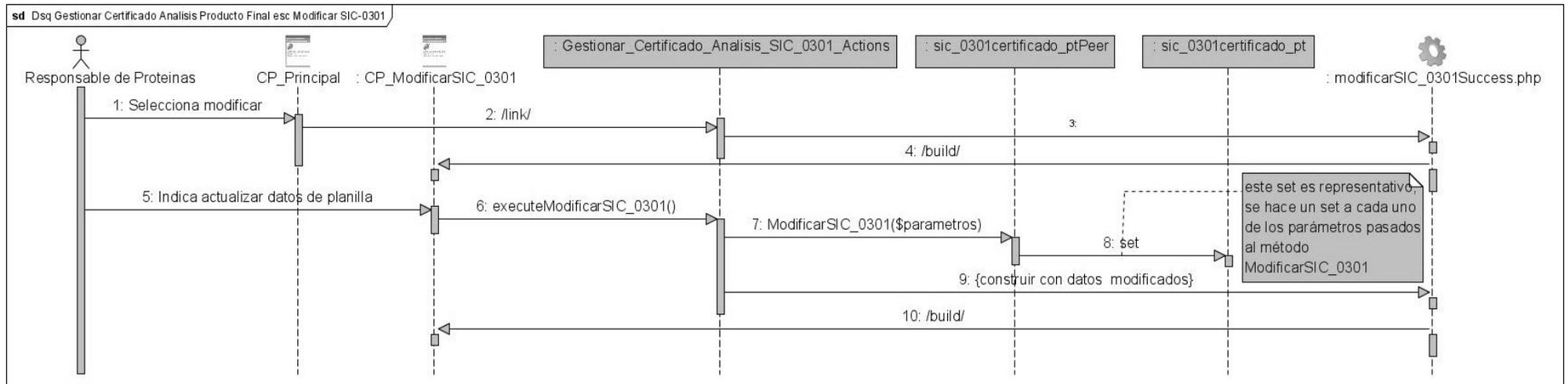


Fig. 64 Diagrama de secuencia CU Gestionar Certificado Análisis Producto Final escenario Modificar SIC-0301

CU Gestionar Certificado Análisis Producto Final escenario Generar Reporte SIC-0301

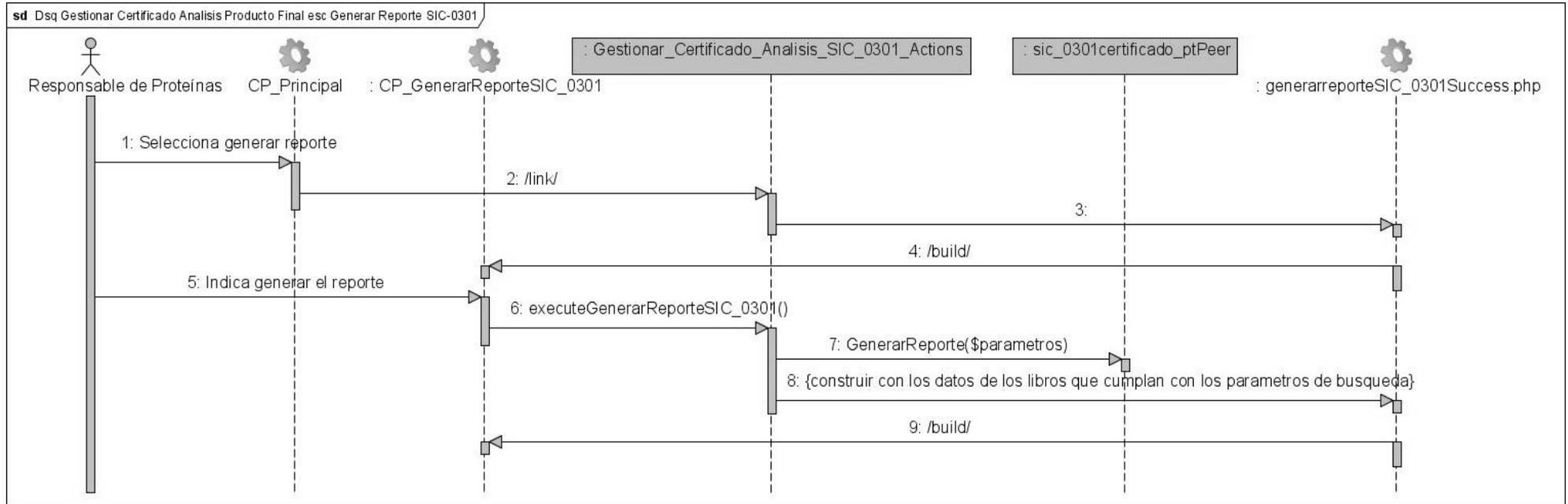


Fig. 65 Diagrama de secuencia CU Gestionar Certificado Análisis Producto Final escenario Generar Reporte SIC-0301

