

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: Análisis y Diseño del módulo Análisis Químico”

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

Autoras: Yadira Sulmainiz Tamarit Quintana

Yalina Lamas Garcia

Tutora: Ing. Lisdany De la Fuente Díaz

Co-Tutora: Lic. Miulkenia Navarro Reyes

Ciudad de La Habana, Junio 2008

“Año 50 de la Revolución”

“El futuro pertenece a quienes creen en la belleza de los sueños”

Eleanor Roosevelt

DECLARACIÓN DE AUTORÍA

Declaramos ser autoras 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 firmamos la presente a los ____ días del mes de Junio del 2008.

Firma de la Autora

Firma de la Autora

Firma de la Tutora

Firma de la Co-Tutora

DATOS DEL CONTACTO

Tutora:

Ing. Ing. Lisdany de la Fuente Díaz.

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

Email: ldelafuente@uci.cu.

Co-Tutora:

Lic. Miulkenia Navarro Reyes.

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

Email: mnavarror@uci.cu.

DEDICATORIA

Yadira Sulmainiz Tamarit Quintana

Dedico este trabajo en especial a mis padres Mariela Quintana Fernández y Jorge Luis Tamarit González, por ser mi fuente de inspiración y preocuparse por mí cada día. ¡Los quiero muchísimo!

A mi padastro Adalberto por ser tan preocupado, buen amigo y estar siempre apoyándome cuando lo necesito.

A mis abuelita Caridad y Fredesvinda y a mi querido abuelo Reynaldo.

En especial a mi tía Yoandra por apoyarme a lo largo de estos cinco años y por su cariño incondicional y a mi tío Roly por ser un gran apoyo también.

A mi hermanita Yaritza y a mi hermanito Jorge.

A mis primas Yoannet, Ivón, Dayana, Yenoelys, y Maribel.

A mis familiares en sentido general, que tanto han hecho por mí para que pudiera llegar a este día.

A Norma Torres Pérez por su cariño y apoyo.

A Guelmis por ser tan paciente y cariñoso conmigo ¡Te quiero muchito!

A mis amistades de la UCI por siempre estar ahí cuando los necesito a Yanelys, Nayvis, Yalina y Adairis, a las que recordaré ¡por siempre!

Yalina Lamas García

Dedico este trabajo a mis padres Cira Clara García Rodríguez y a Rodolfo Lamas Valdés por ser los mejores padres del mundo, por haberme apoyado siempre y haber confiado en mí cuando nadie más lo hizo, por brindarme su cariño y su ternura infinita.

A mis abuelos Hilda y Segundo por brindarme sus consejos y su cariño y por ser otros padres para mí.

A mis primas Daneisy y Daily por su ayuda y su cariño

A mis amigas Yanefys, Adairis, Yadira y Nayvis por aguantarme durante 5 largos años, por estar siempre ahí en los buenos y malos momentos, por saber darme siempre el mejor consejo, nunca las olvidaré.

A todas mis amistades en la UCI y fuera de ella que siempre me han brindado su afecto.

A mis amigos de la infancia que se han preocupado siempre por mí, que me han brindado su cariño incondicional.

A toda mi familia en general que nunca han dudado en brindarme una mano.

A mis primos Franco, José Agustín y Ángel Daniel por alegrarme la vida.

AGRADECIMIENTOS

Agradecemos a nuestros seres más queridos que hicieron posible que se cumplieran nuestros sueños.

A nuestros compañeros y amigos que nos brindaron consuelo cuando el camino se tornó más difícil.

A los profesores de la Universidad de las Ciencias Informáticas, por la formación que nos han dado a lo largo de estos cinco años de la carrera.

A nuestro tutora Lisdany por su asesoría constante y su preocupación por nosotras.

A las muchachas y muchachos del Laboratorio Análisis Químico del Centro de Ingeniería Genética y Biotecnología, por ser tan atentos, y en especial a Rubén por su paciencia sin límites y dedicación en el desarrollo de la tesis.

RESUMEN

Diariamente en la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología (CIGB) se cuenta con gran cantidad de documentos a consultar lo que representa una pérdida considerable de tiempo en procesos de suma importancia. Luego de un estudio referente a los mecanismos de procesamiento y almacenamiento de la información, un grupo de analistas realizó el Modelo de negocio, con el objetivo de informatizar el manejo de ésta información y contribuir al control de la misma, se está desarrollando un Sistema de Gestión de Información de los Laboratorios (LIMS, del inglés Laboratory Information Management System). Este trabajo realiza el análisis y diseño de uno de los laboratorios de dicha dirección:

-El Laboratorio Análisis Químico (LAQ) donde se realizan ensayos a las materias primas y las proteínas empleadas en la producción.

Se identifican las funcionalidades y características que deberá cumplir el módulo y se diseñan las clases del mismo.

Palabras Claves: CIGB, LIMS, Gestión de la Información.

ÍNDICE

DEDICATORIA	I
AGRADECIMIENTOS.....	III
RESUMEN.....	IV
INTRODUCCIÓN.....	1
Fundamentación Teórica	6
Introducción	6
1.1. Gestión de la Información	6
1.1.1. Sistemas de Gestión de la Información.....	7
1.1.2. La Web como Sistema de información.....	8
1.1.3. Sistemas de Gestión de la Información de Laboratorios.....	8
1.2. Metodología y herramientas a utilizar	9
1.2.1. Proceso de Desarrollo de Software: RUP	9
1.2.1.1. Actividades de los flujos de trabajo	11
1.2.2. Lenguaje de modelado: UML	12
1.2.3. Herramientas CASE: Visual Paradigm (Versión 6.1)	13
1.2.3.1. Características fundamentales que ofrece Visual Paradigm.....	13
1.3. Herramienta para la Gestión de Requerimientos	14
1.3.1. RequisitePro.....	15
1.3.2. Open Source Requirement Management Tool (Versión 1.5)	15
1.4. Roles	16
1.5. Artefactos	18
1.5.1. Artefactos a realizar por el Analista del Sistema.....	18
1.5.2. Artefactos a realizar por el Especificador de Requerimientos.....	18
1.5.3. Artefactos a realizar por el Diseñador	19
1.5.4. Artefactos a realizar por el Diseñador de Interfaz de Usuario.....	20
1.5.5. Artefactos a realizar por el Arquitecto	20
1.6. Patrones.....	20
1.6.1. Patrones de Casos de Usos.....	21
1.6.2. Patrones de Diseño.....	21
1.6.2.1. Patrones GRASP.....	22
1.6.2.2. Patrones GOF	22

1.7. Framework: Symfony (Versión 1.0.11)	24
1.7.1. Características de Symfony	24
Conclusiones	26
Características del Sistema	27
Introducción	27
2.1. Requerimientos	27
2.1.1. Requerimientos funcionales y no funcionales del sistema	27
2.1.1.1 Requerimientos funcionales	28
2.1.1.2 Requerimientos no funcionales	39
2.1.2. Actores del sistema a automatizar: Módulo Análisis Químico (MAQ)	41
2.1.3. Casos de Uso del Sistema	42
2.1.4. Matriz de trazabilidad	44
2.1.5. Diagrama de Casos de Uso del Sistema: Laboratorio Análisis Químico (LAQ)	44
2.2. Descripciones textuales resumidas	51
2.3. Descripción ampliada del CU Gestionar Libro de entrada de cuarentena	52
Conclusiones	60
Diseño del Sistema	61
Introducción	61
3.1. Propósitos del diseño	61
3.2. Principios de diseño	62
3.3. Referencia a la Arquitectura del Sistema	63
3.3.1. Vista Lógica	63
3.4. Prototipos de interfaz de usuario	70
3.4.1. Prototipos de interfaz de usuario del Libro de entrada de cuarentena	70
3.5. Mapa de Navegación	72
3.6. Diagramas de Clases del Diseño	75
3.7. Diagramas de secuencia	82
3.8. Validación del diseño realizado	86
Conclusiones	94
CONCLUSIONES	95
RECOMENDACIONES	96
REFERENCIAS BIBLIOGRÁFICAS	97
BIBLIOGRAFÍA	99
ANEXOS	101

Anexo 1. Estructura jerárquica del Área de Calidad del Centro de Ingeniería Genética y Biotecnología	101
Anexo 2. Una iteración de RUP	102
Anexo 3. Fases e Iteraciones de RUP	102
Anexo 4. Artefactos a realizar por el Analista del Sistema	103
Anexo 5. Artefactos a realizar por el Especificador de Requerimientos	103
Anexo 6. Artefactos a realizar por el Diseñador.....	103
Anexo 7. Artefactos a realizar por el Diseñador de Interfaz de Usuario	104
Anexo 8. Funcionamiento del patrón MVC	104
GLOSARIO DE TÉRMINOS	105

INTRODUCCIÓN

En la actualidad las Tecnologías de la Información y las Comunicaciones ocupan un lugar fundamental en el desarrollo de la sociedad y han permitido llevar la globalidad al mundo de la comunicación, facilitando la interconexión entre las personas e instituciones a nivel mundial. Son una parte de las tecnologías emergentes que se identifican con las siglas TIC y que hacen referencia a la utilización de medios informáticos para almacenar, procesar y difundir todo tipo de información. La importancia de las TIC es relevante en el mundo, proveen herramientas que ofrecen la posibilidad de encontrar soluciones novedosas ante los desafíos sociales de hoy.

Debido al desarrollo tecnológico y la convergencia cada vez mayor, las Tecnologías de la Información y las Comunicaciones, han tenido una gran influencia en todos los sectores de la sociedad, siendo así una necesidad para el desarrollo económico y social de cualquier país.

En el país se ha defendido siempre el concepto de que el uso masivo de las TIC no es un fin sino una herramienta poderosa para lograr el desarrollo por tanto se ha llevado a cabo el reto de la informatización de la sociedad, alcanzando resultados satisfactorios en áreas tan esenciales como la Educación, la Salud y la Investigación.

Se desarrollan proyectos de informatización en lugares muy importantes del país, entre ellos se encuentra el Centro de Ingeniería Genética y Biotecnología (CIGB), el cual tiene un papel integrador en la esfera de la Biotecnología Cubana, con alta capacidad científico técnica. Asume la responsabilidad de contribuir directamente en el desarrollo económico y social del País. Sus producciones están destinadas a la salud humana, las producciones agropecuarias, y al medio ambiente.

La calidad, es la imagen del CIGB, los productos desarrollados y elaborados en este centro se caracterizan por su eficacia y seguridad, por tal motivo en la estructura del centro existe la **Dirección de Calidad**.

Su objetivo se pone de manifiesto a través de los Departamentos de **Control de la Calidad y Aseguramiento de la Calidad**.

“Departamento de Control de la Calidad” ejecuta varias funciones, dentro de las cuales se pueden citar 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, a continuación hacemos referencia a los mismos [1]:

1. Grupo de Desarrollo.

2. Grupo de Liberación Analítica.
3. Sección biológica compuesta por cinco laboratorios:
 - Laboratorio de Biología Molecular.
 - Laboratorio de Ensayos Biológicos I.
 - Laboratorio de Ensayos Biológicos II.
 - Laboratorio de Inmunoquímica.
 - Laboratorio de Microbiología (LM).
4. Sección físico-química compuesta por tres laboratorios:
 - Laboratorio Análisis Químico.
 - Laboratorio de Cromatografía y Electroforesis.
 - Laboratorio de Sistemas Críticos.

El “**Laboratorio Análisis Químico** “...realiza diferentes técnicas físico-químicas y bioquímicas para la determinación de impurezas y purezas de las diferentes proteínas que se producen o investigan en el Centro. También se lleva el control analítico de los reactivos y componentes críticos que son utilizados como materia prima en la producción de los productos farmacéuticos en el CIGB. Todas estas técnicas se realizan bajo un estricto cumplimiento de las Buenas Prácticas de Laboratorio garantizando así resultados confiables, lo cual ha sido demostrado en cada inspección realizada por inspectores internos, por la entidad nacional regulatoria (CECMED) y por organizaciones internacionales del nivel de la Organización Mundial de la Salud (OMS).En el laboratorio se realizan alrededor de 25 técnicas analíticas diferentes, las cuales se encuentran validadas.”[2]

“**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 satisfagan los requisitos de calidad establecidos. Este departamento está compuesto por dos Secciones y dos grupos de trabajo [3]:

Sección de Mejoramiento de la Calidad.

- Sección de Inspección, Auditoría y Liberación de lotes.
- Grupo de Inspección y Auditorias.
- Grupo de Liberación de Lotes.

Una función de vital importancia que desarrolla esta sección es el proceso de liberación de los IFA y los productos finales, preparando además toda la documentación que acompaña a los envíos de los productos liberados. **(Ver Anexo 1)**

- Grupo de Documentación.
- Grupo de Metrología.

La información como un soporte de transmisión de conocimientos se convierte en algo casi vital, por lo que en la sociedad se debe hacer frente a sus necesidades, siendo su gestión muy importante para los procesos que se realizan en algunas instituciones donde se maneja una cantidad considerable, como es el caso del CIGB.

Como una solución informática para el desarrollo de aplicaciones que gestionan gran cantidad de información es que surgen a finales del siglo pasado los sistemas de información. "Un sistema de información se puede definir como un sistema formal para recabar, integrar, comprar, analizar y difundir información interna y externa de la empresa en forma oportuna, eficaz y eficiente." [4]

Durante los últimos años las empresas han estado sometidas a fuertes presiones y cambios en la apreciación de su efectividad y nivel de competitividad en el mercado. Unido a esto, se están implantando programas rigurosos de Aseguramiento de la Calidad para dar respuesta a la gran relevancia que está alcanzando en el mercado todo lo referente a la Calidad de las actividades o productos desarrollados por cada compañía.

Con el objetivo de incorporar al entorno del Laboratorio los beneficios y mejoras aportados por los Sistemas de Información, es que a finales del siglo pasado surgen los Sistemas de Gestión de la Información del Laboratorio (LIMS, del inglés Laboratory Information Management System).

"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 y minimizando los errores debidos a la transferencia de información" [5].

Un Sistema de Gestión de la Información del Laboratorio es la herramienta esencial para los laboratorios modernos, y la aplicación informática clave para un amplio rango de industrias y organizaciones, especialmente en la industria farmacéutica, la cual depende en gran medida del éxito de los laboratorios.[6]

En el Laboratorio Análisis Químico, el llenado de planillas se realiza manualmente, además se maneja un gran volumen de información, siendo así muy difícil la gestión de la misma, esto provoca una pérdida considerable de tiempo en procesos de suma importancia, dado básicamente por la forma de búsqueda de la gran cantidad de documentos a consultar. Luego de un estudio realizado a los mecanismos de procesamiento y almacenamiento de la información de este laboratorio, un grupo de analistas realizó el modelamiento del negocio, identificando así los procesos fundamentales que se llevan a cabo. En la actualidad se prosigue con el Análisis y Diseño del módulo Análisis Químico para el Sistema de Gestión de la Información del Laboratorio (LIMS) y se identifica como **problema científico** a resolver ¿Cómo transformar los procesos identificados para el módulo Análisis Químico del Sistema de Gestión de la Información de los Laboratorios en elementos que puedan ser implementados?

El problema planteado tiene como **objeto de estudio**: Proceso de desarrollo de los Sistemas de Gestión de la Información.

El objeto delimita el **campo de acción**: Análisis y Diseño de los Sistemas de Gestión de la Información.

En la búsqueda de la solución al problema planteado se define como **objetivo**: Desarrollar el Análisis y Diseño del módulo: Análisis Químico, para el Sistema de Gestión de la Información de los Laboratorios de la Dirección de Calidad del CIGB.

Resultando como **objetivos específicos**:

- Obtener las funcionalidades y las características del sistema.
- Realizar el diseño del módulo Análisis Químico.

Para cumplir estos objetivos se trazan las siguientes **tareas**:

- Investigación sobre los roles analista y diseñador.
- Estudio del Modelo del Negocio.
- Obtención y clasificación de los requerimientos.
- Identificación de actores y casos de uso.
- Estructuración del modelo de casos de uso.
- Especificación de los casos de uso.
- Realización de CU-diseño: diagramas de clases del diseño y diagramas de interacción.

- Elaboración del Mapa de Navegación.

El trabajo consta de Introducción, 3 Capítulos, Bibliografía y Anexos.

En el Capítulo 1 Fundamentación Teórica: en este capítulo se mencionarán algunos aspectos fundamentales de la Gestión de la Información, dentro de ésta se hace una breve referencia a los Sistemas de Gestión de la Información, la Web como Sistema de Información y los Sistemas de Gestión de Información de Laboratorios. Además se aborda sobre las principales características del Proceso Unificado del Rational (RUP), los roles a desempeñar, los artefactos, el framework y las herramientas a utilizar.

En el Capítulo 2 Características del Sistema: se identifican requerimientos funcionales y no funcionales, los actores del sistema, casos de uso del sistema, se realizan los diagramas de casos de uso del sistema, se describen los casos de uso del sistema, se realiza la matriz de trazabilidad entre los Actores-CU, y entre CU-RF y además se realiza la Vista de Casos de uso.

En el Capítulo 3 Diseño del Sistema: se representan los diagramas de las clases para aplicaciones Web con sus relaciones, los diagramas de interacción, mapa de navegación, se construyen los prototipos no funcionales, se hace referencia a la arquitectura del sistema a través de la Vista Lógica y de Despliegue, y se realiza la validación del Diseño propuesto.

Capítulo 1

Fundamentación Teórica

Introducción

En este capítulo, para lograr una mejor comprensión de la necesidad de este trabajo se aborda de los Sistemas de Gestión de Información de forma general, de los procesos de desarrollo de software más conocidos, de la metodología y herramientas a utilizar, los roles analista del sistema, diseñador, diseñador de interfaz de usuario y arquitecto y los artefactos enmarcados en cada uno de estos, definidos por el proceso unificado de desarrollo (RUP) y los patrones a utilizar.

1.1. Gestión de la Información

En la era de la información, de la explosión de sus tecnologías, se vive la etapa en la que la humanidad ha alcanzado un desarrollo imprevisible; cada día son mayores las diferencias sociales, políticas y económicas. Se habla constantemente sobre la sociedad de la información, es visible el paso de las sociedades industriales a las posindustriales y del conocimiento, donde el factor esencial de progreso es el conocimiento. Esta nueva sociedad, con organizaciones basadas en el aprendizaje, cuyo capital máspreciado es el ser humano, se sustenta en un desarrollo tecnológico sin precedentes, es el punto en el cual las grandes compañías planifican sus productos en función de la gestión de la información y de la viabilidad para su obtención. [7]

La gestión de la información es el proceso de analizar, utilizar, recuperar y almacenar la información que se ha obtenido.

Su función es facilitar información precisa para la toma de decisiones, sin preocuparse por otros aspectos relacionados con el aprendizaje. Tiene una visión más mecanicista, ingenieril, el elemento humano tiene menos importancia.

Algunas de las funciones de la Gestión de Información:

- 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.
- Entrenar a los miembros de la organización en el manejo o la utilización de los recursos informacionales.
- 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.

1.1.1. Sistemas de Gestión de la Información

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.

Almacenamiento de información: El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, porque a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.

Procesamiento de Información: Es la capacidad del Sistema de 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.

Salida de Información: La salida es la capacidad de un Sistema de Información para sacar al exterior la información procesada o bien datos de entrada. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información.

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

La utilización de los LIMS ha favorecido en gran medida la manipulación de los datos que se obtienen en los laboratorios. Entre los beneficios más comunes que tiene su utilización se pudieran mencionar:

- Aumento de la cantidad de información disponible y requisitos de manipulación de la misma.
- Aseguramiento de la calidad.
- Integridad de la información.
- Mejoras en el procesamiento de la información y en la productividad.

A continuación ilustramos algunos ejemplos:

- **Matrix LIMS** se combinan los dos aspectos fundamentales para garantizar el éxito en el proyecto LIMS con la mínima inversión de tiempo y dinero: configurabilidad auténtica y rapidez de implementación. Se suministra con el conjunto más completo de herramientas para lograr

una configurabilidad absoluta sin necesidad de programar códigos a medida. Se adapta a cualquier industria y laboratorio, tanto en los sectores altamente regulados como en los no regulados. Las aplicaciones Matrix LIMS solucionan las necesidades de grandes empresas y también de organizaciones de tamaño pequeño o mediano, desde incluso un único usuario o un departamento.

- **Watson** es un LIMS diseñado de forma específica para ser compatible con los estudios DMPK/Bioanalíticos del desarrollo de fármacos. Fue desarrollado por InnaPhase junto a las principales compañías mundiales farmacéuticas para acelerar el bioanálisis y gestión de muestras del desarrollo de fármacos.
- **Quaass-Lab** (Quality Assurance for Laboratory) es un sistema LIMS (Laboratory Information Management System) y está destinado a la correcta organización, gestión y explotación de toda la información que se origina en un laboratorio de análisis de muestras.

1.2. Metodología y herramientas a utilizar

El Sistema de Gestión de la Información del Laboratorio del Centro de Ingeniería Genética y Biotecnología (CIGB), en el que se continúa trabajando actualmente lleva dos años desarrollándose, y desde ese entonces se definieron las herramientas y metodología a emplear.

1.2.1. Proceso de Desarrollo de Software: RUP

El proceso de desarrollo de software... "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo". [8]

Requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases:

Concepción: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.

Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.

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

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

El proceso de desarrollo de Software escogido, es el Proceso Unificado Rational (Rational Unified Process) que es una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas probadas y una arquitectura configurable. Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Es un marco de desarrollo software iterativo e incremental. [9]

RUP (Rational Unified Process) está basado en 5 principios claves:

- Adaptar el proceso.
- Balancear prioridades.
- Demostrar valor iterativamente.
- Elevar el nivel de abstracción.
- Enfocarse en la calidad.

Rational Unified Process (RUP) tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental. [10]

- **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 CU

relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.

- **Iterativo e Incremental:** El equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

(Ver Anexo 2)

En RUP se han agrupado las actividades en grupos lógicos definiéndose nuevos flujos de trabajo principales. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo: Modelamiento del negocio, Requerimientos, Análisis y diseño, Implementación, Prueba (Testeo), Instalación, Administración del proyecto, Administración de configuración y cambios, Ambiente. **(Ver Anexo 3)**

En el presente trabajo se realizarán fundamentalmente los siguientes flujos de trabajo:

Requerimientos: es uno de los flujos de trabajo más importantes, porque en él se establece qué tiene que hacer exactamente el sistema que se va a construir.

Análisis y diseño: su objetivo es traducir los requisitos a una especificación que describe cómo implementar el sistema.

1.2.1.1. Actividades de los flujos de trabajo

Flujo de trabajo de Requerimientos: como parte de este flujo de trabajo las principales actividades que se realizan son:

- Identificar y clasificar requerimientos.
- Encontrar actores y casos de uso.
- Priorizar casos de uso.
- Detallar casos de uso.

- Estructurar el modelo de casos de uso.

Flujo de trabajo de Análisis y Diseño: como parte de este flujo de trabajo las principales actividades que se realizan son:

- Análisis de la arquitectura: Identificar paquetes y clases del análisis.
- Analizar un caso de uso: Identificar clases cuyos objetos son necesarios para realizar el caso de uso y distribuir el comportamiento entre los objetos que participan en el caso de uso.
- Analizar una clase: Identificar atributos de las clases y relaciones entre ellas.
- Analizar un paquete: Garantizar alta cohesión y bajo acoplamiento de los paquetes y describir las relaciones entre ellos.

1.2.2. Lenguaje de modelado: UML

UML es un lenguaje que proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Este lenguaje indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: permite especificar cuáles son las características de un sistema antes de su construcción.
- Diagrama de colaboración.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Diagrama de estados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.
- Diagrama de actividades.

- Diagrama de componentes.
- Diagrama de despliegue.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (workflow) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware.

Los diagramas que se representan en el presente trabajo son los de casos de uso, clases, secuencia, y despliegue.

1.2.3. Herramientas CASE: Visual Paradigm (Versión 6.1)

”Las Herramientas Computer Aided Software Engineering (CASE), se puede definir como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software.” [11]

En el trabajo se decidió utilizar Visual Paradigm basado en la investigación realizada por las tesis anteriores del proyecto sobre las herramientas CASE más conocidas y utilizadas en el proceso de desarrollo de software, como resultado de esta búsqueda se determinó que era la más apropiada para el desarrollo del proyecto, pero no se contaba con los recursos necesarios, y se utilizó Rational Suite 2003 como herramienta Case. Actualmente se cuenta con las características de hardware para su uso.

La herramienta ayuda al equipo de desarrollo de software a maximizar y acelerar tanto las contribuciones individuales como las de equipo. Es un producto que facilita a las organizaciones diagrama visual y diseño, integrar y desplegar sus aplicaciones empresariales de misión crítica y de sus bases de datos subyacentes.

Apoya los últimos estándares de Java y de UML. Además, Vp-uml se encaja con un IDE de gran alcance. Las transiciones del análisis al diseño y entonces a la puesta en práctica se integran dentro de la herramienta. Un conjunto de idiomas tanto en la generación del código y la Ingeniería Inversa en Java, C ++, CORBA IDL, PHP, XML Schema, Ada y Python.

1.2.3.1. Características fundamentales que ofrece Visual Paradigm

- Entorno de creación de diagramas para UML 2.0.

- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería 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. Herramienta para la Gestión de Requerimientos

Los requisitos constituyen uno de los primeros entregables en el proceso de desarrollo de aplicaciones de software. De hecho, es la aparición de unas determinadas necesidades lo que motiva el desarrollo de un sistema o aplicación, el cual tiene como fin solucionar esas necesidades o deficiencias detectadas. Los requisitos son estas necesidades plasmadas de forma clara.

Resulta que esta parte de gestión de requisitos es una de las más importantes del proceso de desarrollo y que en mayor o menor medida condicionará el éxito del proceso global. El proceso de gestión de requisitos, no se basa únicamente en la deducción de los mismos, sino que implica más tareas, por lo que surge la necesidad de apoyarse en herramientas que den soporte a todas estas tareas, entre las cuales podemos destacar:

- Recogida de requisitos.
- Formalización de requisitos.
- Revisión.
- Gestión.

Las herramientas que dan soporte a estas tareas son las conocidas como Herramientas de Gestión de Requisitos. [12]

1.3.1. RequisitePro

Rational RequisitePro es una solución fácil de usar, es una herramienta de administración de requerimientos que le permite al equipo crear y compartir sus requerimientos utilizando métodos familiares basados en documentos potenciados por la aplicación de las capacidades de una base de datos, tales como la trazabilidad y análisis de impacto. El resultado es una mejor comunicación y administración de requerimientos con una mayor probabilidad de completar los proyectos en tiempo, dentro del presupuesto y superando las expectativas. Los proyectos exitosos comienzan con una buena administración de requerimientos cuanto más efectiva sea su ejecución, mayor será el resultado en calidad y satisfacción del cliente.

1.3.2. Open Source Requirement Management Tool (Versión 1.5)

Es una herramienta de software libre, llevada a cabo por un único desarrollador y que no ofrece ningún soporte empresarial, permite la descripción avanzada de diversos tipos de requisitos y garantiza la trazabilidad entre todos los documentos relacionados con la ingeniería de requisitos (funcionalidades, requisitos, casos de uso, casos de prueba).

Con el desarrollo en equipo y el chequeo de los documentos involucrados en el proyecto, logra que se conserven juntos todos los requerimientos y que puedan ser manipulados por la totalidad de los miembros del equipo.

Los diversos módulos integrados en la herramienta son:

- Administración y Configuración.
- Gestión de documentos de ingeniería de requisitos (funcionalidades, requisitos, casos de uso, casos de prueba).
- Trazabilidad entre documentos de trabajo.
- Informes y estadísticas.

La herramienta presenta las siguientes ventajas:

- La visualización de requisitos en forma jerárquica es intuitiva y fácil de manejar.
- Existen diversas distribuciones, tanto para un equipo en local como para un servidor de aplicaciones J2EE para permitir desarrollo colaborativo.

- Su licencia es GPL.
- Es un desarrollo basado en Java, por lo que es multiplataforma.
- Como herramienta open source de gestión de requisitos no tiene mucha competencia en cuanto a la funcionalidad ofrecida.
- Tiene una buena documentación pese a tratarse de una herramienta muy reciente.
- Existe un gran soporte para mantener la trazabilidad entre los documentos.
- Existen mecanismos que facilitan la importación y exportación de la información en XML
- Los principales inconvenientes que se han observado son los siguientes:
- Es un desarrollo llevado a cabo por una persona individual, por lo que existe el riesgo de que no sea sostenible a lo largo del tiempo.
- No existe un soporte empresarial.
- No es posible generar automáticamente un documento de requisitos para entregar al cliente.
- Algunas funcionalidades no han sido desarrolladas completamente y están a medias.
- La interfaz de usuario es en ocasiones lenta.

Se realizó un estudio de las características fundamentales de estas herramientas para un mayor nivel de conocimiento y se escoge como herramienta para la gestión de requisitos Open Source Requirement Management Tool por las características mencionadas anteriormente.

1.4. Roles

”**Rol:** Un rol es una definición abstracta de un conjunto de actividades realizadas y de artefactos obtenidos. Los roles son realizados típicamente por un individuo, o un conjunto de individuos, trabajando juntos en equipo. Un miembro del equipo de proyecto cumple normalmente muchos roles. Los roles describen cómo los individuos se comportan en el negocio y qué responsabilidades tienen”.
[13]

Los cuatro roles de Analistas que define RUP son:

- Analista de procesos de negocio.
- Diseñador del negocio.

- Analista de sistema.
- Especificador de requisitos.

En el desarrollo del trabajo se cumplirán específicamente dentro del rol Analista: Analista del Sistema y Especificador de requisitos, por las actividades que se realizan en el flujo de trabajo que se desarrollará y dando continuidad al estudio referente de los principales procesos identificados en el modelamiento del negocio, identificados por el **Analista de procesos de negocio y el Diseñador de negocio** que es el rol que se desempeñó en la tesis anterior. Además se cumplirán los roles de: Diseñador, Diseñador de Interfaz de Usuario, el Especificador de requisitos y Arquitecto que se especifican posteriormente.

- **Analista del sistema:** conduce y coordina las necesidades de estimulación y el modelado de casos de uso, por ejemplo, identificar que actores existen y cual caso de uso se necesita cuando interactúan con el sistema. Como uno de los principales roles en el equipo del proyecto, una persona que desempeña este papel debe ser capaz de colaborar eficazmente con otros miembros del equipo.
- **Diseñador:** identifica y define las responsabilidades, operaciones, atributos, y las relaciones de elementos de diseño. Asegura que el diseño está en consonancia con la arquitectura de software, y se detalla a un punto en que la aplicación puede continuar.
- **Diseñador de Interfaz de Usuario:** conduce y coordina los prototipos y el diseño de la interfaz de usuario, no es responsable de la implementación de dicha interfaz, su papel solo se centra en el diseño y la "conformación visual".
- **El especificador de requisitos:** detalla la especificación de una parte de la funcionalidad del sistema describiendo el aspecto de los requisitos de uno o varios casos de uso y otros requisitos de soporte del software. Puede también ser responsable de un paquete casos de uso, y mantiene la integridad de ese paquete. Se recomienda que el especificador de los requisitos responsable de un paquete de casos de uso sea también responsable de sus casos de uso y actores contenidos.
- **Arquitecto:** es responsable de la arquitectura de software, la cual incluye decisiones técnicas que contrastan con el diseño y la implementación del proyecto.

1.5. Artefactos

“Un producto o artefacto es un trozo de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los productos son los resultados tangibles del proyecto, las cosas que va creando y usando hasta obtener el producto final”. [14]

El Proceso Unificado de Desarrollo, para cada ciclo de iteración, plantea el uso de artefactos con el fin de alcanzar un grado de certificación en el desarrollo del software. En este documento solo se hará referencia a los artefactos que obtienen los roles anteriormente mencionados.

1.5.1. Artefactos a realizar por el Analista del Sistema

Los artefactos realizados por el **Analista del Sistema** son:

Modelo de casos de uso: es un modelo de las funciones del sistema y de su entorno, y sirve como un contrato entre el cliente y los desarrolladores. El modelo de casos de uso se utiliza como un insumo esencial a las actividades de análisis, diseño y prueba.

Contiene:

- Paquetes de Casos de Uso.
- Caso de Uso.
- Actor.

Vista de casos de usos: es un extracto del modelo de casos de usos del sistema. Representa los casos de uso significativos para la arquitectura porque describen alguna funcionalidad importante y crítica o algún requisito que deba priorizarse.

Glosario: define términos importantes utilizados en el proyecto. **(Ver Anexo 4)**

1.5.2. Artefactos a realizar por el Especificador de Requerimientos

Los artefactos realizados por el **Especificador de Requerimientos** son:

Caso de uso: son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.

Paquetes de caso de uso: es una colección de casos de uso, actores, relaciones, diagramas, y otros paquetes; se usa para estructurar al modelo del caso de uso dividiéndolo en las partes más pequeñas.

Especificación de requerimientos del software: documenta tanto los requerimientos funcionales del sistema como los requerimientos no funcionales para el software a desarrollar, y está enfocada en la colección y organización de todos los requisitos que rodean el proyecto. **(Ver Anexo 5)**

1.5.3. Artefactos a realizar por el Diseñador

Realización de casos de uso del diseño: está compuesto por los diagramas de clases del diseño y diagramas de interacción. Es una colaboración en el modelo de diseño que describe como se realiza un caso de uso específico, y cómo se ejecuta en términos de casos de uso del diseño.

Diagramas de Clases del Diseño: 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. Son la base para un par de diagramas relacionados: los diagramas de componentes y los diagramas de despliegue. Son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

Contienen normalmente los siguientes elementos:

- Clases.
- Interfaces.
- Colaboraciones.
- Relaciones de dependencia, generalización y asociación.

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 y destaca la ordenación temporal de los mensajes.

Diseño de Clases: es una parte fundamental de la aproximación de un diseño orientado a objetos.

Paquete de diseño: es una colección de clases, relaciones, realizaciones de caso de uso, diagramas, y otros paquetes. Se usa para estructurar el modelo de diseño la división de este en partes más pequeñas. **(Ver Anexo 6)**

1.5.4. Artefactos a realizar por el Diseñador de Interfaz de Usuario

RUP define dos artefactos a realizar por el Diseñador de interfaz de usuario:

Mapa de Navegación: expresa la estructura de los elementos de la interfaz-usuario en el sistema, junto con sus sendas de navegación potencial. Hace fácil de ver cuántos "clicks" tomará a un usuario para conseguir llegar a una pantalla específica.

En el desarrollo del trabajo se confecciona el mapa de navegación para el laboratorio Análisis Químico.

Prototipo de interfaz de usuario: este artefacto representa un ejemplo visual de la interfaz de usuario que se construye para explorar y validar el diseño de interfaz de usuario, permitiéndole al cliente verificar que el sistema satisfaga sus necesidades. **(Ver Anexo 7)**

1.5.5. Artefactos a realizar por el Arquitecto

Vista lógica: representa un subconjunto del artefacto modelo de diseño, la cual representa los elementos de diseño más significativos para la arquitectura del sistema.

Vista de despliegue: se utiliza para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos.

1.6. Patrones

"Un patrón es un modelo que podemos seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos al tratar de lograr ciertos objetivos." [15]

Un patrón es una unidad que se construye a partir de soluciones de problemas semejantes en contextos semejantes y que han sido probadas. Los patrones más generales son más difíciles de aplicar pero a su vez pueden aplicarse a una mayor variedad de problemas, en tanto, los más específicos, en la práctica son más viables pero se reduce su capacidad de utilización. Los patrones identificados en el desarrollo de la tesis se muestran en los epígrafes posteriores.

1.6.1. Patrones de Casos de Usos

Creating, Reading, Updating and Deleting (**Patrón CRUD**): propone identificar un CU, llamado “Información CRUD” o “Administrar Información”, que modela todas las operaciones que se pueden realizar sobre una parte de información de cierto tipo (o sea una misma entidad), tal como crearla, leerla, actualizarla y eliminarla.

Aplicación: debe ser usado cuando todos los flujos contribuyen al mismo valor de negocio, son cortos y sencillos.

Patrón Actores múltiples: Rol común (Multiple Actors: Common Role): dos actores juegan el mismo papel hacia el caso de uso. Este rol es representado por otro actor, heredado por los actores que comparten este rol.

Aplicación: este patrón es aplicable cuando, desde el punto de vista de un caso de uso hay sólo una entidad externa interactuando con cada instancia del caso de uso.

1.6.2. Patrones de Diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general, es una solución estándar para un problema común de programación y una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios. Dentro de los patrones de diseño que se usan para la arquitectura del framework Symfony:

Modelo-Vista-Controlador (**MVC**): que está formado por tres niveles:

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

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación

debe ejecutarse tanto en un navegador estándar como 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). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. [16] **(Ver Anexo 7)**

1.6.2.1. Patrones GRASP

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades), describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones, 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).

1.6.2.2. Patrones GOF

El grupo de GoF clasificaron los patrones en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

- **Creacionales:** tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.

- **Estructurales:** describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos.
- **Comportamiento:** ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos.

Dentro de los patrones estructurales se pone de manifiesto:

Decorator: es otro de los patrones de diseño utilizados por Symfony, es implementado 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.

También conocido como Wrapper, a veces se desea adicionar responsabilidades a un objeto, pero no a toda la clase. Las responsabilidades se pueden adicionar por medio de los mecanismos de herencia, pero este mecanismo no es flexible porque la responsabilidad es adicionada estáticamente. La solución flexible es la de rodear el objeto con otro objeto que es el que adiciona la nueva responsabilidad. Este nuevo objeto es el Decorator, este patrón se relaciona con otros patrones los cuales son: Adapter, Composite y el Strategy.

Aplicación:

- El Decorator se debe usar para:
- 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 Decorator

- 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. [17]

1.7. Framework: Symfony (Versión: 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, 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, porque 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. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación Web.

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) como en plataformas Windows. Symfony está basado en el patrón clásico del diseño Web conocido como arquitectura MVC. A continuación se muestran algunas de sus características.

1.7.1. Características de Symfony

- Symfony se diseñó para que se ajustara a los siguientes requisitos:
- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos.

- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la Web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.
- Automatización de características de proyectos Web. [18]

Conclusiones

En el CIGB se desarrollan actividades de suma importancia, para ello se necesita de una gran eficiencia en cuanto a la manipulación de la información y se hace necesario el desarrollo de un Sistema de Gestión de la Información, como es el caso de los LIMS. Después del análisis de los procesos, se identificaron aquellos que necesitan ser informatizados, para ello fueron seleccionados: como proceso de desarrollo de software RUP, herramienta CASE para la realización de los artefactos Visual Paradigm, para la trazabilidad de los requisitos Open Source Requirement Management Tool y como framework Symfony.

Capítulo 2

Características del Sistema

Introducción

En el presente capítulo se describe cómo se logran los objetivos: definir el ámbito del sistema, una interfaz de usuario para el sistema enfocada a las necesidades y metas del usuario, establecer y mantener un acuerdo entre clientes y otros involucrados sobre lo que el sistema debería hacer, proveer a los desarrolladores un mejor entendimiento de los requerimientos del sistema y de una base para estimar recursos y tiempo de desarrollo del sistema, mediante los requisitos funcionales y no funcionales. Serán mostradas en este capítulo las descripciones textuales resumidas de tres de los casos de uso más significativos del módulo Análisis Químico (MAQ) y solo una descripción textual en su forma ampliada que será la que se tomará de ejemplo para ilustrar los artefactos del flujo de trabajo de Diseño. En el *Expediente de proyecto* serán presentadas las restantes descripciones textuales en su forma ampliada, así como los CUS y la matriz de trazabilidad entre (Actores- CUS) y la dependencia entre (RF-CUS).

2.1. Requerimientos

El flujo de trabajo **Requerimientos** del Proceso Unificado de Desarrollo (RUP, Rational Unified Process) tiene su mayor peso en la fase de Inicio, aunque se continúan refinando los requisitos a lo largo de todo el ciclo de vida. Además es muy importante para la elaboración de un software porque en él los desarrolladores identifican y clasifican los requerimientos, encuentran actores y casos de uso, los priorizan y los detallan, se encargan de prototipar la interfaz de usuario, y estructurar el modelo de casos de uso del sistema.

2.1.1. Requerimientos funcionales y no funcionales del sistema

Los requisitos se pueden clasificar en: funcionales y no funcionales, deben ser descritos y en la fase de construcción deben ser posibles de probar o verificar.

2.1.1.1 Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, los requerimientos funcionales que se identificaron en el módulo Análisis Químico son:

1. Gestionar Registro de preparación de soluciones (SIC-0020).
 - 1.1. Crear nuevo SIC-0020.
 - 1.2. Buscar y visualizar SIC-0020.
 - 1.3. Modificar SIC-0020.
 - 1.3.1 Registrar traza.
 - 1.4. Imprimir SIC-0020.

2. Gestionar Registro de Reensayo (SIC-0814).
 - 2.1. Crear nuevo SIC-0814.
 - 2.2. Buscar y visualizar SIC-0814.
 - 2.3. Modificar SIC-0814.
 - 2.3.1 Registrar traza.
 - 2.4. Imprimir SIC-0814.

3. Gestionar Resultado de la Determinación de proteínas por el método de Lowry (SIC-0107).
 - 3.1. Crear nuevo SIC-0107.
 - 3.2. Buscar y visualizar SIC-0107.
 - 3.3. Modificar SIC-0107.
 - 3.3.1 Registrar traza.
 - 3.4. Imprimir SIC-0107.

4. Gestionar Curva de calibración de Lowry (SIC-0107C).
 - 4.1. Crear nuevo SIC-0107C.

4.2. Buscar y visualizar SIC-0107C.

4.3. Modificar SIC-0107C.

4.3.1 Registrar traza.

4.4. Imprimir SIC-0107C.

5. Gestionar Curva de calibración de carbohidratos (SIC-0112C).

5.1. Crear nuevo SIC-0112C.

5.2. Buscar y visualizar SIC-0112C.

5.3. Modificar SIC-0112C.

5.3.1 Registrar traza.

5.4. Imprimir SIC-0112C.

6. Gestionar Curva de calibración de lípidos (SIC-0113C).

6.1. Crear nuevo SIC-0113C.

6.2 Buscar y visualizar SIC-0113C.

6.3. Modificar SIC-0113C.

6.3.1 Registrar traza.

6.4. Imprimir SIC-0113C.

7. Gestionar Curva de calibración de Tiocianato (SIC-0116C).

7.1. Crear nuevo SIC-0116C.

7.2. Buscar y visualizar SIC-0116C.

7.3. Modificar SIC-0116C.

7.3.1 Registrar traza.

7.4. Imprimir SIC-0116C.

8. Gestionar Curva de calibración de Formaldehído (SIC-0172C).

8.1. Crear nuevo SIC-0172C.

8.2. Buscar y visualizar SIC-0172C.

8.3. Modificar SIC-0172C.

8.3.1 Registrar traza.

8.4. Imprimir SIC-0172C.

9. Gestionar Curva de calibración de Orcinol (SIC-0762C).

9.1. Crear nuevo SIC-0762C.

9.2. Buscar y visualizar SIC-0762C.

9.3. Modificar SIC-0762C.

9.3.1 Registrar traza.

9.4. Imprimir SIC-0762C.

10. Gestionar Resultado de la Determinación de Volumen (SIC-0764).

10.1 Crear nuevo SIC-0764.

10.2 Buscar y visualizar SIC-0764.

10.3 Modificar SIC-0764.

10.3.1 Registrar traza.

10.4 Imprimir SIC-0764.

11. Gestionar Curva de calibración de Grupos sulfhidrilos (SIC-0759C).

11.1. Crear nuevo SIC-0759C.

11.2. Buscar y visualizar SIC-0759C.

11.3. Modificar SIC-0759C.

11.3.1 Registrar traza.

11.4. Imprimir SIC-0759C.

12. Gestionar Curva de calibración de fósforo (SIC-1001C).

12.1. Crear nuevo SIC-1001C.

12.2. Buscar y visualizar SIC-1001C.

12.3. Modificar SIC-1001C.

12.3.1 Registrar traza.

12.4. Imprimir SIC-1001C.

13. Gestionar Curva de calibración de Ácido Siálico (SIC-0760C).

13.1. Crear nuevo SIC-0760C.

13.2. Buscar y visualizar SIC-0760C.

13.3. Modificar SIC-0760C.

13.3.1 Registrar traza.

13.4. Imprimir SIC-0760C.

14. Gestionar Curva de calibración de Biuret (SIC-0710C).

14.1. Crear nuevo SIC-0710C.

14.2. Buscar y visualizar SIC-0710C.

14.3. Modificar SIC-0710C.

14.3.1 Registrar traza.

14.4. Imprimir SIC-0710C.

15. Gestionar Curva de calibración de Tween (SIC-0761C).

15.1. Crear nuevo SIC-0761C.

15.2. Buscar y visualizar SIC-0761C.

15.3. Modificar SIC-0761C.

15.3.1 Registrar traza.

15.4. Imprimir SIC-0761C.

16. Gestionar Curva de calibración del ácido Bicinconónico (SIC-0763C).

16.1. Crear nuevo SIC-0763C.

16.2. Buscar y visualizar SIC-0763C.

16.3. Modificar SIC-0763C.

16.3.1 Registrar traza.

16.4. Imprimir SIC-0763C.

17. Gestionar Libro de entrada de cuarentena.

17.1 Crear Libro de entrada de cuarentena.

17.2 Buscar y visualizar Libro de entrada de cuarentena.

17.3 Modificar Libro de entrada de cuarentena.

17.3.1 Registrar traza.

17.4 Registrar datos en el Libro de entrada de cuarentena.

17.5 Imprimir datos del Libro de entrada de cuarentena.

18. Gestionar Libro del control de desempeño.

18.1 Registrar datos en el libro del control de desempeño.

18.2 Buscar y visualizar libro del control de desempeño.

18.3 Modificar Libro del control de desempeño.

18.3.1 Registrar traza.

18.4 Imprimir datos del Libro del control de desempeño.

19. Gestionar Aviso de Autorización (SIC-0014).

19.1 Crear SIC-0014.

19.2 Buscar y visualizar SIC-0014.

19.3 Modificar datos en SIC-0014.

19.3.1 Registrar traza.

19.4 Imprimir SIC-0014.

20. Gestionar Certificado de Análisis de Materias Primas (SIC-0225).

20.1 Crear SIC-0225.

20.2 Buscar y visualizar SIC-0225.

20.3 Modificar SIC-0225.

20.3.1 Registrar traza.

20.4 Imprimir SIC-0225.

21. Gestionar Aviso de Decisión (SIC-0048).

21.1 Crear SIC-0048.

21.2 Buscar y visualizar SIC-0048.

21.3 Modificar datos en SIC-0048.

21.3.1 Registrar traza.

21.4 Imprimir SIC-0048.

22. Gestionar Resultado del análisis por cromatografía de capa fina (SIC-1000).

22.1 Crear nuevo SIC-1000.

22.2 Buscar y visualizar SIC-1000.

22.3 Modificar datos en SIC-1000.

22.3.1 Registrar traza.

22.4 Imprimir SIC-1000.

23. Gestionar Resultado de la Determinación de PH (SIC-0133).

23.1 Crear nuevo SIC-0133.

23.2 Buscar y visualizar SIC-0133.

23.3 Modificar datos en SIC-0133.

23.3.1 Registrar traza.

23.4 Imprimir SIC-0133.

24. Gestionar Resultado de la Determinación de la concentración de Formaldehído (SIC-0172).

24.1 Crear nuevo SIC-0172.

24.2 Buscar y visualizar SIC-0172.

24.3 Modificar datos en SIC-0172.

24.3.1 Registrar traza.

24.4 Imprimir SIC-0172.

25. Gestionar Resultado de la Determinación de la concentración de aluminio (SIC-0105).

25.1 Crear nuevo SIC-0105.

25.2 Buscar y visualizar SIC-0105.

25.3 Modificar SIC-0105.

25.3.1 Registrar traza.

25.4 Imprimir SIC-0105.

26. Gestionar Resultado de la Determinación de concentración de carbohidratos por el método de Antrona (SIC-0112).

26.1 Crear nuevo SIC-0112.

26.2 Buscar y visualizar SIC-0112.

26.3 Modificar SIC-0112.

26.3.1 Registrar traza.

26.4 Imprimir SIC-0112.

27. Gestionar Resultado de la Determinación del contenido de Lípidos (SIC-0113).

27.1 Crear nuevo SIC-0113.

27.2 Buscar y visualizar SIC-0113.

27.3 Modificar datos en SIC-0113.

27.3.1 Registrar traza.

27.4 Imprimir SIC-0113.

28. Gestionar Resultado de la Determinación de la concentración de Tiocianato (SIC-0116).

28.1 Crear nuevo SIC-0116.

28.2 Buscar y visualizar SIC-0116

28.3 Modificar en SIC-0116.

28.3.1 Registrar traza.

28.4 Imprimir SIC-0116.

29. Gestionar Resultado de la Determinación de la concentración de proteínas Absorbancia (SIC-0118).

29.1 Crear nuevo SIC-0118.

29.2 Buscar y visualizar SIC-0118.

29.3 Modificar en SIC-0118.

29.3.1 Registrar traza.

29.4 Imprimir SIC-0118.

30. Gestionar Resultado de la Determinación de la concentración de Timerosal (SIC-0104).

30.1 Crear nuevo SIC-0104.

30.2 Buscar y visualizar SIC-0104.

30.3 Modificar SIC-0104.

30.3.1 Registrar traza.

30.4 Imprimir SIC-0104.

31. Gestionar Resultado de la Determinación del % de humedad por el método de Karl Fischer (SIC-0129).

31.1 Crear nuevo SIC-0129

31.2 Buscar y visualizar SIC-0129.

31.3 Modificar SIC-0129.

31.3.1 Registrar traza.

31.4 Imprimir SIC-0129.

32. Gestionar Libro de entrada de muestras de producción.

32.1 Crear libro de entrada de muestras de producción.

32.2 Registrar datos en el Libro de entrada de muestras de producción.

32.3 Buscar y visualizar Libro de entrada de muestras de producción.

32.4 Modificar datos en el Libro de entrada de muestras de producción.

32.4.1 Registrar traza.

32.5 Imprimir datos del Libro de entrada de muestras de producción.

33. Gestionar Resultado de la Determinación de Carbohidratos por el método de Orcinol (SIC-0762).

33.1 Crear nuevo SIC-0762.

33.2 Buscar y visualizar SIC-0762.

33.3 Modificar SIC-0762.

33.3.1 Registrar traza.

33.4 Imprimir SIC-0762.

34. Gestionar Resultado de la Determinación de proteínas por el método de Acido Bicinconínico (SIC-0763).

34.1 Crear nuevo SIC-0763.

34.2 Buscar y visualizar SIC-0763.

34.3 Modificar SIC-0763.

34.3.1 Registrar traza.

34.4 Imprimir SIC-0763.

35. Gestionar Resultado de la Determinación colorimétrica de fósforo por el método de Chen (SIC-1001).

35.1 Crear nuevo SIC-1001.

35.2 Buscar y visualizar SIC-1001.

35.3 Modificar SIC-1001.

35.3.1 Registrar traza.

35.4 Imprimir SIC-1001.

36. Gestionar Análisis de los resultados fuera de especificación (SIC-0837).

36.1 Crear nuevo SIC-0837.

36.2 Buscar y visualizar SIC-0837.

36.3 Modificar SIC-0837.

36.3.1 Registrar traza.

36.4 Imprimir SIC-0837.

37. Gestionar Resultado de la Determinación del porcentaje de incorporación del PRP (SIC-1009).

37.1 Crear nuevo SIC-1009.

37.2 Buscar y visualizar SIC-1009.

37.3 Modificar SIC-1009.

37.3.1 Registrar traza.

37.4 Imprimir SIC-1009.

38. Gestionar Resultado de la Determinación de la concentración de Tween (SIC-0761).

38.1 Crear nuevo SIC-0761.

38.2 Buscar y visualizar SIC-0761.

38.3 Modificar SIC-0761.

38.3.1 Registrar traza.

38.4 Imprimir SIC-0761.

39. Gestionar Resultado de la Determinación de ácido Siálico (SIC-0760).

39.1 Crear nuevo SIC-0760.

39.2 Buscar y visualizar SIC-0760.

39.3 Modificar SIC-0760.

39.3.1 Registrar traza.

39.4 Imprimir SIC-0760.

40. Gestionar Resultado de la Determinación de grupo sulfhidrilos (SIC-0759).

40.1 Crear nuevo SIC-0759.

40.2 Buscar y visualizar SIC-0759.

40.3 Modificar SIC-0759.

40.3.1 Registrar traza.

40.4 Imprimir SIC-0759.

41. Gestionar Resultado de la Determinación de la concentración de proteínas por el método de Biuret (SIC-0710).

41.1 Crear nuevo SIC-0710.

41.2 Buscar y visualizar SIC-0710.

41.3 Modificar SIC-0710.

41.3.1 Registrar traza.

41.4 Imprimir SIC-0710.

42. Gestionar Resultado de la Concentración de proteínas en geles de hidróxido de aluminio (SIC-0185).

42.1 Crear nuevo SIC-0185.

42.2 Buscar y visualizar SIC-0185.

42.3 Modificar SIC-0185.

42.3.1 Registrar traza.

42.4 Imprimir SIC-0185.

2.1.1.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades se ven como las características que hacen al producto atractivo, usable, rápido o confiable, los requerimientos no funcionales que se identificaron son:

- **Apariencia o interfaz externa.**

El sistema, tendrá los colores correspondientes al logo del CIGB. Las páginas de la aplicación no se cargarán con mucha información y contendrán solo las imágenes necesarias.

