

Universidad de las Ciencias Informáticas

Facultad 6



Título: Sistema de Gestión de la Información de los Laboratorios de la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología: Implementación del Módulo de Microbiología

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN
CIENCIAS INFORMÁTICAS**

Autor(es): Manuel Marrero Marcané.

Pedro Luis del Campo Gutiérrez.

Tutor: Ing. Alieski Sarmiento Almenares.

Co-Tutora: Lic. Miulkenia Navarro Reyes.

Ciudad de La Habana, Cuba

“Año 50 de la Revolución”

Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte.

Ernesto Guevara de la Serna

DECLARACIÓN DE AUTORÍA

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

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

Manuel Marrero Marcané

Pedro Luis del Campo Gutiérrez

Firma del Autor

Firma del Autor

Ing. Alieski Sarmiento Almenares.

Lic. Miulkenia Navarro Reyes.

Firma del Tutor

Firma de la Co-Tutora

DATOS DE CONTACTO

Tutor:

Ing. Alieski Sarmiento Almenares.

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

asarmiento@uci.cu

Co-Tutora:

Lic. Miulkenia Navarro Reyes.

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

mnavarror@uci.cu

AGRADECIMIENTOS

A nuestros seres más queridos, en especial a nuestros padres, por estar siempre ahí cuando más los necesitamos y por ayudarnos a hacer este sueño realidad.

A nuestros amigos que nos ayudaron cuando el camino se tornó difícil.

A nuestros compañeros de proyecto, por la ayuda brindada para la realización de este trabajo.

A todo aquel que influyó en nuestra formación como profesional, durante este período de estudiante.

A Fidel Castro Ruz por ser el guía impulsor de este sueño hecho realidad y del cual formamos parte.

Agradecer a la Revolución Cubana por la posibilidad de hacer realidad nuestro sueño de convertirnos en ingenieros.

A esta Universidad por todas las enseñanzas, experiencias y recuerdos inolvidables, que quedarán para siempre guardados en nuestra memoria.

A todos, Muchas Gracias...

DEDICATORIA

Pedro Luis del Campo Gutiérrez.

Dedico este trabajo a una mujer especial, la cual ha sido lo más importante en mi vida, a la que le debo lo que soy, por su amor y entrega incondicional; a mi madre, Isaida Andrea Gutiérrez Ruiz. Gracias por existir.

A mi abuela Efrén Ruiz Henández y a mi tía Idania Gutiérrez Ruiz, por formar parte de mi vida y estar siempre allí para mí.

A mi padre Pedro Luis del Campo Gómez, por formar parte de mi vida.

A mis otros dos papas, Jorge Luis Hernández Mejías y José Couzo Medina, por ayudarme en todo lo que ha estado a su alcance y contribuir en mi formación.

A mi prima Beatriz Muñoz Gutiérrez, ojalá y sirva como guía para su vida.

A Virginia por ser la madre que me falta aquí en La Habana y a su familia.

A todos mis amigos de la UCI por estar allí cuando los necesito, especialmente a Yanella Fales Valle, Sonia González del Sol, Elenis Díaz Laurencio, Lisdany De la Fuente Díaz, Yanelis Ramírez Hernández, Juan Manuel Ruiz Godoy, Manuel Marrero Marcané, Alain Meneses Jimenez, Lisbeth León, Ariskien Mendoza y Liyanis Dube Argote.

A todo aquel que ha contribuido en mi formación, como ser humano a lo largo de estos años.

Gracias a todos.

DEDICATORIA

Manuel Marrero Marcané.

Dedico este trabajo realizado, a mis padres queridos, Manuel y Aleida. Que aunque no están aquí presentes sus cuerpos; sus deseos están conmigo augurándome mucha suerte. Ellos con su amor, su dedicación, su firmeza; han sido guía y motor impulsor para que yo pusiera todo mi empeño en llevar adelante este sueño.

A mi novia, Leydí Morales, por estar presente cuando el camino se tornó difícil, darme aliento a seguir adelante y hacerme crecer ante las dificultades.

Gracias a todos.

RESUMEN

Se desarrolla un Sistema para la Gestión de la Información de los Laboratorios (LIMS) para la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología (CIGB), el mismo consta de ocho módulos, dentro de los cuales se encuentra el módulo de Microbiología. En el presente trabajo de diploma se aborda el proceso de implementación de dicho módulo, partiendo de un diseño previo realizado en tesis precedentes, siguiendo la metodología RUP, a través del Rol de Implementador que propone la misma y haciendo uso del framework Symfony, herramienta de actualidad para la programación web, el Sistema Gestor de Bases de Datos (SGBD) PostgreSQL y la herramienta CASE de modelado Visual Paradigm. Se implementa el módulo de Microbiología, debido a que es necesidad del Polo Científico, el uso de esta herramienta para mejorar su desempeño, optimizar y reducir el tiempo a la hora de procesar la información de los laboratorios del Centro.