## 2.6 Evaluación de la calidad del diseño

Después de culminado el flujo de trabajo de Diseño de cualquier software, es importante la evaluación de la calidad del diseño realizado para comprobar si el objetivo fue cumplido y el sistema puede ser implementado sin ambigüedades a partir del mismo.

Existen diferentes formas de validar el diseño; se puede generar código a través de la herramienta CASE utilizada y comprobar si el mismo es de utilidad para los programadores que implementarán el sistema. En el caso de que el sistema ya se esté implementando, es posible comparar el diseño con el código fuente y analizar si existe correspondencia entre ellos.

El módulo que se está diseñando en este trabajo, no se ha comenzado a implementar y el diseño realizado, debido a sus particularidades (específicamente la representación del paquete Modelo), no permite la generación de código de forma efectiva, por lo que no se pueden aplicar las variantes antes expuestas.

Se va a comprobar la calidad del diseño a partir de encuestas a los programadores que van a implementar el módulo de Liberación Analítica, para conocer el grado de satisfacción que los mismos adquirieron como resultado de un análisis del diseño realizado. La encuesta se aplicó a dos programadores, y se midieron tres aspectos. A continuación se pueden observar los resultados por cada aspecto medido:

- *Comprensión del diseño:* Los programadores han estudiado los diagramas de clases del diseño y secuencia realizados entendiendo correctamente el flujo de mensajes que se representa en los mismos, considerándolos además legibles y apropiados para comenzar con la implementación del sistema
- *Correspondencia de los requerimientos no funcionales con el diseño:* Se evidencian en el diseño requisitos no funcionales; como el de Seguridad, ya que cada usuario que se autentifique en el sistema tendrá acceso solamente a la página que le corresponde y el de apariencia o interfaz externa, pues las interfaces son sencillas y no están sobrecargadas de información
- *Descomposición del diseño en partes más manejables:* Los diagramas de clases del diseño se encuentran estructurados en paquetes lo que constituye una ventaja para la organización en la implementación, ya que es posible descomponer los trabajos de implementación en partes más manejable llevadas a cabo por diferentes programadores

## **Conclusiones**

En este capítulo se realizaron los diagramas de clases y secuencia para los casos de uso identificados de acuerdo al patrón arquitectónico utilizado (Modelo-Vista-Controlador). Se confeccionó el mapa de navegación haciendo uso de UML. Se hizo además una breve referencia a la arquitectura a través de las vistas: lógica y de despliegue, y se diseñaron los prototipos de interfaz de usuario no funcionales que van a servir de apoyo a los programadores para la futura implementación del módulo de Liberación Analítica.

## **CONCLUSIONES**

- Como resultado de la investigación realizada se puede concluir que se cumplió el objetivo propuesto. Se diseñaron las clases para el módulo de Liberación Analítica del LIMS según el framework seleccionado para su desarrollo (Symfony), dejándolo listo para la implementación. Los artefactos de diseño obtenidos propiciaron la comunicación entre los analistas y el equipo de desarrollo. Las herramientas y metodologías definidas resultaron útiles para el desarrollo del trabajo y la utilización de la arquitectura MVC facilitó estructurar el diseño de la aplicación Web con vista a la implementación.