- **Usabilidad.**

El sistema debe permitir a los usuarios un acceso fácil y rápido. Podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de una computadora y de un ambiente Web.

- **Rendimiento.**

Los tiempos de respuestas deben ser rápidos, al igual que la velocidad de procesamiento de la información.

- **Soporte.**

Se requiere de la instalación de un servidor Web que interprete código PHP 5 o superior.

Las computadoras clientes requerirán de un navegador capaz de interpretar código Java Script.

- **Portabilidad.**

El sistema podrá ser usado sobre los sistemas operativos Windows y Linux.

- **Seguridad.**

Se debe garantizar que la información sensible sólo pueda ser vista por los usuarios con el nivel de acceso adecuado y que las funcionalidades del sistema se muestren de acuerdo al usuario que esté activo.

El sistema debe contar con protección contra acciones no autorizadas o que puedan afectar la integridad de los datos; y con un registro de trazas de los documentos modificados por los usuarios, para garantizar el control de las operaciones de este tipo.

Se podrá acceder a algunas funcionalidades del sistema desde cualquier computadora personal que esté fuera del CIGB.

- **Políticos-culturales.**

El idioma que se empleará en la aplicación será el español.

El sistema tendrá logotipos e imágenes en correspondencia con el carácter científico y profesional del CIGB.

Algún cambio que se quiera realizar en la aplicación, será tramitado por la dirección del proyecto por parte del CIGB y canalizado por los directivos de Producción de la Universidad de las Ciencias Informáticas.

- **Legales.**

El sistema se dará a conocer a todos los trabajadores de la Dirección de Calidad como la herramienta para gestionar la información de cada uno de los grupos y laboratorios.

Se estará usando para el desarrollo de la aplicación herramientas de software libre con licencia GNU/GPL.

- **Confiabilidad.**

El sistema será usado y administrado solamente por trabajadores de la Dirección de Calidad del CIGB, por lo tanto la información que fluirá en el mismo, será la emitida por cada uno de los grupos y laboratorios.

Podrán acceder a visualizar ciertas informaciones, directivos de otras áreas, con previa consulta a la dirección del proyecto y a los desarrolladores de la aplicación.

El sistema validará la entrada de datos para evitar entradas inadecuadas.

- **Software.**

Se deberá disponer, para instalar la aplicación, del Sistema Operativo Windows 98 o superior, o cualquier distribución de Linux. Las computadoras clientes de los usuarios accederán al sistema usando uno de los siguientes navegadores: Internet Explorer 5.5 o superior, Netscape, Mozilla 1.7 o superior o FireFox 0.9.3 o superior.

Para el servidor de la aplicación el Sistema Operativo recomendado es Windows Server 2003 o superior o Linux.

Se debe instalar un servidor Web Apache 1.3 o superior.

- **Hardware.**

Se deberá contar con impresora y escáner en las computadoras clientes que interactúen con la aplicación.

Se deberán incluir más computadoras personales en los grupos y laboratorios de la Dirección de Calidad.

El servidor debe tener las siguientes características: capacidad de disco duro superior a 80.0 GB, microprocesador Pentium IV superior a 2.0 GHz y como mínimo 1.0 GB de RAM.

- **Restricciones en el diseño y la implementación.**

La lógica de presentación constituirá una capa independiente de la lógica de negocio, centrandose su función en la interfaz de usuario y validaciones de los datos de entrada.

Se utilizarán herramientas de desarrollo que garanticen la calidad de todo el ciclo de desarrollo del producto.

Se usará el lenguaje de programación PHP 5, como framework de desarrollo Symfony (Versión 1.0.11), y como gestor de bases de datos PostgreSQL (Versión 8.2).

2.1.2. Actores del sistema a automatizar: Módulo Análisis Químico (MAQ)

Entre los actores y trabajadores que se identificaron en el negocio se encuentran:

Actores:

- Almacén: provee a Análisis Químico las materias primas para que sean analizadas.

Trabajadores:

- Jefe de Grupo: es quien tiene firma autorizada dentro del Laboratorio Análisis Químico.
- Analista (Inspector, Auditor): son quienes realizan los ensayos a las muestras y documentan todo el proceso.
- Grupos dentro de Calidad:
- Recepción de Muestras y Desarrollo: proveen las muestras a Análisis Químico para la realización de los ensayos y finalmente reciben los resultados.

Después de analizar los actores y trabajadores que intervienen en el negocio del módulo Análisis Químico: se han definido los siguientes actores del sistema para dicho módulo.

Definición de actores del sistema a automatizar: Módulo Análisis Químico (MAQ)

Nombre del actor	Descripción
Analista	Ejecuta las técnicas analíticas y reporta los resultados a Liberación Análítica, cumpliendo con las Buenas Prácticas.
Jefe de Grupo	Puede ejecutar cualquier técnica analítica, es el único autorizado a modificar los registros antes de entregarlos a Liberación Análítica y garantiza que se cumpla con los procedimientos aprobados y con las Buenas Prácticas.

Tabla 2.1 Descripción de los actores del sistema.

2.1.3. Casos de Uso del Sistema

Se muestran a continuación los Casos de Uso del sistema que se identificaron para el módulo Análisis Químico, basados en los requerimientos expuestos con anterioridad:

1. Gestionar Registro de preparación de soluciones (SIC-0020).
2. Gestionar Registro de Reensayo (SIC-0814).
3. Gestionar Resultado de la Determinación de proteínas por el método de Lowry (SIC-0107).
4. Gestionar Curva de calibración de Lowry (SIC-0107C).
5. Gestionar Curva de calibración de carbohidratos (SIC-0112C).
6. Gestionar Curva de calibración de lípidos (SIC-0113C).
7. Gestionar Curva de calibración de Tiocianato (SIC-0116C).
8. Gestionar Curva de calibración de Formaldehído (SIC-0172C).
9. Gestionar Curva de calibración de Orcinol (SIC-0762C).
10. Gestionar Resultado de la Determinación de Volumen (SIC-0764).
11. Gestionar Curva de calibración de Grupos sulfhidrilos (SIC-0759C).
12. Gestionar Curva de calibración de fósforo (SIC-1001C).
13. Gestionar Curva de calibración de Ácido Siálico (SIC-0760C).
14. Gestionar Curva de calibración de Biuret (SIC-0710C).
15. Gestionar Curva de calibración de Tween (SIC-0761C).
16. Gestionar Curva de calibración del ácido Bicinconínico (SIC-0763C).

17. Gestionar Libro de entrada de cuarentena.
18. Gestionar Libro del control de desempeño.
19. Gestionar Aviso de Autorización (SIC-0014).
20. Gestionar Certificado de Análisis de Materias Primas (SIC-0225).
21. Gestionar Aviso de Decisión (SIC-0048).
22. Gestionar Resultado del análisis por cromatografía de capa fina (SIC-1000).
23. Gestionar Resultado de la Determinación de PH (SIC-0133).
24. Gestionar Resultado de la Determinación de la concentración de Formaldehido (SIC-0172).
25. Gestionar Resultado de la Determinación de la concentración de aluminio (SIC-0105).
26. Gestionar Resultado de la Determinación de concentración de carbohidratos por el método de Antrona (SIC-0112).
27. Gestionar Resultado de la Determinación del contenido de Lípidos (SIC-0113).
28. Gestionar Resultado de la Determinación de la concentración de Tiocianato (SIC-0116).
29. Gestionar Resultado de la Determinación de la concentración de proteínas Absorbancia (SIC-0118).
30. Gestionar Resultado de la Determinación de la concentración de Timerosal (SIC-0104).
31. Gestionar Resultado de la Determinación del % de humedad por el método de Karl Fischer (SIC-0129).
32. Gestionar Libro de entrada de muestras de producción.
33. Gestionar Resultado de la Determinación de Carbohidratos por el método de Orcinol (SIC-0762).
34. Gestionar Resultado de la Determinación de proteínas por el método de Acido Bicinconínico (SIC-0763).
35. Gestionar Resultado de la Determinación colorimétrica de fósforo por el método de Chen (SIC-1001).
36. Gestionar Análisis de los resultados fuera de especificación (SIC-0837).
37. Gestionar Resultado de la Determinación del porcentaje de incorporación del PRP (SIC-1009).
38. Gestionar Resultado de la Determinación de la concentración de Tween (SIC-0761).
39. Gestionar Resultado de la Determinación de ácido Siálico (SIC-0760).
40. Gestionar Resultado de la Determinación de grupo sulfhidrilos (SIC-0759).
41. Gestionar Resultado de la Determinación de la concentración de proteínas por el método de Biuret (SIC-0710).
42. Gestionar Resultado de la Concentración de proteínas en geles de hidróxido de aluminio (SIC-0185).

2.1.4. Matriz de trazabilidad

La matriz de trazabilidad es necesaria para poder medir el impacto que un proyecto tendrá al momento de realizar un cambio y así poder estimar y una vez aprobado el cambio, saber que áreas se deben cambiar.

Beneficios que brinda la matriz:

- Ayuda a demostrar que la validación se encuentra completa.
- Posee un gran valor agregado en el momento de las auditorias y las inspecciones mostrando a través de la misma un mapa general de la validación.

Para ver la matriz de trazabilidad del módulo Análisis Químico, remitirse al *Expediente de proyecto*.

2.1.5. Diagrama de Casos de Uso del Sistema: Laboratorio Análisis Químico (LAQ)

Teniendo en cuenta la extensibilidad del sistema se estructuró el diagrama de casos de uso del sistema en paquetes que en su interior contienen los actores del sistema y los Casos de Uso fuertemente relacionados, éstos últimos son todos significativos, por lo que el diagrama de Casos de Uso del sistema coincide con la Vista de casos de uso.

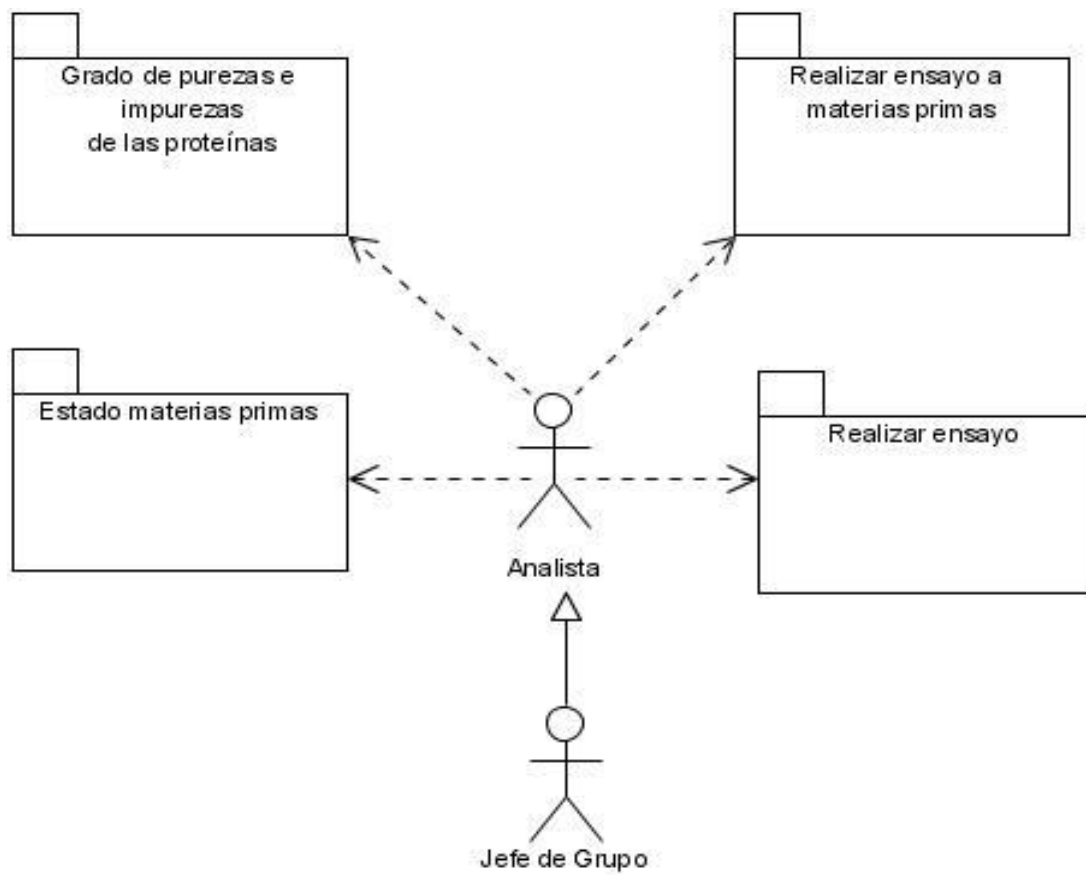


Figura 2.1 Diagrama de Casos de Uso del Sistema.

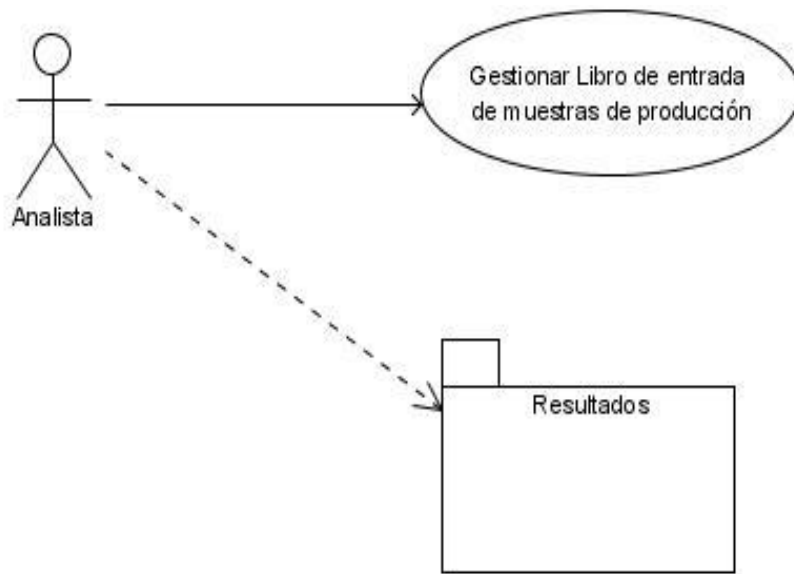


Figura 2.2 Paquete “Grado de purezas e impurezas de las proteínas”.

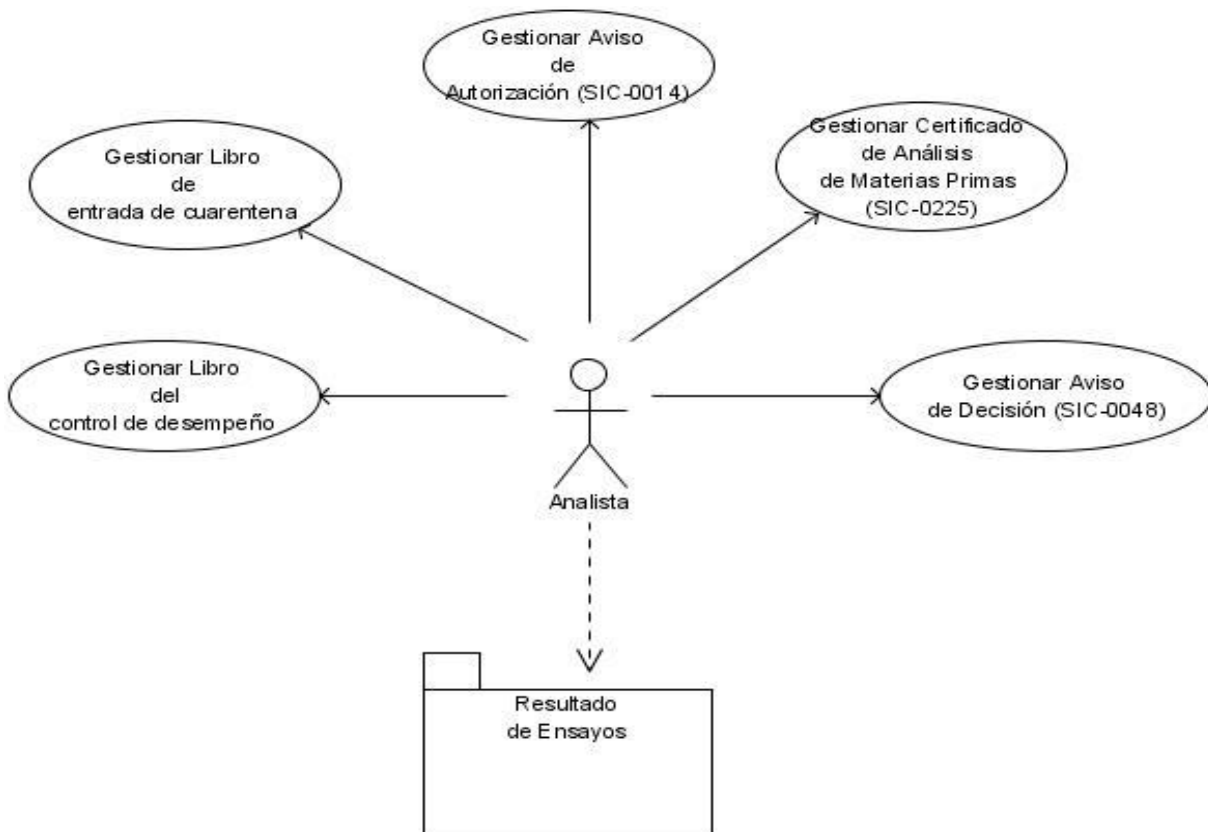


Figura 2.3 Paquete “Estado materias primas”.

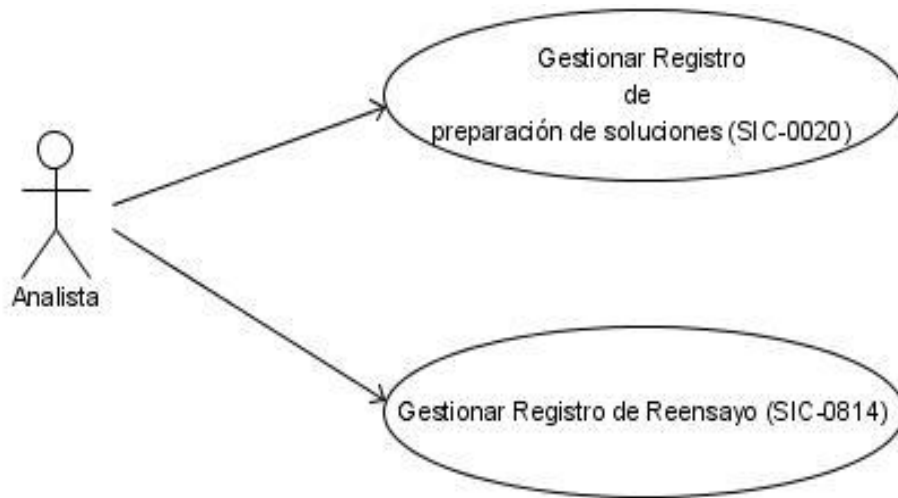


Figura 2.4 Paquete “Realizar ensayo a materias primas”.

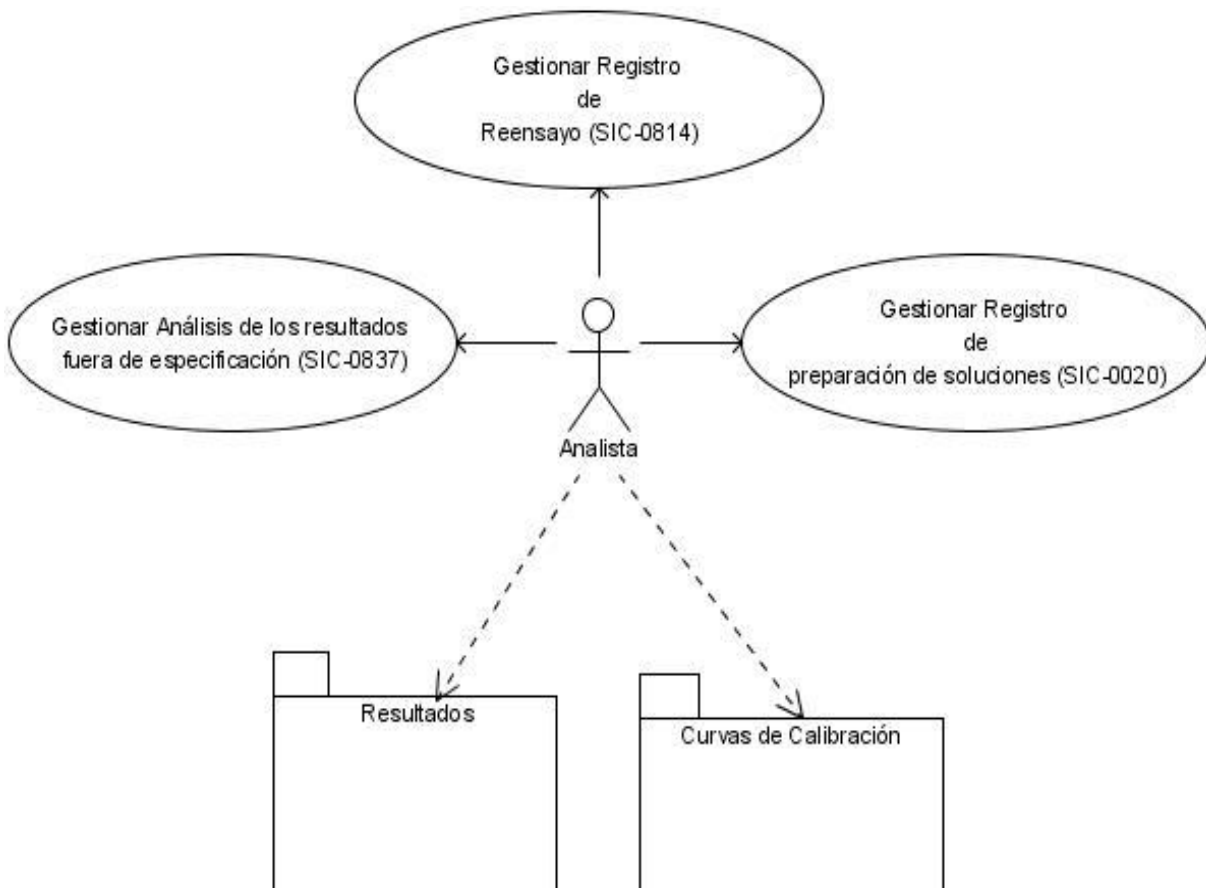


Figura 2.5 Paquete “Realizar ensayo”.

