

Universidad de las Ciencias Informáticas

Facultad 6.



***TÍTULO: “LIMS DE CALIDAD DEL CENTRO DE INGENIERÍA GENÉTICA
Y BIOTECNOLOGÍA: IMPLEMENTACIÓN DEL MÓDULO DE LA
SECCIÓN DE MEJORAMIENTO DE LA CALIDAD.”***

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

Autores: Mariagne Rodríguez Prawl.
Dennis Molinet Guzmán.

Tutora: Ing. Elennis Díaz Laurencio.
Co-Tutora: Lic. Miulkenia Navarro Reyes.

Junio 2008

“Año 50 de la Revolución”

Libre, y para mi sagrado, es el derecho de pensar... La educación es fundamental para la felicidad social; es el principio en el que descansan la libertad y el engrandecimiento de los pueblos.

- Benito Juárez

DECLARACIÓN DE AUTORÍA

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

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

Mariagne Rodríguez Prawl

Dennis Molinet Guzmán

Ing. Elennis Diaz Laurencio

Lic. Miulkenia Navarro Reyes

DATOS DE CONTACTO

Tutoras:

Ing. Elennis Diaz Laurencio
Universidad de las Ciencias Informáticas, Habana, Cuba.
Email: ediaz@uci.cu

Lic. Miulkenia Navarro Reyes
Universidad de las Ciencias Informáticas, Habana, Cuba.
Email: mnavarror@uci.cu

OPINIÓN DEL TUTOR

DEDICATORIA

Dedico este trabajo a mis padres Ana María Prawl y Fredy Rodríguez, por ser mi fuente de inspiración y confianza.

En especial a mis abuelos que aunque no estén conmigo se que están muy orgullosos de mi.

A mi única hermana por estar siempre ahí para mí. Te quiero muchísimo.

A mi novio Jorge Luis por su amistad y amor, por estar en los momentos difíciles; gracias por estar en mi vida.

A mi compañero de tesis, muchas gracias por haberme escogido para este trabajo tan importante, te quiero mucho.

A mi tia Marilyn y a mi tío Raúl por su amor y su cariño, por estar ahí siempre que los he necesitado, a mis primos, a mi cuñado, a mis sobrinitos por ser los angelitos más lindos que tengo. Los quiero mucho.

A Nina, Yudelsi, Jimmy, Irais, Elsia, Tita, Yodelvis, Ricardo, por su cariño.

A mis demás amistades de la UCI por su cariño y dedicación.

A norita y sus padres por recibirme con amor y cariño. Muchas gracias.

A todos aquellos que de alguna manera han transitado por mi vida y han dejado en ella una huella indestructible que no se borrará con el paso del tiempo.

Mariagne Rodríguez Prawl

Dedico este trabajo especialmente a mi madre María A. Guzmán Mederos, que no está físicamente conmigo pero sé que donde quiera que esté estará muy orgullosa de mí. De igual manera a mi padre Rubén Molinet Pérez y mi hermana Darlyn Molinet Guzmán por ser una guía y un ejemplo para mí y por haberme ayudado tanto en mi vida en general. A mis abuelos paternos y maternos Estela, Rubén, y Angela que fueron los que me enseñaron a dar los primeros pasos en la vida.

También a mi novia Mislenis Morenos Pérez por compartir conmigo mis malos y buenos momentos, por preocuparte tanto por mí cuando se que no me lo merezco mucho. Te quiero mucho...Serio, serio, serio.

A mi compañera de tesis que es la mejor compañera de tesis de toda la escuela y sabe que la voy a extrañar cantidad. Te quiero mucho también.

A mis amistades tanto de la UCI como del barrio. A Ricardo, Manuel, El Tommy, Ariesky que lo molesté tanto por la tesis, a Yosmel, Adrián e Iván mis hermanos del barrio. A mis otras hermanas Yoleidis, Lizandra y a Lisbey por compartir tantos momentos conmigo.

A todos aquellos que de alguna manera han transitado por mi vida y han dejado en ella una huella indestructible que no se borrará con el paso del tiempo.

Dennis Molinet Guzmán

AGRADECIMIENTOS

Agradecimiento eterno a la Revolución Cubana y a nuestro Comandante en Jefe Fidel Castro Ruz por brindarnos la posibilidad de estudiar aquí y obtener los conocimientos necesarios para ser futuros profesionales dignos de este país.

Agradecemos de todo corazón a nuestros padres por su confianza en nosotros. Los queremos mucho.

A nuestras tutoras Elennis y Miulkenia por su amistad, guía y apoyo.

A nuestros compañeros de aula por compartir cada momento con nosotros durante estos 5 años.

A nuestros profesores de la UCI, por brindarnos cada día, con paciencia y esmero sus conocimientos que nos ayudaron en nuestra formación.

RESUMEN

En la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología, se gestiona a diario un gran volumen de información relacionada con los ensayos que allí se realizan, la cual es de vital importancia en el control y aseguramiento de la calidad de las producciones del Centro. Para contribuir a la gestión y control de la información que se genera, se está trabajando en la realización de un Sistema de Gestión de Información de los Laboratorios (LIMS) para esta dirección. El presente trabajo aborda la Implementación del Módulo de la Sección de Mejoramiento de la Calidad, partiendo de un diseño previo realizado en un trabajo de diploma como precedencia, siguiendo la metodología RUP, a través del Rol de Implementador que propone la misma y haciendo uso de las herramientas para la programación web de actualidad como el Framework Symfony, el Gestor de Base de Datos (SGBD) PostgreSQL y la Herramienta Case de Modelado Visual Paradigm, con el objetivo de poner en práctica una aplicación informática que permita realizar de forma personalizada para cada usuario según sus responsabilidades, los procesos involucrados en la gestión de la información del módulo.

Palabras claves: LIMS, gestión de la información.

TABLA DE CONTENIDOS

DEDICATORIA.....	I
AGRADECIMIENTOS	III
RESUMEN	IV
INTRODUCCIÓN	1
CAPITULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción.....	5
1.2 Sistemas de Gestión de Información de los Laboratorios.....	5
1.3 Metodologías de Desarrollo	7
1.3.1 Proceso Unificado de Desarrollo de Software (RUP).....	7
1.3.2 Flujo de trabajo de Implementación	10
1.3.3 Objetivos del flujo de trabajo de Implementación:.....	10
1.3.4 Roles y artefactos:.....	10
1.3.5 Actividades que desempeña el Implementador.....	11
1.3.6 Lenguaje de Modelado.....	11
1.4 Herramienta de modelado.....	12
1.5 Lenguajes de Programación	12
1.5.1 JavaScript.....	12
1.5.2 PHP 5	13
1.5.3 Eclipse.....	14
1.6 Framework:	14
1.6.1 Symfony:	14
1.7 Gestores de base de datos a usar: Postgre SQL.....	15
1.8 Conclusiones.....	15
CAPITULO 2: IMPLEMENTACIÓN DEL SISTEMA	16
2.1 Introducción.....	16
2.2 Listado de los Requerimientos	16
2.3 Arquitectura utilizada.....	25

2.4 Diagrama de Componentes	37
2.5 Código Fuente.....	48
2.6 Tratamiento de Errores	74
2.7 Conclusiones.....	77
CONCLUSIONES.....	78
RECOMENDACIONES	79
REFERENCIA BIBLIOGRÁFICA.....	80
BIBLIOGRAFÍA	81
ANEXOS	83
GLOSARIO DE TÉRMINOS.....	99

ÍNDICE DE FIGURAS

Figura 1 Diagrama de componentes del CU Gestionar SIC0098	39
Figura 2 Diagrama de componentes del CU Gestionar SIC0816	40
Figura 3 Diagrama de componentes del CU Gestionar SIC0919	41
Figura 4 Diagrama de componentes del CU Gestionar SIC0963	42
Figura 5 Diagrama de componentes del CU Gestionar SIC0851	43
Figura 6 Diagrama de componentes del CU Gestionar SIC0042	44
Figura 7 Diagrama de componentes del CU Gestionar SIC0023	45
Figura 8 Diagrama de componentes del CU Gestionar SIC0959	46
Figura 9 Diagrama de componentes del CU Gestionar SIC0024	47

INTRODUCCIÓN

En la actualidad las Tecnologías de la Información y las Comunicaciones (TIC) ocupan un lugar importante en el desarrollo de cualquier país del mundo, estas se encargan del estudio, desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de hardware y software como medio de sistema informático.

En estos últimos años, se está evidenciando una verdadera estrategia del país en la introducción de las TICs, complementando su perspectiva tecnológica y humanista, de equidad, participación y justicia social, priorizando el uso en los sectores como la ciencia, la educación y la salud que reversion indirectamente su labor y por tanto benefician a toda la población cubana y la coloca en una situación ventajosa para emprender el reto de la informatización, pues nuestro país está consciente de que una sociedad para ser más eficaz, eficiente y competitiva debe aplicar la informatización en todas sus esferas y procesos.

Para el desarrollo de aplicaciones que gestionan gran cantidad de información, es que a finales del siglo pasado se crean los sistemas de información, que son un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

La gestión de la información es el proceso de analizar, utilizar, recuperar y almacenar la información que se ha obtenido y registrado para permitir a los administradores tomar decisiones documentadas.

En la actualidad la industria farmacéutica depende en gran medida del éxito de los laboratorios, los cuales enfrentan un aumento considerable en su carga de trabajo, una demanda creciente en la generación de resultados elaborados. Presenta además un control riguroso de la validez, adquisición y el tratamiento de los datos con el objetivo principal de mantener la integridad de la información y validez de las decisiones tomadas.

Para ayudar al correcto funcionamiento de estos laboratorios es que surge a finales del siglo pasado los Sistemas de Gestión de la Información de los Laboratorios (LIMS o Laboratory Information Management System). “Un LIMS es un programa de gestión de laboratorios que permite recoger, almacenar, calcular y gestionar datos en una amplia variedad de formas. 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.”[1]

El uso de los LIMS ha favorecido en gran medida la manipulación de los datos que se llevan en los laboratorios. De forma más concreta, se pueden señalar varias áreas fundamentales para la identificación de beneficios, entre los que 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 partir de la década del 80 del siglo pasado, se intensificó en el país la actividad científica aplicada a la salud, que tuvo como colofón la inauguración, el 1ro de julio de 1986, del Centro de Ingeniería Genética y Biotecnología (CIGB), y la creación en 1992 del Polo Científico en el oeste de La Habana. Este Centro marcó la mayoría de edad del pujante movimiento científico de la isla. Su labor ha tenido gran impacto en la biomedicina, salud animal, mejoramiento vegetal y la bioindustria; y está encaminada a contribuir de manera significativa con el desarrollo económico y social del país.

Durante estos 20 años se han desarrollado nuevas vacunas y fármacos para la salud humana que se encuentran actualmente en uso dentro del Sistema de Salud Cubano, así como en diferentes países.

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**.

“El **Departamento de Control de la Calidad** realiza varias funciones entre las cuales están las relacionadas con el muestreo, las especificaciones, los ensayos y la evaluación de la calidad de los productos que se generan en el centro” [2], para el desempeño de las mismas, cuenta con la ayuda de dos grupos de trabajo y dos secciones:

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

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

“El **Departamento de Aseguramiento de la Calidad** garantiza que se lleven a cabo las acciones planificadas y sistemáticas que son necesarias para proporcionar la confianza de que nuestros productos y servicios satisfacen los requisitos de calidad establecidos. Vela por el cumplimiento de las Buenas Prácticas de Producción (BPP), Buenas Prácticas de Laboratorio (BPL) y Buenas Prácticas Clínicas (BPC). Este Departamento está compuesto por dos Secciones y dos grupos de trabajo:” [3]

- **Sección de Mejoramiento de la Calidad (SMC).**
- Sección de Inspección, Auditoría y Liberación de lotes.
 - Grupo de Inspección y Auditorías.
 - Grupo de Liberación de Lotes.
- Grupo de Documentación.
- Grupo de Metrología. (Ver Anexo No. 1)

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

Además de la cantidad de material de oficina empleado para el desempeño laboral de dichos trabajadores, la elaboración de reportes rutinarios u ocasionales se les dificulta considerablemente, dado básicamente por una ineficiente forma de búsqueda en el gran cúmulo de libros a consultar.

Con anterioridad un grupo de analistas realizaron el análisis de los procesos de manipulación de información de esta sección, sin obtener resultados factibles, por lo que se puede concluir que el proceso de gestión de la información de la SMC no fluye de manera eficiente, segura y rápida. Por lo

que se identifica como **problema científico**: ¿Cómo obtener un producto funcional a partir de las clases diseñadas para el módulo de la SMC de la Dirección de Calidad del CIGB?

El problema planteado se enmarca en el **objeto de estudio**: Proceso de Desarrollo de los Sistemas de Gestión de la Información.

El objetivo delimita el **campo de acción**: Implementación de los Sistemas de Gestión de la Información.

Para dar solución al problema se define como **objetivo**: Implementar el módulo de la SMC para el LIMS de la Dirección de Calidad del CIGB.

Para lograr los objetivos, se realizarán las siguientes **tareas**:

- Estudio del Modelamiento del Negocio.
- Estudio del Modelo del Sistema.
- Estudio del Modelo del Diseño.
- Realización de los diagramas de componentes.
- Implementación de los componentes.

El trabajo consta de Introducción, 2 capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografías, anexos y glosario de términos.

• En el Capítulo 1 **Fundamentación Teórica**: queda demostrado la necesidad de realizar un LIMS para facilitar el intercambio, manejo, procesamiento y almacenamiento de la información de los laboratorios del CIGB; se abordan las aplicaciones en las que se apoya para la solución, se brinda información de los artefactos realizados y herramientas CASE enmarcadas en el rol de programador.

• En el Capítulo 2 **Implementación del Sistema**: Se obtienen los artefactos del flujo de trabajo de implementación, fundamentalmente los diagramas de componentes; además se muestra el código fuente más significativo de las principales clases implementadas del módulo.

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo, para lograr una mejor comprensión de la necesidad de este trabajo se realiza una valoración de los Sistemas de Gestión de Información de los Laboratorios así como de los procesos de desarrollo de software más conocidos, de los artefactos y herramientas enmarcados en el rol de programador del proceso de desarrollo. Así como la tecnología del lado del servidor: PHP y del lado del cliente: Java Script y utilizando como framework Symphony.

1.2 Sistemas de Gestión de Información de los Laboratorios.

Muchas empresas se han dedicado a la fabricación de estos Sistemas de Gestión de Información de los Laboratorios para el manejo en particular de cada una de sus propias necesidades. A continuación se mencionan algunos ejemplos de LIMS donde se brinda una breve explicación de las funcionalidades principales de cada uno de ellos.

- **MediLab Interfaces:**

MediLab Interfaces es un sistema de información que permite extraer los datos de los resultados de los equipos de análisis y almacenarlos en el LIMS de su laboratorio. Al implantar una interfaz entre sus equipos de análisis y su LIMS se evitan la transcripción de resultados, trayendo estos como consecuencia la desaparición de los errores de transcripción, disminución del tiempo de reporte de resultados y la reducción de los costos del procesamiento de muestras.

MediLab Interfaces se integra perfectamente a MediLab LIS sin necesidad de realizar modificaciones. MediLab Interfaces también puede integrarse a otro, en este caso es necesario crear un programa de interpretación para que permita colocar los resultados en su LIMS, para ello se debe contar con acceso a la base de datos de su LIMS.

Funcionalidades [4]:

- Extracción de resultados del equipo de análisis hacia el LIMS por lotes o individual.
- Relación de resultados extraídos en un período de tiempo.

Características generales [4]:

- Permite conectar tantos equipos de análisis como el laboratorio requiera.
- Maneja roles de seguridad para solo los usuario autorizados extraigan los resultados.
- Funciona en cualquier sistema operativo: Windows, Linux.
- Posee una interfaz de usuario gráfica bien fácil de usar.
- Puede conectarse a cualquier sistema LIMS.

○ **LabWare LIMS**

Uno de los beneficios claves identificados por nuestros clientes es la habilidad de conectar LabWare LIMS a otros sistemas dentro de la compañía. LabWare excelente experiencia demostrada de interfaces bidireccionales funcionando con éxito, tanto con soluciones comerciales como a sistemas propios de la empresa.

Funcionalidades [5]:

- Importar / Exportar – Una utilidad estándar de importación / exportación permite automáticamente el registro de muestras desde sistemas externos o intercambio de información. Esta utilidad usa XML y puede ser configurado fácilmente usando un editor común de hojas de cálculo.
- LabWare LIMS puede leer y escribir las bases de datos de aplicaciones externas, así que en muchos casos no es necesario hacer mucha configuración sobre las otras aplicaciones.
- Los eventos programados, como informes, el registro de entrada de muestras y las rutinas en LIMS Basic pueden estar preconfiguradas y luego activarse basándose en una planificación.
- LabWare LIMS proporciona un amplio conjunto de funcionalidades para interfasearse con otros sistemas permitiendo que la información sea compartida entre los sistemas y no tener que volver a entrarse a mano.

Características Generales [5]:

- LabWare LIMS presenta acceso equivalente cliente/servidor y Web.
- Se integra sin costuras en cualquier entorno informático.
- La aplicación corre sobre Windows™ y sobre un explorador Web, tal como Microsoft IE, Safari, Opera.
- La plataforma puede instalarse en Windows™, y cualquier versión de Unix o Linux.
- Como base de datos, se pueden usar todos los productos comerciales, desde ORACLE™ o SQL Server™ hasta DB2™.

Luego de haber realizado una investigación sobre los Sistemas de Gestión de la Información de los Laboratorios (LIMS) se decidió que las funcionalidades que presentan cada uno de ellos no se ajusta a las propias del cliente por tanto es necesario el desarrollo de un sistema que si cumpla con cada una de las características del CIGB, entre las que se pueden mencionar:

- La mayoría de lo proveedores de LIMS son de los Estados Unidos.
- Las características del CIGB impiden poder configurar un LIMS para su uso particular, porque se realizan ensayos diferentes a diferentes productos y eso provoca que se necesite un sistema adaptable a las necesidades específicas de la Dirección de Calidad del CIGB.
- Resulta menos costoso desarrollarlo, debido a que los procesos de instalación, mantenimiento y actualización cuestan miles de dólares y el sistema que necesita la Dirección de Calidad del CIGB requiere mantenimiento de por vida por ser tan grande y variable.
- Una vez desarrollado en Cuba puede ser extensible a los centros del Polo Científico y en general a entidades que lo requieran para mejorar y agilizar los procesos relacionados con el control de la calidad de sus producciones.

1.3 Metodologías de Desarrollo

Para el desarrollo de los sistemas de información se debe basar en guías para lograr un software que cumplan con las características mínimas que necesita. El objetivo del uso de metodologías de desarrollo es elevar la calidad del software a través de una mayor transparencia y control sobre el proceso. Hay que producir lo requerido en tiempo y con el costo esperado. Es una labor difícil decidir cuál es la metodología de desarrollo que mejor se adapta a la situación concreta a la que se enfrenta el proyecto en cuestión para obtener los resultados esperados y la reducción del tiempo que se necesita para la toma de decisiones.

Dadas todas estas características la dirección del equipo de desarrollo del proyecto tomó la decisión de aplicar la metodología RUP, teniendo en cuenta las particularidades que presenta este proyecto.

1.3.1 Proceso Unificado de Desarrollo de Software (RUP)

“Un proceso de desarrollo de software define quién hace qué, cómo y cuándo. RUP define cuatro elementos: los roles, que responden a la pregunta ¿Quién?, las actividades que responden a la pregunta ¿Cómo?, los productos, que responden a la pregunta ¿Qué? y los flujos de trabajo de las disciplinas que responden a la pregunta ¿Cuándo?” [6]

RUP 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 proceso bien definido, estructurado y adaptable a las características y necesidades de cada proyecto específico. Es aconsejable utilizar la metodología RUP para proyectos grandes, por que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios. La definición de este proceso está dada por tres características fundamentales: [6]

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

RUP divide el proceso en cuatro fases (Ver anexo 2), dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

Fase de inicio: las iteraciones hacen mayor énfasis en las actividades de modelado de negocio y de requisitos.

Fase de elaboración: las iteraciones se orientan al desarrollo de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientada a la arquitectura.

Fase de construcción: lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Fase de transición: garantiza que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Los roles definidos por RUP son: analistas, desarrolladores, administradores, soporte y producción, especialistas de pruebas y roles adicionales.

Los flujos de trabajo que propone RUP son: [7]

- **Modelado del negocio:** comprende los procesos que se desarrollan en la organización donde se va a implantar el producto.
- **Requerimientos:** establece las funcionalidades con las que debe cumplir el sistema, de modo que los usuarios finales puedan comprender y aceptar los requisitos que se especifiquen.
- **Análisis y Diseño:** traduce los requisitos a una especificación que describe cómo implementar el sistema.
- **Implementación:** implementa las clases y objetos en ficheros fuente, binarios, ejecutables. El resultado final es un sistema ejecutable.
- **Pruebas:** evalúa la calidad del producto que se está desarrollando.
- **Despliegue:** produce con éxito distribuciones del producto y distribuirlo a los usuarios.
- **Gestión del proyecto:** logra un balance al gestionar objetivos, riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.
- **Configuración y control de cambios:** mantiene la integridad de todos los artefactos que se crean en el proceso.
- **Entorno:** brinda soporte al proyecto con las adecuadas herramientas, procesos y métodos; además de especificar las herramientas que se van a necesitar.

Cada fase del RUP concluye con un hito bien definido, punto en el cual se deben tomar ciertas decisiones y alcanzar las metas claves antes de pasar a la siguiente fase. Los hitos para cada una de las fases son:

- Inicio – visión de los objetivos.
- Elaboración – prototipo de la arquitectura.
- Construcción – capacidad operacional inicial.
- Transición – liberación del producto.(Ver anexo 3)

RUP utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) como lenguaje de notación.

La investigación se basa principalmente en la implementación de este módulo, donde el flujo de trabajo de implementación se encuentra en la fase de construcción.

1.3.2 Flujo de trabajo de Implementación

La implementación se inició con el resultado del diseño. Este flujo de trabajo tiene como actividades fundamentales: la implementación de clases y objetos en ficheros fuentes, binarios y ejecutables. El resultado final de este flujo de trabajo es un sistema ejecutable; planificar las integraciones del sistema necesarias en cada iteración y la distribución del sistema asignando componentes ejecutables a nodos en el diagrama de despliegue. Implementar las clases y subsistemas encontrados durante el diseño.