## **RECOMENDACIONES**

Luego de haber concluido el presente trabajo de diploma se recomienda:

- Se extienda la experiencia del desarrollo del LIMS a los demás centros del Polo Científico
- Que se integre el módulo de Liberación analítica a los restantes módulos que van a constituir el LIMS una vez que esté terminado el diseño e implementación de la totalidad de los mismos
- Se incluya el CU Gestionar Número de parte, luego de haber definido aspectos de la integración que se requieren para el funcionamiento del mismo

## REFERENCIAS BIBLIOGRÁFICAS

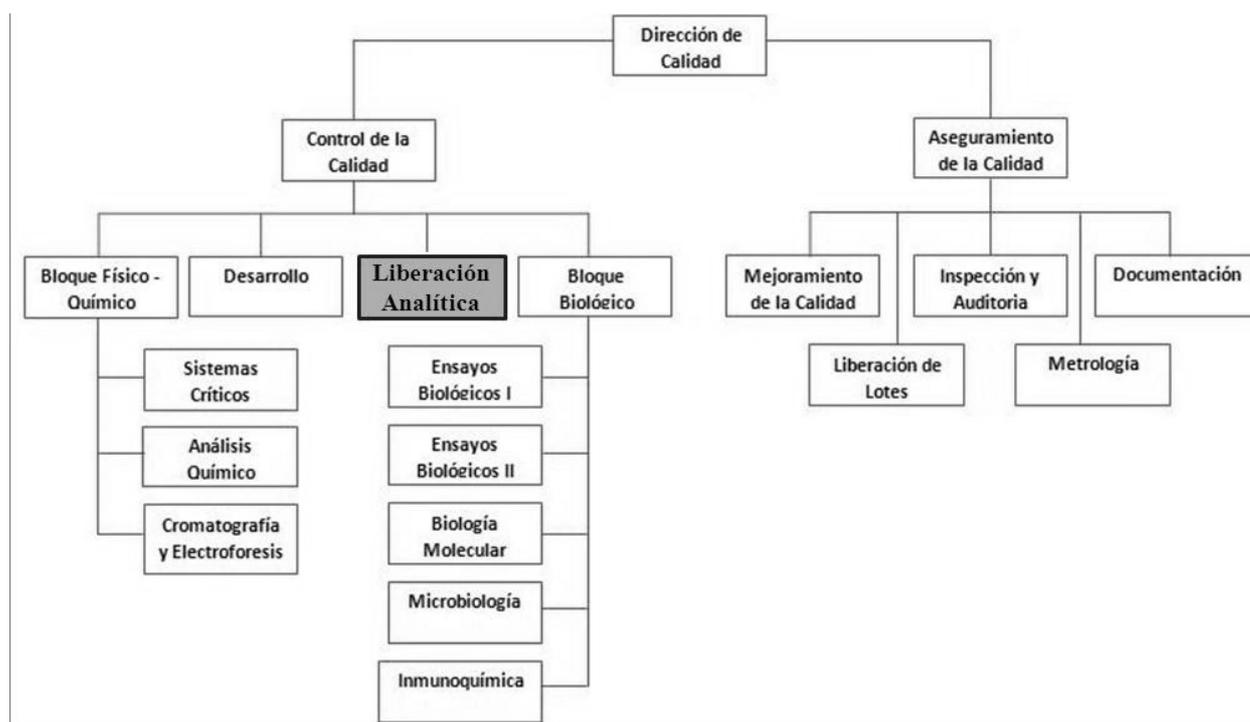
1. **CIGB**. Centro de Ingeniería Genética y Biotecnología. [En línea] 2003. [Citado el: 28 de febrero de 2008.] <http://www.cigb.edu.cu/pages/default.htm>.
2. **CIGB**. Departamento de Aseguramiento de la Calidad. [En línea] 2003. [Citado el: 28 de febrero de 2008.] <http://www.cigb.edu.cu/pages/acalidad.htm>.
3. **CIGB**. Departamento de Control de Calidad. [En línea] 2003. [Citado el: 28 de febrero de 2008.] <http://www.cigb.edu.cu/pages/ccalidad.htm>.
4. **CIGB**. Grupo de Recepción de Muestras y Manipulación de Expedientes. [En línea] 2003. [Citado el: 28 de febrero de 2008.] <http://www.cigb.edu.cu/pages/grmme.htm>.
5. **LIC. ARIDAY BALLESTEROS, LIC. YUDMILA MARÍA SOLER SANCHEZ, LIC. MARITZA AGUILERA MONTERO, LIC. ENRIQUE LÓPEZ ÁRIAS, LIC. ORLANDO DELGADO GONZALEZ.** *Importancia de la gestión de la información y el conocimiento en el proceso de cambio organizacional.* 2008.
6. **Yimian de Lyz Contreras Díaz, Soleydi Rivero Amador.** 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. Febrero, 2007.
7. **Melis, Olga Cuevas i.** InfoLab. [En línea] 2007. [Citado el: 28 de febrero de 2008.] (<http://www.onlabweb.com/lims.htm>).
8. Starlims. [En línea] 2007. [Citado el: 28 de febrero de 2008.] [http://www.starlims.com/es/LIMS\\_for\\_Pharmaceutical\\_Division\\_es.htm](http://www.starlims.com/es/LIMS_for_Pharmaceutical_Division_es.htm).
9. **INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE.** Departamento de Ingeniería y Gestión de Software, Universidad de las Ciencias Informáticas. 2007-2008. Conferencia
10. **Aurora Vizcaíno , Felix ÓscarGarcía , Ismael Caballero.** Una herramienta CASE para ADOO Visual Pradigm. [En línea] [Citado el: 25 de marzo de 2008.] [http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1\\_VP.pdf](http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf).
11. **Fabien Potencier, François Zaninotto.** *Symfony La Guía Definitiva.*
12. **Mario Rodríguez Martín, David Sánchez Fernández, Roberto Santa Escolástica Villoria.** *Patrones de Diseño: Decorator*