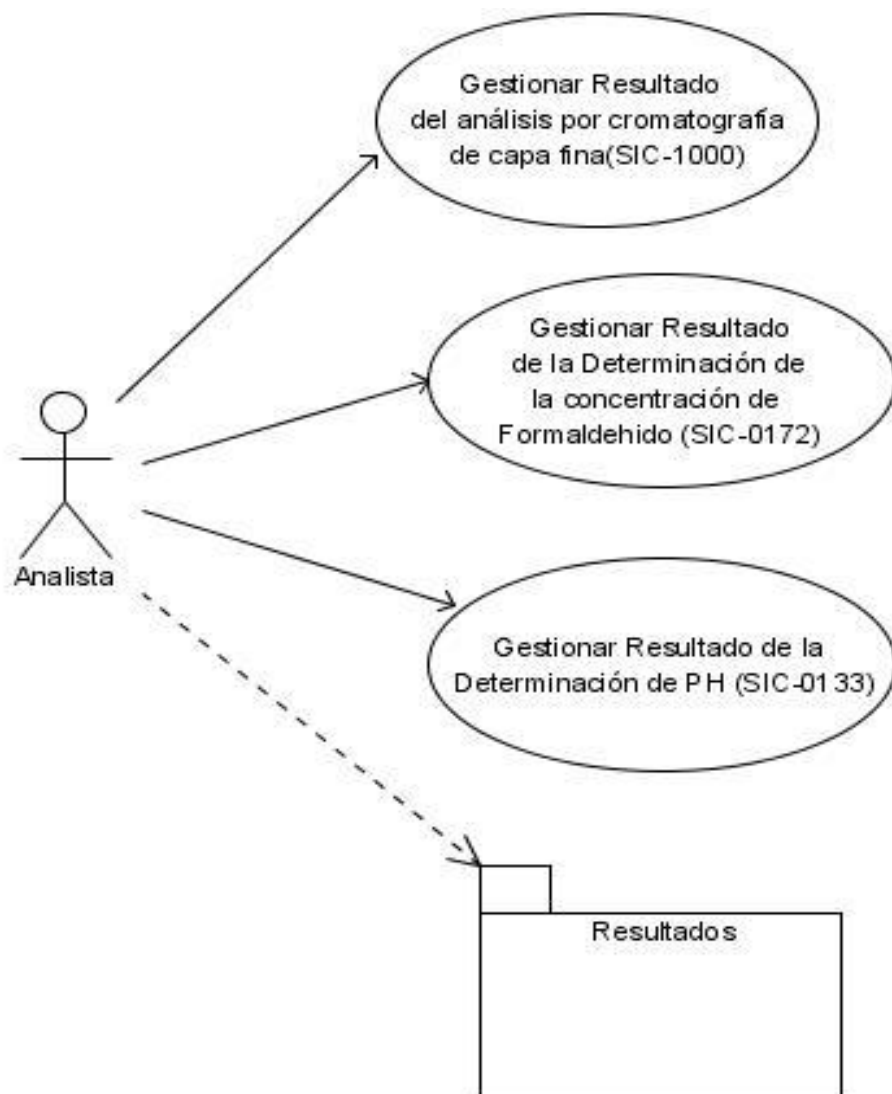


Figura 2.6 Paquete "Resultado de Ensayos".

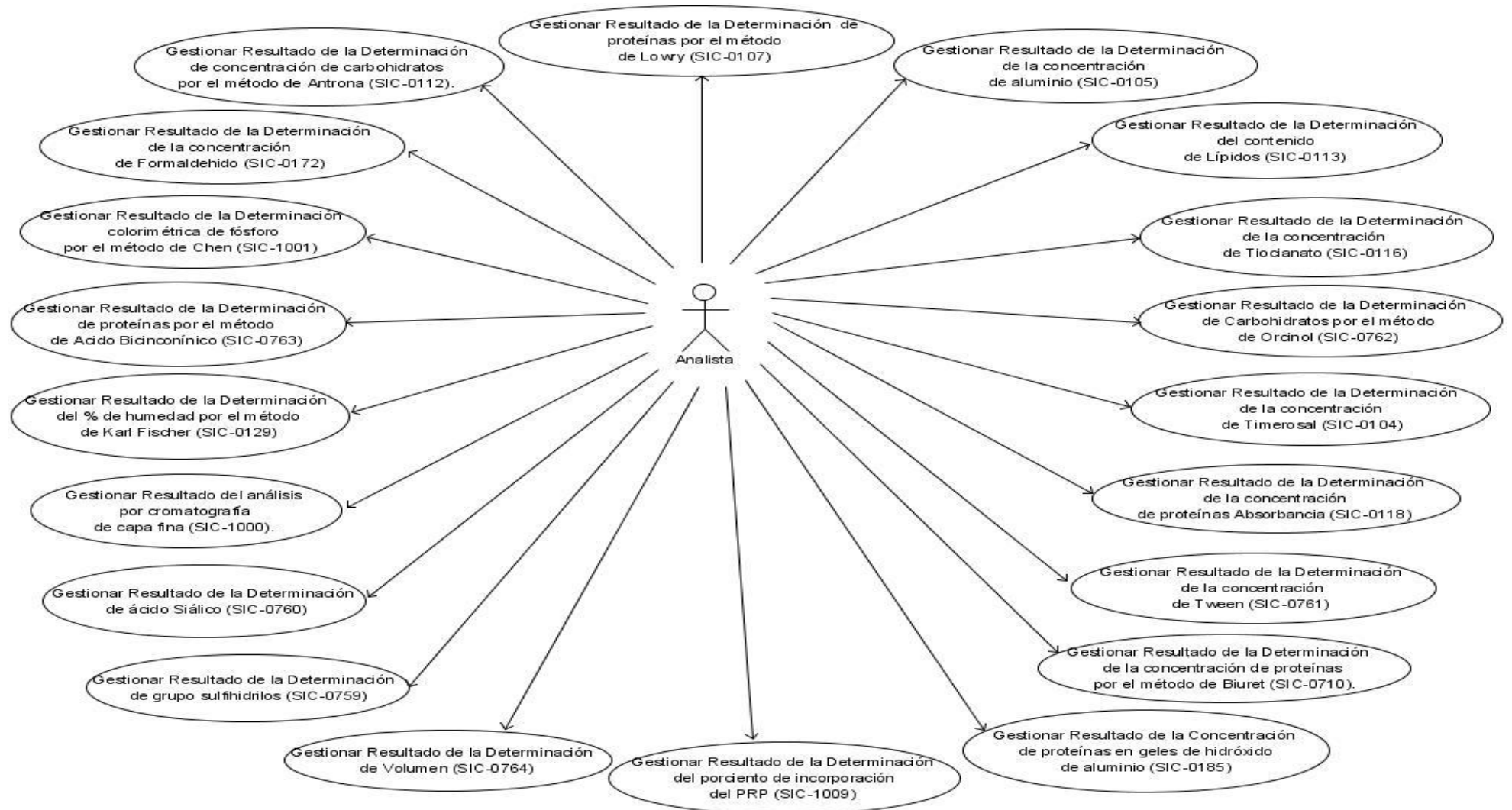


Figura 2.7 Paquete "Resultados".

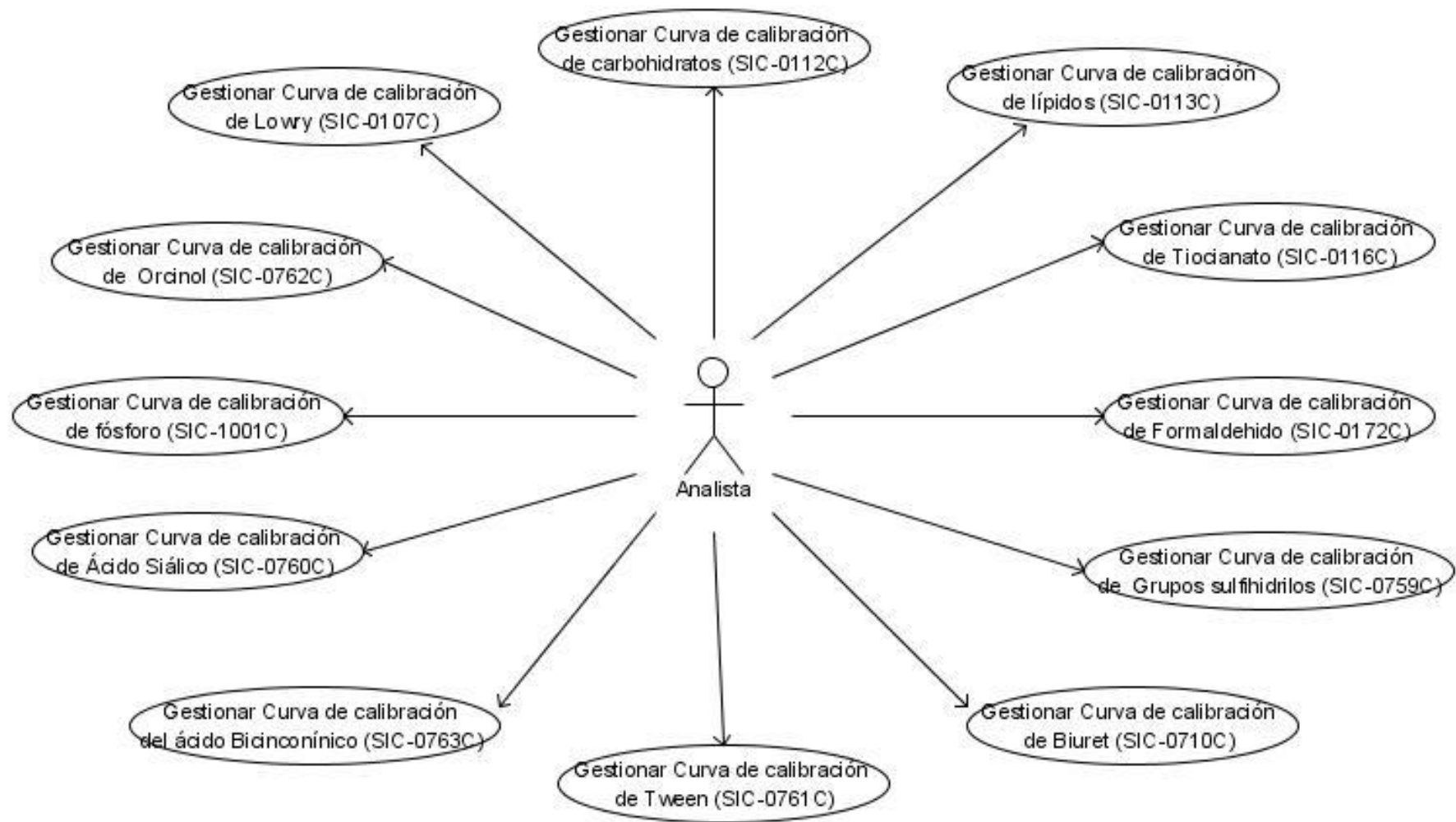


Figura 2.8 Paquete "Curvas de Calibración".

2.2. Descripciones textuales resumidas

A continuación se muestran las descripciones textuales en su forma resumida de los Casos de Uso: Gestionar Registro de preparación de soluciones (SIC-0020), Gestionar Libro del control de desempeño y Gestionar Libro de entrada de muestras de producción.

Nombre del Caso de Uso	Gestionar Registro de preparación de soluciones (SIC-0020).
Actores	Analista (Inicia).
Propósito	Crear el SIC-0020, modificar el SIC-0020; buscar y visualizar el SIC-0020; imprimir SIC-0020.
Resumen	<p>El caso de uso se inicia cuando el Analista va a realizar alguna de las siguientes acciones relacionadas con el Registro de Preparación de Soluciones :</p> <ul style="list-style-type: none"> - Crear nuevo SIC-0020. - Buscar y visualizar SIC-0020. <p>El sistema muestra la interfaz correspondiente según la solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.</p>

Tabla 2.2 Descripción del CU “Gestionar Registro de preparación de soluciones (SIC-0020)”.

Nombre del Caso de Uso	Gestionar Libro del control de desempeño.
Actores	Analista (Inicia).
Propósito	Registrar datos, modificar sus datos, buscar datos.
Resumen	<p>El caso de uso se inicia cuando el Analista va a realizar alguna de las siguientes acciones relacionadas con el Libro del control de desempeño:</p> <ul style="list-style-type: none"> - Registrar en el libro. - Buscar y visualizar libro. <p>El sistema muestra la interfaz correspondiente a la solicitud y</p>

	ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.
--	--

Tabla 2.3 Descripción del CU "Gestionar Libro del control de desempeño".

Nombre del Caso de Uso	Gestionar Libro de entrada de muestras de producción.
Actores	Analista (Inicia).
Propósito	Crear un nuevo libro, registrar datos, modificar sus datos, buscar datos y visualizar libro, imprimir datos.
Resumen	<p>El caso de uso se inicia cuando el Analista va a realizar alguna de las siguientes acciones relacionadas con el Libro de entrada de muestras de producción:</p> <ul style="list-style-type: none"> - Crear nuevo libro. - Registrar en el libro. - Buscar y visualizar libro. <p>El sistema muestra la interfaz correspondiente a la solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p>

Tabla 2.4 Descripción del CU "Gestionar Libro de entrada de muestras de producción".

2.3. Descripción ampliada del CU Gestionar Libro de entrada de cuarentena

Nombre del Caso de Uso	Gestionar Libro de entrada de cuarentena.
Actores	Analista (Inicia).
Propósito	Crear un nuevo libro, registrar datos, modificar sus datos, buscar datos.
Resumen	<p>El caso de uso se inicia cuando el Analista va a realizar alguna de las siguientes acciones relacionadas con el Libro de Entrada de Cuarentena:</p> <ul style="list-style-type: none"> - Crear nuevo libro. - Registrar en el libro. - Buscar libro. <p>El sistema muestra la interfaz correspondiente a la solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se</p>

	ejecuta alguna de las acciones solicitadas.	
Referencias	17.1, 17.2, 17.3, 17.3.1, 17.4, 17.5.	
Precondiciones	- Que el Analista se haya autenticado en la aplicación.	
Poscondiciones	Libro creado, datos insertados en el libro, libro buscado, libro modificado.	
Requerimientos especiales	-	
Flujo normal de los eventos		
Acción del actor	Respuesta del Sistema	
<p>1. El Analista, decide realizar una de las siguientes acciones:</p> <ul style="list-style-type: none"> - Crear nuevo libro. - Registrar en el libro. - Busca libro. 	<p>2. El sistema, en dependencia de la acción solicitada, hace lo siguiente:</p> <ul style="list-style-type: none"> • Si decide Crear nuevo libro, ir a la Sección “Crear nuevo libro”. • Si decide Registrar en el libro, ir a la Sección “Registrar datos en el libro”. • Si decide Buscar libro, ir a la Sección “Buscar y visualizar”. 	
Sección “Crear nuevo libro”		
Acción del actor	Respuesta del Sistema	
	1. Muestra la interfaz correspondiente para crear un libro.	
<p>2. El analista provee los datos necesarios para crear un libro:</p> <ul style="list-style-type: none"> • “Año” <p>Si el analista va a registrar los datos de una o varias entradas de cuarentena provee los datos:</p> <ul style="list-style-type: none"> • “Folio”. • “Fecha de Entrada”. 	<p>3. Verifica que el campo “Año” tenga valor y que sea un número de 4 dígitos. Ej: 2008.</p> <p>Si fueron introducidos los datos de una o varias entradas de cuarentena el sistema verifica que los campos obligatorios para registrar la entrada estén llenos y cumplan con el formato requerido</p>	

<ul style="list-style-type: none"> • “Materia Prima“. • “Lote CIGB“. • “Proveedor“. • “Lote Proveedor“. • “No. Envase“. • “U/M“. • “Fecha de Vencimiento“. • “Recibe“. • “Decisión“. • “Fecha“. <p>Solicita crear el libro: “Crear“.</p>	<p>:</p> <ul style="list-style-type: none"> • “Folio“. • “Fecha de Entrada“. • “Materia Prima“. • “Lote CIGB“. • “Proveedor“. • “Lote Proveedor“. • “No. Envase“. • “U/M“. • “Fecha de Vencimiento“. • “Recibe“. • “Decisión“. • “Fecha“. <p>-Verifica en tiempo de ejecución que ninguno de los valores introducidos en el campo “Folio” existan en la BD.</p> <p>-En el caso de ser: “Fecha de Entrada” cumple con el formato: Día/Mes “Fecha de Vencimiento” cumple con el formato: Mes/Año “Fecha” cumple con el formato: Día/Mes</p> <p>-En los campos:</p> <ul style="list-style-type: none"> - “Materia Prima“. - “Proveedor“. - “Recibe“. - “Decisión“. <p>Solo se puede introducir textos, en los restantes campos dígitos.</p>
	<p>4. Crea el libro.</p>

	5. Muestra un mensaje: “El libro ha sido creado satisfactoriamente”.
Flujos Alternos Sección “Crear nuevo libro”	
Acción del actor	Respuesta del Sistema
2.1 El analista no provee el dato para crear el libro, sale de la sección.	<p>3.1 Si el campo “Año” no tiene valor emite un mensaje de error “Faltan datos para crear el libro”.</p> <p>No se introdujeron los datos de una o varias entradas de cuarentena, ir a la acción 4.</p> <p>- Si fueron introducidos los datos de una o varias entradas de cuarentena verificar:</p> <ul style="list-style-type: none"> • Si alguno de los valores introducidos en el campo “Folio” existe en la BD muestra un mensaje de error: “Folio existente”. • Si algún campo de los especificados está vacío emite un mensaje de error: “Faltan datos para crear el libro”. • En caso de que haya algún error en el formato especificado para los campos, emite un mensaje de error: “Formato no válido”.
Sección “Registrar datos en el libro”	
Acción del actor	Respuesta del Sistema
	<p>1. Muestra la interfaz correspondiente para insertar datos.</p> <p>- En el campo “Año” se muestra una lista con todos los años en que se haya creado un libro, almacenados en la BD.</p>

<p>2. El analista provee los datos de una o varias entradas de cuarentena:</p> <ul style="list-style-type: none"> • “Año” • “Folio”. • “Fecha de Entrada”. • “Materia Prima”. • “Lote CIGB”. • “Proveedor”. • “Lote Proveedor”. • “No. Envase”. • “U/M”. • “Fecha de Vencimiento”. • “Recibe”. • “Decisión”. <p>“Solicita registrar datos en el libro: “Registrar”.</p>	<p>3. Verifica que los campos obligatorios para registrar la entrada estén llenos y cumplan con el formato requerido :</p> <ul style="list-style-type: none"> • “Año” • “Folio”. • “Fecha de Entrada”. • “Materia Prima”. • “Lote CIGB”. • “Proveedor”. • “Lote Proveedor”. • “No. Envase”. • “U/M”. • “Fecha de Vencimiento”. • “Recibe”. • “Decisión”. • “Fecha”. <p>-Verifica en tiempo de ejecución que ninguno de los valores introducidos en el campo “Folio” existan en la BD.</p> <p>-La Fecha debe cumplir con el formato (Día/Mes/Año). Ej: 10/02/08.</p> <p>-En los campos:</p> <ul style="list-style-type: none"> - “Materia Prima”. - “Proveedor”. - “Recibe”. - “Decisión”. <p>Solo se puede introducir textos, en los restantes campos dígitos.</p>
	<p>4. Inserta los datos en el libro, donde el campo “Año” coincida con el valor seleccionado por el analista.</p>

	5. Muestra un mensaje de operación satisfactoria: "Se han registrado los datos".
Flujos alternos Sección "Registrar datos en el libro"	
2.1 El analista no provee los datos, sale de la sección. El analista no solicita insertar datos, sale de la sección.	3.1 Si falta algún dato, emite un mensaje de error:"Faltan datos para registrar la entrada". -En caso de que haya algún error en el formato de los datos, emite un mensaje de error:"Formato no válido". - Si alguno de los valores introducidos en el campo "Folio" existe en la BD muestra un mensaje de error: "Folio existente".
Sección "Buscar y visualizar libro"	
Acción del actor	Respuesta del Sistema
	1. Muestra la interfaz correspondiente a la búsqueda de libros. - El campo "Lote CIGB" debe mostrar sin repetir ninguno, los valores que se han almacenado en la BD.
2. El analista provee los parámetros para la búsqueda. • "Lote CIGB". Solicita buscar: "Buscar".	3. Verifica que el campo para realizar la búsqueda tenga valor: • "Lote CIGB".
	4. Busca en el o los libros las entradas de cuarentena registradas que cumplen con el parámetro especificado.

	<p>5. Muestra todas las entradas de cuarentena encontradas. Da la posibilidad de Modificar cada una.</p>
<p>6. Solicita una de las opciones: - Modificar una entrada de cuarentena registrada: <i>"Modificar"</i>. - Imprimir el resultado de la búsqueda: <i>"Imprimir"</i>.</p>	<p>7. Según lo solicitado por el analista, se realiza una de las siguientes operaciones: - Si seleccionó la opción Modificar: ir a Sección <i>"Modificar datos del libro"</i>. - Si seleccionó la opción Imprimir: imprime el resultado de la búsqueda.</p>
Flujos alternos Sección "Buscar y visualizar libro"	
<p>2.1 Si no provee los datos y no desea buscar sale de la sección.</p>	<p>3.1 Si no proporcionó el parámetro para efectuar la búsqueda, emite un mensaje de error: "No se especificaron parámetros para realizar la búsqueda".</p>
	<p>5.1 Si no obtiene resultados al concluir la búsqueda muestra un mensaje: <i>"No hay resultados para esta búsqueda"</i></p>
<p>6.1 No solicita ninguna de las opciones, sale de la sección.</p>	
Sección "Modificar datos del libro"	
Acción del actor	Respuesta del Sistema
	<p>1. Muestra la nueva interfaz correspondiente para modificar los datos. -En el campo "Año" se muestra el año del libro en el que se seleccionó realizar la modificación de alguna de las entradas de cuarentenas registradas.</p>

<p>2. Provee los nuevos valores de los campos a modificar.</p> <ul style="list-style-type: none"> • “Folio”. • “Fecha de Entrada”. • “Materia Prima”. • “Lote CIGB”. • “Proveedor”. • “Lote Proveedor”. • “No. Envase”. • “U/M”. • “Fecha de Vencimiento”. • “Recibe”. • “Decisión”. • “Fecha”. <p>Solicita modificar datos “Modificar”.</p>	<p>3. Verifica que los campos obligatorios estén llenos.</p> <ul style="list-style-type: none"> • “Año”. • “Folio”. • “Fecha de Entrada”. • “Materia Prima”. • “Lote CIGB”. • “Proveedor”. • “Lote Proveedor”. • “No. Envase”. • “U/M”. • “Fecha de Vencimiento”. • “Recibe”. • “Decisión”. • “Fecha”.
	4. Registra la actualización en la BD.
	5. Actualiza registro de trazas.
	6. Muestra un mensaje: “Los datos han sido modificados satisfactoriamente”.
Flujos alternos Sección “Modificar datos del libro”	
2.1 No provee los datos, sale de la sección.	3.1 Si quedó algún campo vacío, emite un mensaje de error: “Datos incompletos”.
Prototipo	Figura 3.3, Figura 3.4, Figura 3.5 y Figura 3.6

Tabla 2.5 Descripción del CU “*Gestionar Libro de entrada de cuarentena*”.

Conclusiones

En este capítulo se obtuvieron los requerimientos no funcionales y los funcionales críticos. Se identificaron los actores y casos de uso del sistema y se realizó el diagrama de casos de uso permitiendo mostrar la relación entre actores y casos de uso. Además se describieron los casos de uso, para la posterior construcción de los prototipos, y se realizó la matriz de trazabilidad entre Actores-CU y RF-CU.

Capítulo 3

Diseño del Sistema

Introducción

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a una arquitectura estable y sólida, y crear un plano del modelo de implementación. En el presente capítulo se hace referencia a la arquitectura del sistema a través de la Vista Lógica y de Despliegue, se presentan de los prototipos no funcionales los correspondientes al caso de uso "Gestionar Libro de entrada de cuarentena", además el diagrama de secuencia correspondiente a este caso de uso. De los diagramas de clases del diseño se muestran los correspondientes a los casos de uso "Gestionar Registro de preparación de soluciones", "Gestionar Libro de control de desempeño", "Gestionar Libro de entrada de muestras de producción" y "Gestionar Libro de entrada de cuarentena". Se confecciona el mapa de navegación.

Para ver el resto de los artefactos que se obtuvieron y no fueron presentados en el trabajo remitirse al *Expediente de proyecto*.