PALABRAS CLAVES

LIMS.

CIGB.

Módulo.

Microbiología.

Implementación.

Framework.

ÍNDICE

AGRADECIMIENTOS	I
DEDICATORIA.....	II
DEDICATORIA.....	III
RESUMEN.....	IV
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 LIMS EXISTENTES EN EL MERCADO.	6
1.2 TECNOLOGÍAS DE DESARROLLO Y HERRAMIENTAS UTILIZADAS.....	7
1.2.1 Proceso Unificado de Desarrollo de Software (RUP, Rational Unified Process).....	8
1.2.2 Herramienta CASE	12
1.2.3 Lenguajes de programación	13
1.2.4 Framework.	16
1.2.5 Sistema Gestor de Bases de Datos PostgreSQL (Versión: 8.2).....	17
1.2.6 Servidor Web Apache (Versión: 2.2.6).	17
1.2.7 Entorno de Desarrollo Integrado (IDE) Eclipse (Versión: 3.3.1.1).	18
CONCLUSIONES.....	18
CAPÍTULO 2: IMPLEMENTACIÓN DEL SISTEMA	19
2.1 REQUISITOS FUNCIONALES:.....	19
2.2 DESCRIPCIÓN DEL CASO DE USO.	21
2.3 REALIZACIÓN DE DIAGRAMAS DE COMPONENTES.	28
2.4 ESTRUCTURA DE LA ARQUITECTURA DE SYMFONY.....	41
2.5 CÓDIGO FUENTE.	41
2.5.1 Capa Control.	41
2.5.2 Capa Modelo.	54
2.5.3 Capa Vista.....	79
CONCLUSIONES.....	83
CONCLUSIONES GENERALES.....	84
RECOMENDACIONES.....	85
REFERENCIAS BIBLIOGRÁFICAS.....	86

BIBLIOGRAFÍA.....	88
ANEXOS.....	91
GLOSARIO DE TÉRMINOS.....	92

ÍNDICE DE FIGURAS

Figura 1. Iteración de RUP.	8
Figura 2. Fases e Iteraciones de la Metodología RUP.	9
Figura 3. Fases e Hitos en RUP.	10
Figura 4. Diagrama de componentes CU Gestionar SIC_0020.	29
Figura 5. Diagrama de componentes CU Gestionar SIC_0142.	30
Figura 6. Diagrama de componentes CU Gestionar SIC_0159.	31
Figura 7. Diagrama de componentes CU Gestionar SIC_0162.	32
Figura 8. Diagrama de componentes CU Gestionar SIC_0198.	33
Figura 9. Diagrama de componentes CU Gestionar SIC_0199.	34
Figura 10. Diagrama de componentes CU Gestionar SIC_0711.	35
Figura 11. Diagrama de componentes CU Gestionar SIC_0714.	36
Figura 12. Diagrama de componentes CU Gestionar SIC_0719.	37
Figura 13. Diagrama de componentes CU Gestionar SIC_0805.	38
Figura 14. Diagrama de componentes CU Gestionar SIC_0806.	39
Figura 15. Diagrama de componentes CU Gestionar SIC_0807.	40
Figura 16. Patrón MVC.	41
Figura 17. Interfaz visual de la función Crear del CU Gestionar SIC_0020.	45
Figura 18. Interfaz visual de la función Modificar del CU Gestionar SIC_0020.	49
Figura 19. Interfaz visual de la función Reporte del CU Gestionar SIC_0020.	50
Figura 20. Interfaz visual de la función Visualizar Reporte del CU Gestionar SIC_0020.	51
Figura 21. Interfaz visual de la función Visualizar del CU Gestionar SIC_0020.	52
Figura 22. Interfaz visual de la función Buscar del CU Gestionar SIC_0020.	54
Figura 23. Interfaz visual de validación de errores.	83
Figura 24. Estructura jerárquica del Área de Calidad del Centro de Ingeniería Genética y Biotecnología.	91
Figura 25. Trabajadores y artefactos.	91

INTRODUCCIÓN

En la actualidad las Tecnologías de la Información y las Comunicaciones (TIC) ocupan un lugar significativo en el proceso de desarrollo de la sociedad y la economía a nivel mundial. El concepto de las TIC nace con la convergencia tecnológica de la electrónica, el software y las infraestructuras de las telecomunicaciones. Las TIC proporcionan herramientas que ofrecen la posibilidad de encontrar soluciones novedosas ante los desafíos que se presentan a diario en la sociedad.