1.3.3 Objetivos del flujo de trabajo de Implementación:

- Planificar las integraciones del sistema necesarias en cada iteración.
- Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue.
- Implementar las clases y subsistemas encontrados durante el diseño.
- Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones del sistema.

1.3.4 Roles y artefactos:

Implementador: El rol de Implementador es el responsable de desarrollar y probar los componentes, de conformidad con las normas del proyecto aprobado, para su integración en grandes subsistemas. También es responsable de desarrollar y probar los componentes de prueba y los correspondientes subsistemas.

Un Implementador se le puede asignar la responsabilidad de la aplicación de una parte estructural del sistema, (ya sea una clase o la implementación del subsistema), o de una parte funcional del sistema, (tales como la realización de casos de uso).

Artefactos que debe obtener el Implementador (Ver anexo 4):

- Modelo de Implementación: Representa la composición física de la implementación en términos de subsistemas de implementación y elementos de implementación (carpetas y ficheros, incluyendo código fuente, datos y ejecutables). El modelo de implementación es opcional. Si se decide crear este modelo las decisiones que debe tomar son: cómo relacionar el modelo de implementación y el modelo de diseño y cuáles elementos de implementación son suficientemente importantes para modelarlos.

- **Subsistemas de Implementación:** Una colección de componentes y otros subsistemas de implementación usados para estructurar el modelo de implementación y dividirlos en pequeñas partes que pueden ser integradas y probadas de forma separada.
- **Diagrama de Componentes:** Muestran la estructura de los componentes, incluidos los clasificadores que especifiquen los componentes, y los artefactos que las aplican. También se pueden utilizar para demostrar el alto nivel de la estructura del Modelo de Implementación en términos de subsistemas de implementación, y las relaciones entre los elementos de implementación. El uso más importante del Diagrama de Componentes de RUP es mostrar el alto nivel de la estructura del Modelo de Implementación.

En este trabajo el artefacto que se va a obtener es el Diagrama de Componentes.

1.3.5 Actividades que desempeña el Implementador

El implementador es el encargado de desarrollar y probar los componentes, de acuerdo con las normas aprobadas por el proyecto, para su integración en grandes subsistemas. Es responsable para desarrollar y probar los componentes de la prueba y los subsistemas correspondientes.

1.3.6 Lenguaje de Modelado.

El Proceso Unificado de Desarrollo utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) como lenguaje de notación. “UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema software.” [6]

UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. Es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

Un lenguaje de modelado contiene vistas, diagramas, símbolos o elementos de modelo y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos. (Ver Anexo 5.)

Vistas: Las vistas muestran diferentes aspectos del sistema modelado. Una vista es una abstracción que consiste en un número de diagramas y todos esos diagramas juntos muestran una "fotografía" completa del sistema.

Diagramas: Los diagramas son las gráficas que describen el contenido de una vista.

Símbolos o Elementos de modelo: Los conceptos utilizados en los diagramas son los elementos del modelo que representan conceptos comunes orientados a objetos, como clases, objetos, mensajes, y sus relaciones.

Reglas o Mecanismos generales: Proveen comentarios extras o información acerca del elemento de modelo; además proveen mecanismos de extensión para adaptar o extender UML a un método o proceso específico, organización o usuario.

1.4 Herramienta de modelado.

Visual Paradigm (versión 6.1) es una herramienta CASE que utiliza “UML”: como lenguaje de modelado y soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Se integra al entorno de desarrollo que se utiliza en este proyecto:

- Eclipse

Está diseñada para dar soporte a arquitectos de sistemas, diseñadores, desarrolladores, analistas de procesos de negocio y modeladores de datos en los procesos de desarrollo de software. Permite modelar diferentes diagramas como: diagramas de clases, de interacción, de componentes.

Visual Paradigm ofrece: [8]

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

1.5 Lenguajes de Programación

1.5.1 JavaScript

Para cumplir algunos requisitos funcionales del sistema se utilizó JavaScript en su versión 2.0, que es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintáxis semejante a la del lenguaje Java y el lenguaje C.

Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Las ventajas que ofrece JavaScript como lenguaje de programación son:

- JavaScript no requiere un tiempo de compilación, los scripts se pueden desarrollar en un período de tiempo relativamente corto. A esto podemos añadirle las características de interfaz como, por ejemplo, cuadro de diálogo, formularios y otros elementos GUI (Interfaz Gráfico de Usuario), son gestionados por el navegador y por el código HTML.
- Aunque JavaScript tiene muchas similitudes con Java, no incluye la sintáxis y reglas complejas de Java.
- Los programas JavaScript tienden a ser pequeños y compactos (en comparación con los applets de Java), no requieren mucha memoria ni tiempo adicional de transmisión. Además, al incluirse dentro de las mismas páginas HTML se reduce el número de accesos independientes a la red.
- Lenguaje de scripting seguro y fiable.
- El código JavaScript se ejecuta en el cliente. [9]

1.5.2 PHP 5

Como PHP (versión 5.2.5) es uno de los lenguajes de creación de páginas Web dinámicas más utilizados del mercado, para el desarrollo de esta aplicación se utiliza PHP 5, que cuenta con innumerables mejoras que consolidan su éxito y nos ofrece un nuevo abanico de posibilidades a la hora de Programar Orientados a Objetos, y posibilita un mejor trabajo en proyectos de gran magnitud, continuando como el lenguaje rápido y flexible que es.

Los avances que han surgido en PHP 5 son:

- Rendimiento mejorado y mejor utilización de la memoria.
- Más seguridad a través de los Filtros, que permiten a los programadores leer los campos de formularios de forma segura.
- Posibilidad de controlar el progreso de subida de ficheros.
- Programación orientada a objetos real.
- Más rápido y más estable.

1.5.3 Eclipse

Eclipse (versión 3.3.1.1) es una plataforma de desarrollo de código abierto. Es usado para múltiples aplicaciones como por ejemplo de escritorio, aplicaciones basadas en framework así como para la construcción de entornos de desarrollo (IDEs) que puedan ser utilizados para la construcción de aplicaciones Web, aplicaciones Java de todo tipo y programas C++, además ofrece un autocompletado de código para el desarrollo con el framework Symfony. Es un IDE de desarrollo multiplataforma ejecutándose en varios sistemas operativos como Windows y Linux.

1.6 Framework:

En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

1.6.1 Symfony:

Symfony (versión 1.0.11) es un completo framework diseñado para optimizar, por 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. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación Web.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios Web de comercio electrónico de primer nivel. 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. A continuación se muestran algunas de sus características.

Características:

Symfony se diseñó para que se ajustara a los siguientes requisitos [10]:

- 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, adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

1.7 Gestores de base de datos a usar: Postgre SQL

PostgreSQL (versión 8.2) es un SGBD Objeto-Relacional basado en el proyecto POSTGRES, de la universidad de Berkeley. Es una derivación libre de este proyecto, y utiliza el lenguaje SQL. Fue pionero en muchos de los conceptos del sistema objeto-relacional actual.

Este proyecto lleva más de una década de desarrollo, siendo hoy día, el sistema libre más avanzado con diferencia, soportando la gran mayoría de las transacciones SQL y control concurrente. [11]

PostgreSQL es un gestor multiplataforma, confiable, estable, con gran escalabilidad, control de concurrencia y funcionalidades que lo destacan como uno de los SGBD más potentes en la actualidad.

1.8 Conclusiones

En este capítulo se demuestra la necesidad de realizar un Sistema de Gestión de Información de los Laboratorios (LIMS) que permita facilitar el intercambio, manejo, procesamiento y almacenamiento de la información. Además se presentaron las actividades y artefactos que realizan los trabajadores en el flujo de trabajo de implementación, según la metodología de desarrollo de software definida, así como las distintas herramientas y tecnologías a utilizar para la obtención de una aplicación que cumpla con todos los requerimientos exigidos.

CAPITULO 2: IMPLEMENTACIÓN DEL SISTEMA

2.1 Introducción

En este capítulo se muestra el listado de las funcionalidades que han sido implementadas; se explica la arquitectura de Symfony, que es muy importante porque el código fuente se relaciona con él, es decir al utilizar este framework se programa en capas diferentes de acuerdo con la estructura del mismo. Además se obtienen los artefactos del rol de implementador que propone RUP, específicamente el diagrama de componentes; se muestran también los códigos fuentes más significativos de las principales clases implementadas en el módulo conformado por 9 Casos de Usos del Sistema. Este código, en parte, se realizó haciendo uso del framework Symfony, esta herramienta mediante su funcionamiento generó 116 clases (clases no peer, clases peer, base peer, y base no peer) por lo que hay 29 de cada tipo, también se hace un tratamiento de errores para este código.

2.2 Listado de los Requerimientos

A continuación se muestra el listado de requerimientos que han sido definidos en el módulo y que da lugar a la realización de este trabajo.

CUS Gestionar Registro de Retirada de Productos del Mercado (SIC-0024)

- 1.1. Crear nuevo Registro de Retirada de Productos del Mercado.
- 1.2. Modificar datos del Registro de Retirada de Productos del Mercado.
 - 1.2.1. Registrar traza.
- 1.3. Buscar y visualizar Registro de Retirada de Productos del Mercado.
- 1.4. Generar un reporte del Registro de Retirada de Productos del Mercado.
 - 1.4.1. Imprimir reporte.
- 1.5. Imprimir Registro de Retiradas de Productos del Mercado.

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

- 2.1. Crear nuevo Registro de Control de Cambios.
- 2.2. Registrar datos en el Registro de Control de Cambios.
- 2.3. Modificar datos del Registro de Control de Cambios.
 - 2.3.1. Registrar traza.
- 2.4. Imprimir registro de Control de Cambios.
- 2.5. Generar reporte del Registro de Control de Cambios.
 - 2.5.1. Imprimir reporte.

CUS Gestionar Reporte de Límites (SIC-0963)

- 3.1. Crear nuevo Reporte de Límites.
- 3.2. Registrar datos en el Reporte de Límites.

3.3. Modificar datos en Reporte de Límites.

3.3.1. Registrar traza.

3.4. Buscar y visualizar Reporte de Límites.

3.5. Imprimir Reporte de Límites.

3.6 Generar reporte de Reporte de Límites.

3.6.1. Imprimir reporte.

CUS Gestionar Programa de Monitoreo Ambiental (SIC-0098)

4.1. Crear nuevo Programa de Monitoreo Ambiental.

4.2. Modificar datos del Programa de Monitoreo Ambiental.

4.2.1. Registrar traza.

4.3. Buscar y visualizar Programa de Monitoreo Ambiental.

4.4. Imprimir Programa de Monitoreo Ambiental.

4.5. Generar reporte del Programa de Monitoreo Ambiental

4.5.1. Imprimir reporte.

CUS Gestionar Programa de Informes de Tendencias (SIC-0919)

5.1. Crear nuevo Programa de Informes de Tendencias.

5.2. Registrar datos en el Programa de Informes de Tendencias.

5.3. Modificar datos del Programa de Informes de Tendencias.

5.3.1. Registrar traza.

5.4. Buscar y visualizar Programa de Informes de Tendencias.

5.5. Imprimir Programa de Informes de Tendencias.

5.6. Generar reporte del Programa de Informes de Tendencias.

5.6.1. Imprimir reporte.

CUS Gestionar Registro de Devoluciones (SIC-0023)

6.1. Crear nuevo Registro de Devoluciones.

6.2. Modificar datos del Registro de Devoluciones.

6.2.1. Registrar traza.

6.3. Buscar y visualizar Registro de Devoluciones.

6.4. Imprimir Registro de Devoluciones.

6.5. Generar un reporte.

6.5.1. Imprimir Reporte

CUS Gestionar Notificación de Quejas (SIC-0042)

7.1. Crear el Registro de Quejas y Reclamaciones.

7.2. Modificar datos del Registro de Quejas y Reclamaciones.

7.2.1. Registrar traza.

7.3. Buscar y Visualizar Registro de Quejas y Reclamaciones.

7.4. Imprimir Registro de Quejas y Reclamaciones.

7.5. Generar Reporte del Registro de Quejas y Reclamaciones.

7.5.1. Imprimir reporte.

CUS Gestionar Registro de Reportes de Desviaciones (SIC-0851)

8.1. Crear Registro de Reportes de Desviaciones.

8.2. Modificar datos en el Registro de Reportes de Desviaciones.

9.2.1. Registrar traza.

8.3. Buscar y visualizar Registro de Reportes de Desviaciones.

8.4. Imprimir Registro de Reportes de Desviaciones.

8.5. Generar reporte del Registro de Reportes de Desviaciones.

8.5.1. Imprimir reporte.

CUS Gestionar Cierre de Queja (SIC-0959)

9.1. Crear Cierre de Quejas.

9.2. Modificar datos en el Cierre de Quejas.

9.2.1. Registrar traza.

9.3. Buscar y visualizar Cierre de Quejas.

9.4. Imprimir Cierre de Quejas.

La siguiente tabla muestra una descripción del CUS Gestionar Programa de Monitoreo Ambiental, que ha dado lugar a la implementación de este módulo, para lograr que los programadores conozcan como fluye la información y como se ejecutan cada uno de los procesos que se realizan en este módulo. Las demás descripciones se encuentran en el expediente de proyecto.

Nombre del Caso de Uso	Gestionar Programa de Monitoreo Ambiental
Actores	Analista
Propósito	Crear nuevo programa de monitoreo ambiental, modificar e imprimir el programa, generar reporte del programa e imprimirlo.
Resumen	El caso de uso se inicia cuando el Analista de Mejora va a realizar alguna de las siguiente operaciones relacionadas con la administración del Programa de Monitoreo Ambiental:

	<ul style="list-style-type: none"> • Crear nuevo Programa de Monitoreo Ambiental. • Buscar y visualizar Programa de Monitoreo Ambiental. • Generar reportes del Programa de Monitoreo Ambiental. <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p>
Referencias	R4.1, R4.2, R4.3, R4.3.1, R4.4, R4.5, R4.5.1.
Precondiciones	El analista se haya autenticado en la aplicación.
Poscondiciones	El sistema crea nuevo programa de monitoreo ambiental, modifica e imprime el programa, genera reporte del programa e imprime.
Flujo normal de los eventos	
Acción del actor	Respuesta del Sistema
<p>1. El analista de mejora desea realizar una de las siguientes operaciones:</p> <ul style="list-style-type: none"> • Crear nuevo Programa de Monitoreo Ambiental. • Buscar y visualizar Programa de Monitoreo Ambiental. • Generar reportes del Programa de Monitoreo Ambiental. 	<p>2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> • Si decide crear un nuevo programa, ir a Sección “Crear nuevo Programa de Monitoreo Ambiental”. • Si decide buscar un programa, ir a la Sección “Buscar y visualizar Programa de Monitoreo Ambiental”. • Si decide Generar e Imprimir un reporte del SIC, ir a la Sección “Generar reportes del Programa de Monitoreo Ambiental”.
Sección “Crear nuevo Programa de Monitoreo Ambiental”	

Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz correspondiente a la creación de un nuevo programa.
<p>2. Inserta los datos en el registro:</p> <ul style="list-style-type: none"> • Folio • Producción. • Área. • NP. (Número de Partes) • Local. • Clase. • Zona de Trabajo. • Características. • Método. • Límite Alerta y Límite Acción • Frecuencia. • Instrumentos/PPO. • Punto Muestreo (Referencia) • Responsable. • Revisado por. • Aprobado por. • Terminado. 	<p>3. Verifica que todos los datos necesarios han sido introducidos.</p> <ul style="list-style-type: none"> • Folio. • Producción. • Área. • NP. • Local. • Clase. • Zona de Trabajo. <p>En tiempo de ejecución el sistema busca y genera los campos:</p> <ul style="list-style-type: none"> • Realizado por: • Fecha. Este campo carga la fecha del sistema.
	4. Si el campo Terminado está seleccionado, el sistema actualiza el campo "Estado" estableciéndole el valor de "Terminado".
5. El analista solicita crear el programa.	6. El sistema crea el programa.
	7. El sistema muestra el programa creado y brinda la posibilidad de imprimirlo y/o modificarlo.
8. El analista escoge una de las siguientes opciones:	9. En dependencia de lo que escoge el analista el sistema realiza lo siguiente:

<ul style="list-style-type: none"> • Imprimir programa. • Modificar programa. 	<ul style="list-style-type: none"> • Si decide imprimir, el sistema imprime el programa. • Si decide modificar va a la Sección “Modificar datos en el Programa de Monitoreo Ambiental ”
Flujo Alternativo Sección “Crear nuevo Programa de Monitoreo Ambiental”	
Acción del actor	Respuesta del Sistema
2.1. Si el analista no introduce los datos, sale de la sección.	3.1. Si alguno de los campos no es introducido, el sistema muestra el mensaje “No se puede crear el Programa”.
	4.1. Si el campo Terminado no está seleccionado, el sistema actualiza el campo “Estado” estableciéndole el valor de “No Terminado”.
5.1. Si el analista no selecciona crear el programa, sale de la sección.	
8.1. Si el analista no desea hacer ninguna de las opciones brindadas, sale de la sección.	
Sección “Modificar datos en el Programa de Monitoreo Ambiental”	
Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz correspondiente a la modificación de los datos del programa.
2. El analista realiza los cambios necesarios y/o actualizaciones pertinentes. <ul style="list-style-type: none"> • Terminado. 	3. El sistema verifica que los campos necesarios tengan valor.
	4. Si el campo Terminado está seleccionado, el sistema actualiza el campo “Estado” estableciéndole el valor de “Terminado”.

	5. El sistema actualiza el programa en la BD.
	6. Se actualiza el registro de trazas.
	7. El sistema muestra el programa modificado y brinda la posibilidad de imprimirlo y/o modificarlo..
8. El analista escoge una de las siguientes opciones: <ul style="list-style-type: none"> • Imprimir programa. • Modificar programa. 	9. En dependencia de lo que escoge el analista el sistema realiza lo siguiente: <ul style="list-style-type: none"> • Si decide imprimir, el sistema imprime el programa. • Si decide modificar va a la acción 1.
Flujo Alternativo Sección “Modificar datos en el Programa de Monitoreo Ambiental”	
Acción del actor	Respuesta del Sistema
2.1. Si el analista no desea modificar los datos, sale de la sección.	3.1 El sistema muestra un mensaje de error, para que sean llenados los datos correctamente.
	4.1. Si el campo Terminado no está seleccionado, el sistema actualiza el campo “Estado” estableciéndole el valor de “No Terminado”.
8.1. Si el analista no desea realizar ninguna de las opciones brindadas, sale de la sección.	
Sección “Buscar y visualizar Programa de Monitoreo Ambiental”	
Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz correspondiente a la búsqueda de programas.
2. El analista de mejora introduce los datos que le permitirán la búsqueda del programa.	3. Verifica existencia del programa asociado a los criterios de búsqueda introducidos.

<ul style="list-style-type: none"> • Folio. • NP.(Número de Parte) • Área. • Zona de Trabajo. • Clase. • Planta. 	
	4. El sistema realiza la búsqueda.
	<p>5. El sistema muestra el resultado en forma de listado en una tabla que muestra los siguientes campo:</p> <ul style="list-style-type: none"> • Zona de Trabajo. • NP. • Método. <p>Brinda la posibilidad de visualizar el programa encontrado.</p>
6. El analista selecciona el reporte que desea visualizar.	7. El sistema muestra el reporte seleccionado en una nueva interfaz y brinda la posibilidad de imprimir y/o modificar.
<p>8. El analista escoge una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Imprimir programa. • Modificar programa. 	<p>9. En dependencia de lo que escoge el analista el sistema realiza lo siguiente:</p> <ul style="list-style-type: none"> • Si decide imprimir, el sistema imprime el programa. • Si decide modificar va a la Sección “Modificar datos en el Programa de Monitoreo Ambiental ”
Flujo Alternativo Sección “Buscar y visualizar Programa de Monitoreo Ambiental”	
Acción del actor	Respuesta del Sistema
2.1 Si el analista introduce incorrectamente alguno de los criterios de búsqueda.	3.1. Si no existe el programa asociado al o los criterios de búsqueda, el sistema muestra el siguiente mensaje

	“No se ha encontrado programa asociado al criterio de búsqueda introducido”.
6.1. Si el analista no desea visualizar ninguno de los programas mostrados, sale de la sección.	
8.1. Si el analista no desea realizar ninguna de las opciones que se brindan, sale de la sección.	
Sección “Generar reporte del Programa de Monitoreo Ambiental”	
Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz correspondiente a la generación de reportes.
<p>2. El analista selecciona o inserta los datos para generar el reporte.</p> <ul style="list-style-type: none"> • Producción. • Área. • Clase. <p>El analista indica los parámetros que desean que se visualicen en el reporte a obtener.</p> <ul style="list-style-type: none"> • Todos. • Zona de Trabajo. • Área. • Clase. • NP. • Método. • Características. • Responsable. • Frecuencia. • Instrumento/PPO. 	3. El sistema verifica que han sido introducidos al menos uno de los datos necesarios para generar el reporte del programa.
	4. El sistema genera el reporte según

	los criterios indicados y brinda la posibilidad de imprimirlo.
4. El analista confirma la impresión del reporte obtenido.	5. Imprime el reporte.
	6. El sistema posibilita la generación de un nuevo reporte.
Flujo Alternativo Sección “Generar reporte del Programa de Monitoreo Ambiental”	
Acción del actor	Respuesta del Sistema
2.1. Si el analista no desea insertar al menos un criterio de búsqueda, sale de la sección.	3.1. Si falta al menos un dato para generar el reporte el sistema muestra el siguiente mensaje “Necesita introducir al menos un dato para generar el reporte”.
4.1. Si el analista no desea imprimir el reporte, sale de la sección.	

2.3 Arquitectura utilizada

Para el desarrollo de la aplicación se decidió utilizar el Framework Symfony. Este introduce el patrón arquitectónico Modelo-Vista-Controlador (MVC), que está formado por 3 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. (Ver Anexo 6)

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como en un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email). 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. [10]

Capa Control:

Las acciones son el corazón de la aplicación, puesto que contienen toda la lógica de la aplicación. Estas utilizan el modelo y definen variables para la vista. La clase que representa las acciones de un módulo se encuentra en el archivo actions.class.php, en el directorio actions/ del módulo.