## BIBLIOGRAFÍA

1. **Alieski Sarmiento Almenares, Elián Cutiño Díaz.** *LIMS DE CALIDAD DEL CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA: ANÁLISIS DEL GRUPO DE RECEPCIÓN DE MUESTRAS Y MANIPULACIÓN DE EXPEDIENTES.* Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2006.
2. Centro de Ingeniería Genética y Biotecnología. [En línea] 2003. [Citado el: 28 de febrero de 2008.] <http://www.cigb.edu.cu/pages/default.htm>
3. **Conallen, Jim.** *Building Web Applications with UML.* Second Edition. Object Technology Series by Addison Wesley. 2002.
4. **Conallen, Jim.** *UML Extension for Web Applications.* Marzo 1999.
5. *FASE DE INICIO. FLUJO DE ANÁLISIS Y DISEÑO. DIAGRAMAS DE INTERACCIÓN.* Departamento de Ingeniería y Gestión de Software, Universidad de las Ciencias Informáticas. 2007-2008. Conferencia.
6. *INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE.* Departamento de Ingeniería y Gestión de Software, Universidad de las Ciencias Informáticas. 2007-2008. Conferencia.
7. **Navarro, Jose Angel Franco.** *UML en acción. Modelando Aplicaciones Web.*
8. Paradigma MVC. [En línea] <http://aldeacafe.com.mx/programacion/patrones/mvc.html>.
9. Patrones de diseño. [En línea] [Citado el: 28 de febrero de 2008.] <http://mit.ocw.universia.net/6.170/6.170/f01/pdf/lecture-12.pdf>.
10. Patrones para asignación de responsabilidades. [En línea] <https://s3.amazonaws.com/ppt-download/gonzalorojas-12-uml-patrones-de-diseno1574.ppt>.
11. Starlims. [En línea] 2007. [Citado el: 28 de febrero de 2008.] [http://www.starlims.com/es/LIMS\\_for\\_Pharmaceutical\\_Division\\_es.htm](http://www.starlims.com/es/LIMS_for_Pharmaceutical_Division_es.htm).
12. The Model-View-Controller Design Pattern. [En línea] [http://developer.apple.com/documentation/Cocoa/Conceptual/CocoaFundamentals/CocoaDesignPatterns/chapter\\_5\\_section\\_4.html](http://developer.apple.com/documentation/Cocoa/Conceptual/CocoaFundamentals/CocoaDesignPatterns/chapter_5_section_4.html).