Debido al auge que ha tenido la implantación y utilización de las TIC en todo el mundo, se presentan como una necesidad para el desarrollo económico y social de cualquier país. Por lo que nuestro país no ha estado ajeno a esta evolución de la informática y las comunicaciones que se ha llevado a cabo desde la segunda mitad del pasado siglo.

En Cuba se ha determinado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las TIC, así como lograr una cultura digital como una de las características imprescindibles de la sociedad. El proyecto de desarrollo económico y social llevado a cabo por el pueblo cubano, lo sitúa en una situación ventajosa para emprender el reto de la informatización y el tránsito hacia una sociedad basada en el conocimiento. Cuba está consciente de que una sociedad para ser más eficaz, eficiente y competitiva debe aplicar la informatización en todas sus esferas y procesos.

A partir de la década del 80 del pasado siglo, tuvo lugar en Cuba una explosión en la actividad científica en el campo de la salud, llevándose a cabo un intenso proceso inversionista y de formación de personal.

A consecuencia de esto en el año 1992 surgió en la capital cubana, el Polo Científico al oeste de La Habana, el cual marcó la mayoría de edad del pujante movimiento científico de la Isla dentro de este se encuentra enmarcado el Centro de Ingeniería Genética y Biotecnología (CIGB), colofón de tal proceso; el cual fue inaugurado el 1ro de julio de 1986.

La labor desplegada por el personal científico que labora en ese Centro ha tenido un gran impacto en la biomedicina, salud animal, mejoramiento vegetal y la bioindustria. También está encaminada a contribuir de manera significativa con el desarrollo económico y social del país. Durante estos 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 de los productos elaborados allí, es la imagen que proyecta el centro para con sus clientes. Los productos desarrollados y elaborados se caracterizan por su eficacia y seguridad, todo esto se logra gracias a la labor desempeñada por la Dirección de Calidad del centro, la cual realiza sus

funciones a través del Departamento de Control de la Calidad (DCC) de conjunto con el Departamento de Aseguramiento de la Calidad (DAC). (Ver Anexo. 1)

El DCC ejecuta varias funciones, dentro de las cuales podemos citar las relacionadas con el muestreo, las especificaciones, los ensayos y la evaluación de la calidad de los productos que se generan en el Centro. [1] Para el desempeño de las mismas, cuenta con la ayuda de dos grupos de trabajo e igual número de secciones, a continuación hacemos referencia a los mismos:

Grupo de Desarrollo.

Grupo de Recepción de Muestras y Manipulación de Expedientes.

Sección biológica compuesta por cinco laboratorios:

- Laboratorio de Biología Molecular.
- Laboratorio de Ensayos Biológicos I.
- Laboratorio de Ensayos Biológicos II.
- Laboratorio de Inmunoquímica.

Laboratorio de Microbiología (LM).

“El **LM** es el único laboratorio para estos fines que existe en el Centro, en el mismo se realiza el control microbiológico de todos los procesos de producción al controlar la pureza y viabilidad de los bancos de células, así como la posible contaminación en la etapa fermentativa, purificación y control de los Ingredientes Farmacéuticos Activos (IFA). Lleva a cabo el monitoreo microbiológico ambiental, control microbiológico de las muestras procedentes de los sistemas auxiliares. Se procesan además muestras de productos finales estériles y no estériles. Todos los medios de cultivo utilizados son controlados para promoción del crecimiento y esterilidad. El trabajo es realizado según las Buenas Prácticas de Laboratorio (BPL).” [2]

Sección físico-química compuesta por tres laboratorios:

- Laboratorio Análisis Químico.
- Laboratorio de Cromatografía y Electroforesis.
- Laboratorio de Sistemas Críticos. [1]

El DAC garantiza que se lleven a cabo las acciones planificadas y sistemáticas que son necesarias para proporcionar la confianza de que los productos y servicios satisfagan los requisitos de calidad establecidos. Vela por el cumplimiento de las Buenas Prácticas de Producción (BPP), BPL y Buenas

Prácticas Clínicas (BPC). Este departamento está compuesto por dos secciones y dos grupos de trabajo:

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

Una función de vital importancia que desarrolla esta sección es el proceso de liberación de los IFA y los productos finales, también se prepara toda la documentación que acompaña a los envíos de los productos liberados.

- Grupo de Documentación.
- Grupo de Metrología.

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 no son más que 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.