A continuación se muestra el código fuente de la función Crear en la clase action.class del módulo SIC0023 en la cual se entran los valores y se guarda en la base de datos. (Ver Anexo 7)

```

public function executeCrearSIC0023()
{
    if ($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $this->mensaje='';
        return sfView::SUCCESS;
    }
else
    {
        $folio=$this->getRequestParameter('crear_fol');
        if(RegistrodevolucionesPeer::ExisteFolio($folio))
        {
            $this->mensaje='Ese Folio Ya Existe';
            return sfView::SUCCESS;
        }
        else
        {
            $nom_prod=$this->getRequestParameter('nom_prod');
            $f_not=$this->getRequestParameter('f_not');
            $cant_prod=$this->getRequestParameter('cant_prod');
            $cod_dev=$this->getRequestParameter('cod_dev');
            $cons_dosis=$this->getRequestParameter('cons_dosis');
            $concen_dosis=$this->getRequestParameter('concen_dosis');
            $n_lote=$this->getRequestParameter('n_lote');
            $form_farm=$this->getRequestParameter('form_farm');
            $desc_prob=$this->getRequestParameter('desc_prob');
            $cr_ident=$this->getRequestParameter('cr_ident');
            $decis=$this->getRequestParameter('decis');
            $real_por=$this->getRequestParameter('real_por');
            $cargo_real=$this->getRequestParameter('cargo_real');
            $f_reali=$this->getRequestParameter('f_reali');
            $revi_por=$this->getRequestParameter('revi_por');
            $cargo_revi=$this->getRequestParameter('cargo_revi');
            $f_revi=$this->getRequestParameter('f_revi');
            $terminado=$this->getRequestParameter('terminado');
        }
    }
}

```

```

$id=ControlplanillaPeer::ControlPlanilla('', '', $real_por, $f_reali, '', $revi_por, $f_revi);
RegistrodevolucionesPeer::CrearSIC0023 ($folio, $f_not, $nom_prod, $cant_prod, $cod_dev,
$concen_dosis, $cons_dosis, $n_lote, $form_farm, $desc_prob, $cr_ident, $decis, $cargos_revi,
$cargos_real, $terminado, $id, 0);

$lote=$this->getRequestParameter('slote');
$arr_lote= explode('*', $lote);
$cant=$this->getRequestParameter('scant');
$arr_cant= explode('*', $cant);
for($i=0;$i<count($arr_lote)-1;$i++)
{
    LotesasociadosPeer::CrearLotesAsoc($folio,$arr_lote[$i],$arr_cant[$i]);
}
$doc=$this->getRequestParameter('sdoc');
$arr_doc= explode('*', $doc);
for($i=0;$i<count($arr_doc)-1;$i++)
{
    DocumentosasociadosPeer::CrearDocAsoc($folio,$arr_doc[$i]);
}
$nombre=$this->getRequestParameter('snomb');
$arr_nombre= explode('*', $nombre);
$carg=$this->getRequestParameter('scarg');
$arr_carg= explode('*', $carg);
for($i=0;$i<count($arr_nombre)-1;$i++)
{
    ParticipantescomisonPeer::CrearParticipanTCom($folio,$arr_nombre[$i],$arr_carg[$i],null);
}
$this->redirect('SIC0023/VisualizarSIC0023?enviar='.$folio.'&enviar1='.$id);
}
}
}

```

En el execute CrearSIC0023 primeramente se pregunta si le pasas valores por el método POST a través de la condición "if (\$this->getRequest()->getMethod() != sfRequest::POST)". De no suceder esto se visualiza la página CrearSIC0023Success.php sin ningún problema, pero de suceder se toman los valores entrados necesarios para insertar en las tablas "Controlplanilla" y "Registrodevoluciones", que se hace a través de las funciones "ControlPlanilla()" y "CrearSIC0023()" respectivamente, en estas se devuelven las llaves primarias de cada tabla. Seguidamente se capturan los valores de los campos ocultos, donde cada campo va a tener muchos valores que van a estar dividido por un (*), por lo que la función "explode('*', \$valor)" divide estos valores y los guarda en un arreglo, cada uno en una posición distinta del arreglo. Estos se insertan en las tablas "Lotesasociados", "Documentosasociados", "Participantescomison" con las llaves primarias anteriormente mencionadas como llave foránea utilizando las funciones "CrearLotesAsoc", "CrearDocAsoc" y "CrearParticipanTCom" respectivamente y finalizando se envían las llaves primarias a la página VisualizarSIC0023Success.php para ser visualizada toda la información referente a estas llaves guardadas en las tablas de la Base de Datos.

En el siguiente código se muestra la función Visualizar en la clase action.class del módulo SIC0023, en la cual pasándole la llave primaria busca todos los valores que se desean mostrar.

```
public function executeVisualizarSIC0023()
{
    $folio = $this->getRequestParameter('enviar');
    $id = $this->getRequestParameter('enviar1');

    $arr_par=ParticipantescomisonPeer::BuscarPorFolio($folio);
    $arr_cont_plan=ControlplanillaPeer::BuscarPlanilla($id);
    $arr_reg=RegistrodevolucionesPeer::BuscarPorFolio($folio);
    $arr_lote=LotesasociadosPeer::BuscarPorFolio($folio);
    $arr_doc=DocumentosasociadosPeer::BuscarPorFolio($folio);

    $par_arr=ParticipantescomisonPeer::ConvArrayObj($arr_par);
    $lote_arr=LotesasociadosPeer::ConArrayObj($arr_lote);
    $reg_arr=RegistrodevolucionesPeer::ConArrayObj($arr_reg);
    $doc_arr=DocumentosasociadosPeer::ConArrayObj($arr_doc);

    $cont_pani_arr=ControlplanillaPeer::ConvArrayObj($arr_cont_plan);

    $this->ver1=$cont_pani_arr;
    $this->ver=$reg_arr;
    $this->ver2=$doc_arr;
    $this->ver3=$par_arr;
    $this->ver4=$lote_arr;
}
```

En el executeVisualizarSIC0023 se reciben las dos llaves primarias y se buscan todos los datos de las tablas “Participantescomison”, “Controlplanilla”, “Registrodevoluciones”, “Lotesasociados” y “Documentosasociados” con las funciones “BuscarPorFolio(\$folio)” y “BuscarPlanilla(\$id)” que se encuentran implementadas en las clases PEER que genera el propel de cada tabla de la Base de Datos. Estos datos se guardan en arreglos de objetos, luego se llama a la función “ConvArrayObj(\$arr_obj)” para transformarlos en arreglos de arreglos y mostrarlos en la página VisualizarSIC0023Success.php.

Aquí se muestra el código fuente de la función Buscar en la clase action.class del módulo SIC0023, en la cual entrándole tres patrones de búsqueda devuelve los SIC que presenten dichos patrones.

```

public function executeBuscarSIC0023()
{
    if ($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $this->ver=1;
        $ver_fol=array('Select...');
        $ver_cod=array('Select...');
        $ver_def=array('Select...');

        $lista_fol=RegistrodevolucionesPeer::BuscarFolios();
        $lista_cod=RegistrodevolucionesPeer::BuscarCodigos();

        for($i=0;$i<count($lista_fol);$i++)
            array_push($ver_fol,$lista_fol[$i]);
        for($i=0;$i<count($lista_cod);$i++)
            array_push($ver_cod,$lista_cod[$i]);

        $this->list_folios=$ver_fol;
        $this->list_cod=$ver_cod;
        $this->mensaje='';
    }
    else
    {
        $ver=array();
        $ver_fol=array('Select...');
        $ver_cod=array('Select...');

        $lista_fol=RegistrodevolucionesPeer::BuscarFolios();
        $lista_cod=RegistrodevolucionesPeer::BuscarCodigos();

        for($i=0;$i<count($lista_fol);$i++)
            array_push($ver_fol,$lista_fol[$i]);
        for($i=0;$i<count($lista_cod);$i++)
            array_push($ver_cod,$lista_cod[$i]);

        $this->list_folios=$ver_fol;
        $this->list_cod=$ver_cod;

        $folio =$this->getRequestParameter('select_fol');
        $codigo=$this->getRequestParameter('select_cod');
        $anno=$this->getRequestParameter('select_anno');
        if($folio==0 && $codigo==0 && $anno==0)
        {

```

```

    {
        $this->ver=2;
        $this->mensaje='Introducir al menos un par&acute;metro';
    }
    else
    {
        $this->mensaje='';
        $c=new criteria;
        if($folio>0)
        {
            $c->add(RegistrodevolucionesPeer::FOLIO,$ver_fol[$folio]);
        }
        if($codigo>0)
        {
            $c->add(RegistrodevolucionesPeer::CODIGODEVOLUCION,$ver_cod[$codigo]);
        }

        $arreglo=RegistrodevolucionesPeer::doSelect($c);

        if($anno>0)
        {
            $lista_anno =array('Select...','1980','1981','1982','1983','1984','1985',
            '1986','1987','1988','1989','1990','1991','1992','1993','1994','1995','1996',
            '1997','1998','1999','2000','2001','2002','2003','2004','2005','2006','2007',
            '2008');
            $anno_f=$lista_anno[$anno];
            for($i=0;$i<count($arreglo);$i++)
            {
                $anno= substr($arreglo[$i]->getFechanotificacion(), 0, 4);
                if($anno==$anno_f)
                array_push($ver,$arreglo[$i]);
            }
        }
        else
        $ver=$arreglo;

        $this->ver=RegistrodevolucionesPeer::ConArrayObjl($ver);
    }
}
}

```

En el execute BuscarSIC0023 primeramente se pregunta si le llegan valores por el método POST a través de la condición “`if ($this->getRequest()->getMethod() != sfRequest::POST)`”. De ser esto negativo se buscan todos los folios y códigos que hay en la base de datos y se guardan en los arreglos `$lista_fo` y `$lista_cod` respectivamente, para ser seleccionado luego en el formulario de la página BuscarSIC0023. Si se confirma que le entran valores por el método POST, se toman estos a través de la función `getRequestParameter` y se verifica si al menos uno de ellos está seleccionado correctamente, si esto se cumple se crea un objeto de tipo `Criteria()` y se pregunta cual fue el dato entrado, adicionándole a dicho objeto las consultas de búsqueda de estos datos, y guardando el resultado en el arreglo de objeto `$arreglo`. Seguidamente se le pasa este arreglo a la función `ConArrayObj1($ver)` que se encuentra en la clase `RegistrodevolucionesPeer`, para convertirlo en un arreglo de arreglos y se envía este ultimo a la clase `BuscarSIC0023Success.php` para ser mostrados los valores que lo conforman. Si no se cumple entonces para un mejor funcionamiento de esta aplicación, se realizó un tratamiento de errores, que es muy importante ya que esto permite rectificar los errores que se cometen a la hora de entrar los datos, posibilitando que esto trabaje correctamente.

En esta capa nada más se muestra estos códigos explicados, más adelante en el epígrafe de Código Fuente están todos los demás que no se explican ya que todos tienen la misma forma de implementación.

Capa Modelo:

Las bases de datos son relacionales. PHP 5 y Symfony están orientados a objetos. Para acceder de forma efectiva a la base de datos desde un contexto orientado a objetos, es necesaria una interfaz que traduzca la lógica de los objetos a la lógica relacional. Dicha interfaz se llama Object Relational Mapping (ORM), la misma permite llamar a los métodos de un objeto de datos desde varias partes de la aplicación e incluso desde diferentes aplicaciones, además de permitir añadir métodos accesorios en los objetos que no tienen relación directa con una tabla.

Para crear el modelo de objetos de datos que utiliza Symfony, se debe traducir el modelo relacional de la base de datos a un modelo de objetos de datos. Para realizar esa traducción, el ORM necesita una descripción del modelo relacional, que se llama “esquema”. En el esquema se definen las tablas, sus relaciones y las características de sus columnas. El esquema se utiliza para construir las clases del modelo que necesita la capa del ORM. Para reducir el tiempo de ejecución de la aplicación, estas clases se generan mediante una tarea de línea de comandos llamada `propel-build-model`. Al ejecutar ese comando (`symfony propel-build-model`), se analiza el esquema y se generan las clases base del modelo, dígame clases base peer y base no peer, que se almacenan en el directorio `lib/model/om/` del

proyecto. Además, se crean las verdaderas clases del modelo de datos, clases peer y no peer en el directorio lib/model/.

Las 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. Sus métodos devuelven normalmente un objeto o una colección de objetos de la clase objeto relacionado.

A continuación se muestra el código de una clase peer, aquí se encuentran las funciones principales programadas.

```
class Sic0098Peer extends BaseSic0098Peer
{
    public static function CrearSIC0098($folio,$prod,$area,$np,$local,$clase,$ztrabajo,
    $real_pon,$rezi_por,$apro_por,$fec_real,$fec_apro,$fec_revi,$term)
    {

        $sic0098 = new Sic0098();
        $sic0098->setIdfolio($folio);
        $sic0098->setProduccion($prod);
        $sic0098->setArea($area);
        $sic0098->setNp($np);
        $sic0098->setLocal($local);
        $sic0098->setClase($clase);
        $sic0098->setZonaTrabajo($ztrabajo);
        $sic0098->setAprobadoPor($apro_por);
        $sic0098->setRealizadoPor($real_pon);
        $sic0098->setRevisadoPor($rezi_por);
        $sic0098->setFechaRealizacion($fec_real);
        $sic0098->setFechaAprobacion($fec_apro);
        $sic0098->setFechaRevision($fec_revi);

        if($term==1)
        $sic0098->setTerminado('1');
        else
        $sic0098->setTerminado('0');

        $sic0098->save();
    }
    public static function BuscarFolios()//Completo
    public static function BuscarPlantas()//Completo
```

Principales funciones que brinda el Symfony en las clases basePeer utilizadas para el desarrollo de esta aplicación (BaseSIC0098Peer.php):

```
public static function doSelect(Criteria $criteria, $con = null)
{
return Sic0098Peer::populateObjects(Sic0098Peer::doSelectRS($criteria, $con));
}
```

```
public static function retrieveByPK($pk, $con = null)
{
    if ($con === null) {
        $con = Propel::getConnection(self::DATABASE_NAME);
    }
    $criteria = new Criteria(Sic0098Peer::DATABASE_NAME);
    $criteria->add(Sic0098Peer::IDFOLIO, $pk);
    $v = Sic0098Peer::doSelect($criteria, $con);
    return !empty($v) > 0 ? $v[0] : null;
}
```

Principales funciones que brinda el Symfony en las clases base utilizadas para el desarrollo de esta aplicación (BaseSIC0098.php):

```
public function getIdfolio()
{
    return $this->idfolio;
}
public function getProduccion()
{
    return $this->produccion;
}
public function getArea()
{
    return $this->area;
}
public function getNp()
{
    return $this->np;
}
public function getLocal()
{
    return $this->local;
}
public function getClase()
{
    return $this->clase;
}
public function getZonatrabajo()
{
    return $this->zonatrabajo;
}
public function getTerminado()
{
    return $this->terminado;
}
public function getRealizadopor()
```

```

    {
        return $this->realizadopor;
    }
    public function getRevisadopor()
    {
        return $this->revisadopor;
    }
    public function getAprobadopor()
    {
        return $this->aprobadopor;
    }
    public function getFecharealizacion()
    {
        return $this->fecharealizacion;
    }
    public function getFecharevision()
    {
        return $this->fecharevision;
    }
    public function getFechaaprobacion()
    {
        return $this->fechaaprobacion;
    }
    public function setIdfolio($v)
    {
        if ($v !== null && !is_string($v)) {
            $v = (string) $v;
        }
        if ($this->idfolio !== $v) {
            $this->idfolio = $v;
            $this->modifiedColumns[] = Sic0098Peer::IDFOLIO;
        }
    }
    public function setProduccion($v)
    {
        if ($v !== null && !is_string($v)) {
            $v = (string) $v;
        }
        if ($this->produccion !== $v) {
            $this->produccion = $v;
            $this->modifiedColumns[] = Sic0098Peer::PRODUCCION;
        }
    }
    public function setArea($v)
    {
        if ($v !== null && !is_string($v)) {
            $v = (string) $v;
        }
        if ($this->area !== $v) {
            $this->area = $v;
            $this->modifiedColumns[] = Sic0098Peer::AREA;
        }
    }
    public function setNp($v)
    {

```

```

        if ($v !== null && !is_string($v)) {
            $v = (string) $v;
        }
        if ($this->np !== $v) {
            $this->np = $v;
            $this->modifiedColumns[] = Sic0098Peer::NP;
        }
    }
    public function setLocal($v)
    {
        if ($v !== null && !is_string($v)) {
            $v = (string) $v;
        }
        if ($this->local !== $v) {
            $this->local = $v;
            $this->modifiedColumns[] = Sic0098Peer::LOCAL;
        }
    }
    public function setClase($v)
    {
        if ($v !== null && !is_string($v)) {
            $v = (string) $v;
        }
        if ($this->clase !== $v) {
            $this->clase = $v;
            $this->modifiedColumns[] = Sic0098Peer::CLASE;
        }
    }
    public function setZonatrabajo($v)
    {
        if ($v !== null && !is_string($v)) {
            $v = (string) $v;
        }
        if ($this->zonatrabajo !== $v) {
            $this->zonatrabajo = $v;
            $this->modifiedColumns[] = Sic0098Peer::ZONATRABAJO;
        }
    }
    public function setTerminado($v)
    {
        if ($this->terminado !== $v) {
            $this->terminado = $v;
            $this->modifiedColumns[] = Sic0098Peer::TERMINADO;
        }
    }
    public function setRealizadopor($v)
    {
        if ($v !== null && !is_string($v)) {
            $v = (string) $v;
        }
        if ($this->realizadopor !== $v) {
            $this->realizadopor = $v;
            $this->modifiedColumns[] = Sic0098Peer::REALIZADOPOR;
        }
    }

```

```

}
public function setRevisadopor($v)
{
    if ($v !== null && !is_string($v)) {
        $v = (string) $v;
    }

    if ($this->revisadopor !== $v) {
        $this->revisadopor = $v;
        $this->modifiedColumns[] = Sic0098Peer::REVISADOPOR;
    }
}
public function setAprobadopor($v)
{
    if ($v !== null && !is_string($v)) {
        $v = (string) $v;
    }

    if ($this->aprobadopor !== $v) {
        $this->aprobadopor = $v;
        $this->modifiedColumns[] = Sic0098Peer::APROBADOPOR;
    }
}
public function setFecharealizacion($v)
{
    if ($v !== null && !is_string($v)) {
        $v = (string) $v;
    }

    if ($this->fecharealizacion !== $v) {
        $this->fecharealizacion = $v;
        $this->modifiedColumns[] = Sic0098Peer::FECHAREALIZACION;
    }
}
public function setFecharevision($v)
{
    if ($v !== null && !is_string($v)) {
        $v = (string) $v;
    }

    if ($this->fecharevision !== $v) {
        $this->fecharevision = $v;
        $this->modifiedColumns[] = Sic0098Peer::FECHAREVISION;
    }
}
public function setFechaaprobacion($v)
{
    if ($v !== null && !is_string($v)) {
        $v = (string) $v;
    }

    if ($this->fechaaprobacion !== $v) {
        $this->fechaaprobacion = $v;
        $this->modifiedColumns[] = Sic0098Peer::FECHAAPROBACION;
    }
}
}

```

```
public function save($con = null)
{
    if ($this->isDeleted()) {
        throw new PropelException("You cannot save an object that has been deleted.");
    }

    if ($con === null) {
        $con = Propel::getConnection(Sic0098Peer::DATABASE_NAME);
    }

    try {
        $con->begin();
        $affectedRows = $this->doSave($con);
        $con->commit();
        return $affectedRows;
    } catch (PropelException $e) {
        $con->rollback();
        throw $e;
    }
}
```

Capa Vista:

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

Normalmente se trabaja con las plantillas (que son la presentación de los datos de la acciones que se está ejecutando) y con el layout, que contiene el código HTML común a todas las páginas y pequeñas representaciones de código PHP, son llamadas a los diversos helpers disponibles.

Para mejorar la reutilización de código, se extraen trozos de las plantillas y se transforman en componentes y elementos parciales. De esta forma, el layout se modifica para definir zonas en las que se insertan componentes externos.

2.4 Diagrama de Componentes

Para lograr una implementación lo más efectiva posible se debe contar con un buen diseño, lo cual responde en gran medida al buen funcionamiento de la aplicación.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

Dado que los diagramas de componentes muestran los componentes software que constituyen una parte reusable, sus interfaces y sus interrelaciones, en muchos aspectos se puede considerar que un diagrama de componentes es un diagrama de clases a gran escala. Cada componente en el diagrama debe ser documentado con un diagrama de componentes más detallado, un diagrama de clases, o un diagrama de casos de uso. [12]

El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando:

- Los subsistemas de implementación y sus dependencias a la hora de importar código.
- Organizar los subsistemas de implementación en capas.

Lo que distingue a un diagrama de componentes de otros tipos de diagramas es su contenido.

Normalmente contienen componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.

Seguidamente se muestran los diagramas de componentes desarrollados, cada diagrama está compuesto por un paquete (View), en el cual se encuentran todas páginas success.php con las cuáles interactúa el usuario, el layout.php es la parte global en toda la aplicación, es decir, todo el código Hypertext Markup Language (HTML) común para todas las páginas. También está el componente Java Script dentro del cual se localiza todo el código java script utilizado. Otro componente utilizado es el index que es el controlador, este se relaciona con las páginas Success.php y con la actions.class.php, este último mencionado es otro componente donde se encuentra toda la implementación de la capa control, de la cual esta forma parte y esta relacionado con las páginas Success. Además se tiene otro componente, el subsistema Model, que en él se encuentran todas las clases objetos de la capa modelo, aquí Symfony generó 116 clases (clases no peer, clases peer, base peer, y base no peer) por lo que hay 29 de cada tipo. Otro subsistema representado es del componente Symfony, dentro del cual se hayan todos los archivos que contienen las funcionalidades que brinda el framework, a través de este se realiza la conexión a la base de datos; también se relaciona con los componentes YML, siendo estos los encargados de la seguridad y validación de la aplicación, otra relación que éste posee es con el layout.php. Por último se encuentra el componente Base de Datos, que no es más que la base de datos del módulo. Todas las relaciones son de dependencia, demostrando esto la estrecha relación existente entre cada uno de los componentes representados.