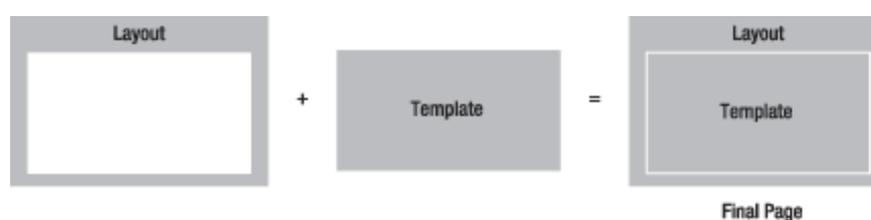
## ANEXOS

**Anexo 1:** Estructura jerárquica del Área de Calidad del Centro de Ingeniería Genética y Biotecnología.

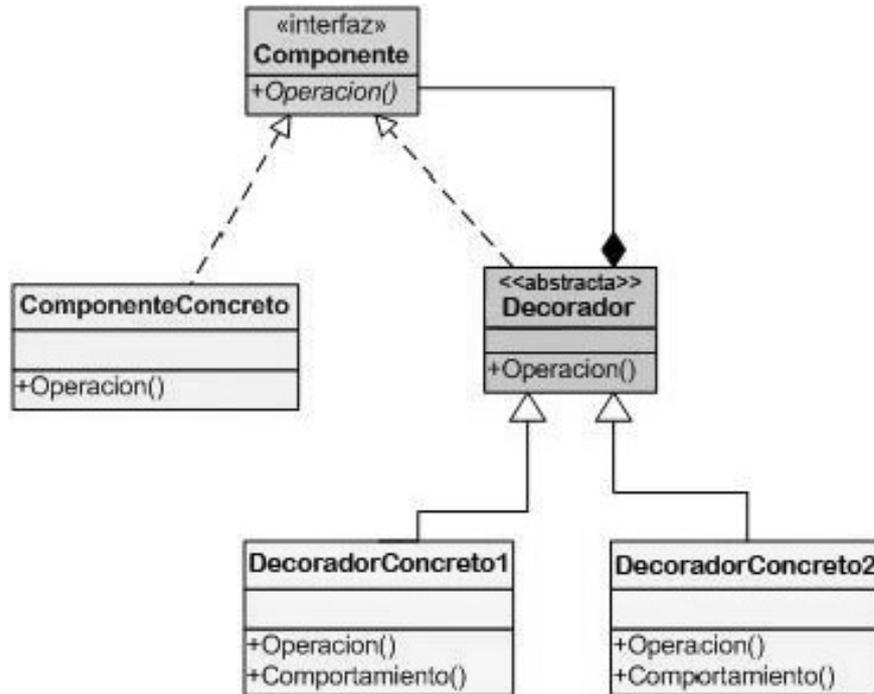


**Fig.66** Estructura jerárquica del Área de Calidad del Centro de Ingeniería Genética y Biotecnología

**Anexo 2:** Plantilla decorada con un layout.



**Fig. 67** Plantilla decorada con un layout

**Anexo 3:** Participantes y estructura del patrón de diseño decorador.**Fig. 68** Participantes y estructura del patrón de diseño decorador

**Anexo 4:** Prototipos de interfaz de usuario del CU Gestionar Muestras.

Registrar recepción de nueva muestra

CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA				
DIRECCION DE CALIDAD				
LIBRO DE RECEPCION DE MUESTRAS				
El Producto proviene de <input type="text"/> Productos <input type="text"/> Etapas <input type="text"/>				
Lote	Almacenamiento	Cantidad	F. Recepción	Lote Origen
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/> Buscar				
F. Fabricación Lote	Entrega	Recibe	Producción	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Observaciones	Firma	Autorizado		
<input type="text"/>		<input type="radio"/> Si <input type="radio"/> No		
Análisis		Nombre Análisis		
<input type="checkbox"/> Análisis 1 <input type="checkbox"/> Análisis 1 <input type="button" value="Agregar"/>		<input type="text"/> <input type="button" value="Guardar"/>		
<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>				

Fig. 69 Prototipo Registrar Muestra

Buscar y visualizar muestra

CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA							
DIRECCION DE CALIDAD							
LIBRO DE RECEPCION DE MUESTRAS							
Lote:	<input type="text"/>	Fuente :	<input type="text"/>	Producto:	<input type="text"/>	Etapa:	<input type="text"/>
F. de Recepción:	<input type="text"/> <input type="text"/>	Análisis:	<input type="text"/>	F. Confección SIC	<input type="text"/>	<input type="text"/>	<input type="text"/>
							<input type="button" value="Buscar"/> <input type="button" value="Cancelar"/>
Lote:	Fuente :	Producto:	Etapa:	F. de Recepción:	Análisis:	F. Confección SIC	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Visualizar