3.1. Propósitos del diseño

Concretamente se puede definir como propósitos del diseño [19]:

- 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 se utilizan interfaces como elementos de sincronización entre diferentes equipos de desarrollo.

- Para modelar el diseño se utilizan las clases del diseño que son 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 operaciones, atributos, tipos, visibilidad, se pueden especificar con la sintaxis del lenguaje elegido.
- Los métodos de una clase del diseño tienen correspondencia directa con el correspondiente método en la implementación de las clases.

3.2. Principios de diseño

UML posee una extensión para el modelado de aplicaciones Web, propuesta por Conallen [20], dicha extensión es usada para el diseño de las clases. Los estereotipos que usa son:



<<Server Page>> Representa la página Web que tiene código que se ejecuta en el servidor.

Este código interactúa con recursos en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página.

Esta clase sólo puede tener relaciones con objetos en el servidor, una relación 1:1 con un fichero en el servidor. En las aplicaciones en PHP se corresponde con un fichero .php



<<Client Page>> Una instancia de Página Cliente es una página Web, con formato HTML;

mezcla de datos, presentación y lógica. Son interpretadas por el browser. Cada página cliente solo puede ser construida por una página servidor.



<<Form>> Grupo de elementos de entrada que son parte de una página cliente. Se relaciona

directamente con la etiqueta de igual nombre del HTML. Sus atributos son los elementos de entrada del formulario (Text Field, Text Area, Button, Label, Radio Button, Radio Group, Select, Check Box y Hidden Fields). [21]

3.3. Referencia a la Arquitectura del Sistema

"La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución." [22]

La arquitectura de un sistema consta de múltiples vistas, asociadas a diferentes dimensiones o perspectivas del sistema.

A continuación se muestra la representación de las dos vistas: Vista Lógica y Vista de Despliegue.

3.3.1. Vista Lógica

La Vista Lógica es un subconjunto del artefacto Modelo de diseño, la cual representa los elementos de diseño más importantes para la arquitectura del sistema. Describe las clases más importantes, su organización en paquetes y subsistemas, éstos en capas y las realizaciones de casos de uso más importantes.

Para estructurar la vista del sistema, no se identificaron en el módulo Análisis Químico 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. Existen paquetes formados por otros paquetes y éstos tendrán como todos, su diagrama de clases. Los paquetes más generales que se identificaron son: Determinar_Grado_Purezas_Impurezas_Proteinas, Realizar_Ensayo_Materias_Primas, Realizar_Ensayo, y Determinar_Estado_de_Materias_Primas, en los que se agrupan los casos de uso arquitectónicamente significativos relacionados con la finalidad del paquete. A continuación se muestra la Vista Lógica en la **Figura 3.1**.

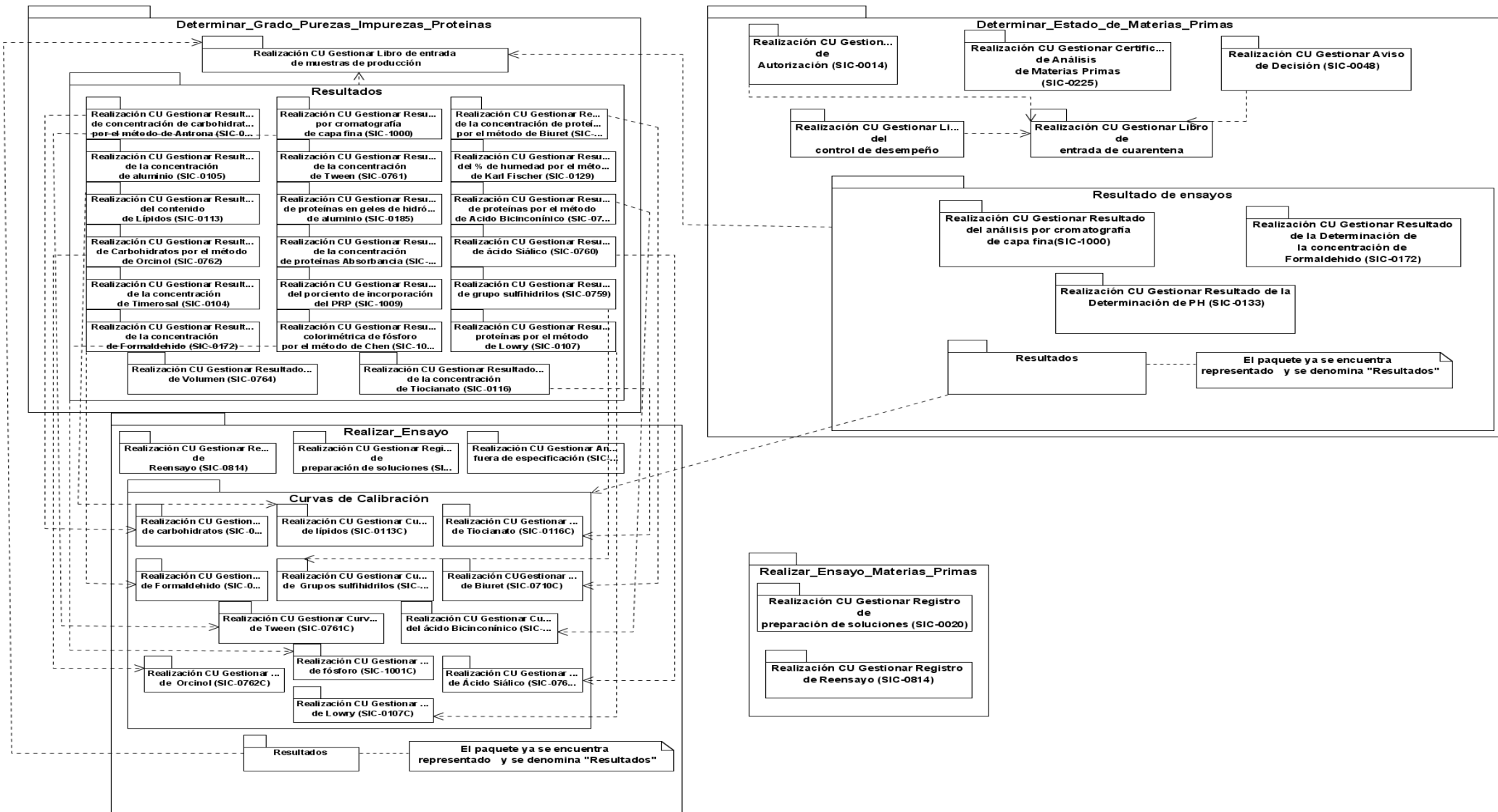


Figura 3.1 Vista Lógica.

Para un mejor entendimiento de la Vista Lógica se presentan los paquetes y el contenido de cada uno de ellos, con las realizaciones de los casos de uso correspondientes a estos, no se encuentran representadas las relaciones con los otros paquetes, solo las relaciones dentro de los mismos, y los paquetes por los que están compuestos.

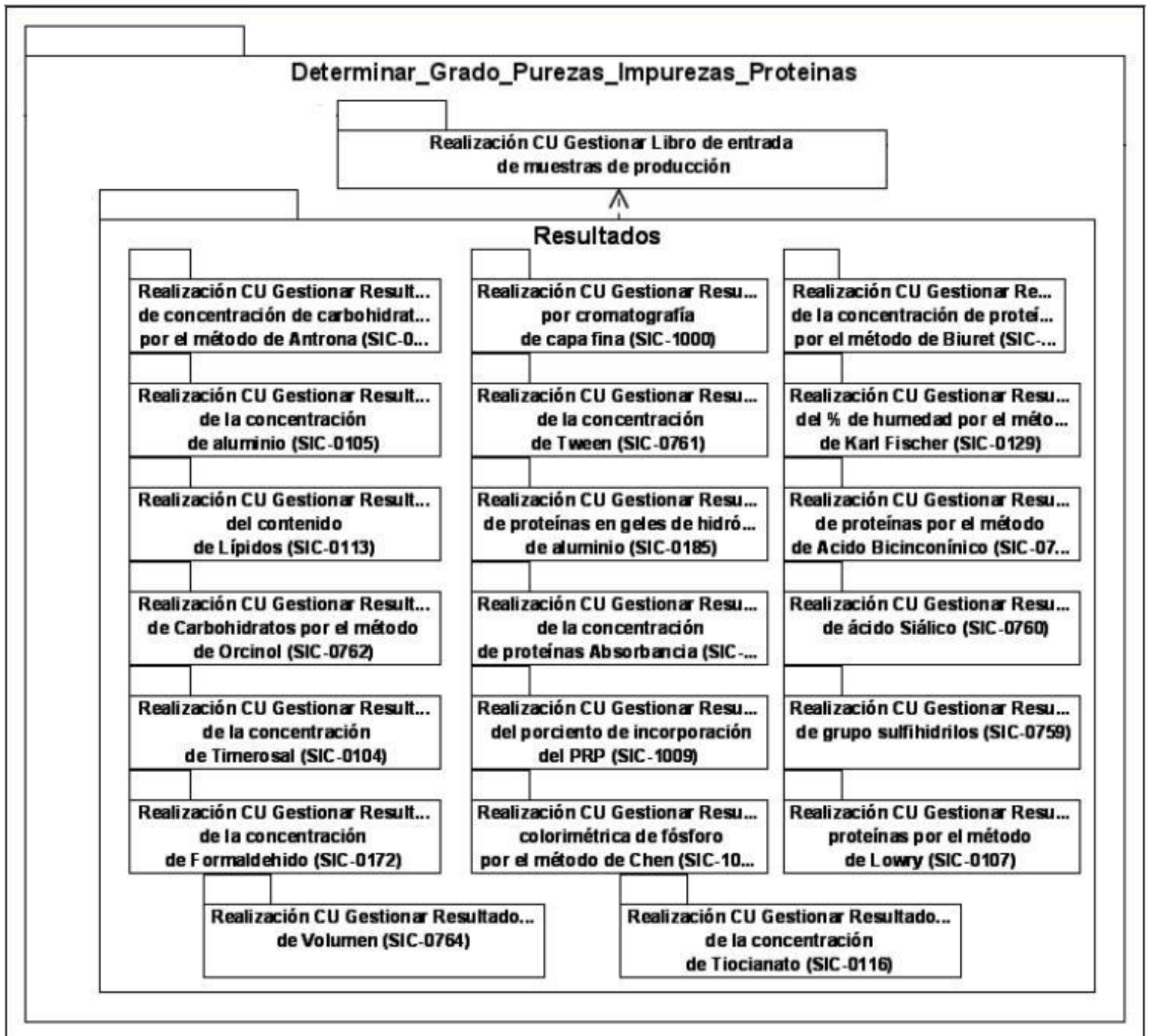


Figura 3.2 Paquete "Determinar_Grado_Purezas_Impurezas_Proteinas".



Figura 3.3 Paquete *“Realizar_Ensayo_Materias_Primas”*.

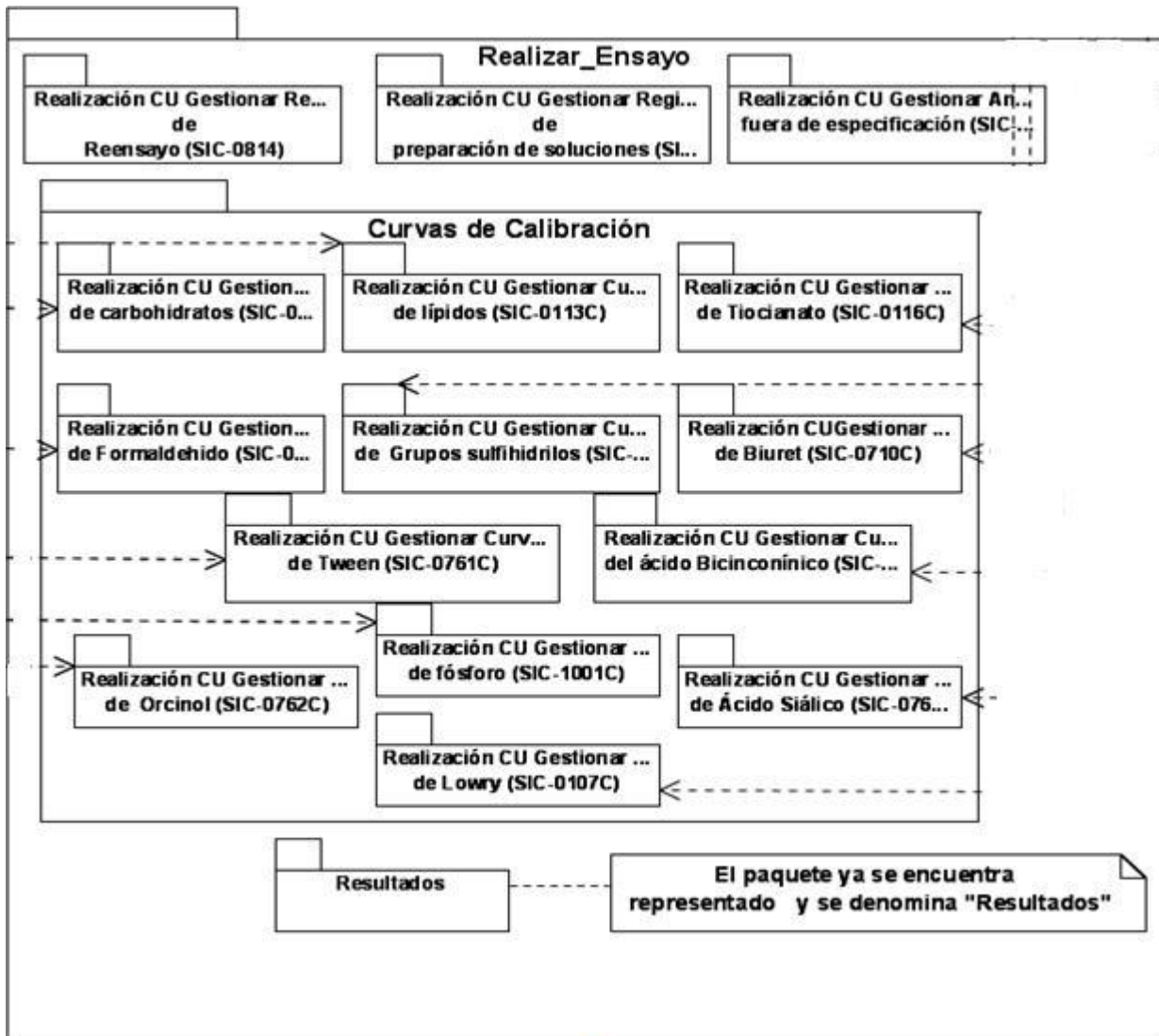


Figura 3.4 Paquete "Realizar_Ensayo".

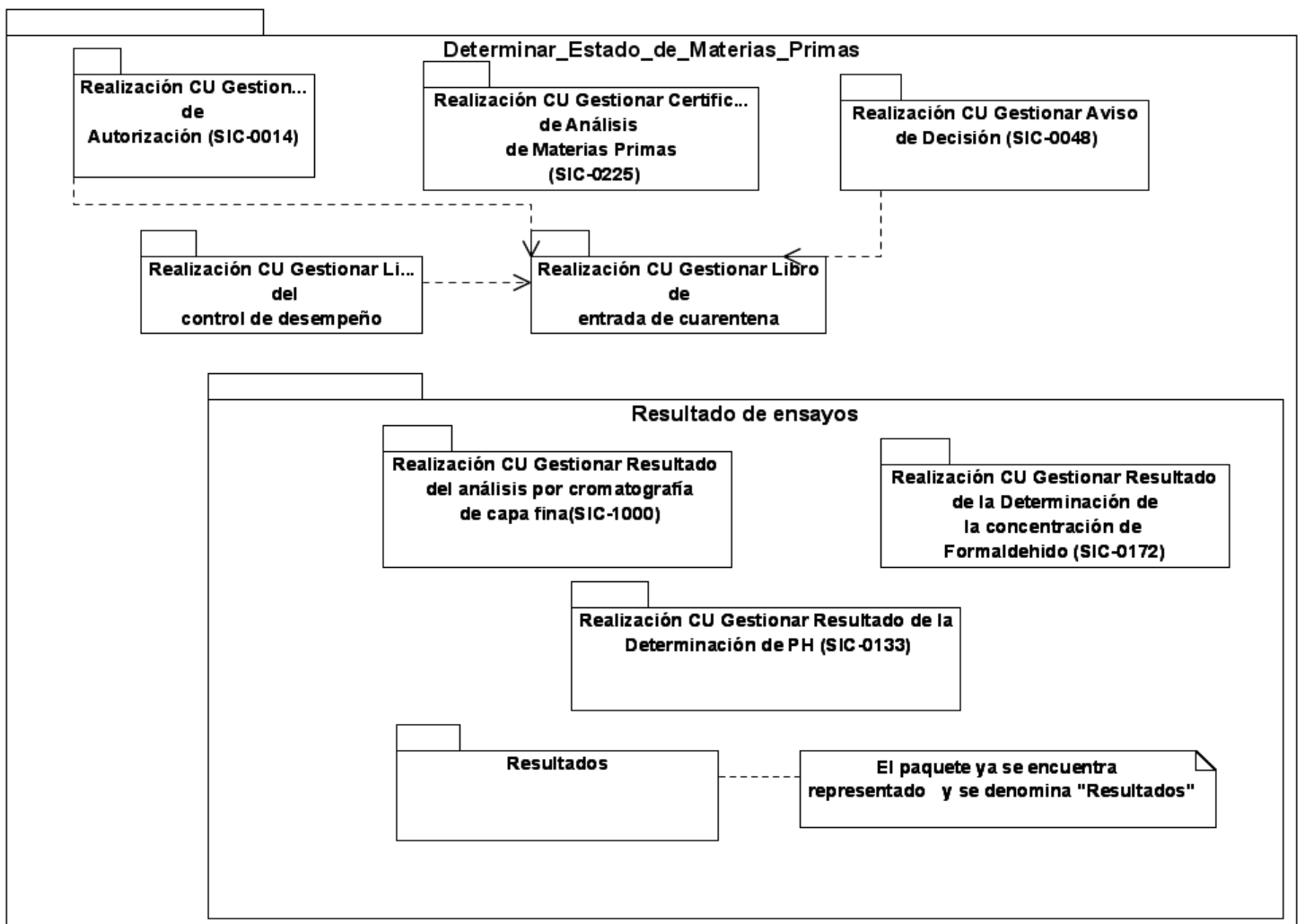


Figura 3.5 Paquete “Determinar_Estado_de_Materias_Primas”.

A continuación en la Figura 3.6, se muestra la realización del CU Gestionar libro de entrada de cuarentena del paquete “Determinar_Estado_de_Materias_Primas” Figura 3.5, la cual está conformada por su diagrama de clases del diseño y de interacción, en cada uno de estos se evidencia el patrón de diseño clásico (Modelo-Vista-Controlador) a partir del uso de Symfony como framework para desarrollar la aplicación web.



Figura 3.6 "Realización CU Gestionar Libro de entrada de cuarentena".

3.3.2. Vista de Despliegue

Vista Despliegue: Muestra la distribución física del sistema (ordenadores, dispositivos) y sus conexiones, y se plasma en el diagrama de Despliegue.

La vista se encuentra representada a través del diagrama de despliegue que se muestra a continuación, que se encuentra estructurado de la siguiente manera:

Los nodos representan un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar, se cuenta con un Servidor de Aplicaciones conectado a la PC Cliente, a través del protocolo HTTPS. Además se tendrá mediante una conexión USB una impresora para la impresión de documentos y un scanner para la digitalización de información que se encuentra en formato duro.

Para el almacenamiento de los datos de la aplicación se utiliza un servidor de Base de Datos representado como un nodo y para lograr la conexión del sistema con la base de datos se utiliza TCP / IP + SSL como protocolo de comunicación.

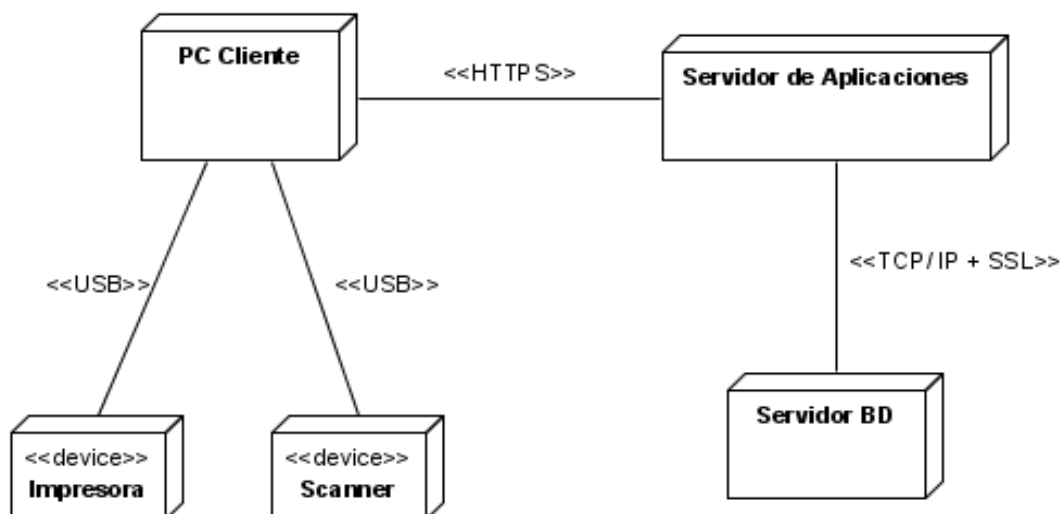


Figura 3.7 Vista de Despliegue.

3.4. Prototipos de interfaz de usuario

Los prototipos de interfaz de usuario son una representación de la funcionalidad contenida en los casos de uso; de manera que permita que el usuario verifique que el sistema va a satisfacer sus necesidades. De esta manera se válida que se esté cumpliendo con los requerimientos exigidos antes de que se realice todo el esfuerzo necesario para el desarrollo.

Para ver prototipos no funcionales de los restantes Casos de uso del Sistema remitirse al *Expediente de proyecto*, a continuación se muestra los prototipos correspondientes al Libro de entrada de cuarentena.

3.4.1. Prototipos de interfaz de usuario del Libro de entrada de cuarentena

CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA

DIRECCIÓN DE CALIDAD

LIBRO DE ENTRADA DE CUARENTENA

Año:

REGISTRAR

Folio	Fecha Entrada	Materia Prima	Lote CIGB	Proveedor	Lote Proveedor	No.Envase	U/M	Fecha de Vencimiento	Recibe	Decisión	Fecha
	/ /							-			/

Figura 3.8 Prototipo no funcional "Crear Libro de entrada de cuarentena".

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA

DIRECCIÓN DE CALIDAD

LIBRO DE ENTRADA DE CUARENTENA

Año:

REGISTRAR

Folio	Fecha Entrada	Materia Prima	Lote CIGB	Proveedor	Lote Proveedor	No.Envase	U/M	Fecha de Vencimiento	Recibe	Decisión	Fecha
	/							-			/

Figura 3.9 Prototipo no funcional "Registrar datos en el Libro de entrada de cuarentena".

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA

DIRECCIÓN DE CONTROL DE CALIDAD

LIBRO DE ENTRADA DE CUARENTENA

PARÁMETROS DE LA BÚSQUEDA

Lote CIGB

RESULTADO DE LA BÚSQUEDA

Folio	Fecha Entrada	Materia Prima	Lote CIGB	Proveedor	Lote Proveedor	No.Envase	U/M	Fecha de Vencimiento	Recibe	Decisión	Fecha	
***	*/*	***	***	***	***	***	***	** _ **	***	***	*/*	Modificar

Figura 3.10 Prototipo no funcional "Busca y Visualizar Libro de entrada de cuarentena".

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA

DIRECCIÓN DE CALIDAD

LIBRO DE ENTRADA DE CUARENTENA

Año:

DATOS A MODIFICAR

Folio	Fecha Entrada	Materia Prima	Lote CIGB	Proveedor	Lote Proveedor	No.Envase	U/M	Fecha de Vencimiento	Recibe	Decisión	Fecha
***	:/	*	*	***	*	*	*	*_*	***	*	/

Figura 3.11 Prototipo no funcional "Modificar Libro de entrada de cuarentena".

3.5. Mapa de Navegación

Un mapa de navegación es la representación del sitio a través de una secuencia de interfaces. Para la construcción del mapa de navegación se utilizó UML, la navegabilidad se estructuró de la siguiente manera: a partir de la página principal (Index) se podrá acceder a los diferentes laboratorios, entre ellos el Laboratorio Análisis Químico, una vez en él, estarán disponibles las páginas que contienen información detallada de los procesos que se realizan en el laboratorio (Determinar_Estado_de_Materias_Primas, Determinar_Grado_Purezas_Impurezas_Proteinas, Realizar_Ensayo y Realizar_Ensayo_Materias_Primas), y a través de ellas tener acceso a la gestión de los diferentes documentos del laboratorio. Como parte de la navegabilidad del sitio se podrá ir de las diferentes páginas que representan los SIC al laboratorio Análisis Químico, y de este último se podrá volver a la página principal (Index) y viceversa.