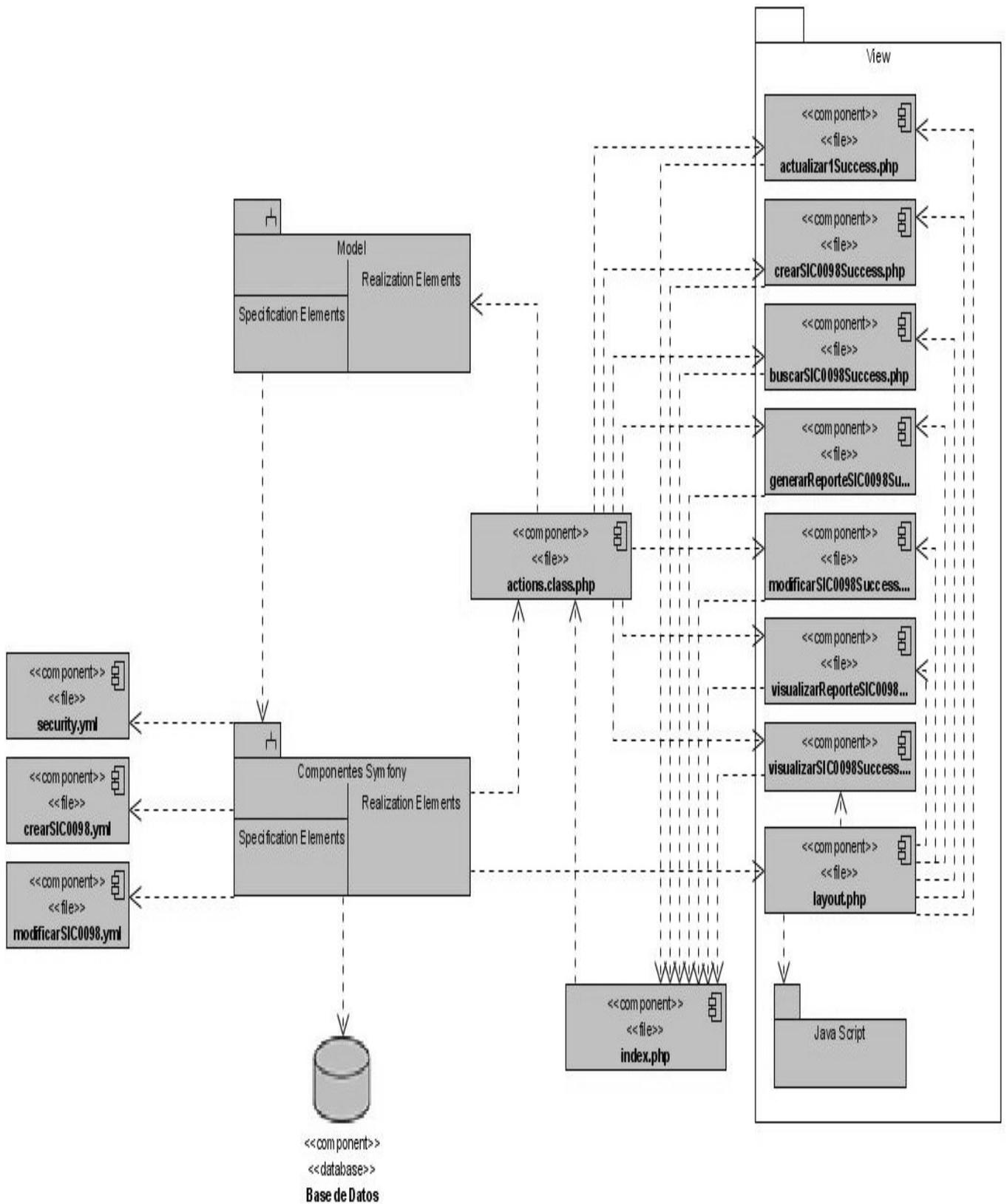


Figura 1 Diagrama de componentes del CU Gestionar SIC0098

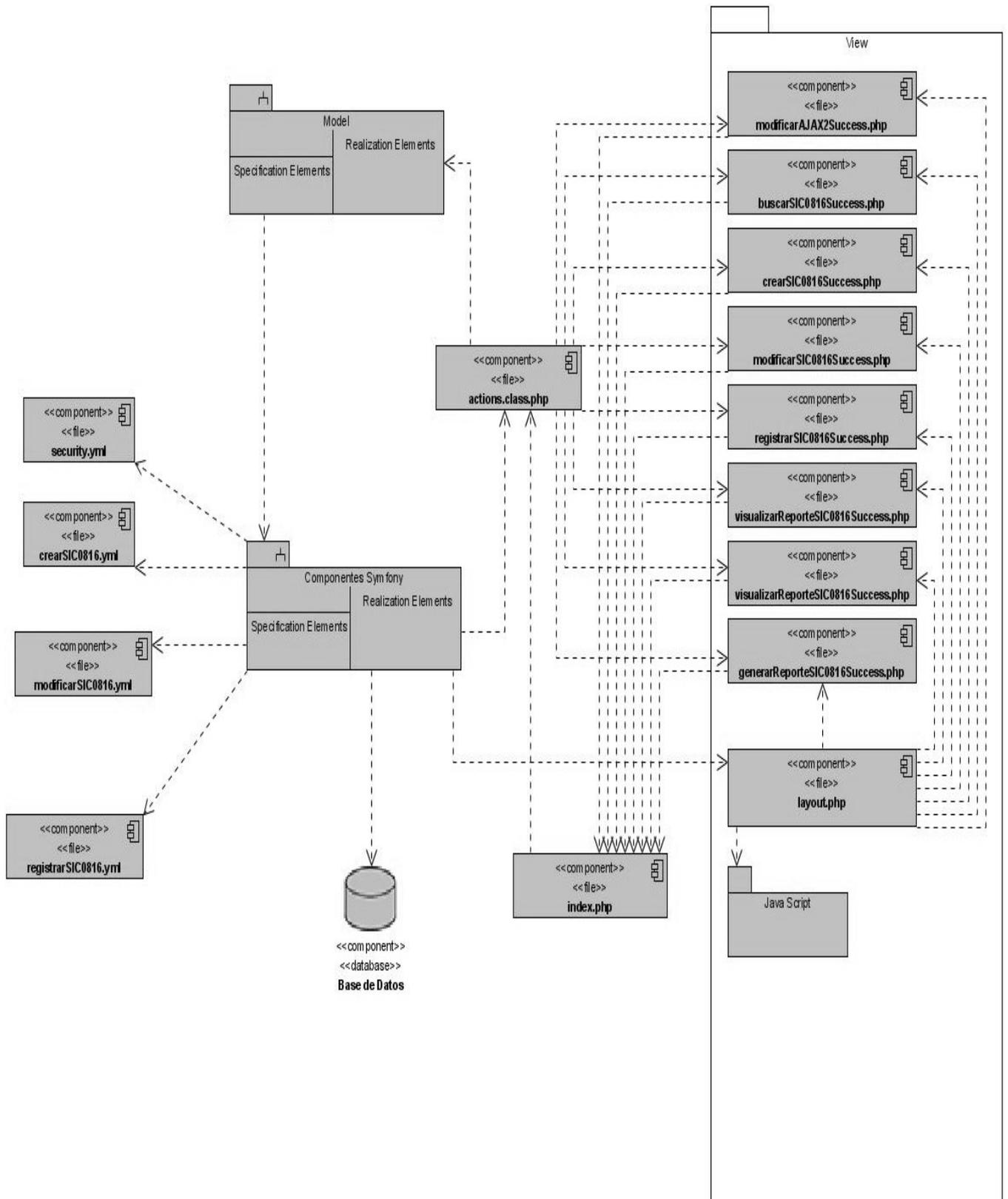


Figura 2 Diagrama de componentes del CU Gestionar SIC0816

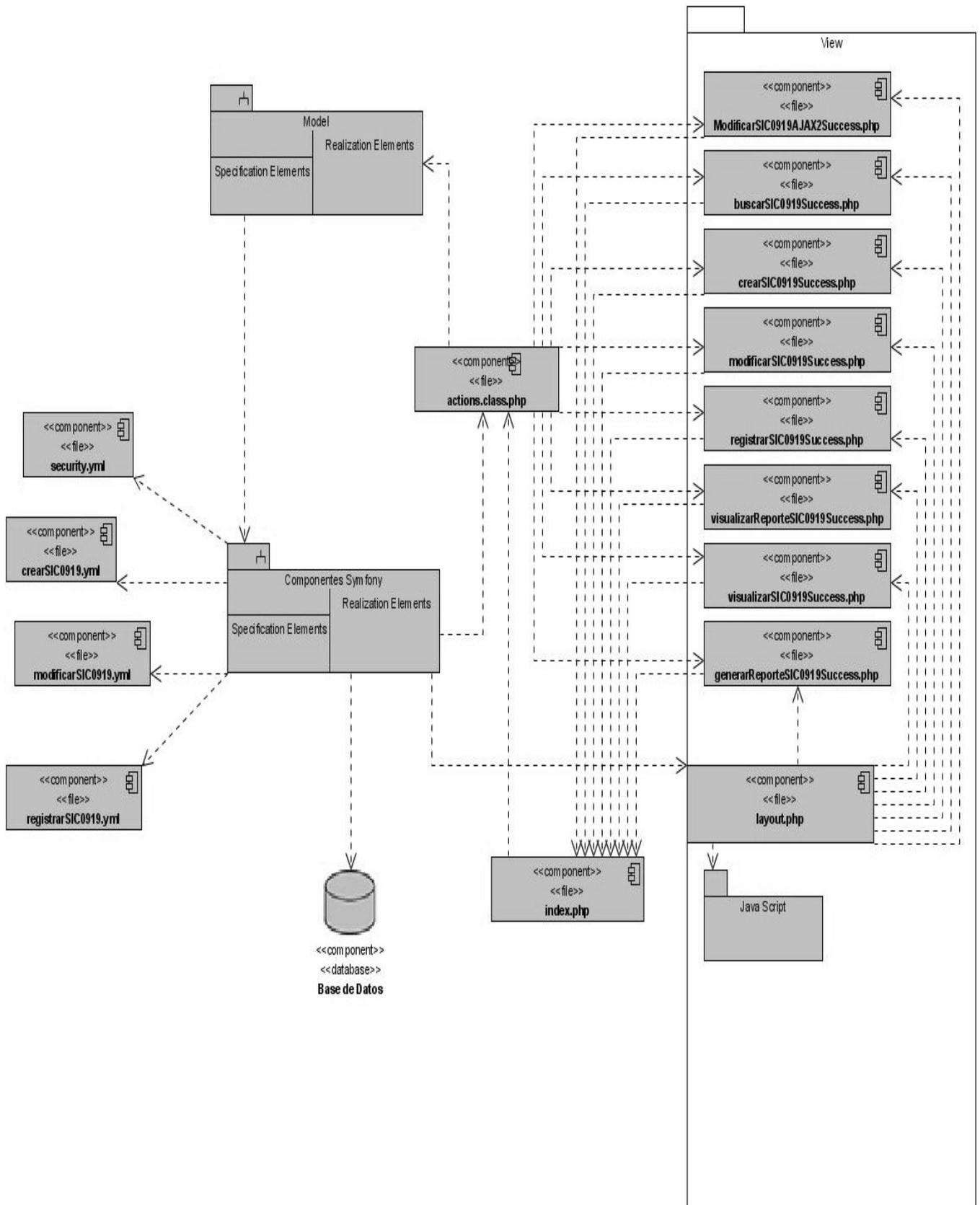


Figura 3 Diagrama de componentes del CU Gestionar SIC0919

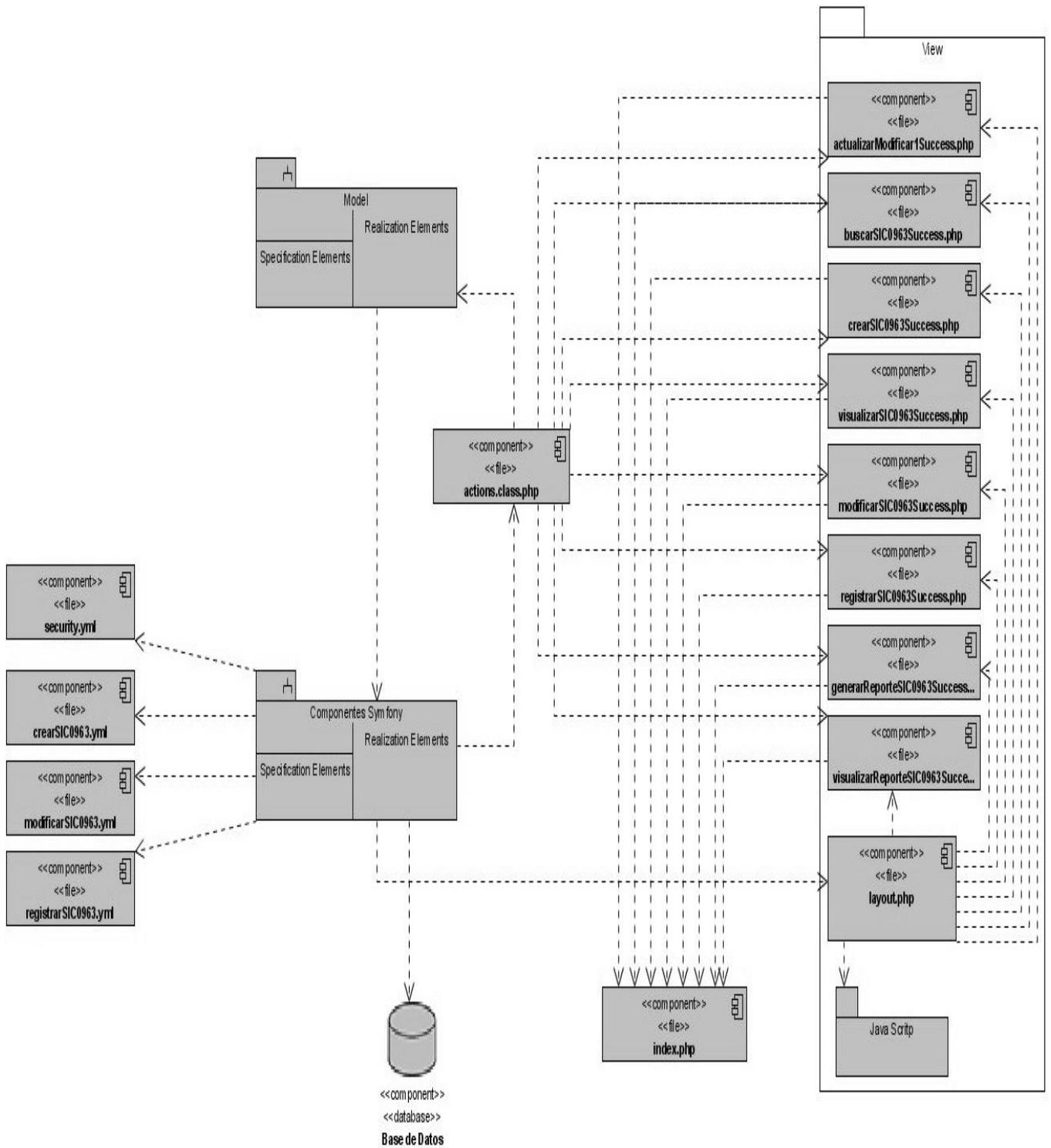


Figura 4 Diagrama de componentes del CU Gestionar SIC0963

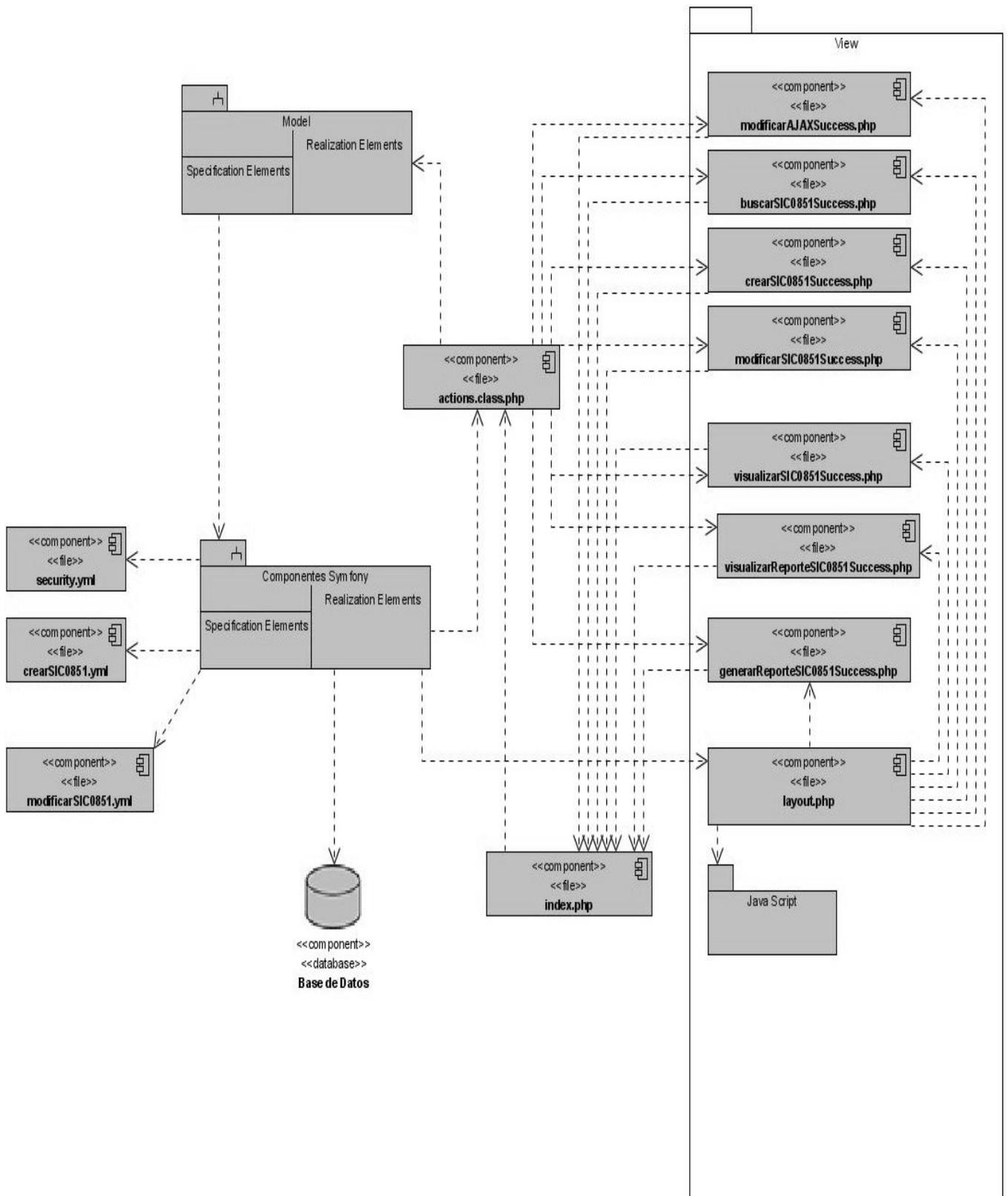


Figura 5 Diagrama de componentes del CU Gestionar SIC0851

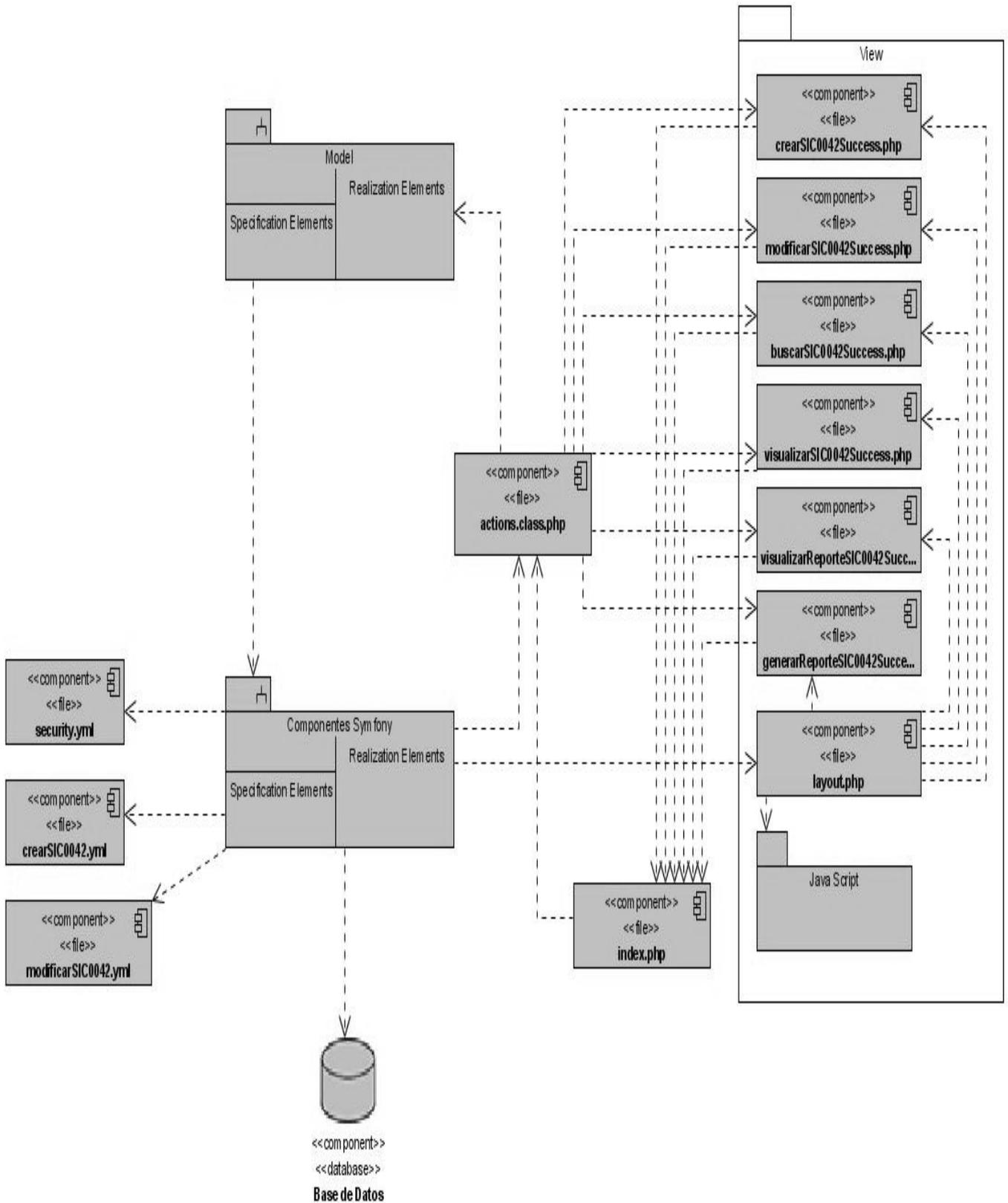


Figura 6 Diagrama de componentes del CU Gestionar SIC0042

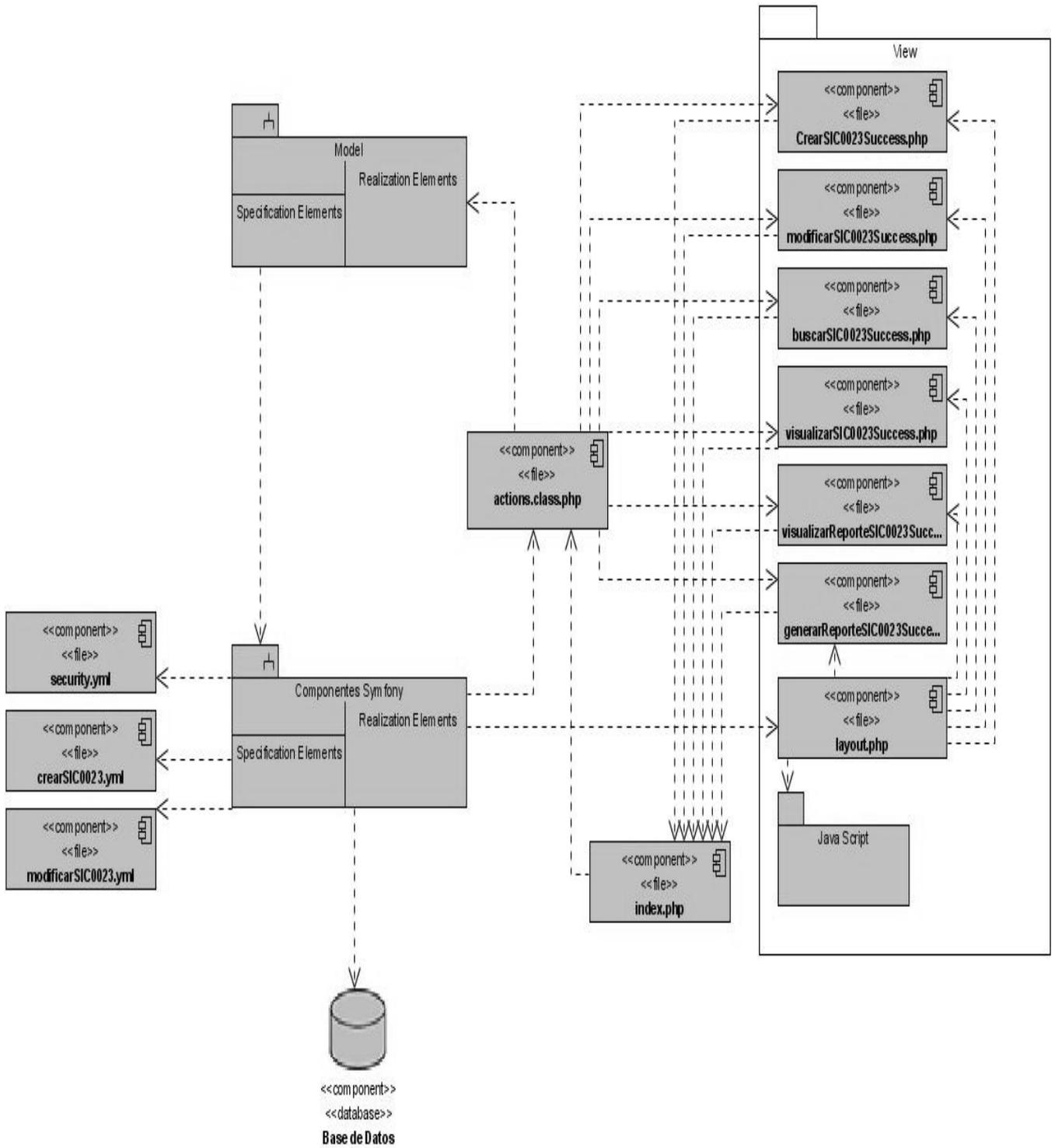


Figura 7 Diagrama de componentes del CU Gestionar SIC0023

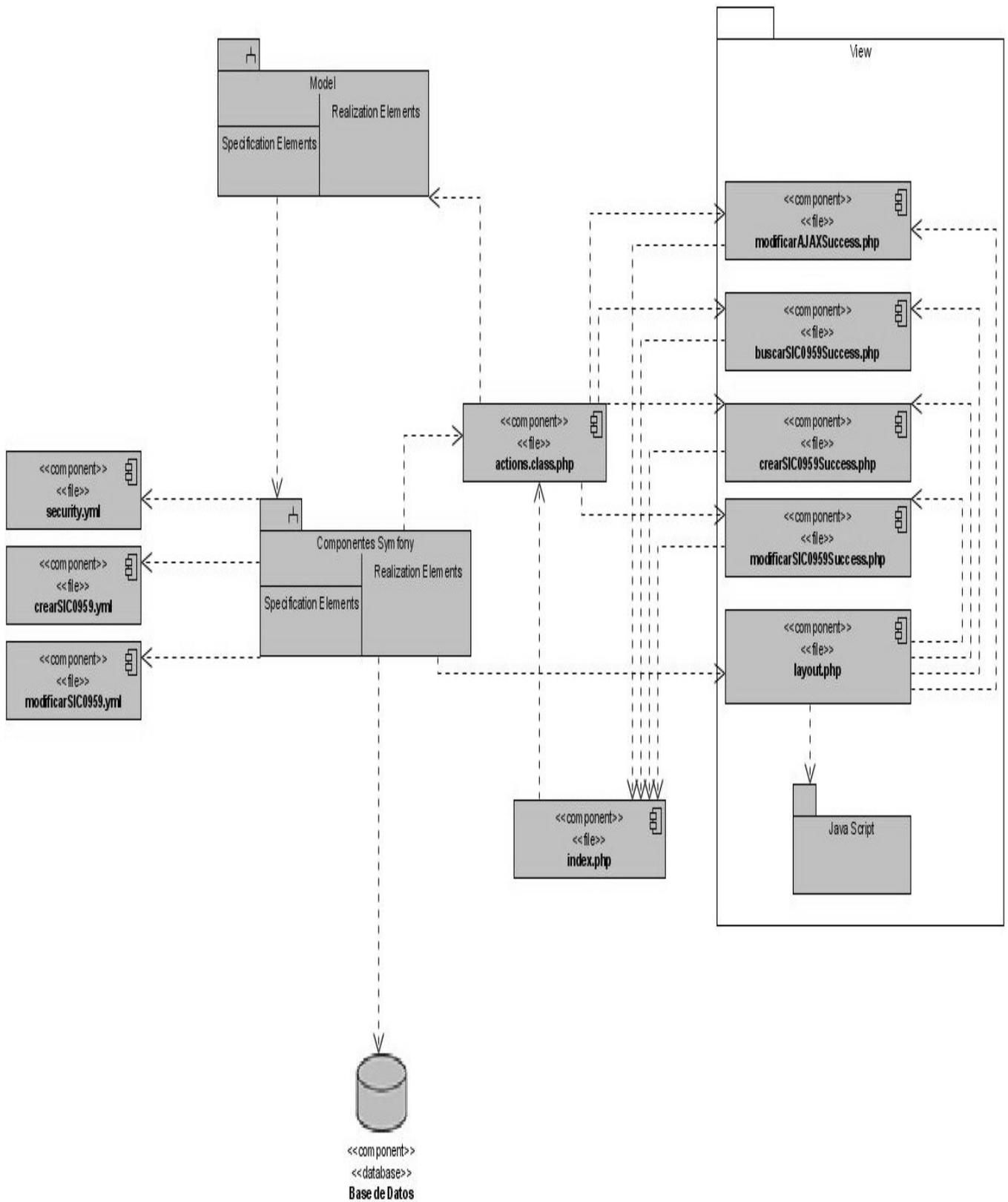


Figura 8 Diagrama de componentes del CU Gestionar SIC0959

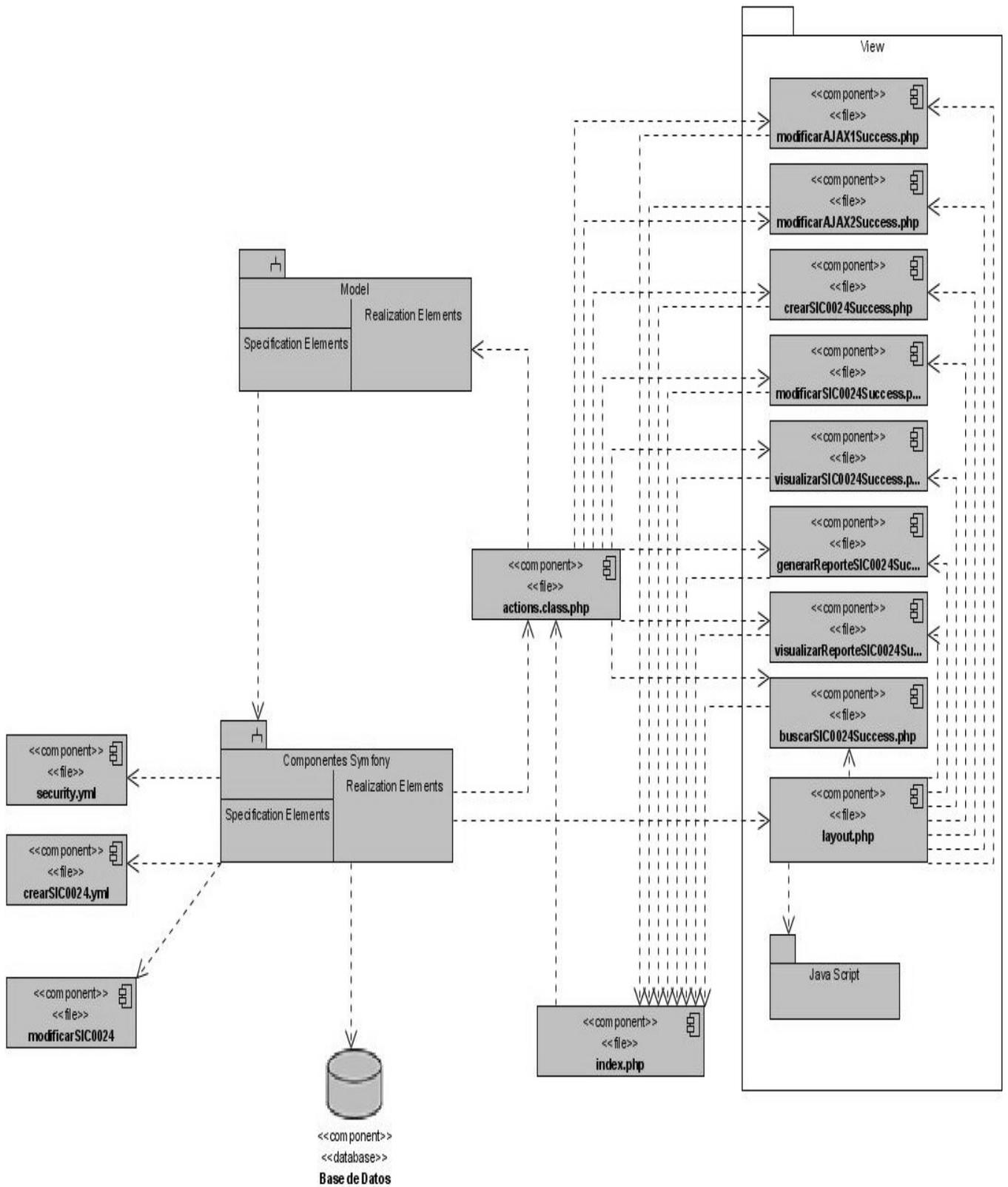


Figura 9 Diagrama de componentes del CU Gestionar SIC0024

2.5 Código Fuente

En este epígrafe se muestra el código fuente del módulo, los CUS se dividen en dos paquetes uno es el de las No Conformidades, y el otro es Mejora y Control de la Calidad. En el primero están implementados los SIC 0023, 0024, 0042, 0959 y en el segundo están los SIC 0098, 0816, 0963, 0919,0851. A continuación se muestra dos códigos por cada CUS, por que casi todos los casos de usos que hay presentan las mismas funcionalidades, es decir, un buscar, visualizar, generar reporte, crear, registrar, visualizar reporte, imprimir, modificar y todos se implementan de forma similar.

A continuación se muestra el código fuente de la función Buscar en la clase action.class del módulo SIC0098, en el cuál pasándole tres patrones de búsqueda devuelve los SIC que presenten dichos patrones.

```

public function executeBuscarSIC0098()
{
    if ($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $folios=array("Select....");
        $plantas=array("Select....");
        $clases=array("Select....");
        $fo=Sic0098Peer::BuscarFolios();
        $plan=Sic0098Peer::BuscarPlantas();
        $clas=Sic0098Peer::BuscarClase();
        for($i=0;$i<count($fo);$i++)
            array_push($folios,$fo[$i]);
        for($i=0;$i<count($clas);$i++)
            array_push($clases,$clas[$i]);
        for($i=0;$i<count($plan);$i++)
            array_push($plantas,$plan[$i]);
        $this->mensaje="";
        $this->bus_folio=$folios;
        $this->bus_clase=$clases;
        $this->bus_planta=$plantas;
        $this ->arreglo=1;

        return sfView::SUCCESS;
    }
}

```

```

}
else
{
    $folios=array("Select....");
    $plantas=array("Select....");
    $clases=array("Select....");
    $fo=Sic0098Peer::BuscarFolios();
    $plan=Sic0098Peer::BuscarPlantas();
    $clas=Sic0098Peer::BuscarClase();
    for($i=0;$i<count($fo);$i++)
        array_push($folios,$fo[$i]);
    for($i=0;$i<count($clas);$i++)
        array_push($clases,$clas[$i]);
    for($i=0;$i<count($plan);$i++)
        array_push($plantas,$plan[$i]);
    $this->bus_folio=$folios;
    $this->bus_clase=$clases;
    $this->bus_planta=$plantas;
    $posicion_folio=$this->getRequestParameter('bus_folio');
    $posicion_planta=$this->getRequestParameter('bus_planta');
    $posicion_clase=$this->getRequestParameter('bus_clase');
    if($posicion_folio==0 && $posicion_planta==0 && $posicion_clase==0)
    {
        $this->mensaje='Entre al menos un criterio de busqueda';
        $this ->arreglo=2;
    }
    else
    {
        $this->mensaje="";
        $c = new Criteria();
        if($posicion_folio!=0)
        {
            $folio_buscar=$folios[$posicion_folio];
            $c->add(Sic0098Peer::IDFOLIO,$folio_buscar);

```

```

    }
    if($posicion_planta!=0)
    {
        $planta_buscar=$plantas[$posicion_planta];
        $c->add(Sic0098Peer::LOCAL,$planta_buscar);
    }
    if($posicion_clase!=0)
    {
        $clase_buscar=$clases[$posicion_clase];
        $c->add(Sic0098Peer::CLASE,$clase_buscar);
    }
    $sic0098=Sic0098Peer::doSelect($c);
    $this->arreglo=Sic0098Peer::BuscarSIC($sic0098);

}
} (Ver. Anexo 8)

```

El código fuente que se muestra aquí es de la función VisualizarReporte en la clase action.class del módulo SIC0098 seleccionando un producto, una clase, o un área visualiza el reporte según los parámetros seleccionados.

```

public function executeVisualizarReporteSIC0098()
{
    $arreglo=array();
    $arreglo_f=array();
    $lista_prod1=Sic0098Peer::BuscarProd();
    $lista_area1=Sic0098Peer::BuscarArea();
    $lista_clase1=Sic0098Peer::BuscarClase();
    $mostrar_clase1="";
    $mostrar_area1="";
    $mostrar_producto1="";
    $posicion_prod=$this->getRequestParameter('select_prod');
    $posicion_clase=$this->getRequestParameter('select_clase');
    $posicion_area=$this->getRequestParameter('select_area');
}

```

```

$todos=$this->getRequestParameter('todos');
$area=$this->getRequestParameter('area');
$ztrab=$this->getRequestParameter('ztrab');
$clase=$this->getRequestParameter('clase');
$np=$this->getRequestParameter('np');
$metodo=$this->getRequestParameter('metodo');
$Caract=$this->getRequestParameter('Caract');
$responsable=$this->getRequestParameter('responsable');
$frec=$this->getRequestParameter('frec');
$int=$this->getRequestParameter('int');
    if($posicion_area>0 || $posicion_clase>0 ||$posicion_prod>0)
    {
        $c = new Criteria();
        if($posicion_prod!=0)
        {
            $prod_buscar=$lista_prod1[$posicion_prod-1];
            $c->add(Sic0098Peer::PRODUCCION,$prod_buscar);
            $mostrar_producto1='Producto: '.$prod_buscar;
        }
    }
if($posicion_area!=0)
    {
        $area_buscar=$lista_area1[$posicion_area-1];