La gestión de la información permite determinar la información que se precisa, recoger y analizar la información, registrar y recuperar dicha información cuando sea necesaria además de utilizarla y divulgarla.

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 nacen a finales del siglo pasado los **Sistemas de Gestión de la Información del Laboratorio (LIMS**, del inglés 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. Los LIMS representan una importante herramienta

para la gestión global de un laboratorio en un entorno de calidad, agilizando temas de registro de datos primarios, archivo, trazabilidad, etc. y minimizando los errores debidos a la transferencia de información.”[3]

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.

Es considerable el volumen de información generada y que debe ser procesada por los trabajadores del LM. Esta información plasmada en documentos debe ser revisada, supervisada y aprobada, en la mayoría de los casos por los superiores, lo que genera una pérdida de tiempo considerable, cuando se requiere 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.

Luego de un estudio realizado en el CIGB, particularmente en la Dirección de Calidad y con el objetivo de incorporar nuevas tecnologías de la información, para conseguir hacer más factibles los mecanismos de procesamiento y almacenamiento de la información, se arribó a la conclusión de la necesidad inmediata de desarrollar un Sistema de Gestión de la Información de los Laboratorios. Producto a que los especialistas del Laboratorio de Microbiología manipulan un considerable volumen de documentación se concluyó mejorar la gestión de la información de esta área y se añadió como un módulo a desarrollar para el LIMS. Un grupo de analistas concluyó en julio del 2007, el análisis del problema en cuestión e identificaron un grupo de requerimientos. En la actualidad se trabaja en un módulo para el Sistema de Gestión de la Información del Laboratorio (LIMS) y se identifica como **problema científico**: ¿Cómo obtener un producto funcional a partir de las clases diseñadas para el módulo “Laboratorio de Microbiología” del LIMS Control de la Calidad del CIGB?

Este problema se enmarca en el **objeto de estudio**: Proceso de desarrollo de los sistemas de gestión de la información. El objeto de estudio delimita el **campo de acción**: Implementación de los Sistemas de Gestión de la Información. Para la solución del problema se plantea como **objetivo**: Implementar el

módulo del Laboratorio de Microbiología, para el Sistema de Gestión de la Información de los laboratorios de la Dirección de Calidad del CIGB.

Para lograr el objetivo planteado, se definen las siguientes tareas a cumplir:

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

El presente documento se divide en dos capítulos:

En el Capítulo 1 “**Fundamentación Teórica**”, se brinda información referente a los Sistemas de Gestión de la Información de Laboratorios que se comercializan en diferentes mercados del mundo; se abordan las principales características del proceso de desarrollo utilizado. Además se brinda información de los artefactos, que desarrolla el rol de Programador según RUP, con la utilización de las herramientas de desarrollo y modelado definidas por el proyecto, de las cuales se brinda información de sus características y propiedades.

En el Capítulo 2. “**Implementación del Sistema**”, se muestra una serie de diagramas de componentes, los mismos son los artefactos realizados por el Rol de Implementador que propone la Metodología RUP, así como parte del código fuente más importante, usado en la implementación de este módulo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo, con el objetivo de lograr una mejor comprensión de la necesidad de este trabajo, se realiza una valoración de los Sistemas de Gestión de la Información de los Laboratorios conocidos hasta el momento, de los artefactos enmarcados en el Rol de Programador del Proceso Unificado de Desarrollo capaz de desarrollar dichos sistemas. Del mismo modo se hace referencia a las herramientas y tecnologías utilizadas, del lado del servidor se utilizó PHP y del lado del cliente Java Script y se utiliza el framework Symfony.

1.1 LIMS existentes en el mercado.

Muchas empresas se han consagrado a la fabricación de los Sistemas de Gestión de Información de los Laboratorios para el manejo en particular de cada una de sus propias necesidades. Entre los diversos LIMS desarrollados hasta el momento, podemos mencionar:

Matrix LIMS es un Sistema de Gestión de Información de Laboratorios capaz de combinar dos aspectos fundamentales, configurabilidad auténtica y rapidez de implementación para garantizar el éxito en el proyecto.

Matrix LIMS tiene como características principales:

- La flexibilidad de configuración.
- Es adecuado para laboratorios pequeños y grandes organizaciones.
- Es compatible con varias bases de datos comerciales (Oracle y Microsoft SQL Server).
- Presenta una interfaz de usuario sencilla.
- Tiene tres opciones de producto: Matrix Express, Matrix Plus y Matrix Enterprise para brindar respuesta a las necesidades de cualquier laboratorio.[4]

La **Suite Pharma LIMS**, es una serie de aplicaciones LIMS destinadas para las industrias farmacéuticas y diseñadas para realizar sus operaciones dentro de los entornos compatibles con la norma 21 CFR Parte 11 (indica de qué manera las firmas electrónicas y registros computacionales pueden tener el mismo valor que las firmas autógrafas y los reportes en papel).