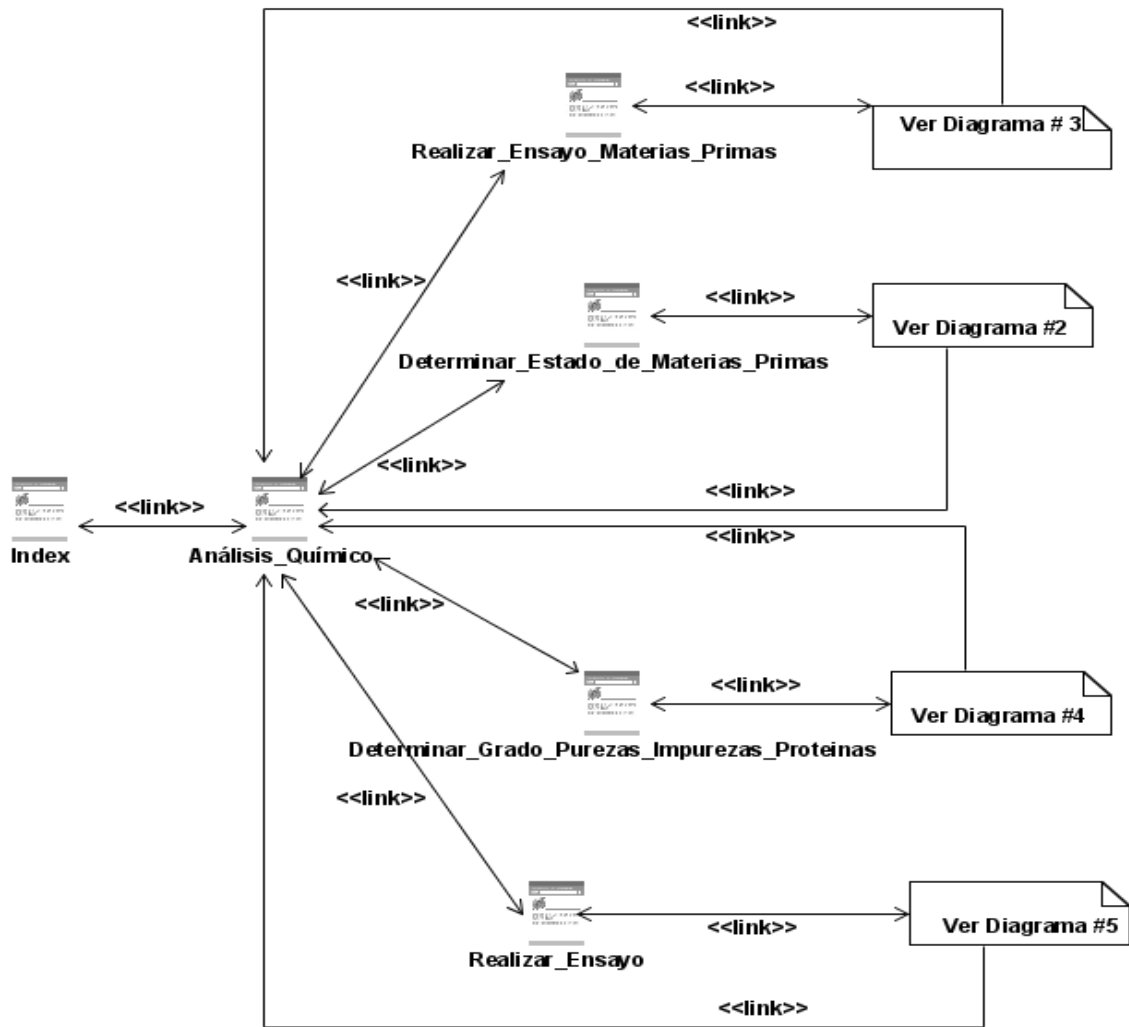


Figura 3.12 Mapa de Navegación.

Se presenta la **Figura 3.13** como un ejemplo donde se observan las relaciones que existen entre las acciones que se podrán realizar cuando se selecciona un SIC, se podrá seleccionar una de las opciones de: Crear o Buscar SIC y a partir de este último se podrá visualizar para posteriormente poder modificar. Solo se muestra este diagrama porque los restantes son muy extensos, remitirse al *Expediente de proyecto*.

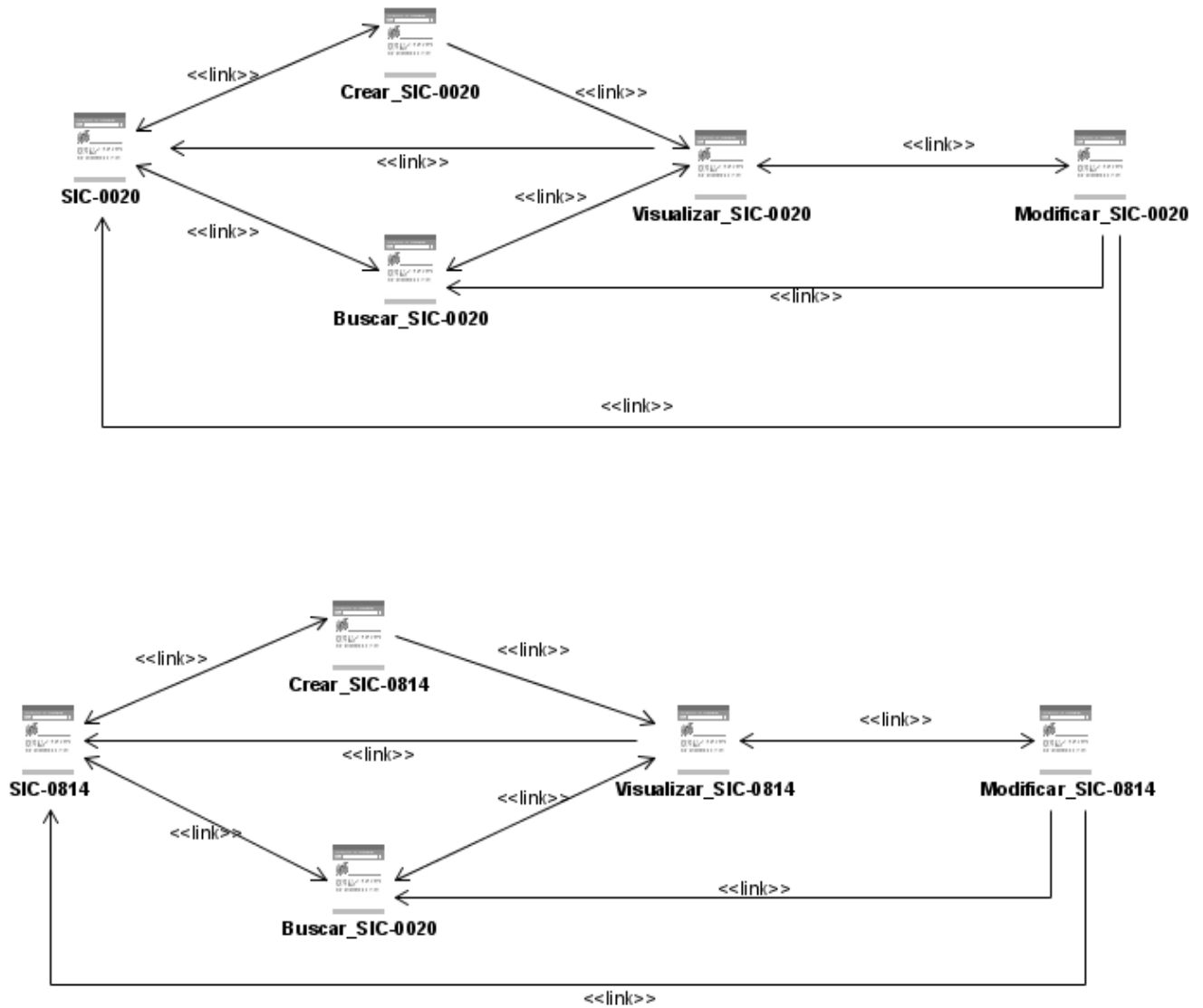


Figura 3.13 Diagrama 3 "Realizar_Ensayo_Materias_Primas".

3.6. Diagramas de Clases del Diseño

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que son diagramas “estáticos” porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases “conocen” a qué otras clases o qué clases “son parte” de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

A continuación se muestran los diagramas de clases del diseño de los casos de uso presentados en el capítulo 2. En la Vista se encuentran: las páginas clientes, los formularios, páginas servidoras y el layout, aquí se representa la información con la que trabaja la aplicación y la lógica de negocio, en el trabajo se presentan los diferentes libros, registros y SIC (sistema de Información y control) con los que se trabaja diariamente en el Centro de Ingeniería y Biotecnología (CIGB). El Controlador se encuentra conformado por el controlador frontal que es el único punto de entrada a la aplicación, carga la configuración y determina la acción a ejecutarse, y las acciones: verifican la integridad de las peticiones, preparan los datos requeridos por la capa de presentación y contienen la lógica de la aplicación. La lógica de negocio de las aplicaciones Web depende de su modelo de datos, en el se encuentran todas las clases con sus atributos y las relaciones existentes entre ellas y las clases que genera Symfony: Objeto, ObjetoPeer, BaseObjeto y BaseObjetoPeer.

- 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.

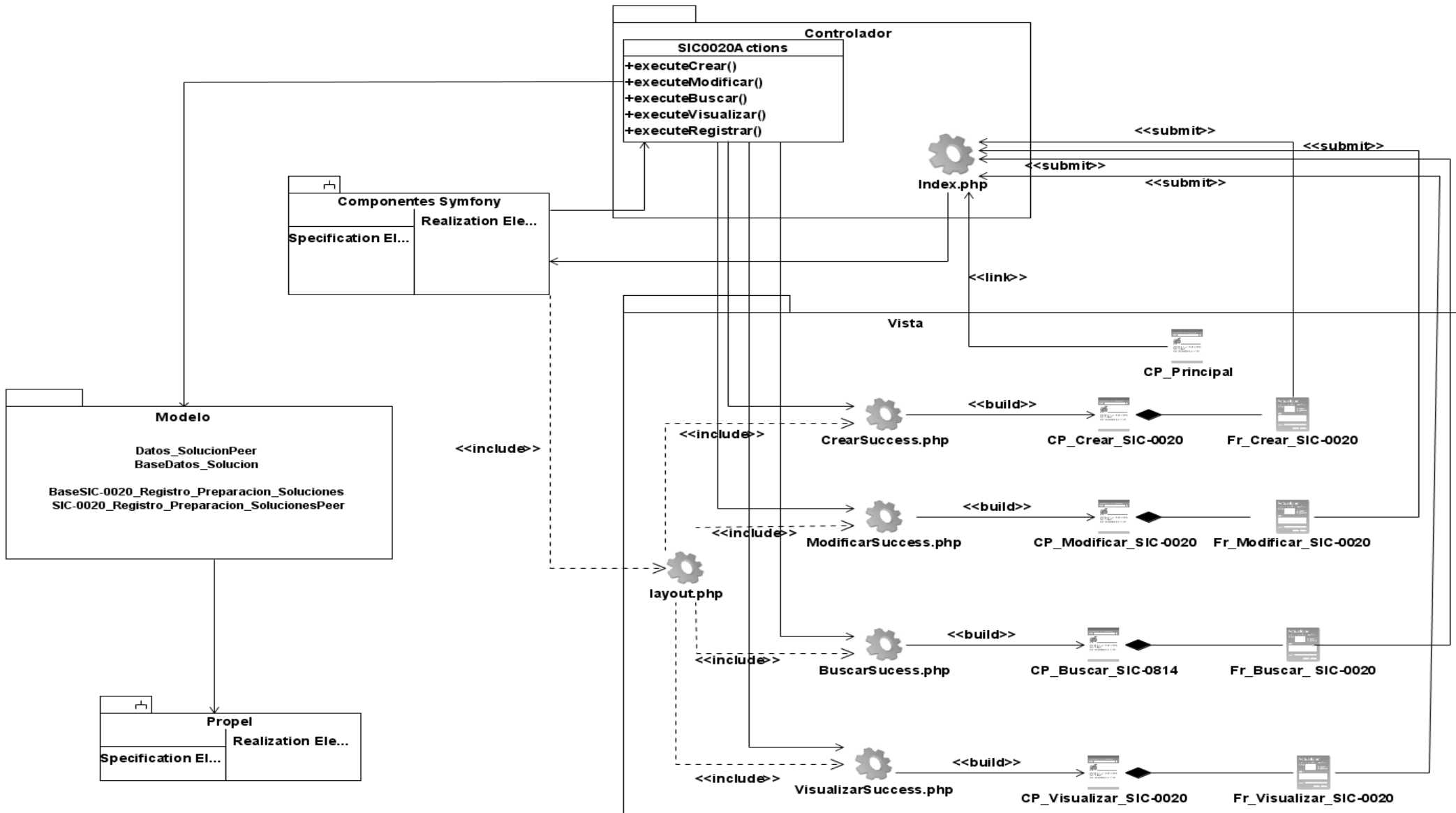


Figura 3.14 Diagrama de Clase: CU "Gestionar Registro de preparación de soluciones (SIC-0020)".

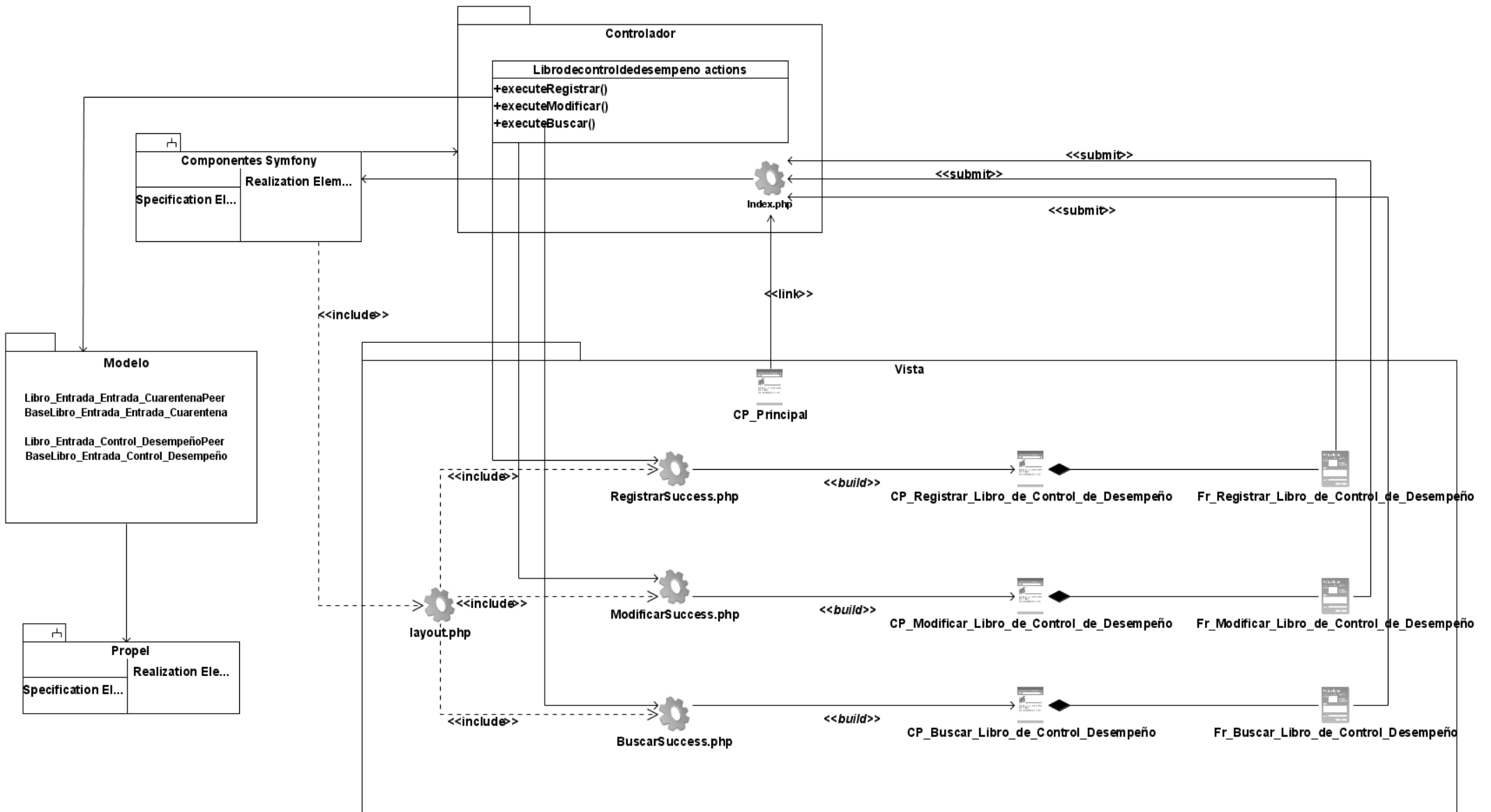


Figura 3.15 Diagrama de Clase: CU "Gestionar Libro del control de desempeño".

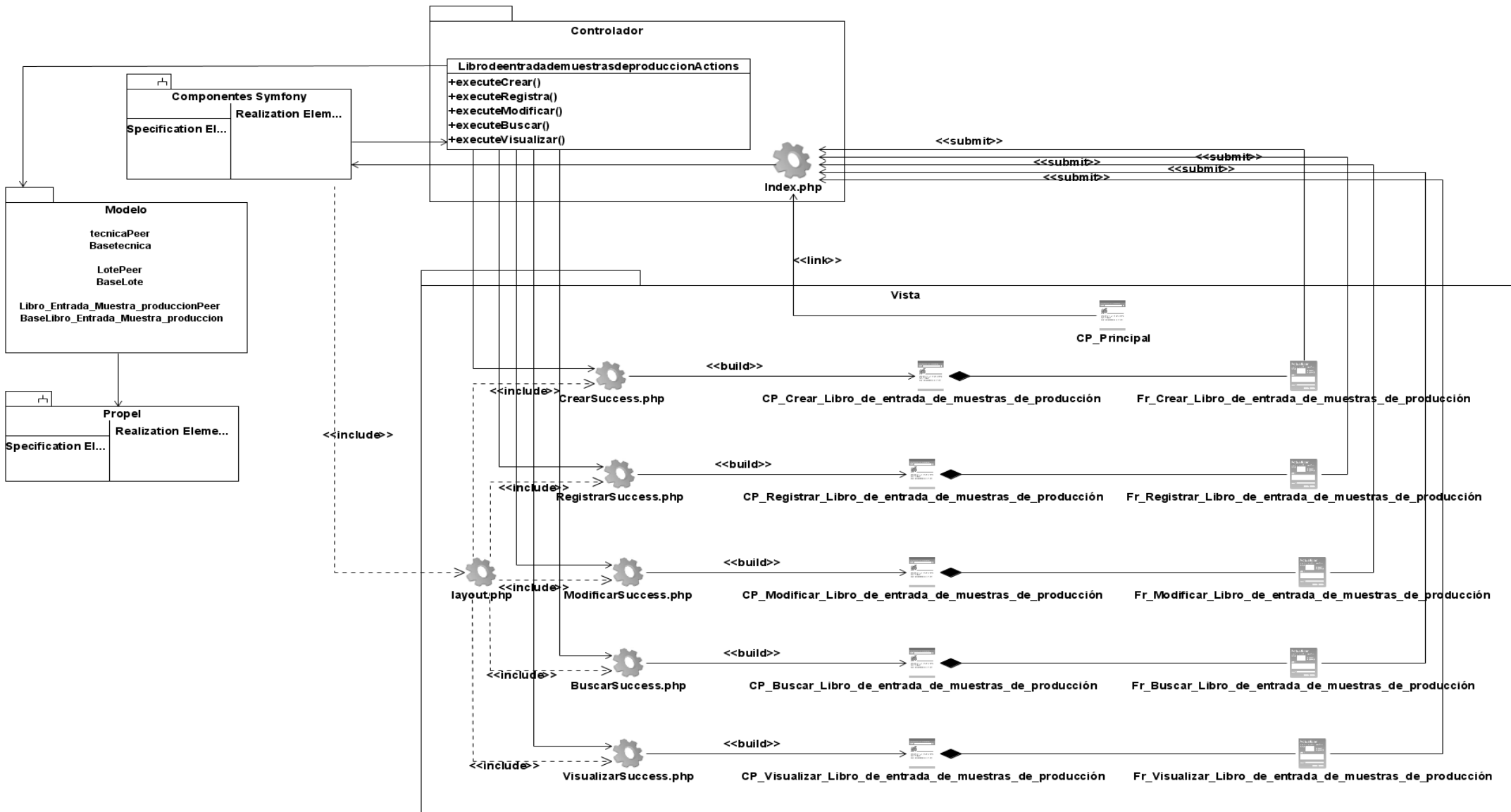


Figura 3.16 Diagrama de Clase: CU "Gestionar Libro de entrada de muestras de producción".

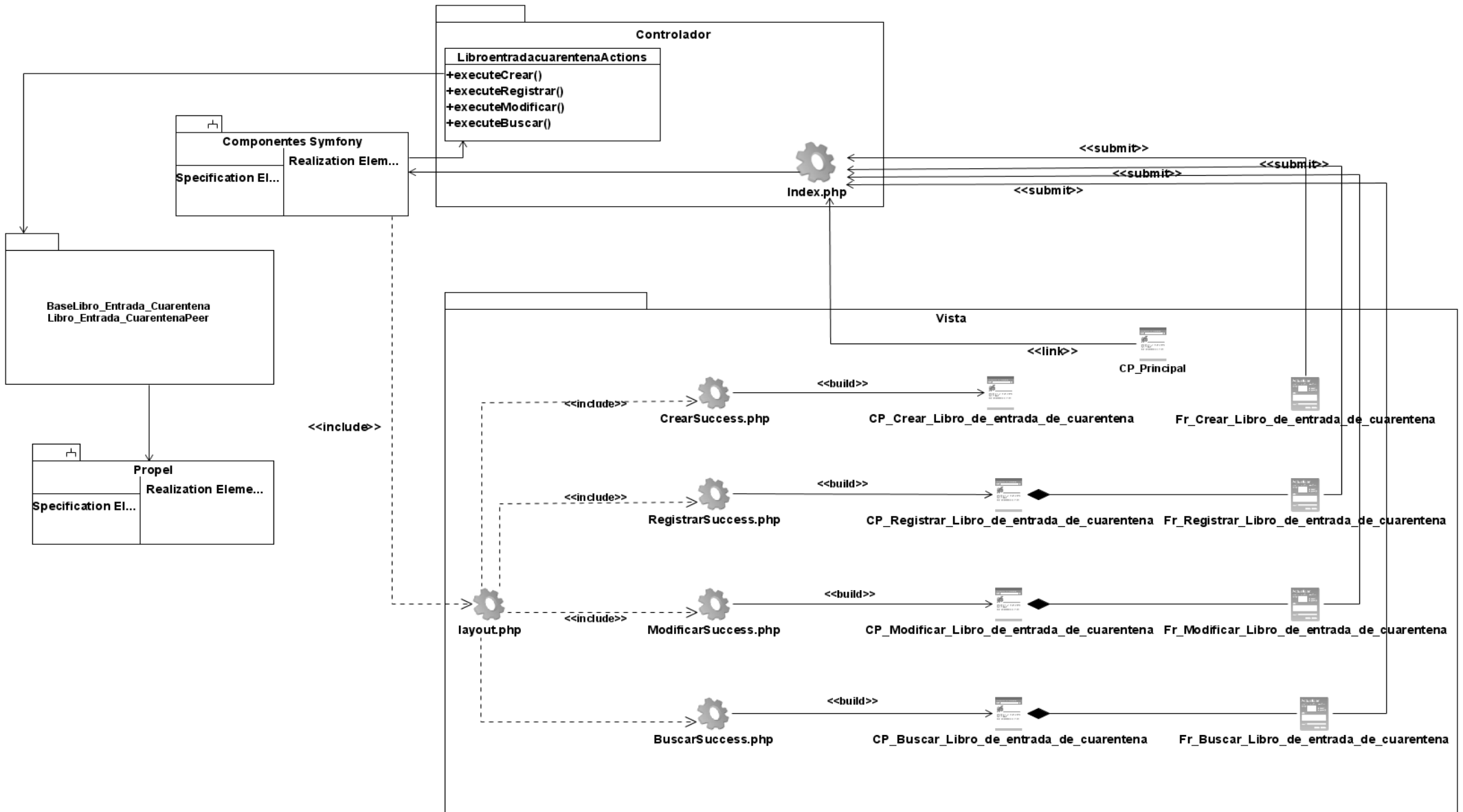


Figura 3.17 Diagrama de Clase: CU "Gestionar Libro de entrada de cuarentena".

Para todos los diagramas de clases se realizó un único paquete denominado Modelo para lograr un mejor entendimiento de los mismos, en el cual se representan las clases generadas por Symfony por cada una de las tablas de la Base de Datos y las relaciones existentes entre ellas. En el trabajo solo se mostrará del Modelo que contiene 204 clases, una pequeña representación (**Figura 3.18**) relacionada con el diagrama de clases del diseño correspondiente al CU “Gestionar Libro de entrada de cuarentena” y en la (**Figura 3.19**) y (**Figura 3.20**) se muestran los atributos y métodos de las clases Libro_Entrada_CuarentenaPeer y BaseLibro_Entrada_Cuarentena que fueron presentadas en su forma minimizada en la (**Figura 3.18**).