        $c->add(Sic0098Peer::AREA,$area_buscar);
        $mostrar_area1='Area: '.$area_buscar;
    }
if($posicion_clase!=0)
    {
        $clase_buscar=$lista_clase1[$posicion_clase-1];
        $c->add(Sic0098Peer::CLASE,$clase_buscar);
        $mostrar_clase1='Clase: '.$clase_buscar;
    }
$this->mostrar_clase=$mostrar_clase1;
$this->mostrar_area= $mostrar_area1;
$this->mostrar_producto=$mostrar_producto1;

```

```

    $sic0098=Sic0098Peer::doSelect($c);
    $arreglo1=array();
    $arreglo2=array();
for($i=0;$i<count($sic0098);$i++)
    {
    $aux=array();
    array_push($aux,$sic0098[$i]->getIdFolio());
    $fo=$sic0098[$i]->getIdFolio();
    $d= new Criteria();
    $d->add(MonitoreoambientalPeer::SIC0098IDFOLIO ,$fo);
    $lolo=MonitoreoambientalPeer::doSelect($d);
for($j=0;$j<count($lolo);$j++)
    array_push($arreglo2,$lolo[$j]);
    array_push($aux,$sic0098[$i]->getProduccion());
    array_push($aux,$sic0098[$i]->getArea());
array_push($aux,$sic0098[$i]->getNp());
array_push($aux,$sic0098[$i]->getLocal());
array_push($aux,$sic0098[$i]->getClase());
array_push($aux,$sic0098[$i]->getZonaTrabajo());
array_push($aux,$sic0098[$i]->getAprobadoPor());
array_push($aux,$sic0098[$i]->getFechaRealizacion());
array_push($aux,$sic0098[$i]->getFechaRevision());
array_push($aux,$sic0098[$i]->getFechaAprobacion());
array_push($aux,$sic0098[$i]->getTerminado());
array_push($arreglo1,$aux);
    }
    $val=MonitoreoambientalPeer::BuscarArregloMonitoreo($arreglo2);
    $this->ver=$val;
    if($todos==1)
    {
    $clase=1;
    $Caract=1;
    $np=1;
    $metodo=1;

```

```

$responsable=1;
$area=1;
$frec=1;
$ztrab=1;
$int=1;
}
if($clase==1)
{
$yo=array('Clase');
for($i=0;$i<count($arreglo1);$i++)
{
array_push($yo,$arreglo1[$i][5]);
}
array_push($arreglo,$yo);
}
if($np==1)
{
$yo=array('NP');
for($i=0;$i<count($arreglo1);$i++)
array_push($yo,$arreglo1[$i][3]);
array_push($arreglo,$yo);
}
if($metodo==1)
{
$yo=array('Metodo');
for($i=0;$i<count($val);$i++)
array_push($yo,$val[$i][2]);
array_push($arreglo_f,$yo);
}
if($Caract==1)
{
$yo=array('Caracteristicas');
for($i=0;$i<count($val);$i++)
array_push($yo,$val[$i][1]);
}

```

```

        array_push($arreglo_f,$yo);
    }
    if($responsable==1)
    {
        $yo=array('Responsable');
        for($i=0;$i<count($val);$i++)
            array_push($yo,$val[$i][8]);
        array_push($arreglo_f,$yo);
    }
    if($frec==1)
    {
        $yo=array('Frecuencia');
        for($i=0;$i<count($val);$i++)
            array_push($yo,$val[$i][5]);
        array_push($arreglo_f,$yo);
    }
    if($int==1)
    {
        $yo=array('Int');
        for($i=0;$i<count($val);$i++)
            array_push($yo,$val[$i][6]);
        array_push($arreglo_f,$yo);
    }
    if($area==1)
    {
        $yo=array('Area');
        for($i=0;$i<count($arreglo1);$i++)
            array_push($yo,$arreglo1[$i][2]);
        array_push($arreglo,$yo);
    }
    if($ztrab==1)
    {
        $yo=array('Z. de Trabajo');
        for($i=0;$i<count($arreglo1);$i++)

```

```

        array_push($yo,$arreglo1[$i][6]);
        array_push($arreglo,$yo);
    }
    $this->arreglover=$arreglo;
    $this->arreglover_f=$arreglo_f;
    }
    else
    {
        $men='Entre bien los datos';
        $this->redirect('SIC0098/GenerarReporteSIC0098?mensaje='.$men);
    }
} (Ver anexo 9)

```

En los SIC se tienen campos que hay que adicionarlos muchas veces sin actualizar la página por lo que a continuación se muestra un código Java Script que adiciona todos esos datos a un campo oculto resolviendo dicho problema.

```

<script>
function save()
{
    if(NoAste('crear_caract')&&    NoVacio('crear_caract')    &&    NoAste('crear_met')    &&
    NoVacio('crear_met') && NoAste('crear_alerta') && NoVacio('crear_alerta') && NoAste('crear_accion')
&&    NoVacio('crear_accion')    &&    NoAste('crear_frec')    &&    NoVacio('crear_frec')    &&
SoloNumeros('crear_intru') && NoVacio('crear_intru') && NoAste('crear_ref')&& NoVacio('crear_ref') &&
NoAste('crear_respon')&& NoVacio('crear_respon'))
    {
        var carac= document.getElementById('crear_caract');
        var caracteristica= document.getElementById('caracteristica');
        caracteristica.value= caracteristica.value + carac.value + '*';
        carac.value="";
        var met= document.getElementById('crear_met');
        var metodo= document.getElementById('metodo');
        metodo.value= metodo.value + met.value + '*';
        met.value="";
        var aler= document.getElementById('crear_alerta');

```

```

var alerta= document.getElementById('alerta');
alerta.value= alerta.value + aler.value + '*';
aler.value="";
var acci= document.getElementById('crear_accion');
var accion= document.getElementById('accion');
accion.value= accion.value + acci.value + '*';
acci.value="";
var frec= document.getElementById('crear_frec');
var frecuencia= document.getElementById('frecuencia');
frecuencia.value= frecuencia.value + frec.value + '*';
frec.value="";
var intru= document.getElementById('crear_intru');
var instrumento= document.getElementById('instrumento');
instrumento.value= instrumento.value + intru.value + '*';
intru.value="";
var ref= document.getElementById('crear_ref');
var referencia= document.getElementById('referencia');
referencia.value= referencia.value + ref.value + '*';
ref.value="";
var respon= document.getElementById('crear_respon');
var responsable= document.getElementById('responsable');
responsable.value= responsable.value + respon.value + '*';
respon.value="";
    }
    else
    {
        alert('Entre los datos correctamente');
    }
}
</scrip>