Fig. 70 Prototipo Buscar Muestra

<b>CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA</b>				
<b>DIRECCION DE CALIDAD</b>				
<b>LIBRO DE RECEPCION DE MUESTRAS</b>				
<b>F. Recepción</b>	<b>Almacenamiento</b>	<b>Cantidad</b>	<b>Lote</b>	<b>Lote Origen</b>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<b>F. Fabricación Lote</b>	<b>Entrega</b>	<b>Recibe</b>	<b>Producción</b>	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
<b>Observaciones</b>			<b>Autorizado</b>	
<input type="text"/>			<input type="radio"/> Si <input type="radio"/> No	
<b>Analisis</b>			<b>Firma</b>	
<input type="text"/>			<input type="text"/>	
<input type="button" value="Modificar"/> <input type="button" value="Nueva Búsqueda"/> <input type="button" value="Imprimir"/> <input type="button" value="Cancelar"/>				

Fig. 71 Prototipo Visualizar Muestra

## Modificar datos de muestra

CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA				
DIRECCION DE CALIDAD				
LIBRO DE RECEPCION DE MUESTRAS				
<b>F. Recepción</b>	<b>Almacenamiento</b>	<b>Cantidad</b>	<b>Lote</b>	<b>Lote Origen</b>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<b>F. Fabricación Lote</b>	<b>Entrega</b>	<b>Recibe</b>	<b>Producción</b>	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
<b>Observaciones</b>			<b>Autorizado</b>	
<input type="text"/>			<input type="radio"/> Si	<input type="radio"/> No
<b>Analisis</b>			<b>Firma</b>	
<input type="text"/>			<input type="text"/>	
			<input type="button" value="Actualizar"/>	<input type="button" value="Cancelar"/>

Fig. 72 Prototipo Modificar Muestra

Buscar y visualizar expediente analítico

<b>CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA</b>							
<b>DIRECCION DE CALIDAD</b>							
<b>LIBRO DE RECEPCION DE MUESTRAS</b>							
Lote:	<input type="text"/>	Fuente :	<input type="text"/>	Producto:	<input type="text"/>	Etapa:	<input type="text"/>
F. de Recepción:	<input type="text"/>	-	<input type="text"/>	Análisis:	<input type="text"/>	F. Confección SIC	<input type="text"/>
							<input type="button" value="Buscar"/> <input type="button" value="Cancelar"/>
Lote:	Fuente :	Producto:	Etapa:	F. de Recepción:	Análisis:	F. Confección SIC	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Visualizar"/>

Fig. 73 Prototipo Buscar y Visualizar Expediente Analítico

Revisar expediente analítico

<b>CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA</b>			
<b>DIRECCION DE CALIDAD</b>			
<b>EXPEDIENTE ANALITICO</b>			
Expedientes Analíticos	<input type="button" value="Visualizar"/>	Listado de Planillas	
<input type="button" value="v"/>	<input type="button" value="v"/>	<input type="button" value="Visualizar"/>	<input type="button" value="Cancelar"/>

Fig. 74 Prototipo Revisar Expediente Analítico

## GLOSARIO

**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.

**Caso de uso:** Especificación de las secuencias de acciones, incluyendo secuencias variantes y una descripción de un conjunto de secuencias de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

**Control de la Calidad:** Técnicas y actividades de carácter operativo utilizadas para satisfacer los requisitos de calidad.

**Expediente analítico:** Conjunto de todos los resultados analíticos y las investigaciones analíticas realizadas a un lote según se han descrito previamente en una especificación de acuerdo con las normas nacionales o internacionales establecidas para ese producto en cuestión.

**Framework:** Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

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

**Libro de registros del producto:** Es el libro donde se registra el lote para todos los ensayos que se le han solicitado realizar en Control de la Calidad. De esta manera se centra la actividad y se establece un control para el destino de los resultados.

**Metodología:** Un sistema de principios y normas generales de organización y estructuración teórico-práctica de actividades.

**Muestra:** Es cantidad mínima, estadísticamente representativa, que se toma del total del lote, para poder realizar todos los resultados analíticos según se han descrito en las especificaciones del producto.

**SIC:** Sistema de Información y Control.

**Validación:** Acción documentada que demuestra, de acuerdo con los principios de las Buenas Prácticas de Fabricación, que cualquier procedimiento, proceso, equipo, material, actividad o sistema realmente brinda los resultados esperados.

This document was created with Win2PDF available at <http://www.daneprairie.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.