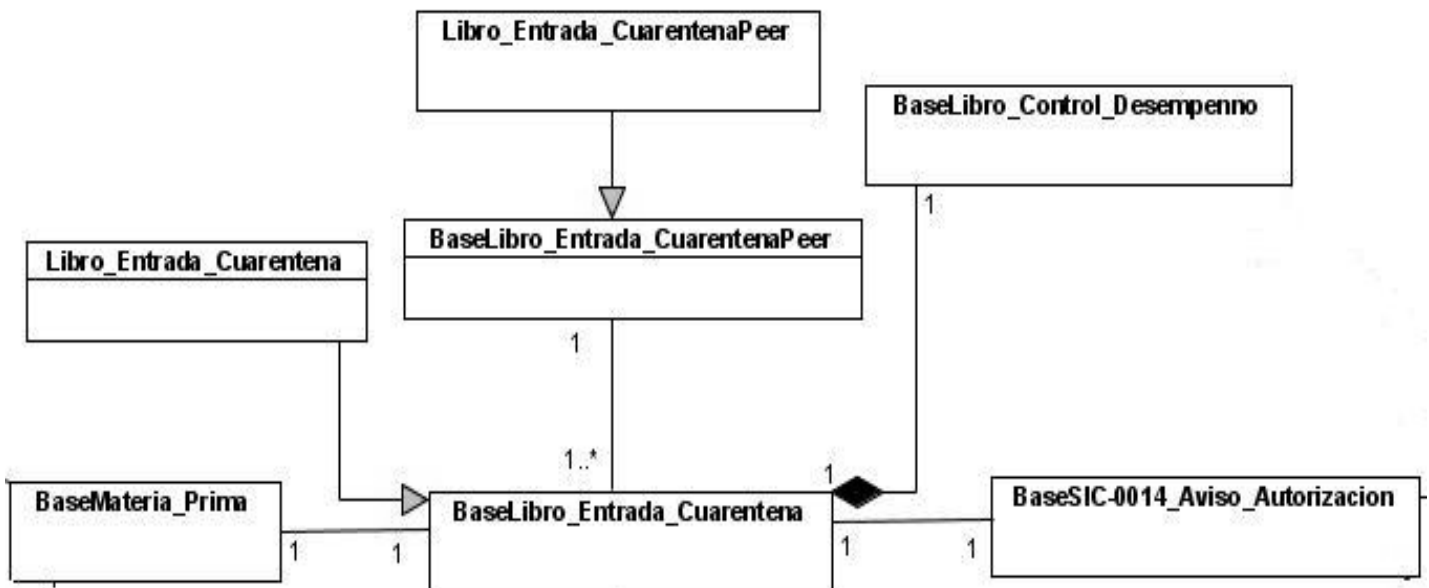


Figura 3.18 Representación de una parte del modelo.

Se muestran las clases: BaseLibro_Entrada_Cuarentena y Libro_Entrada_CuarentenaPeer en su forma extendida.

Libro_Entrada_CuarentenaPeer
+CrearLibro()
+Buscar_lote()
+Buscar(\$libro)
+BuscarPorFolio(\$libro)
+BuscarMostrar(\$id)
+BuscarArreglo(\$array_obj)
+Buscar_Anno()
+Existefolio(\$folio)
+doSelect(\$c)
+ParaSelect(\$lote, \$posicion_lote)

Figura 3.19 Clase “Libro_Entrada_CuarentenaPeer”.

BaseLibro_Entrada_Cuarentena
⊖ \$peer
⊖ \$id_libro
⊖ \$anno
⊖ \$folio
⊖ \$materia_prima
⊖ \$lote_cigb
⊖ \$proveedor
⊖ \$lote_proveedor
⊖ \$no_envase
⊖ \$u_m
⊖ \$recibe
⊖ \$decision
⊖ \$dia_entrada
⊖ \$mes_entrada
⊖ \$anno_entrada
⊖ \$mes_venc
⊖ \$anno_venc
⊖ \$dia_dec
⊖ \$mes_dec
⊖ getIdLibro()0
⊖ getAnno()0
⊖ getFolio()0
⊖ getMateriaPrima()0
⊖ getLoteCigb()0
⊖ getProveedor()0
⊖ getLoteProveedor()0
⊖ getNoEnvase()0
⊖ getUM()0
⊖ getRecibe()0
⊖ getDecision()0
⊖ getDiaEntrada()0
⊖ getMesEntrada()0
⊖ getAnnoEntrada()0
⊖ getMesVenc()0
⊖ getAnnoVenc()0
⊖ getDiaDec()0
⊖ getMesDec()0
⊖ setIdLibro(\$v)0
⊖ setAnno(\$v)0
⊖ setFolio(\$v)0
⊖ setMateriaPrima(\$v)0
⊖ setLoteCigb(\$v)0
⊖ setProveedor(\$v)0
⊖ setLoteProveedor(\$v)0
⊖ setNoEnvase(\$v)0
⊖ setUM(\$v)0
⊖ setRecibe(\$v)0
⊖ setDecision(\$v)0
⊖ setDiaEntrada(\$v)0
⊖ setMesEntrada(\$v)0
⊖ setAnnoEntrada(\$v)0
⊖ setMesVenc(\$v)0
⊖ setAnnoVenc(\$v)0
⊖ setDiaDec(\$v)0
⊖ setMesDec(\$v)0
⊖ save(\$con = null)0
⊖ doSelect(\$c)0

Figura 3.20 Clase “BaseLibro_Entrada_Cuarentena”.

3.7. Diagramas de secuencia

El diagrama de secuencia de un sistema es una representación que muestra, en determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos internos del sistema, a continuación están expuestos los escenarios: *Escenario Registrar*, *Escenario Crear* y *el Escenario Modificar* que corresponden al CU “*Gestionar Libro de entrada de cuarentena*”. En el caso del escenario: *Buscar y Visualizar*, se muestra una explicación detallada en el epígrafe posterior al evaluar la calidad del diseño realizado.

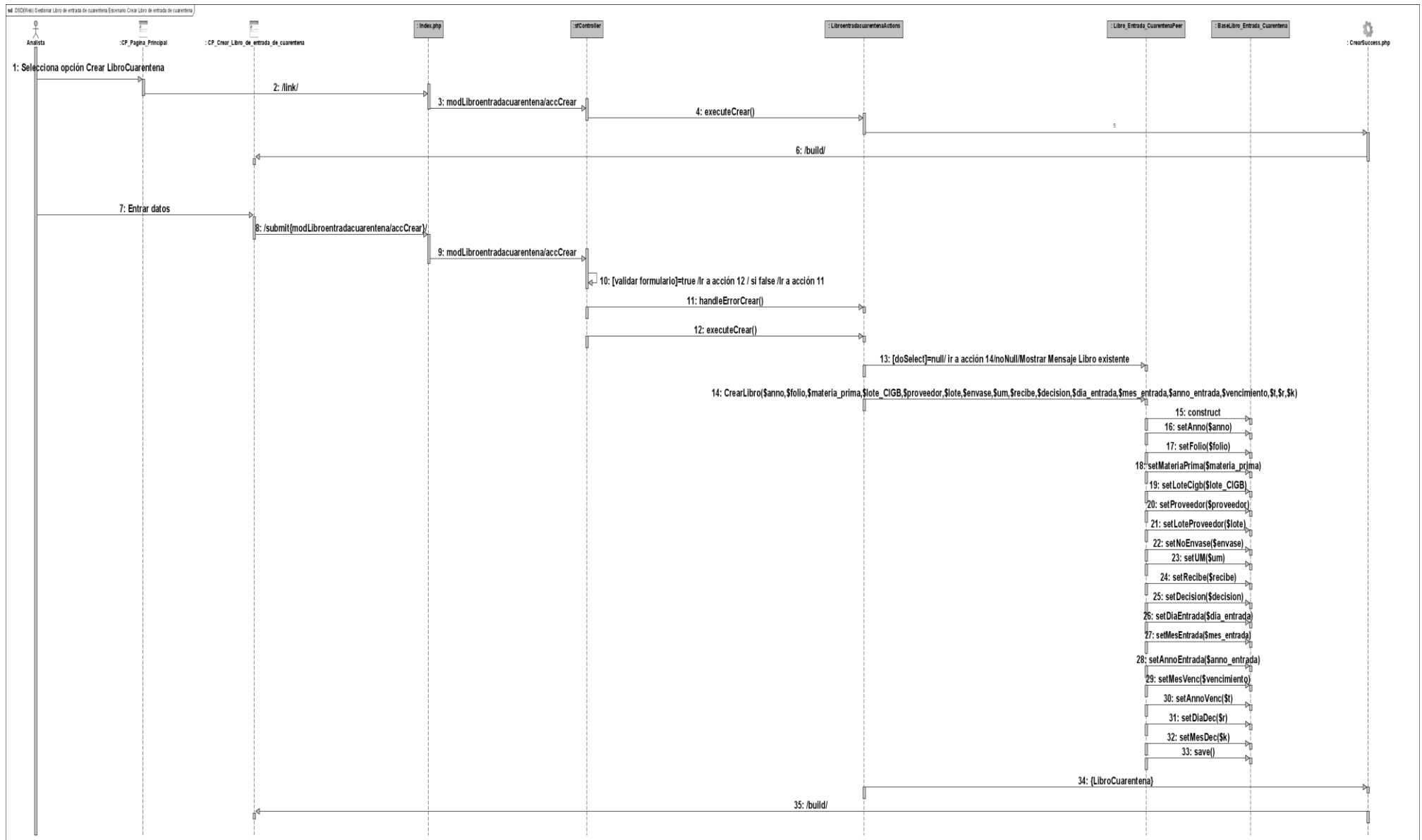


Figura 3.21 Diagrama de Secuencia (Escenario Crear): CU “Gestionar Libro de entrada de cuarentena”.

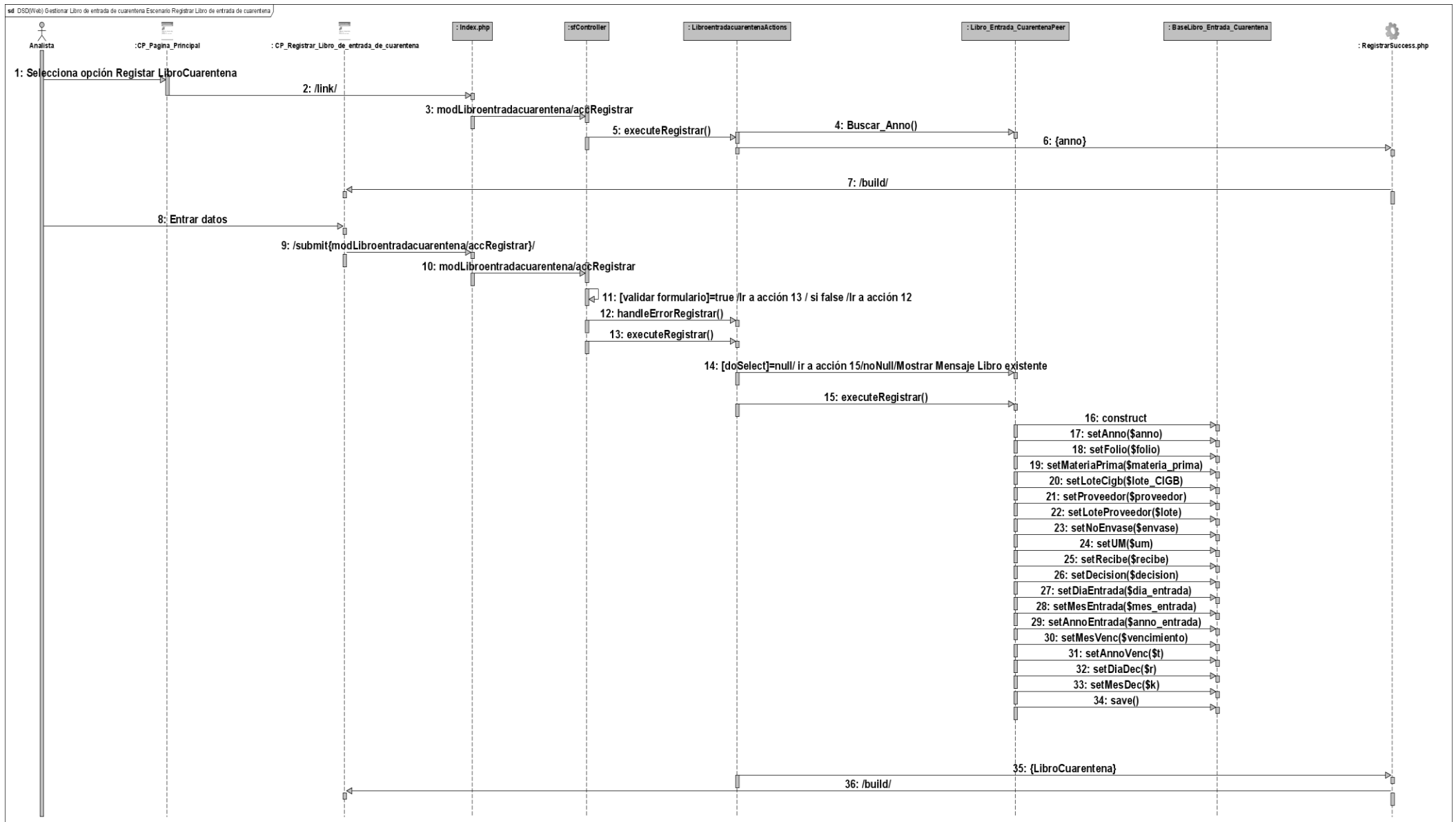


Figura 3.22 Diagrama de Secuencia (Escenario Registrar): CU “Gestionar Libro de entrada de cuarentena”.

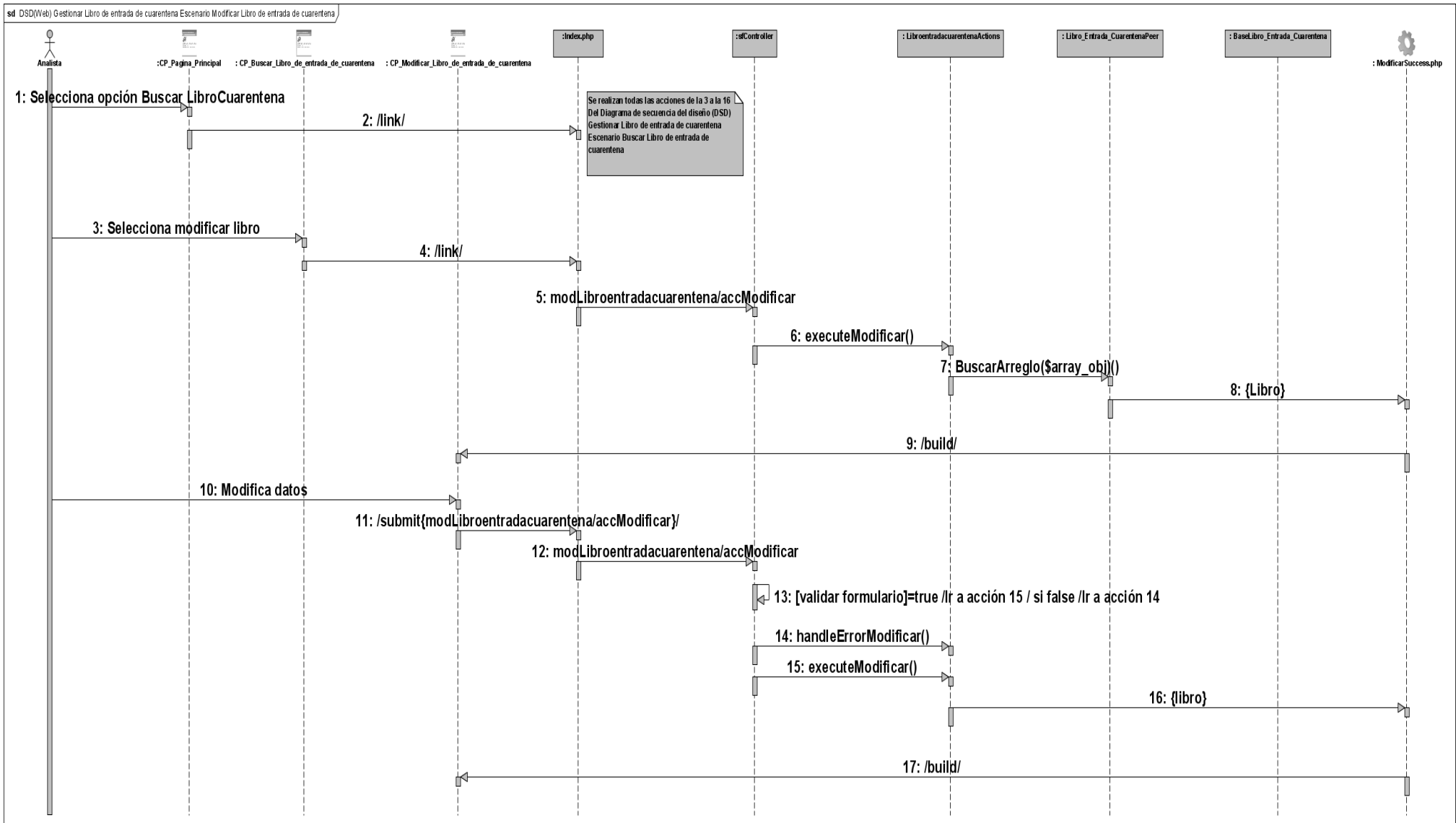


Figura 3.23 Diagrama de Secuencia (Escenario Modificar): CU “Gestionar Libro de entrada de cuarentena”.

3.8. Validación del diseño realizado

Para la validación del diseño realizado del módulo de Análisis Químico, se tomó una porción de lo implementado, con el objetivo de establecer una correspondencia entre lo implementado y lo diseñado. Además se aplicó una encuesta a los dos programadores donde se midieron tres puntos claves: *Comprensión del diseño*, *Correspondencia de los requerimientos no funcionales con el diseño* y *Descomposición del diseño en paquetes*.

A continuación se presentan el diagrama de secuencia (Escenario Buscar y Visualizar) correspondiente al CU “Gestionar Libro de entrada de cuarentena” y un fragmento del código de la aplicación Web que se encuentra estrechamente relacionado con este caso de uso para establecer la correspondencia existente entre ellos. Para ello se muestra el orden de las acciones del diagrama de secuencia desencadenadas a partir de que el actor solicita en la página principal la opción Buscar luego de haberse autenticado en la aplicación a partir de la (*Acción 8*) y se enumeran las líneas de código con el mismo número que identifica las acciones en el diagrama.

Cuando el usuario indica Buscar libroCuarentena (**Acción 8**) del diagrama de secuencia dicha página como resultado hace un link al Index.php (controlador frontal) y ésta a su vez le envía al sfController (componente de Symfony que es el encargado de verificar la seguridad del sistema y validar los formularios) el nombre del módulo (Libroentradacuarentena) y la acción a realizar (Buscar) (**Acción 3**), luego sfController le indica a la clase Actions(LibroentradacuarentenaActions) cual es la acción que debe ejecutar (**Acción 4**). Esta clase le envía un mensaje con el método Buscar_lote () (**Acción 5**) a la clase Libro_Entrada_CuarentenaPeer dicho método lo que hace es buscar todos los lotes almacenados en la BD y devolverlos en un arreglo que se ilustra en el diagrama con el nombre de lote, luego la clase Actions (LibroentradacuarentenaActions) le envía a la clase BuscarSuccess.php dicho arreglo (**Acción 6**) para que esta construya la página CP_Buscar_Libro_de_entrada_de_cuarentena (**Acción 7**) con todos los lotes que se encontraron. Posteriormente el Analista escoge el lote le indica buscar (**Acción 8**) a la CP_Buscar_Libro_de_entrada_de_cuarentena que está compuesta por el formulario (Fr_Buscar_Libro_de_entrada_cuarentena) y a través de éste le envía un submit a la index.php (**Acción 9**), ésta a su vez le envía a sfController el módulo (Libroentradacuarentena) y la acción (Buscar) a realizar (**Acción 10**), luego sfController le indica a la clase Actions(LibroentradacuarentenaActions) cual es la acción que debe ejecutar executeBuscar() (**Acción 11**), y luego valida el formulario (**Acción 12**), si encuentra algún problema lanza un error, en caso contrario ejecuta la acción doSelect(\$c) (**Acción 13**) que dado un criterio devuelve todos los valores que cumplan con este luego. La clase Actions le envía a la clase BuscarSuccess.php un arreglo

(libroCuarentena) (**Acción 14**) con los datos de la entrada de cuarentena que coincida con el lote seleccionado en la clase, posteriormente la clase BuscarSuccess.php construye nuevamente la CP_Buscar_Libro_de_entrada_de_cuarentena (**Acción 15**) con todos los datos que se encontraron y da la posibilidad de modificar los datos del libro correspondiente a la entrada de cuarentena. Finalmente si el usuario escoge la opción Modificar (**Acción 16**) se realizan todas la acciones representadas en el *Diagrama de Secuencia (Escenario Modificar)* (**Acción 17**) **Figura 3.23**.

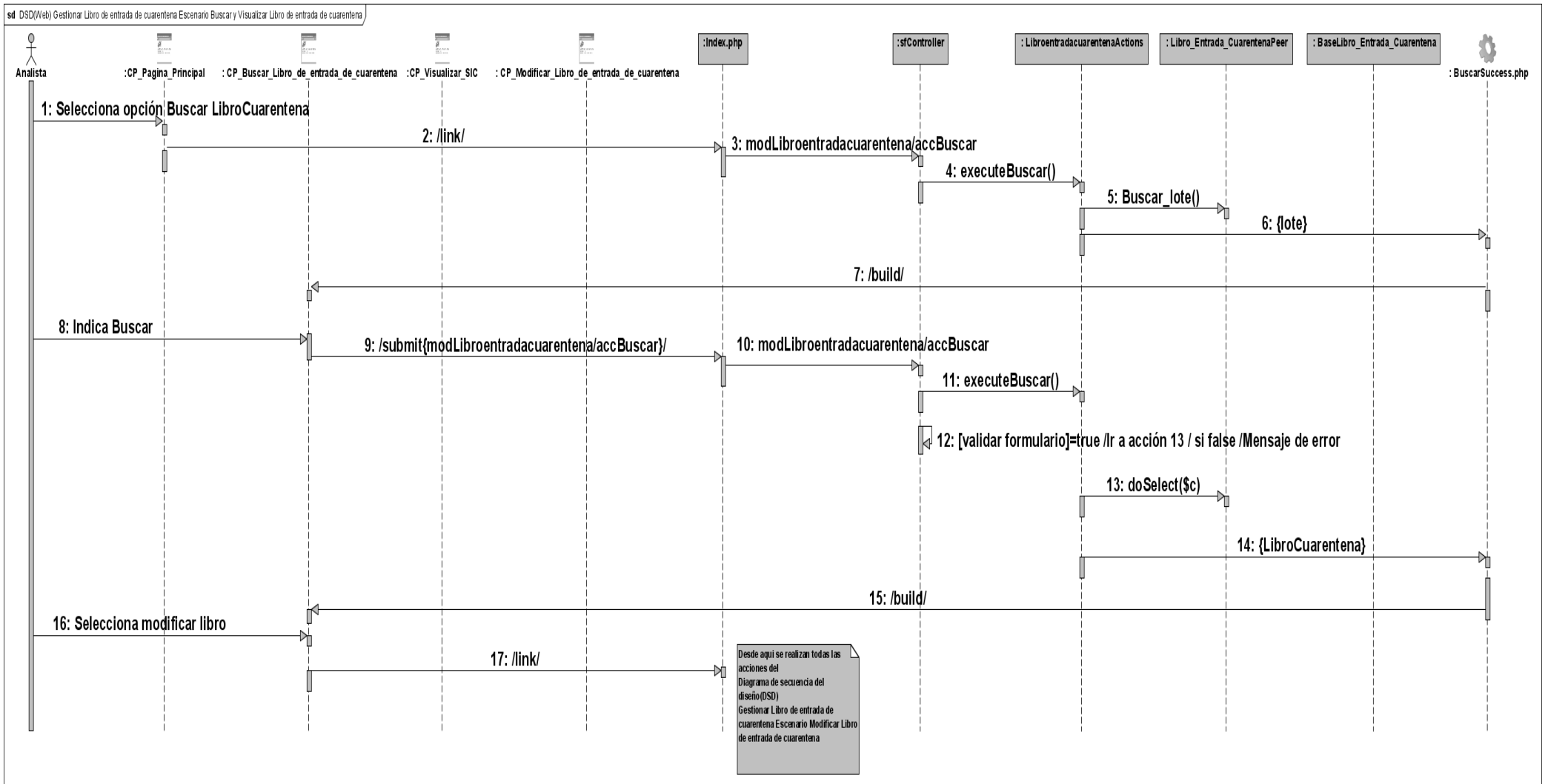


Figura 3.24 Diagrama de Secuencia (Escenario Buscar y Visualizar): CU “Gestionar Libro de entrada de cuarentena”.