```

Aquí se expone el código fuente de la función Crear en la clase action.class del módulo SIC0816 en el cuál se pasa el folio y este a su vez se guarda en la base de datos.

```
public function executeCrearSIC0816()
{
    if ($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $this->mensaje="";
        return sfView::SUCCESS;
    }
    else
    {
        $folio=$this->getRequestParameter('crear_folio');
        if(ControlcambiosPeer::ExisteFolio($folio))
        {
            $this->mensaje='Ese Folio Ya existe';
            return sfView::SUCCESS;
        }
        else
        {
            $cont=new Controlcambios();
            $cont->setFolio($folio);
            $cont->save();
            $this->mensaje='Folio entrado correctamente';
            return sfView::SUCCESS;
        }
    }
}
}(Ver anexo 10)
```

Seguidamente se muestra el código fuente de la función Registrar en la clase action.class del módulo SIC0816 en el cuál se pasan los valores y se guarda en la base de datos.

```
public function executeRegistrarSIC0816()
{
    if ($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $sic=ControlcambiosPeer::BuscarFolios();
```

```

$aux=array('Select...');
for($i=0;$i<count($sic);$i++)
array_push($aux,$sic[$i]);
$this->list_folio=$aux;
$this->mensaje="";
return sfView::SUCCESS;
}
else
{
$sic=ControlcambiosPeer::BuscarFolios();
$aux=array('Selest...');
for($i=0;$i<count($sic);$i++)
array_push($aux,$sic[$i]);
$pos=$this->getRequestParameter('reg_folio');
$codigo=$this->getRequestParameter('codigo');
$arr_codigo= explode('*', $codigo);
$recep=$this->getRequestParameter('fecha_recep');
$arr_recep= explode('*', $recep);
$clase1=$this->getRequestParameter('clase1');
$arr_menor= explode('*', $clase1);
$clase2=$this->getRequestParameter('clase2');
$arr_mayor= explode('*', $clase2);
$clase1_1=$this->getRequestParameter('enrevi');
$arr_enrev= explode('*', $clase1_1);
$clase1_2=$this->getRequestParameter('pendiente');
$arr_pendiente= explode('*', $clase1_2);
$clase1_3=$this->getRequestParameter('enreg');
$arr_enreg= explode('*', $clase1_3);
$final=$this->getRequestParameter('des_final');
$arr_final= explode('*', $final);
$conc=$this->getRequestParameter('fecha_conc');
$arr_conc= explode('*', $conc);
    if(!Sic0816Peer::ExisteFolio($aux[$pos]))
        for($i=0; $i< count($arr_codigo)-1; $i++)

```

```

        {
            Sic0816Peer::CrearSIC0816($aux[$pos],$sarr_codigo[$i],$sarr_final[$i],$sarr_enreg[$i],$sarr
            _enrev[$i],$sarr_pendiente[$i],$sarr_conc[$i],$sarr_recep[$i],$sarr_mayor[$i],$sarr_menor[$i],0);
            $this->ver='entro';
        }
        $obser=$this->getRequestParameter('reg_observ');
        Observacionessic0816Peer::CrearObserCCambio($aux[$pos],$obser);
        Sic0816Peer::InsertFolio($aux[$pos]);
        $this->redirect('SIC0816/VisualizarSIC0816?enviar='.$aux[$pos]);
    } } (Ver anexo 11)

```

Se muestra a continuación el código fuente de la función Visualizar en la clase action.class del módulo SIC0919, en el cuál pasándole la llave primaria busca todos los valores que se desean mostrar.

```

public function executeVisualizarSIC0919()
{
    $folio=$this->getRequestParameter('enviar');
    Sic0919Peer::InsertFolio($folio);
    $sarr1=Sic0919Peer::BuscarSic0919PorFolio($folio);
    $sic=Sic0919Peer::BuscarArregloSic0919($sarr1);
    $sarr2=InformetendenciasPeer::retrieveByPK($folio);
    $sic2=InformetendenciasPeer::BuscarArregloInfTendencia($sarr2);
    $this->dev=$sic;
    $this->dev2=$sic2;
} (Ver anexo 12)

```

A continuación se expone el código fuente de la función VisualizarReporte en la clase action.class del módulo SIC0919 seleccionando el año, un mes de inicio y fin visualiza el reporte según los parámetros seleccionados.

```

public function executeVisualizarReporteSIC0919()
{
    $mesini=$this->getRequestParameter('select_mesini');
    $mesfin=$this->getRequestParameter('select_mesfin');
    $select_anno=$this->getRequestParameter('select_anno');

```

```

if( $select_anno==0 && ( $mesini==0 || $mesfin==0))
{
$men= 'Entre Bien Los Datos';
$this->redirect('SIC0919/GenerarReporteSIC0919?mensaje=.'.$men);
}
$lista_anno=array('Select...','1980','1981','1982','1983','1984'
,'1985','1986','1987','1988','1989','1990','1991','1992','1993','1994','1995','1996','1997','1998','1999','200
0','2001','2002','2003','2004','2005','2006','2007','2008');
if($select_anno==0)
{
$c= new Criteria();
$annos_bus=InformetendenciasPeer::doSelect($c);
}
else
{
$busc_anno=$lista_anno[$select_anno];
$annos_bus=InformetendenciasPeer::BuscarPorAnno($busc_anno);
}
$annos=InformetendenciasPeer::BuscarArregloInfTendencia1($annos_bus);
$this->ver=$annos;
$arreglo=array();
$c = new Criteria();
$sic0919=Sic0919Peer::doSelect($c);
$sic=Sic0919Peer::BuscarArregloSic0919($sic0919);
$arreglo4=array();
$arreglo2=array();
if($mesini>$mesfin && $mesfin!=0 )
$this->redirect('SIC0919/GenerarReporteSIC0919');
$anno=$this->getRequestParameter('anno');
$mes=$this->getRequestParameter('mes');
$des=$this->getRequestParameter('desviacion');
$todos=$this->getRequestParameter('todos');
if($anno!=1 && $mes!=1 && $des!=1 && $todos!=1)
$this->redirect('SIC0919/GenerarReporteSIC0919');

```

```

if($this->getRequestParameter('todos')==1)
{
    $anno=1;
    $mes=1;
    $des=1;
}
$array1=Sic0919Peer::BuscarPorMeses($mesini,$mesfin);
$sic=Sic0919Peer::BuscarArregloSic0919($array1);
$this->ver21=$sic;
    $aver=array();
if(count($annos)>0)
{
for($i=0;$i<count($annos);$i++)
{
    $arreglo8="";
for($j=0;$j<count($sic);$j++)
{
if($sic[$j][1]==$annos[$i][0])
{
array_push($arreglo,$sic[$j]);
$arreglo8=$annos[$i][1];
}
}
if($arreglo8!="")
array_push($aver, $arreglo8);
}
}
if($mes==1)
{
$aux=array('Mes');
if(count($arreglo)>0)
{
for($i=0;$i<count($arreglo);$i++)
array_push($aux,$arreglo[$i][2]);
}
}
}

```

```

array_push($arreglo4,$aux);
}
else
array_push($arreglo4,'No Existe PIT en este Anno');
}
if($anno==1)
{
$aux=array('A&ntilde;o');
for($i=0;$i<count($aver);$i++)
array_push($aux,$aver[$i]);
array_push($arreglo2,$aux);
}
if($des==1)
{
$aux=array('Desviacion');
if(count($arreglo)>0)
{
for($i=0;$i<count($arreglo);$i++)
array_push($aux,$arreglo[$i][3]);
array_push($arreglo4,$aux);
}
}
$this->lolo=$arreglo2;
$this->lolo1=$arreglo4;
$this->meees=$mesini;
$this->meees1=$mesfin;
//$this->ver1=$bd->BuscarArregloT2Mes1($select_anno,1,4);
} (Ver anexo 13)

```

El código fuente que se muestra es el de la función Registrar en la clase action.class del módulo SIC0963 en el cuál se pasan los valores y se guarda en la base de datos.

```

public function executeRegistrarSIC0963()
{

```

```

if ($this->getRequest()->getMethod() != sfRequest::POST)
{
    $arr=Reportessic0963Peer::BuscarFolio();
    $dev=array('Select...');
    for($i=0;$i<count($arr);$i++)
    array_push($dev,$arr[$i]);
    $this->list_folio=$dev;
    $this->mensaje="";
    return sfView::SUCCESS;
}
else
{
    $fol=$this->getRequestParameter('select_fol');
    $arr=Reportessic0963Peer::BuscarFolio();
    $dev=array('Select...');
    for($i=0;$i<count($arr);$i++)
    array_push($dev,$arr[$i]);
    $this->list_folio=$dev;
    $folio=$arr[$fol-1];
    if(!Sic0963Peer::ExisteFolio($folio))
    {
        $plan=$this->getRequestParameter('splanta');
        $arr_planta= explode('*', $plan);
        $fecha=$this->getRequestParameter('sfecha');
        $arr_fecha= explode('*', $fecha);
        $area=$this->getRequestParameter('sarea');
        $arr_area= explode('*', $area);
        $caract=$this->getRequestParameter('scaract');
        $arr_caract= explode('*', $caract);
        $local=$this->getRequestParameter('slocal');
        $arr_local= explode('*', $local);
        $pm=$this->getRequestParameter('spm');
        $arr_pm= explode('*', $pm);
        $nombre=$this->getRequestParameter('snombre');
    }
}

```

```

    $arr_nombre= explode('*', $nombre);
    $ai=$this->getRequestParameter('sl_ai');
    $arr_ai= explode('*', $ai);
    $ac=$this->getRequestParameter('sl_ac');
    $arr_ac= explode('*', $ac);
    for($i=0;$i<count($arr_area)-1;$i++)
Sic0963Peer::CrearSIC0963($folio,$arr_planta[$i],$arr_local[$i],$arr_fecha[$i],$arr_area[$i],$arr_caract
[$i],$arr_pm[$i],$arr_nombre[$i],$arr_ac[$i],$arr_ai[$i]);
    $obser=$this->getRequestParameter('obser');
    $rev_pr=$this->getRequestParameter('rev_por');
    $f_real=$this->getRequestParameter('f_real');
    $reali_por=$this->getRequestParameter('reali_por');
    $fe_rev=$this->getRequestParameter('fe_rev');
    $term1=$this->getRequestParameter('term');
    $this->eee=$term1;
    $term=0;
    if($term1==1)
    $term=1;
    Observacionessic0963Peer::CrearObserSIC0963($obser,$reali_por,$rev_pr,$fe_rev,$f_real,$fol
io,$term);
    $nen=Sic0963Peer::EntrarFolio($folio);
    $this->redirect('SIC0963/visualizarSIC0963?codigo='.$folio);
    }
    $this->mensaje ="
    }
    }(Ver anexo 14)

```

A continuación se muestra el código fuente de la función Modificar en la clase action.class del módulo SIC0963 en el cuál se modifican los datos seleccionados.

```

public function executeModificarSIC0963()
{
if ($this->getRequest()->getMethod() != sfRequest::POST)
{

```

```

$fol=$this->getRequestParameter('codigo');
$array1=Reportessic0963Peer::BuscarPorFolio($fol);
$array2=Sic0963Peer::BuscarParaVisualizar($fol);
$array3=Observacionessic0963Peer::BuscareParaVisauliazar1($fol);
$this->ver=Sic0963Peer::ConvertirObjetoSIC0963($array2);
$this->ver1=Reportessic0963Peer::ConvertirObjetoReporSIC0963($array1);
$this->ver2=Observacionessic0963Peer::ConvertirObjetoobserSIC0963($array3);
return sfView::SUCCESS;
}
else
{
$folio=$this->getRequestParameter('folio');
$plan=$this->getRequestParameter('splanta');
$arr_planta= explode('*', $plan);
$fecha=$this->getRequestParameter('sfecha');
$arr_fecha= explode('*', $fecha);
$area=$this->getRequestParameter('sarea');
$arr_area= explode('*', $area);
$caract=$this->getRequestParameter('scaract');
$arr_caract= explode('*', $caract);
$local=$this->getRequestParameter('slocal');
$arr_local= explode('*', $local);
$pm=$this->getRequestParameter('spm');
$arr_pm= explode('*', $pm);
$nombre=$this->getRequestParameter('snombre');
$arr_nombre= explode('*', $nombre);
$ai=$this->getRequestParameter('sl_ai');
$arr_ai= explode('*', $ai);
$ac=$this->getRequestParameter('sl_ac');
$arr_ac= explode('*', $ac);
for($i=0;$i<count($arr_area)-1;$i++)
Sic0963Peer::CrearSIC0963($folio,$arr_planta[$i],$arr_local[$i],$arr_fecha[$i],$arr_area[$i],$arr_caract
[$i],$arr_pm[$i],$arr_nombre[$i],$arr_ac[$i],$arr_ai[$i]);
$obser=$this->getRequestParameter('obser');

```

```

$rev_pr=$this->getRequestParameter('rev_por');
$f_real=$this->getRequestParameter('f_real');
$reali_por=$this->getRequestParameter('reali_por');
$fe_rev=$this->getRequestParameter('fe_rev');
$term1=$this->getRequestParameter('term');
$this->eee=$term1;
$term=0;
if($term1==1)
$term=1;
Observacionessic0963Peer::ModificarObserSIC0963($obser,$reali_por,$rev_pr,$fe_rev,$f_real,
$folio,$term);
$this->redirect('SIC0963/visualizarSIC0963?codigo='.$folio);
}
}(Ver anexo 15)

```

Seguidamente se muestra el código fuente de la función Visualizar en la clase action.class del módulo SIC0851, en el cuál pasándole la llave primaria busca todos los valores que se desean mostrar.

```

public function executeVisualizarSIC0851()
{
    $prod=$this->getRequestParameter('enviar');
    $arr_c=ControldesviacionesPeer::BuscarProd($prod);
    $arr_sic=Sic0851Peer::BuscarProducto($prod);
    $this->ver=Sic0851Peer::ConvArrayObj($arr_sic);
    $this->ver1=ControldesviacionesPeer::ConvArrObj($arr_c);
}
}(Ver anexo 16)

```

Este es el código fuente de la función VisualizarReporte en la clase action.class del módulo SIC0851 seleccionando el año, un mes de inicio y fin visualiza lo buscado según los parámetros seleccionados.

```

public function executeVisualizarReporteSIC0851()
{
    $dev=array();
    $arr_eta=array();
}

```

```

$etap_pos=$this->getRequestParameter('select_etapa');
$f_init=$this->getRequestParameter('fecha_inicial');
$f_fin=$this->getRequestParameter('fecha_final');
$todo=$this->getRequestParameter('todos');
$cod=$this->getRequestParameter('cod');
$desc=$this->getRequestParameter('desc');
$prod=$this->getRequestParameter('prod');
if($etap_pos==0 &&($f_fin==" || $f_init=="))
{
    $this->redirect('SIC0851/GenerarReporteSIC0851?mensaje=Entre Bien Los Datos');
}
else
{
    $etap_arr=Sic0851Peer::BuscarEtapas();
    $c=new Criteria();
    if($etap_pos>0)
    {
        $aux=$etap_arr[$etap_pos-1];
        $c->add(Sic0851Peer::ETAPA,$aux);
    }
    $sic=Sic0851Peer::doSelect($c);
    $arr_aux=array();
    if($f_fin!=" && $f_init!=")
    {
        $fecha_i=str_replace('-',",$f_init);
        $fecha_f=str_replace('-',",$f_fin);
        //$this->ver1=$fecha_i;
        //$this->ver2=$fecha_f;
        for($i=0;$i<count($sic);$i++)
        {
            $fecha= str_replace('-',",$sic[$i]->getFechanotificacionrealizacion());
            if($fecha>=$fecha_i && $fecha<=$fecha_f)
                array_push($arr_aux,$sic[$i]);
        }
    }
}

```

```

}
else
$arr_aux=$sic;
$ver=Sic0851Peer::ConvArrayObj($arr_aux);
if($todo!=1 && $cod!=1 && $desc!=1 && $prod!=1 )
$this->redirect('SIC0851/GenerarReporteSIC0851?mensaje=Entre Bien Los Parametros a
mostrar');
else
{
if($todo==1)
{
    $cod=1;
    $desc=1;
    $prod=1;
}
if($cod==1)
{
    $aux=array('Codigo/SIC0097');
for($i=0;$i<count($ver);$i++)
array_push($aux,$ver[$i][1]);
array_push($dev,$aux);
}
if($desc==1)
{
    $aux=array('Desc. de la Desv');
for($i=0;$i<count($ver);$i++)
array_push($aux,$ver[$i][5]);
array_push($dev,$aux);
}
if($prod==1)
{
    $aux=array('Producto');
for($i=0;$i<count($ver);$i++)
array_push($aux,$ver[$i][6]);

```

```
array_push($dev,$aux);
    }
    }
    $this->ver=$dev;    }}}
```

?> (Ver anexo 17)

A continuación se muestra el código fuente de la función Crear en la clase action.class del módulo SIC0042 en el cuál se pasan los valores y se guarda en la base de datos.

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

```

$car_per1=$this->getRequestParameter('car_per1');
$inst=$this->getRequestParameter('inst');
$direc=$this->getRequestParameter('direc');
$vias_cont=$this->getRequestParameter('vias_cont');
$desc_prob=$this->getRequestParameter('desc_prob');
$real_por=$this->getRequestParameter('real_por');
$carg_real=$this->getRequestParameter('carg_real');
$f_real=$this->getRequestParameter('f_real');
$revi_por=$this->getRequestParameter('revi_por');
$carg_rev=$this->getRequestParameter('carg_rev');
$f_rev=$this->getRequestParameter('f_rev');
$terminado=$this->getRequestParameter('terminado');
$id=ControlplanillaPeer::ControlPlanilla(",",$real_por,$f_real,"",$revi_por,$f_rev);
$llave=NotificacionquejasPeer::CrearSIC0042($cod_prop,$nom_prod,$for_farm,$f_not,$present,$lote_in
nv,$via_com,$pers_not,$car_per,$persona_not,$car_per1,$inst,$direc,$vias_cont,$desc_prob,$carg_r
ev,$carg_real,$id,0);
$this->ver=$llave;
$this->redirect('SIC0042/VisualizarSIC0042?enviar='.$cod_prop.'&enviar1='.$id);
} } } (Ver anexo 18)

```

El código fuente que se muestra es de la función Crear en la clase action.class del módulo SIC0959 en el cuál se pasa el folio y este a su vez se guarda en la base de datos.

```

public function executeCrearSIC0959()
{
if ($this->getRequest()->getMethod() != sfRequest::POST)
{
$this->list_cod=NotificacionquejasPeer::BuscarCodigos();
$this->mensaje="";
return sfView::SUCCESS;
}
else
{
$select_cod=$this->getRequestParameter('select_cod');
$nombre=$this->getRequestParameter('nombre');

```

```

$radiobuton=$this->getRequestParameter('radiobuton');
$cod_ant=$this->getRequestParameter('cod_ant');
$conclusion=$this->getRequestParameter('conclusion');
$queja=$this->getRequestParameter('queja[]');
$repos=$this->getRequestParameter('repos[]');
$reali_por=$this->getRequestParameter('reali_por');
$carg_real=$this->getRequestParameter('carg_real');
$f_real=$this->getRequestParameter('f_real');
$rev_por=$this->getRequestParameter('rev_por');
$carg_rev=$this->getRequestParameter('carg_rev');
$f_rev=$this->getRequestParameter('f_rev');
$aprob=$this->getRequestParameter('aprob');
$carg_aprob=$this->getRequestParameter('carg_aprob');
$f_apro=$this->getRequestParameter('f_apro');
$terminado=$this->getRequestParameter('terminado');
$list_cod=NotificacionquejasPeer::BuscarCodigos();
$doc_cod=$this->getRequestParameter('sdoc_cod');
$sarr_doc_cod= explode('*', $doc_cod);
$doc_des=$this->getRequestParameter('sdoc_des');
$sarr_doc_des= explode('*', $doc_des);
$doc_fec=$this->getRequestParameter('sdoc_fec');
$sarr_doc_fec= explode('*', $doc_fec);
$nomb=$this->getRequestParameter('snomb');
$sarr_nomb= explode('*', $nomb);
$carg=$this->getRequestParameter('scarg');
$sarr_carg= explode('*', $carg);
if(CierrequejaPeer::ExisteCodigo($list_cod[$select_cod]))
{
$this->list_cod=NotificacionquejasPeer::BuscarCodigos();
$this->mensaje='Ya se hizo un Cierre de Quejas con eseCodigo';
return sfView::SUCCESS;
}
else

```

```

        {
            $id=ControlplanillaPeer::ControlPlanilla(0,$aprob,$reali_por,$f_real,$f_apro,$rev_por,$f_rev);
            CierrequejaPeer::CrearCierreQueja($list_cod[$select_cod],$nombre,$radiobuton,$cod_ant,$conclusion,$queja[0],$repos[0],$terminado,$id,0);
            for($i=0;$i<count($arr_nomb)-1;$i++)
                {
                    ParticipantescomisonPeer::CrearParticipanTCom(null,$arr_nomb[$i],$arr_carg[$i],$list_cod[$select_cod]);
                }
            for($i=0;$i<count($arr_doc_cod)-1;$i++)
                DocumentosPeer::CrearDoc($arr_doc_cod[$i],$arr_doc_des[$i],$arr_doc_fec[$i],$list_cod[$select_cod]);
            $this->redirect('SIC0959/VisualizarSIC0959?enviar='.$list_cod[$select_cod].'&enviar1='.$id);
        }
    }
}(Ver anexo 19)

public function executeRegistrarAJAX()
{
    $list_cod=NotificacionquejasPeer::BuscarCodigos();
    $cod=$this->getRequestParameter('cod');
    if($cod>0)
        $this->cod=NotificacionquejasPeer::BuscarNombre($list_cod[$cod]);
    else
        $this->cod='Entre unCodigo';
}

```

A continuación se expone el código fuente de la función Modificar en la clase action.class del módulo SIC0959 en el cuál se modifican los datos seleccionados.

```

public function executeModificarSIC0959()
{
    if ($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $codigo=$this->getRequestParameter('enviar');
    }
}

```

```

$id=$this->getRequestParameter('enviar1');
$cierre=CierrequejaPeer::BuscarPorCodigo($codigo);
$part=ParticipantescomisonPeer::BuscarPorCodigo($codigo);
$doc=DocumentosPeer::BuscarPorCodigo($codigo);
$cplanilla=ControlplanillaPeer::BuscarPlanilla($id);
$this->vercier=CierrequejaPeer::ConvArrObjet($cierre);
$this->verpart=ParticipantescomisonPeer::ConvArrayObj($part);
$this->verdoc=DocumentosPeer::ConvArrObj($doc);
$this->verplan=ControlplanillaPeer::ConvArrayObj($cplanilla);
$this->mensaje="";
return sfView::SUCCESS;
}
else
{
$codigo=$this->getRequestParameter('idcierre');
$id=$this->getRequestParameter('idpani');
/*$cierre=CierrequejaPeer::BuscarPorCodigo($codigo);
$part=ParticipantescomisonPeer::BuscarPorCodigo($codigo);
$doc=DocumentosPeer::BuscarPorCodigo($codigo);
$cplanilla=ControlplanillaPeer::BuscarPlanilla($id);
$this->vercier=CierrequejaPeer::ConvArrObjet($cierre);
$this->verpart=ParticipantescomisonPeer::ConvArrayObj($part);
$this->verdoc=DocumentosPeer::ConvArrObj($doc);
$this->verplan=ControlplanillaPeer::ConvArrayObj($cplanilla);*/
$nombre=$this->getRequestParameter('nombre');
$radiobuton=$this->getRequestParameter('radiobuton');
$cod_ant=$this->getRequestParameter('cod_ant');
$conclusion=$this->getRequestParameter('conclusion');
$queja=$this->getRequestParameter('queja[]');
$repos=$this->getRequestParameter('repos[]');
$reali_por=$this->getRequestParameter('reali_por');
$carg_real=$this->getRequestParameter('carg_real');
$f_real=$this->getRequestParameter('f_real');
$rev_por=$this->getRequestParameter('rev_por');

```

```

$carg_rev=$this->getRequestParameter('carg_rev');
$f_rev=$this->getRequestParameter('f_rev');
$aprob=$this->getRequestParameter('aprob');
$carg_aprob=$this->getRequestParameter('carg_aprob');
$f_apro=$this->getRequestParameter('f_apro');
$terminado=$this->getRequestParameter('terminado');
$doc_cod=$this->getRequestParameter('sdoc_cod');
$arr_doc_cod= explode('*', $doc_cod);
$doc_des=$this->getRequestParameter('sdoc_des');
$arr_doc_des= explode('*', $doc_des);
$doc_fec=$this->getRequestParameter('sdoc_fec');
$arr_doc_fec= explode('*', $doc_fec);
$nomb=$this->getRequestParameter('snomb');
$arr_nomb= explode('*', $nomb);
$carg=$this->getRequestParameter('scarg');
$arr_carg=explode('*', $carg);
$id=ControlPlanillaPeer::ControlPlanilla($id,$aprob,$reali_por,$f_real,$f_apro,$rev_por,$f_rev);
CierrequejaPeer::CrearCierreQueja($codigo,$nombre,$radiobuton,$cod_ant,$conclusion,$queja
[0],$repos[0],$terminado,$id,1);
for($i=0;$i<count($arr_nomb)-1;$i++)
{
ParticipantescomisonPeer::CrearParticipanTCom(null,$arr_nomb[$i],$arr_carg[$i],$codigo);
}
for($i=0;$i<count($arr_doc_cod)-1;$i++)
DocumentosPeer::CrearDoc($arr_doc_cod[$i],$arr_doc_des[$i],$arr_doc_fec[$i],$codigo);
$this->redirect('SIC0959/VisualizarSIC0959?enviar='.$codigo.'&enviar1='.$id);
(Ver anexo 20)
}
}

```

2.6 Tratamiento de Errores

La aplicación tiene tratamientos de errores, pues los datos que van a ser introducidos por el usuario serán validados en el lado del servidor mediante archivos .yml vinculados a los formularios de las interfaces. Una vez que es detectado un error, ya sea un dato escrito de forma errónea, o algún dato

que no fue introducido, se lanzará un mensaje de error, indicándole cuál es el dato que está mal o que falta y dirigirá al usuario al campo en el cuál está el error, el usuario podrá seguir introduciendo los datos en el formulario. Además una vez que se hayan escrito correctamente en el formulario todos los datos se enviarán finalmente a la base de datos para ser guardados, contribuyendo de esta forma a que no existan errores en la base de datos.

En este módulo se validó la entrada de datos. Aquí se muestra a continuación varios ejemplos de tratamiento de errores del módulo SIC0919:

```

$this->list_folios=$ver_fol;
$this->list_cod=$ver_cod;

$folio =$this->getRequestParameter('select_fol');
$codigo=$this->getRequestParameter('select_cod');
$anno=$this->getRequestParameter('select_anno');
if($folio==0 && $codigo==0 && $anno==0)
{
    $this->ver=2;
    $this->mensaje='Introducir al menos un parámetro';
}
else
{

```

En este ejemplo se valida el BuscarSIC0919.

En este fragmento de código se capturan los datos entrados en el formulario con la función `getRequestParameter` ('id_campo') después se verifica si al menos se entró un parámetro de búsqueda, si esta condición no se cumple se lanza un mensaje: "Introducir al menos un parámetro" el cuál se muestra en la interfaz (Ver Anexo 21) y si se cumple prosigue con la búsqueda.

```

}
if($mensaje!='')
echo '<div align="center" class="Estilo1 Estilo6">'.$mensaje.'</div>';
if(count($ver_arreglo)==0)
echo
'<div align="center" class="Estilo1 Estilo6"> No se han encontrados registros
asociados a los criterios de búsqueda indicados</div>';
}

```

Después de haber entrado los parámetros de búsqueda y a haberse realizado, se envía el arreglo `$ver_arreglo` con los datos encontrados para visualizar al `BuscarSIC0919Success.php`. En este segmento de código se valida si el arreglo viene vacío, si esto se cumple se muestra el mensaje que aparece en dicha página. (Ver Anexo 22). Sino muestra los datos del arreglo.

```

{
    if ($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $this->mensaje='';
        return sfView::SUCCESS;
    }
else
    {
        $folio=$this->getRequestParameter('crear_fol');
        if(RegistroidevolucionesPeer::ExisteFolio($folio))
        {
            $this->mensaje='Ese Folio Ya Existe';
            return sfView::SUCCESS;
        }
    }
}

```

En este fragmento de código después de haber pasado un folio, se verifica con la función ExisteFolio() si el mismo ya está en la Base de Datos, de ser esto verdadero se envía un mensaje(Ver Anexo 23). Sino se adicionan los parámetros a la Base de Datos.

```

<script>
function save()
{
    if(NoAste('mes') && MayorC('mes') && SoloNumeros('desviacion') && NoVacio('desviacion'))
    {
        var mes= document.getElementById('mes');
        var meses= document.getElementById('meses');
        meses.value= meses.value + mes.value + '*';
        mes.value='0';

        var desv= document.getElementById('desviacion');
        var desviacion= document.getElementById('desv');
        desviacion.value= desviacion.value + desv.value + '*';
        desv.value='';

    }
    else
    {
        if(!MayorC('mes'))
        alert('Entre un mes');
        if(!SoloNumeros('desviacion'))
        alert('Entre numeros en desviacion');
    }
}
}

```

En este ejemplo hay campos que hay que adicionarlos muchas veces sin actualizar la página, por tal motivo se utiliza Java Script porque este adiciona estos datos a un campo oculto y resuelve dicha

situación, aquí también se validan los campos ya que si no entra los datos correctamente se lanza un mensaje. (Ver Anexo 24).

2.7 Conclusiones

En este capítulo fueron mostrados una serie de artefactos dígase, diagramas de componentes referentes al flujo de trabajo de implementación, específicamente se realizaron 9, anteriormente se exponen estos artefactos. Además se muestra un número de funciones desarrolladas para la implementación propuesta, siendo estas las de mayor importancia para el funcionamiento de la aplicación. Simultáneamente se pone parte del código Java Script utilizado para la validación de los campos, a la hora de pasar los datos de cada SIC.

CONCLUSIONES

- Como resultado de la investigación realizada se puede concluir que se cumplió el objetivo propuesto. Se implementaron las clases para el módulo de la SMC según el framework seleccionado para su desarrollo (Symfony). Las herramientas y metodologías definidas resultaron útiles para el desarrollo del trabajo y la utilización de la arquitectura MVC facilitó la organización para la implementación.

RECOMENDACIONES

- Trabajar en la seguridad del Módulo.

REFERENCIA BIBLIOGRÁFICA

1. **Cuevas, Olga.** Onlabweb. *Onlabweb*. [Online] diciembre 6, 2007. [Cited: diciembre 25, 2007.] <http://www.onlabweb.com/lims.htm>.
2. **CIGB.** Centro de Ingeniería Genética y Biotecnología. *Centro de Ingeniería Genética y Biotecnología*. [Online] 2003. [Cited: diciembre 25, 2007.] <http://www.cigb.edu.cu/pages/ccalidad.htm>.
3. —. Centro de Ingeniería Genética y Biotecnología. *Centro de Ingeniería Genética y Biotecnología*. [Online] 2003. [Cited: diciembre 25, 2007.] <http://www.cigb.edu.cu/pages/acalidad.htm>.
4. **Medilab.** Sistemas de Información para el Manejo de Laboratorios Médicos Eficientes. *Sistemas de Información para el Manejo de Laboratorios Médicos Eficientes*. [Online] [Cited: enero 5, 2008.] <http://medilabit.com/servicios.htm>.
5. **LabWare.** Interfases con otros sistemas. *Interfases con otros sistemas*. [Online] 2005. [Cited: enero 5, 2008.] <http://www.labware.com/LWWebES.nsf/lp/es0407>.
6. **Departamento de Ingeniería y Gestión de Software, Universidad de las Ciencias Informáticas.** *INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE*. s.l. : Conferencia, 2007-2008.
7. **Vizcaíno, Aurora.** Prácticas Ingeniería del Software. *Prácticas Ingeniería del Software*. [Online] [Cited: febrero 1, 2008.] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf.
8. **Valdelli, Ilario.** Curso JavaScript. [Online] 2006. [Cited: febrero 2008, 1.] <http://www.htmlpoint.com/javascript/corso/index.html>.
9. **Potencier, Fabien.** Symfony, la Guía Definitiva. *Symfony, la Guía Definitiva*. [Online] [Cited: febrero 4, 2008.] <http://www.librosweb.es/symfony/>.
10. **Pecos, Daniel.** PostGreSQL vs. MySQL. *PostGreSQL vs. MySQL*. [Online] [Cited: febrero 4, 2008.]
11. **Vilas, Ana Fernández.** Diagrama de Componentes. [Online] [Cited: febrero 4, 2008.] <http://www-gris.det.uvigo.es/~avilas/UML/node49.html>.

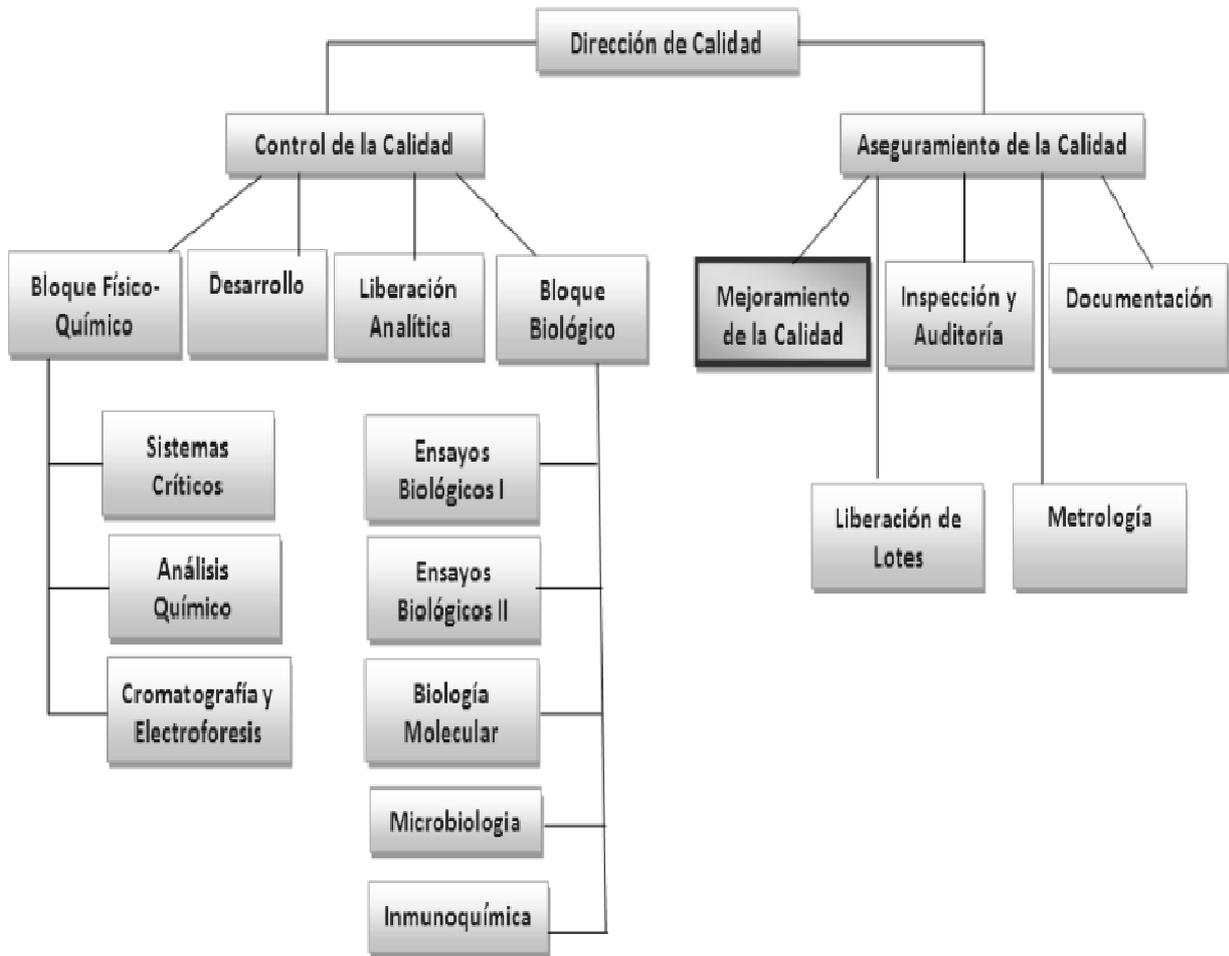
BIBLIOGRAFÍA

- **CIGB. Centro de Ingeniería Genética y Biotecnología. Centro de Ingeniería Genética y Biotecnología.** [Online] 2003. [Cited: noviembre 7, 2007.] <http://www.cigb.edu.cu/pages/ccalidad.htm>.
- **.CIGB. Departamento de Aseguramiento de Calidad.** Departamento de Aseguramiento de Calidad. [Online] 2003. [Cited: noviembre 7, 2007.] <http://www.cigb.edu.cu/pages/acalidad.htm>.
- **CIGB.** Centro de ingeniería genética y biotecnológica. [Online] 2003. [Cited: noviembre 11, 2007.] <http://www.cigb.edu.cu/pages/calidad.htm> .
- **Cuevas, Olga.** Infolab. Infolab. [Online] [Cited: noviembre 7, 2007.] <http://www.onlabweb.com/lims.htm>.
- **Conallen, Jim.** *UML Extension for Web Applications*. Marzo 1999 [Cited: noviembre 7, 2007.]
- **EMS SQL Manager for PostgreSQL.** SQL Manager.net. SQL Manager.net. [Online] 2007. [Cited: noviembre 10, 2007.] <http://www.sqlmanager.net/products/postgresql/manager>.
- **Fernández, Ana.** Diagrama de Componentes. Diagrama de Componentes. [Online] marzo 20, 2001. [Cited: abril 5, 2008.] <http://www-gris.det.uvigo.es/~avilas/UML/node49.html>.
- **INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE.** Departamento de Ingeniería y Gestión de Software, Universidad de las Ciencias Informáticas. 2007-2008. Conferencia. [Cited: marzo 10, 2008.]
- **Lago, Ramiro.** [Online] enero 2006. [Cited: marzo 10, 2008.] <http://www.proactiva-calidad.com/java/herramientas/eclipse/introduccion.html>.
- **. Markiewicz, Marcus eduardo.** El Desarrollo del Framework Orientado al Objeto. El Desarrollo del Framework Orientado al Objeto. [Online] [Cited: enero 15, 2008.] <http://www.acm.org/crossroads/espanol/xrds7-4/frameworks.html>.
- **Nieto, Elennis Diaz y Liusmila.** LIMS DE CALIDAD DEL CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA: ANÁLISIS DE LA SECCIÓN DE MEJORAMIENTO DE LA CALIDAD Y DEL GRUPO DE DESARROLLO. De la Habana : s.n., 2007.
- **Onlaweb.** Onlaweb. [Online] diciembre 6, 2007. [Cited: noviembre 14, 2007.] <http://www.onlabweb.com/lims.htm>.
- **Paradigma MVC.** [En línea] <http://aldeacafe.com.mx/programacion/patrones/mvc.html> [Cited: noviembre 20, 2007.]
- **Pecos, Daniel.** PostGreSQL vs. MySQL. PostGreSQL vs. MySQL. [Online] [Cited: noviembre 20, 2007.] http://www.netpecos.org/docs/mysql_postgres/.

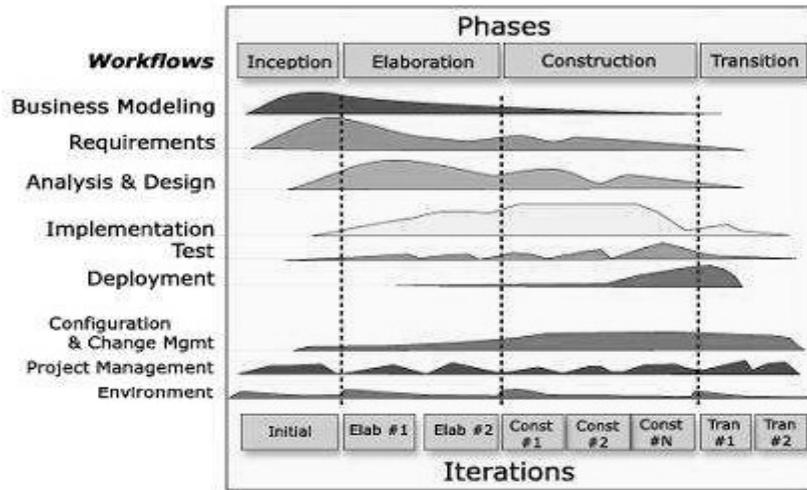
- **Programación en castellano.** Programación en castellano. [Online] [Cited: mayo 5, 2008.] <http://www.programacion.net/>.
- **Rational unified process.** [Online] [Cited: diciembre 12, 2007.] **Rational unified process.**
- **Symfony, la guía definitiva.** [Online] [Cited: enero 15, 2008.] http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html.
- **Universidad De Valencia.** Rational Unified Process (RUP). Rational Unified Process (RUP). [Online] [Cited: noviembre 14, 2007.]
- **Visual Parading para UML.** Visual Parading para UML. [Online] marzo 5, 2007. [Cited: enero 10, 2008.] <http://www.freedownloadmanager.org/es/downloads>
- **Visual Paradigm.** Visual Paradigm. [Online] [Cited: mayo 5, 2008.] <http://www.visual-paradigm.com/product/vpuml/>.

ANEXOS

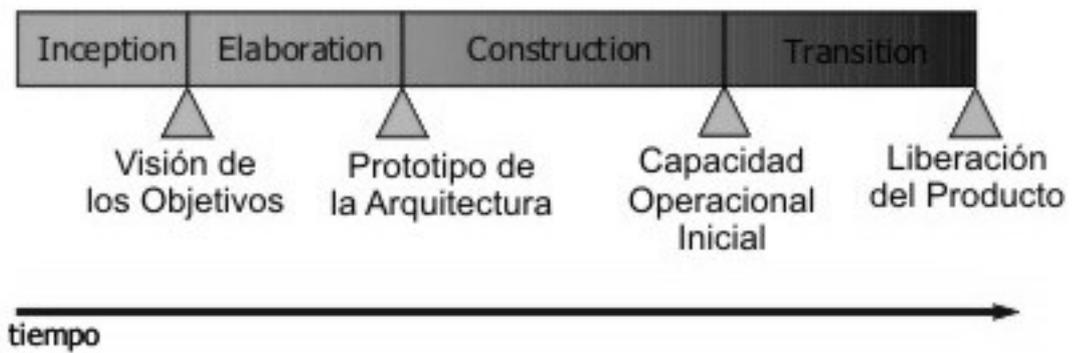
Anexo 1. Organigrama del Departamento de Calidad del CIGB.



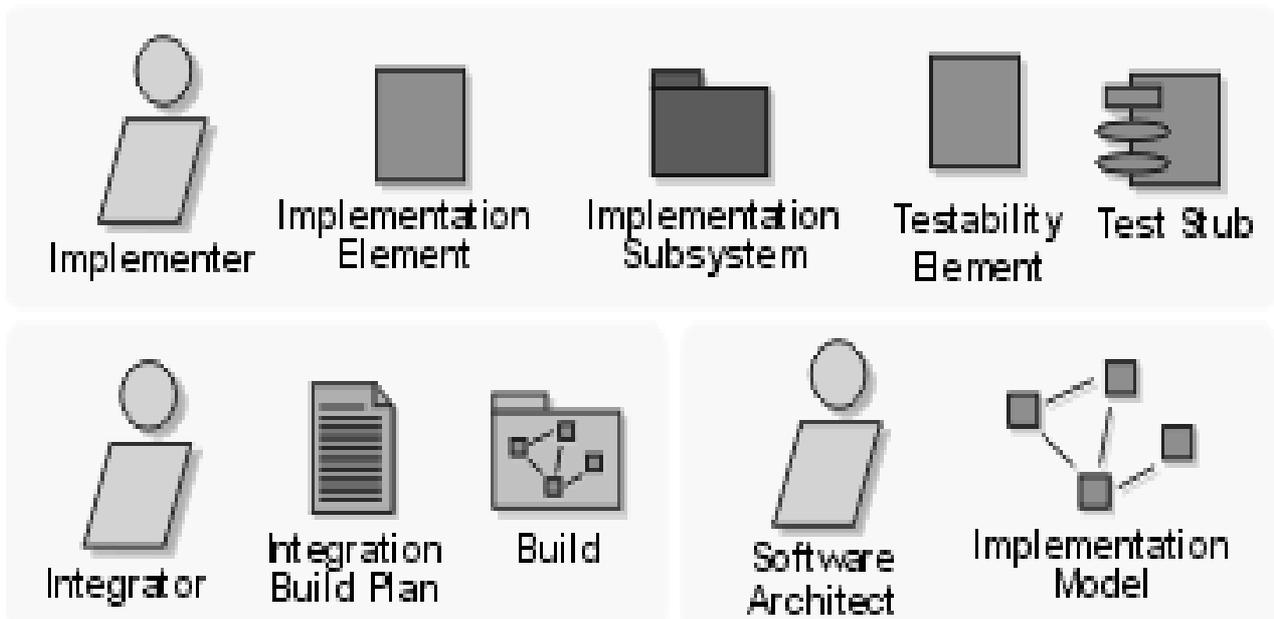
Anexo 2. Fases de RUP.