Fragmento de código correspondiente a la clase *Actions.php*

El método que se muestra a continuación correspondiente a la clase `LibroEntradaCuarentenaActions` es para efectuar la búsqueda primeramente carga el Lote-CIGB que es el único criterio de búsqueda del libro de entrada de cuarentena mediante el método `Lote_CIGB()` que devuelve los valores insertados en la base de datos sin repetir ninguno de estos con ayuda del método `doSelect($c)`, que devuelve todos los objetos que se encuentran en las tablas en dependencia del objeto tipo `Criteria` que se le pase por parámetro, luego se comprueba si la posición del Lote-CIGB en el combo-box es distinta de cero, entonces se ha seleccionado un valor y en este caso no se emite ningún error, en caso contrario si, posteriormente se realiza la búsqueda con el valor del parámetro escogido haciendo una consulta a la base de datos, y como resultado de la búsqueda realizada devuelve los libros que cumplan con lo especificado, si no se encuentra ninguno, se emite un mensaje de error, a continuación se muestra el código correspondiente.

```

class LibroEntradaCuarentenaActions extends sfActions
public function executeBuscar () (Acción 4 y 11)

    {
        $this->mensaje='';
        $this->arreglo=-1;
        $lote=array("Select....");
        $fo= LibroEntradaCuarentenaPeer::Buscar_Lote();
        for($i=0;$i<count($fo);$i++)
        array_push($lote,$fo[$i]);
        $this->bus_lote=$lote;
        $this->mensaje='';
        if ($this->getRequest()->getMethod() != sfRequest::POST)
        {
            $this->mensaje='';
            return sfView::SUCCESS;
        }else
        {
            $posicion_lote=$this->getRequestParameter('lote');
            $this->posicion_lote1=$posicion_lote;
        }
    }

```

```

        if($posicion_lote==0 )                (Acción 12)
    {
        $this->arreglo=-1;
        $this->mensaje='Entre el Criterio de Búsqueda';
        return sfView::SUCCESS;
    }else{
        $c = new Criterias();                (Acción 13)
        if($posicion_lote!=0)
        {
            $lote_buscar=LibroEntradaCuarentenaPeer::ParaSelect
            ($lote,$posicion_lote);
            $c->add(LibroEntradaCuarentenaPeer::LOTE_CIGB,$lote_buscar);}
            $libro=LibroEntradaCuarentenaPeer::doSelect($c); (Acción 14)
            $dev1=LibroEntradaCuarentenaPeer::Buscar($libro);
            $this->arreglo=$dev1; } } }
    
```

Tabla 3.1 “class *LibroentradaCuarentenaActions*”.

Fragmento de código correspondiente a la clase *LibroEntradaCuarentenaPeer.php*

El siguiente método `Buscar_lote ()` correspondiente a la clase `BaseLibroEntradaCuarentenaPeer` como su nombre lo indica es mostrar todos lotes sin repetir ninguno de los ya insertados en el mismo, estos se guardan en dos arreglos y se recorren los mismos comprobando que estos valores en las posiciones no coincidan y como resultado se devuelve un arreglo con los lotes sin repetir ninguno.

```

class LibroEntradaCuarentenaPeer extends BaseLibroEntradaCuarentenaPeer
public static function Buscar_lote ()                (Acción 5)
{
    $arr=array();
    $arr1=array();
    $c = new Criterias();
    $r=LibroEntradaCuarentenaPeer::doSelect($c);
    for ($i=0;$i<count($r);$i++)
    
```

```

array_push($arr,$r[$i]->getLoteCigb());

array_push($arr1,$arr[0]);

for($i=1;$i<count($arr);$i++)

{ $aux=true;

for($j=0;$j<count($arr1);$j++)
{ $aux=true;
if($arr[$i]==$arr1[$j])
break;

else $aux=false;}

if($aux==false)

array_push($arr1,$arr[$i]);}

return $arr1;}

```

Tabla 3.2 “*class LibroEntradaCuarentenaPeer*”.

El siguiente método `Buscar($libro)` es invocado por la clase `Actions` en el execute `buscar` y devuelve todos los campos en un arreglo de arreglos.

```

public static function Buscar($libro)
{ $dev=array();

for($i=0;$i<count($libro);$i++)

{ $aux=array();

array_push($aux,$libro[$i]->getAnno());

array_push($aux,$libro[$i]->getFolio());

array_push($aux,$libro[$i]->getMateriaPrima());

array_push($aux,$libro[$i]->getLoteCigb());

array_push($aux,$libro[$i]->getProveedor());

```

```

array_push($aux,$libro[$i]->getLoteProveedor());
array_push($aux,$libro[$i]->getNoEnvase());
array_push($aux,$libro[$i]->getUM());
array_push($aux,$libro[$i]->getRecibe());
array_push($aux,$libro[$i]->getDecision());
array_push($aux,$libro[$i]->getDiaEntrada());
array_push($aux,$libro[$i]->getMesEntrada());
array_push($aux,$libro[$i]->getAnnoEntrada());
array_push($aux,$libro[$i]->getMesVenc());
array_push($aux,$libro[$i]->getAnnoVenc());
array_push($aux,$libro[$i]->getDiaDec());
array_push($aux,$libro[$i]->getMesDec());
array_push($aux,$libro[$i]->getIdLibro());
array_push($dev,$aux); }
return $dev; }

```

Tabla 3.3 “function Buscar (\$libro)”.

La encuesta aplicada a los programadores arrojó los siguientes resultados:

Aspecto # 1: Comprensión del diseño.

En el caso de los artefactos: diagramas de clases del diseño y los diagramas de secuencia, fueron muy importantes para el comienzo de la implementación, y les facilitó a los programadores un mejor entendimiento de los atributos, las clases a implementar, los métodos contenidos en cada una de las clases, además de las relaciones que se establecen entre ellas, y han entendido el flujo de mensajes que se representa en los mismos, considerándolos legibles y apropiados para comenzar con la implementación del sistema.

Aspecto # 2: Correspondencia de los requerimientos no funcionales con el diseño.

En el caso del requisito no funcional de Seguridad se evidencia en el diseño realizado puesto que el usuario antes de realizar acceder a la información que necesita consultar, se debe autenticar para

garantizar que la información sensible sólo pueda ser vista por los usuarios con el nivel de acceso adecuado.

Aspecto # 3: Descomposición del diseño en paquetes.

Los diagramas del diseño estaban estructurados en paquetes lo que propició la organización en la implementación pues fue posible descomponer los trabajos de implementación en partes más manejables.

Conclusiones

Se desarrolló la totalidad de los diagramas de clases del diseño para los 42 casos de uso identificados y solo cuatro diagramas de interacción, específicamente diagramas de secuencia, además se realizaron los prototipos funcionales para los 42 casos de uso, se realizó la validación del diseño arrojando un resultado satisfactorio. Se confeccionó el mapa de navegación para garantizar una adecuada navegabilidad por parte del usuario y se obtuvo como una referencia a la arquitectura del sistema la Vista Lógica y de Despliegue.

CONCLUSIONES

Como resultado de la investigación realizada se puede concluir que se cumplieron los objetivos propuestos, se realizó un estudio de los procesos que se desarrollan en el laboratorio Análisis Químico, logrando una familiarización con los mismos, lo que permitió identificar claramente los requerimientos del sistema y se diseñaron todas las clases para este módulo del LIMS, dejándolo listo para la implementación.

La utilización de RUP como proceso de desarrollo de software permitió estructurar el trabajo de forma organizada en sus diferentes etapas. El uso de UML resultó de gran apoyo para la mejor comprensión de diagramas de clases y la realización de las estructuras necesarias para la construcción del sistema.

La arquitectura Modelo-Vista-Controlador facilitó estructurar el diseño de la aplicación Web con vista a la implementación.

RECOMENDACIONES

- Se extienda la experiencia del desarrollo del LIMS a los demás centros del Polo Científico.
- Se use RUP como guía de desarrollo de software y como lenguaje de modelado UML, para modelar los artefactos correspondientes a las futuras tesis.
- Se use para modelar los artefactos de los demás flujos de trabajo la herramienta CASE Visual Paradigm porque es de interés para la facultad migrar al Sistema Operativo Linux.

REFERENCIAS BIBLIOGRÁFICAS

1. CIGB, Laboratorio de Análisis-Químico. 2003. [Cited: 11 20, 2007.]
http://www.cigb.edu.cu/pages/aq_calidad.htm.
2. CIGB, Departamento de Control de Calidad. 2003. [Cited: 11 20, 2007.]
<http://www.cigb.edu.cu/pages/ccalidad.htm>.
3. CIGB, Departamento de Aseguramiento de la Calidad. 2003. [Cited: 11 20, 2007.]
<http://www.cigb.edu.cu/pages/acalidad.htm>.
4. Sistema de información. [Cited: 11 21, 2007.] <http://eclases.tripod.com/id28.html>.
5. Sistema de Gestión de la Información del Laboratorio. [Cited: 11 21, 2007.]
<http://www.labware.com/lweuseminars.nsf/h/Latin%20Am>.
6. Proceso de desarrollo de software. [Cited: 12 10, 2007.]
<http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#IngSoft>.
7. LIMS o "Laboratory Information Management System". [Cited: 11 21, 2007.]
<http://www.onlabweb.com/lims.htm>.
8. Herramientas de Gestión de Requisitos. [Cited: 12 12, 2007.] <http://www.ines.org.es/vulcano/wp-content/uploads/2007/09/d6-estudio-de-herramientas-de-certificacion-tid.pdf>.
9. Herramienta CASE. [Cited: 12 12, 2007.]
<http://www.elprisma.com/direc678/redirect.asp?id=13324&idsesion=248810045>.
10. Gestión de la Información. [Cited: 6 3, 2008.]
http://bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm.
11. El Proceso Unificado Rational. [Cited: 12 10, 2007.]
<http://www.rational.com.ar/herramientas/rup.html> <http://hancocchi.net/el-rol-del-analista-en-rup/>.
12. Características de RUP. [Cited: 12 10, 2007.]
<http://teleformacion.uci.cu/mod/resource/view.php?id=6655>.
13. Rol. [Online] [Cited: 12 13, 2007.] <http://hancocchi.net/el-rol-del-analista-en-rup/>.
14. Un producto o artefacto. 2003. [Cited: 12 13, 2007.] Ayuda extendida del Rational Rose Enterprise Edition 2003.
15. Patrón. [Online] [Cited: 12 16, 2007.] <http://teleformacion.uci.cu/mod/resource/view.php?id=10963>.

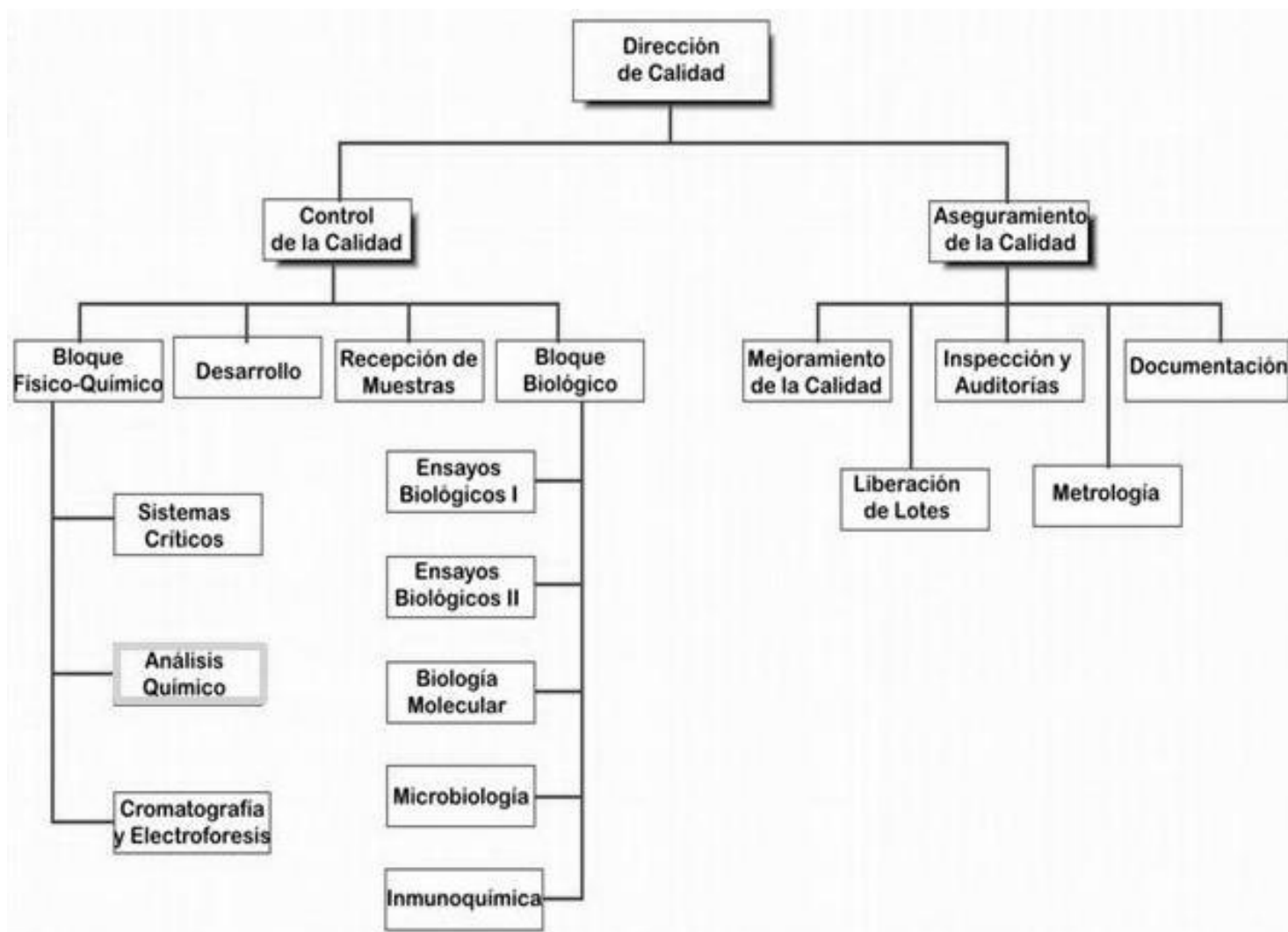
16. Patrón MVC. [Cited: 5 22, 2008.] http://www.librosweb.es/symfony/capitulo2/el_patron_mvc.html.
17. Mario Rodríguez Martín, David Sánchez Fernández, Roberto Santa Escolástica Villoria. Patrones de Diseño: Decorator.
18. Jacobson, Ivar and Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo Volumen I, The Addison Wesley Longman Inc. 1999.
19. Conallen, Jim. UML Extension for Web Applications v0.91. 1999.
20. Díaz, Alieski Sarmiento Almenares y Elian Cutiño. LIMS DE CALIDAD DEL CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA: ANÁLISIS DEL GRUPO DE RECEPCIÓN DE MUESTRAS Y MANIPULACIÓN DE EXPEDIENTES. 2006.
21. Arquitectura de Software. [Cited: 5 25, 2008.]
http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.msp.
22. Framework a utilizar: Symfony. [Cited: 5 22, 2008.]
http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html.

BIBLIOGRAFÍA

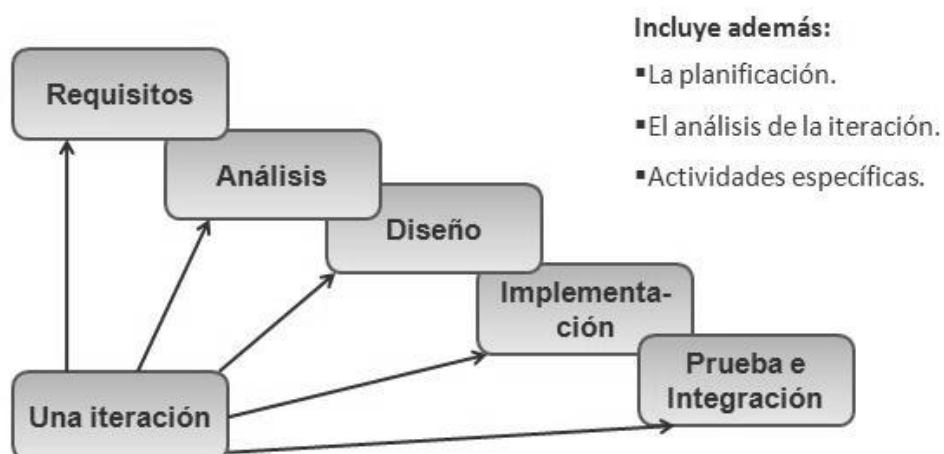
- <http://www.onlabweb.com/lims.htm>. (15/11/2007)
- <http://teleformacion.uci.cu/mod/resource/view.php?id=8865>.(16/11/2007)
- <http://www.cibersociedad.net/archivo/articulo.php?art=218>. (22/11/2007)
- <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.doc>. (12/12/2007)
- <http://www.addlink.es/productos.asp?pid=472>. (14/12/2007)
- <http://www.prnewswire.co.uk/cgi/news/release?id=122726>.(14/12/2007)
- <http://www.quaass.com/web/productos.htm>. (14/12/2007)
- <http://docs.kde.org/kde3/es/kdesdk/umbrello/14/12/2007uml-elements.html>.(15/12/2007)
- <http://www.modelizacion.com/productos.asp?pid=472>. (15/12/2007)
- *Ayuda extendida del Rational Rose Enterprise Edition 2003*. (15/12/2007)
- <http://www.visual-paradigm.com/product/vpuml>. (15/12/2007)
- 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 [Book] / auth. Lic. Yimian de Lyz Contreras Díaz Soleydi Rivero Amador. - 2007.
- 1 3, 2008. - <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>. (03/01/2007)
- <http://mit.ocw.universia.net/6.170/6.170/f01/pdf/lecture-12.pdf>. (01/01/2007)
- Universidad Politécnica de Valencia. Rational Unified Process (RUP) [Disponible en http://java.ciberaula.com/articulo/disenio_patrones_j2ee].(03/02/2008)
- <http://ceds.nauta.es/informes/case01b.htm>. (03/03/2008)
- <http://teleformacion.uci.cu/mod/resource/view.php?id=10963>. (03/03/2008)
- Universidad de Chile Departamento de Ciencias de la Computación CC61J - Taller de UML, tema: UML - Diagramas de interacción / auth. Guerrero Luis A..
- <http://miguelmadero.blogspot.com/2007/04/importancia-de-tdd-contra-la-matriz-de.html>.(06/04/2008)

- <http://www.svshome.com/newsletter/boletin03.html>. (06/04/2008)
- <http://www.rational.com.ar/herramientas/requisitepro.html>. (06/04/2008)

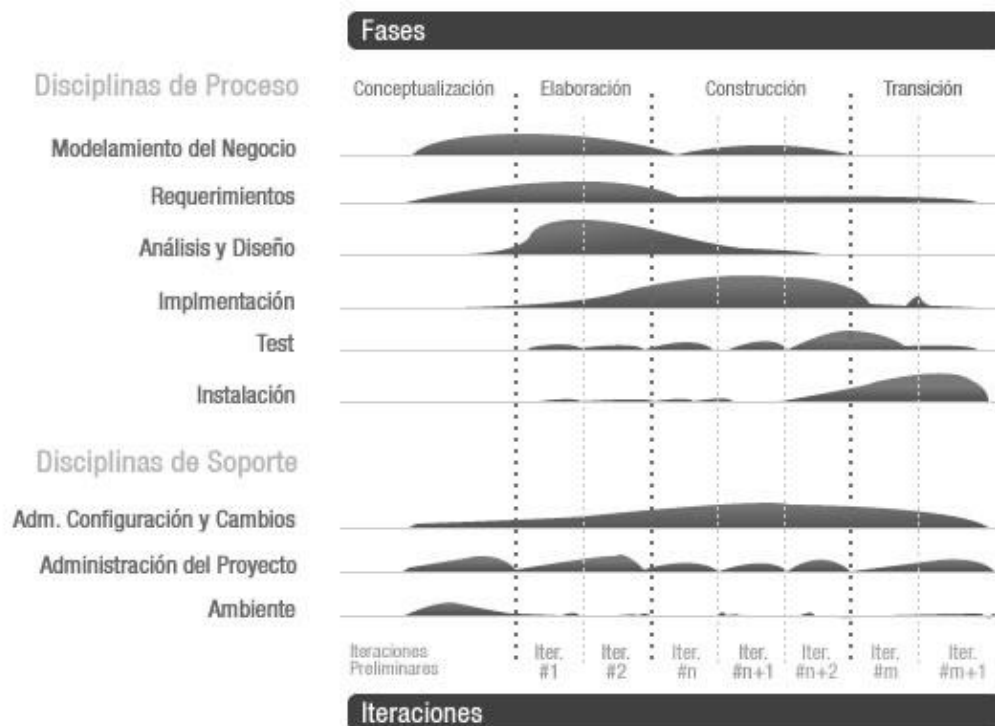
ANEXOS

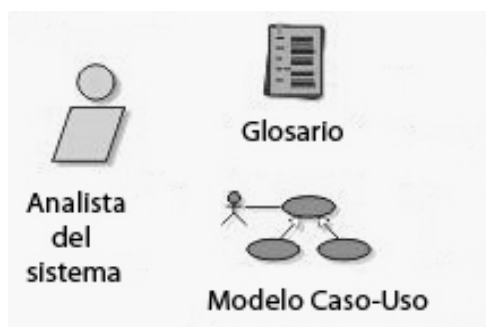
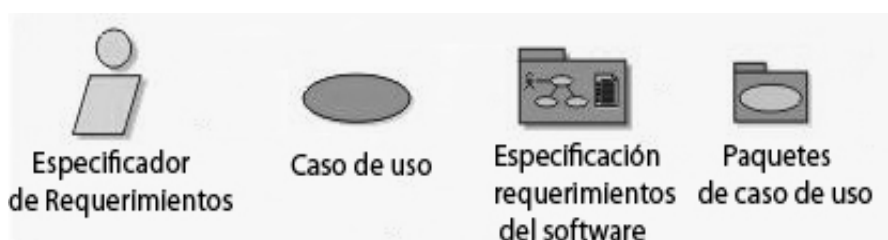
Anexo 1. Estructura jerárquica del Área de Calidad del Centro de Ingeniería Genética y Biotecnología

Anexo 2. Una iteración de RUP



Anexo 3. Fases e Iteraciones de RUP

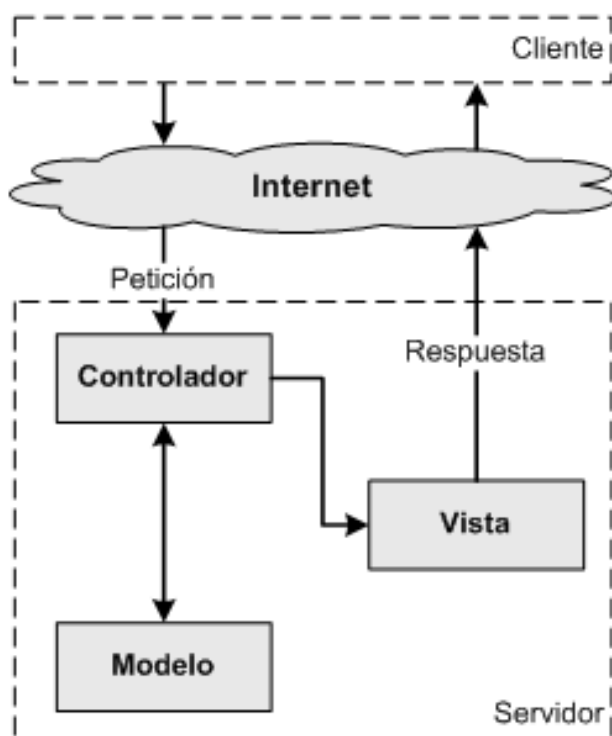


Anexo 4. Artefactos a realizar por el Analista del Sistema**Anexo 5. Artefactos a realizar por el Especificador de Requerimientos****Anexo 6. Artefactos a realizar por el Diseñador**

Anexo 7. Artefactos a realizar por el Diseñador de Interfaz de Usuario



Anexo 8. Funcionamiento del patrón MVC



GLOSARIO DE TÉRMINOS

Aseguramiento de la Calidad: parte de la gestión de la calidad orientada a proporcionar confianza en que se cumplirán los requisitos de la calidad.

Nota: El concepto de Aseguramiento de la Calidad abarca todos los aspectos que individual o colectivamente influyen en la calidad de un producto e incorpora no solo las Buenas Prácticas de Fabricación, sino también otros factores o elementos que van más allá del alcance de esta regulación, tales como el diseño y desarrollo de los productos.

AQ: Análisis Químico.

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.

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.

Calidad: grado en que un conjunto de características inherentes cumplen con los requisitos.

Control de la Calidad: parte de la gestión de la calidad orientada al cumplimiento de los requisitos de la calidad.

Ensayo: es la aplicación de un análisis a una o varias muestras.

GPL: (del inglés General Public License) Licencia pública general. Es una licencia que está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Hardware: componentes físicos que constituyen las computadoras y demás dispositivos periféricos.

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 elaborado en un solo proceso o en una serie de procesos, de tal manera que pueda esperarse que sea homogéneo.

Muestra: pequeña parte que es representativa de un lote en un tiempo y condiciones específicas o determinadas.

PHP: (del inglés Hyper Text Pre-Processor / Personal Home Pages). Es un lenguaje de programación usado frecuentemente para la creación de contenido para sitios web con los cuales se puede programar las páginas html y los códigos de fuente.

Producto: es el resultado de una investigación al que finalmente se le establecen dos nombres: uno genérico y otro comercial. Está destinado al uso que su investigador haya propuesto y definido.

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.