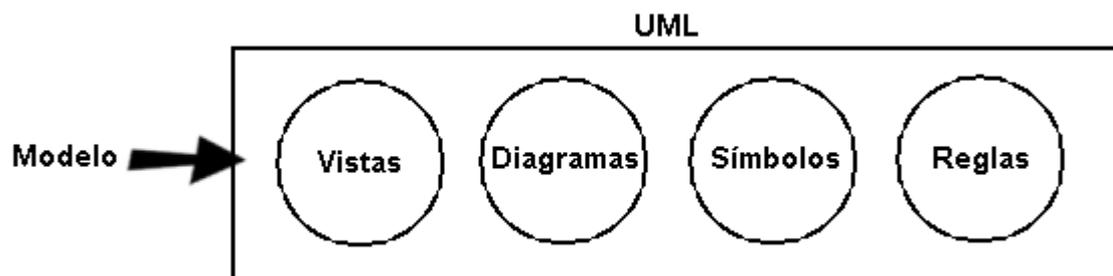
Anexo 3. Hitos de las Fases.



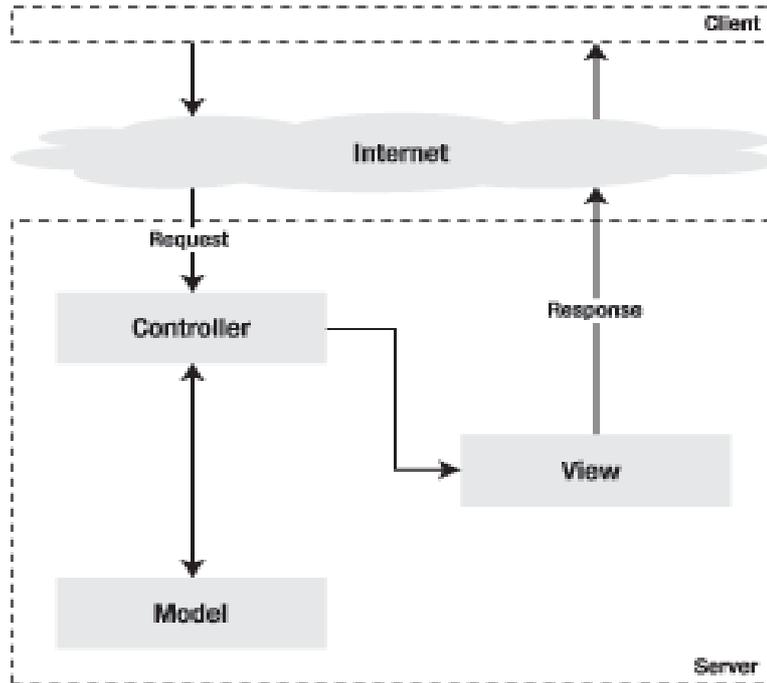
Anexo 4. Artefactos a realizar.



Anexo 5. Representación gráfica del Lenguaje de Modelado.



Anexo 6. Representación gráfica del MVC.



Anexo 7. Representación gráfica del prototipo Crear del SIC 0023.

http://localhost:5901/CIGB_dev.php/SIC0023/crearSIC0023

Go Bookmarks 2 blocked Check AutoLink AutoFill Send to

Sf 1.0.11 vars & config logs & msgs 3942.6 KB 1384 n

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA		SIC-0023	PPO 4.10.124.97
DIRECCIÓN DE CALIDAD		Edición 04	FOLIO: 9
REGISTRO DE DEVOLUCIONES		Fecha de notificación: 2008-05-03	
Nombre del producto:	nombre	Cantidad de producto:	2
Código de la devolución:	5	Concentración/ Dosis:	4
No. de lote (s):	1	Forma farmacéutica:	forma
Descripción del problema: descrip problema			
Lotes		Cantidad (Inspección CIGB)(bbs)	
34		89	
Add ver			
DOCUMENTOS (Código, descripción y ubicación)			
documentos			
Add ver			
PARTICIPANTES EN LA COMISIÓN			
Nombre		Cargo	
nombre		cargo	
Add ver			
Conclusiones de la investigación:			
conclusiones			
Decisión: decision			
Realizado por: realizado		Firma:	
Cargo: cargo		Fecha: 2008-05-03	
Revisado por: revisado		Firma:	
Cargo: cargo		Fecha: 2008-05-03	

Terminado

Crear Cancelar

Anexo 8. Representación gráfica del prototipo Buscar del SIC 0098.

http://localhost:5901/CIGB_dev.php/SIC0098/BuscarSIC0098

Google G

Go

Bookmarks

2 blocked

Check

AutoLink

AutoFill

Send to

Sf 1.0.11 vars & config logs & msgs 4 4749.9 KB

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA
DIRECCIÓN DE CALIDAD
Búsqueda del Programa de Monitoreo Ambiental

Folio: 1 Planta: 2 Clase: Select....

Buscar Cancelar

Zona de Trabajo	Área	NP	
1	2	1	Visualizar

Anexo 9. Representación gráfica del prototipo Visualizar Reporte del SIC0098.

http://localhost:5901/CIGB_dev.php/SIC0098/VisualizarReporteSIC0098

G

Go

Bookmarks

2 blocked

Check

AutoLink

AutoFill

Send to

Sf 1.0.11 vars & config logs & msgs 5 518

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA
DIRECCIÓN DE CALIDAD
Programa de Monitoreo Ambiental
Producto: sd

Clase	IP	Area	Z. de Trabajo		
sd	sd	sd	s		
Metodo	Caracteristicas	Responsable	Frecuencia	Int	
sd	sd	34	sd	43	
4	5	45	45	45	
55	555	5	5	5	
22	22	2	2	2	
22	22	2	2	2	

Imprimir

Anexo 10. Representación gráfica del prototipo Crear del SIC 0816

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA DIRECCIÓN DE CALIDAD	SIC-0816	PPO 4.10.096.98
	Edición 03	
Registro de Control del Cambio		Folio: 98

Folio entrado correctamente

Anexo 11. Representación gráfica del prototipo Registrar del SIC 0816.

Registrar datos en el Registro de Control de Cambios

Folio: 98

Código/Nombre del Cambio	Fecha de Recepción	Clasificación		En Revisión	Pendiente de documentación	En Reg. y Reg.	Decisión Final	Fecha de conclusión
		Me.	Ma.					
codigo	2005-05-08	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	decisión	2005-05-08

Observaciones:
gtgft

Ese Codigo Ya Existe

Anexo 12. Representación gráfica del prototipo Visualizar del SIC 0919.

http://localhost:5901/CIGB_dev.php/SIC0919/VisualizarSIC0919/enviar/222

Go Bookmarks 2 blocked Check AutoLink AutoFill Send to

Sf 1.0.11 vars & config logs & msgs 3 4824.5 KB 1596

CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA DIRECCION DE CALIDAD		SIC-0919	PPO 4.10.056.93
		Edición 01	Folio: 222
PROGRAMA DE INFORMES DE TENDENCIAS		Año: 1994	Pág 1 de 1
Desviaciones del Programa			
Mes	Desviación	Firma del Jefe de Grupo	
1	1		
6	111		
12	12		

Imprimir Modificar

Anexo 13. Representación gráfica del prototipo Visualizar Reporte del SIC 0919.

http://localhost:5901/CIGB_dev.php/SIC0919/VisualizarReporteSIC0919

Go Bookmarks 2 blocked Check AutoLink AutoFill Send to

Sf 1.0.11 vars & config logs & msgs 3 4840.5 KB 1103

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA DIRECCIÓN DE CALIDAD Programa de Informes de Tendencias	
Mes	Desviacion
12	12
12	12
1	1
1	1
6	111
12	12
Año	
1994	
1994	

Imprimir

Anexo 14. Representación gráfica del prototipo Registrar del SIC 0963.



Registrar datos en el Reporte de Limites

Folio: 444

Periodo de Revision: 4444

Planta	Fecha M.	Area	Local	Características	PM	Nombre	L.AI (valor)	L.Ac (valor)	
planta	2008-05-03	area	local	racteristicas	pm	nombre	9	9	
								Add	ver
Observaciones:									
observaciones									
	NOMBRE Y APELLIDOS			FIRMA		FECHA			
Realizado por:	realizado			-		2008-05-03			
Revisado por:	revisado			-		2008-05-03			

Terminado

Registrar

Anexo 15. Representación gráfica del prototipo Modificar del SIC 0963.

Dirección http://localhost:5901/CIGB_dev.php/SIC0963/ModificarSIC0963/codigo/44 Ir

Google Go 2 blocked

CENTRO DE INGENIERERÍA GENÉTICA Y BIOTECNOLOGÍA						SIC-0963	PPO 4.10.140.03	
DIRECCIÓN DE CALIDAD						Edición 01		
Reportes de resultados fuera de los límites establecidos, detectados en las bases de datos del PMA							Pag 1 de 1	
Periodo Revisado: 44							Folio: 44	
Planta	Fecha M.	Área	Local	Características	PM	Nombre	L.Ai (valor)	L.Ac (valor)
11	2008-05-03 <input type="text"/>	11	11	1 <input type="text"/>	1	11 <input type="text"/>	11 <input type="text"/>	11 <input type="text"/>
11	2005-05-08 <input type="text"/>	11	11	1 <input type="text"/>	11	1 <input type="text"/>	11 <input type="text"/>	11 <input type="text"/>
22	2005-05-08 <input type="text"/>	22	22	1 <input type="text"/>	22	22 <input type="text"/>	22 <input type="text"/>	22 <input type="text"/>
33	2005-05-08 <input type="text"/>	33	33	1 <input type="text"/>	33	33 <input type="text"/>	33 <input type="text"/>	33 <input type="text"/>
1	2008-05-03 <input type="text"/>	1	1	1 <input type="text"/>	1	1 <input type="text"/>	1 <input type="text"/>	1 <input type="text"/>
								<input type="button" value="Modificar"/>
	2008-05-03 <input type="text"/>			<input type="text"/>		<input type="text"/>	<input type="text"/>	<input type="text"/>
								<input type="button" value="Add"/> <input type="button" value="ver"/>
Observaciones:								
observaciones <input type="text"/>								
	NOMBRE Y APELLIDOS			FIRMA		FECHA		
Realizado por:	<input type="text" value="realizado"/>			-----		2008-05-03 <input type="text"/>		
Revisado por:	<input type="text" value="revisado"/>			-----		2008-05-03 <input type="text"/>		

Terminado

Anexo 16. Representación gráfica del prototipo Visualizar del SIC 0851.

http://localhost:5901/CIGB_dev.php/SIC0851/visualizarSIC0851/enviar/producto1

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA
DIRECCIÓN DE CALIDAD

SIC-0851 **PPO 4.10.122.96**
Edición 03 **Folio: 12**

REGISTRO DE REPORTE DE DESVIACIONES

Producto:	producto1			
Código SIC-0097	Fecha de Notificación	Etapa	No. lote	Descripción de la Desviación
codigo	2008-05-28	etapa	no_lote	ident

Modificar Imprimir

Anexo 17. Representación gráfica del prototipo Visualizar Reporte del SIC 0851.

http://localhost:5901/CIGB_dev.php/SIC0851/visualizarReporteSIC0851

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA
DIRECCIÓN DE CALIDAD
Registro de Reportes de Desviaciones

Reporte del Registro de Reportes de Desviaciones

Codigo/SIC0097	Desc. de la Desv	Producto
codigo	ident	producto1
codigo	ident	prod1

Imprimir

Anexo 18. Representación gráfica del prototipo Crear del SIC 0042.

http://localhost:5 Detener dev.php/SIC0042/crearSIC0042 Ir

Go 2 blocked Check AutoLink AutoFill Send to

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA		SIC-0042	PPO 4.10.008.91
DIRECCIÓN DE ASEGURAMIENTO DE LA CALIDAD		Edición 08	Pág. 1 de 1
NOTIFICACIÓN DE QUEJA			
Nombre del producto	nombre		
Forma farmacéutica	forma	Fecha de notificación:	2008-05-03 ...
Presentación	presentacion		
Lotes involucrados	lotes		
Vía de comunicación por la que se recibió	via		
Persona que recibe la comunicación	persona		
Cargo	cargo		
DATOS DEL DEMANDANTE			
Persona que comunica	persona		
Cargo	cargo		
Institución	institucion		
Dirección	direccion		
Vías de contacto	vias		
DESCRIPCIÓN DEL PROBLEMA			
descripcion			
Código propuesto:	1		
Realizado por:	REALIZADO	Firma:	
Cargo:	cargo	Fecha:	2008-05-03 ...
Revisado por:	revisado	Firma:	
Cargo:	cargo	Fecha:	2008-05-03 ...

Terminado

Anexo 19. Representación gráfica del prototipo Crear del SIC 0959.

http://localhost:5901/CIGB_dev.php/SIC0959/crearSIC0959

Go Bookmarks 2 blocked Check AutoLink AutoFill Send to Ir

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA		SIC-0959	PPO 4.10.008.91
DIRECCIÓN DE ASEGURAMIENTO DE LA CALIDAD		Edición 01	Pág. 1 de 1
CIERRE DE QUEJA			
Nombre del producto:	nom prod	Código de la queja:	333
Reportado anteriormente:	Si <input checked="" type="radio"/> No <input type="radio"/>		Select...
Código de la queja anterior:	CÓDIGO		4
Documentos			333
Código	Descripción		76
código	descr	20	34
			2
			7
			222
			Add ver
Participantes en la comisión			
Nombre	Cargo		
nombre	cargo		
			Add ver
CONCLUSIONES DE LA INVESTIGACIÓN			
conclusiones			
La queja PROCEDE?: Si <input checked="" type="radio"/> No <input type="radio"/>		La reposición de producto PROCEDE?: Si <input type="radio"/> No <input checked="" type="radio"/>	
Realizado por:	realizado	Firma:	
Cargo:	cargo	Fecha:	2008-06-10
Revisado por :	revisado	Firma:	
Cargo :	cargo	Fecha:	2008-06-20
Aprobado por :	aprobado	Firma:	
Cargo :	cargo	Fecha:	2008-06-28

Terminado

Crear Cancelar

Anexo 20. Representación gráfica del prototipo Modificar del SIC 0959.



CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA		SIC-0959	PPO 4.10.008.91
DIRECCIÓN DE ASEGURAMIENTO DE LA CALIDAD		Edición 01	Pág. 1 de 1
CIERRE DE QUEJA			
Nombre del producto:	nom prod	Código de la queja:	333
Reportado anteriormente:	SI		
Código de la queja anterior: CÓDIGO			
Documentos			
Código	Descripción	Fecha	
ee	eee	2008-06-09 ...	
			Modificar
			...
			Add ver
Participantes en la comisión			
Nombre		Cargo	
ee		eee	
			Modificar
			...
			Add ver
CONCLUSIONES DE LA INVESTIGACIÓN			
<div style="border: 1px solid black; padding: 5px; min-height: 40px;"> conclusiones </div>			
La queja PROCEDE?: Si <input checked="" type="radio"/> No <input type="radio"/>		La reposición de producto PROCEDE?: Si <input type="radio"/> No <input checked="" type="radio"/>	
Realizado por:	realizado	Firma:	-----
Cargo:		Fecha:	2008-06-10 ...
Revisado por :	revisado	Firma:	-----
Cargo :		Fecha:	2008-06-20 ...
Aprobado por :	aprobado	Firma:	-----
Cargo :		Fecha:	2008-06-28 ...

Terminado:
Modificar
Imprimir

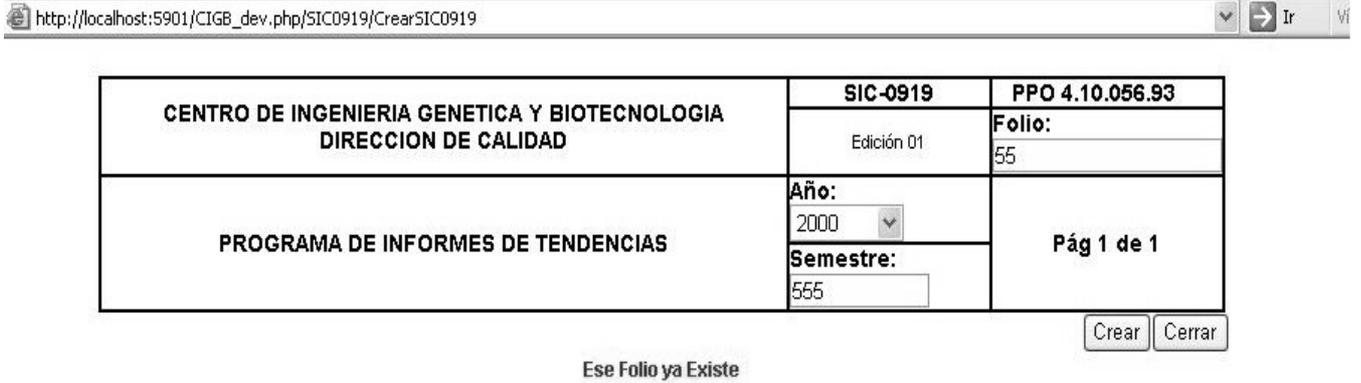
Anexo 21. Representación gráfica de la validación de la página BuscarSIC0919 (1).

The screenshot shows a web browser window with the URL `http://localhost:5901/CIGB_dev.php/SIC0919/BuscarSIC0919`. The page title is "CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA DIRECCIÓN DE CALIDAD Búsqueda del Programa de Informes de Tendencias". The search form contains three dropdown menus: "Folio:" (Select...), "Mes:" (Selec...), and "Año:" (Select...). There are "Buscar" and "Cerrar" buttons. Below the form, a message states "Debe introducir al menos un parámetro para realizar la búsqueda". At the bottom, there is a table with columns "Año", "Folio", and "Semestre".

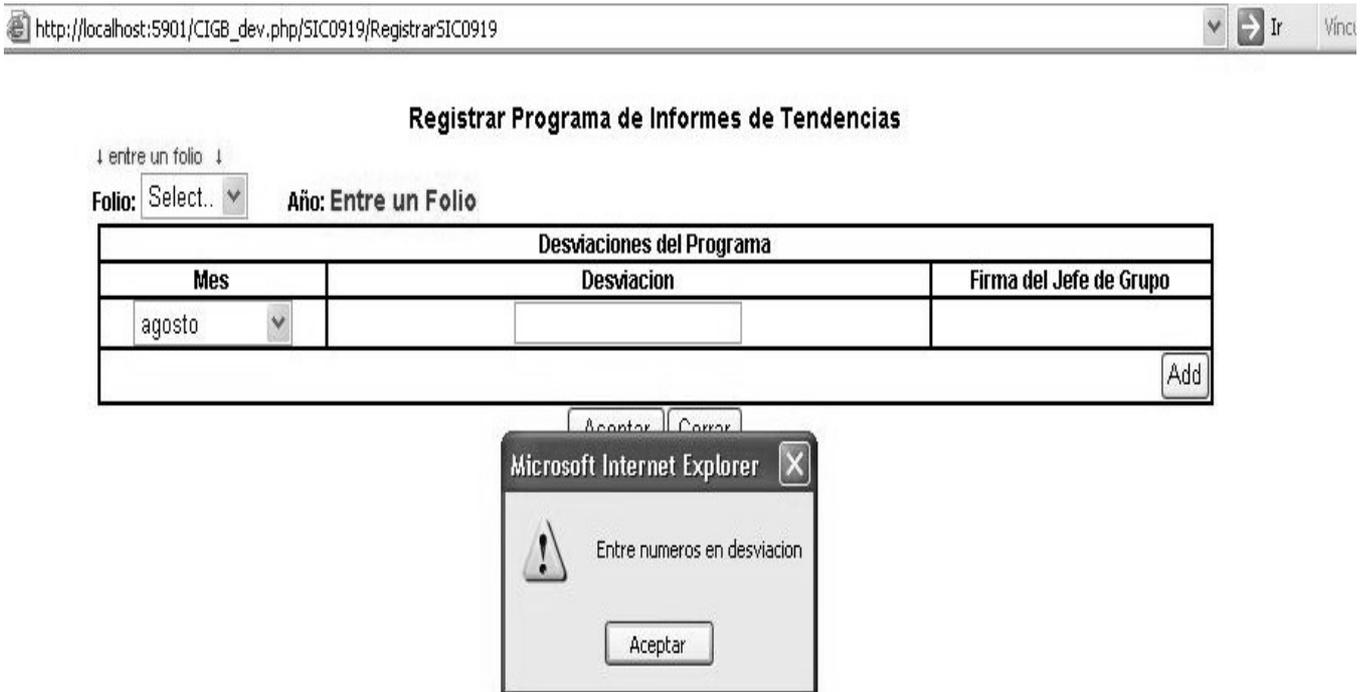
Anexo 22. Representación gráfica de la validación de la página BuscarSIC0919 (2).

The screenshot shows the same web browser window as in Anexo 21. The search form now has "Folio:" set to "888" and "Mes:" set to "noviembre". The "Año:" dropdown remains "Select...". The "Buscar" and "Cerrar" buttons are present. Below the form, a message states "No se han encontrados registros asociados a los criterios de búsqueda indicados". At the bottom, there is a table with columns "Año", "Folio", and "Semestre".

Anexo 23. Representación gráfica de la validación de la página CrearSIC0919.



Anexo 24. Representación gráfica de la validación de la página RegistrarSIC0919.



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 calidad.

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.

Biotecnología: conocimientos y avances biológicos aplicados a procesos tecnológicos o de interés industrial.

Biomedicina: medicina clínica basada en los principios de las ciencias naturales.

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

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

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

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

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

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

SIC: Sistema Informativo de Calidad.

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