Esta Suite incluye entre sus aplicaciones LIMS a:

- Newton LIMS para el lanzamiento de pruebas, control de calidad, formulación, estabilidad y desarrollo analítico.
- Watson LIMS para bioanálisis preclínicos y clínicos.
- Galileo LIMS para las pruebas In Vitro. [5]

Ejemplo de arquitectura de los LIMS.

Podemos citar a **STARLIMS Corporation** el cual tiene una arquitectura distintiva multi-capa está diseñada para eliminar periódicas reinversiones costosas y pérdidas de información durante las actualizaciones o mejoras. La independencia de los componentes en los sistemas multi-capa facilita una completa partición de las tareas de desarrollo y mantenimiento. La capa Lógica del Negocio es el corazón del sistema, allí se almacenan las “Reglas del Negocio” que definen la manera como STARLIMS se comporta en la empresa. Esta capa contiene adicionalmente los metadatos que definen los objetos STARLIMS que son ensamblados para ejecutar funciones específicas. Estos incluyen registro de muestras, entrada de resultados, procesos de aprobación, sistema de numeración de muestras, cálculos automáticos basados en el ingreso de resultados y la lógica tras la programación de reportes, intervalos de muestreo automatizados, impresión de etiquetas, muestreo múltiple, mantenimiento de instrumentos, entrenamiento del personal, niveles de reactivos, optimización del flujo de trabajo y activadores de las bases de datos.

El uso de una capa Lógica de Negocio separada permite a STARLIMS conformar cualquier laboratorio y proceso de negocio. En vista a que la lógica del negocio es única para cada organización, cada capa de lógica de Negocio STARLIMS se convierte en un espejo de las operaciones de la organización. La capa Base de Datos contiene las herramientas de toda la empresa para el mantenimiento de las bases de datos, puede haber más de una capa de este tipo.

El vendedor puede proveer nuevas características avanzadas a la capa de Tecnología. Estas características mejoradas pueden ser rápidamente validadas y distribuidas a través de la empresa sin comprometer los componentes Reglas del Negocio o la Base de Datos. STARLIMS representa un nuevo paradigma en las conductas costo/funcionamiento de las aplicaciones LIMS, es más fácil implementar, validar, certificar y mantener, lo cual reduce el Costo Total de Propiedad a lo largo del ciclo de vida del LIMS. [6]

Debido a que los anteriores LIMS desarrollados en países extranjeros, tienen un alto valor monetario y la situación actual de Cuba como país bloqueado, no permite acceder a productos en el mercado internacional de tan alto valor, se decidió desarrollar uno que tenga similitud con algunas de las características de los anteriores expuestos.

1.2 Tecnologías de desarrollo y herramientas utilizadas.

A continuación se muestran las herramientas y tecnologías, seleccionadas para la posterior implementación del módulo “Laboratorio de Microbiología”, el cual forma parte del LIMS de Calidad del CIGB. Luego de un estudio realizado por los líderes del proyecto y de haber efectuado las

comparaciones pertinentes, las cuales permitieron evaluar las posibilidades de cada una de las alternativas analizadas, se decidió utilizar las herramientas que más adelante se exponen. Queda justificada la elección de cada una de ellas, al detallar características significativas y ventajas de su empleo.

1.2.1 Proceso Unificado de Desarrollo de Software (RUP, Rational Unified Process).

El Proceso Unificado de Desarrollo está bien definido, estructurado y adaptable a las características y necesidades de cada proyecto en específico. La definición de este está dada por tres características fundamentales: dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental.

Que RUP esté dirigido por casos de uso significa que el proceso de desarrollo sigue una trayectoria a través de flujos de trabajos generados por casos de uso, estos describen la funcionalidad del sistema, en términos de su importancia para el usuario.

Que RUP esté dirigido por los casos de uso no indica que se desarrollen de manera aislada de la arquitectura sino que se desarrollan ambos a la vez. La arquitectura involucra los elementos más significativos del sistema y está influenciada por las plataformas del software, los sistemas operativos y los sistemas de gestión de bases de datos.

RUP es un proceso iterativo e incremental que se encarga de dividir el trabajo en partes más pequeñas o en mini proyectos, esto permite el equilibrio entre casos de uso y arquitectura durante cada mini proyecto. Cada mini proyecto se puede ver como una iteración del cual se obtiene un incremento, esto provoca un aumento del producto. (Ver Figura. 1)

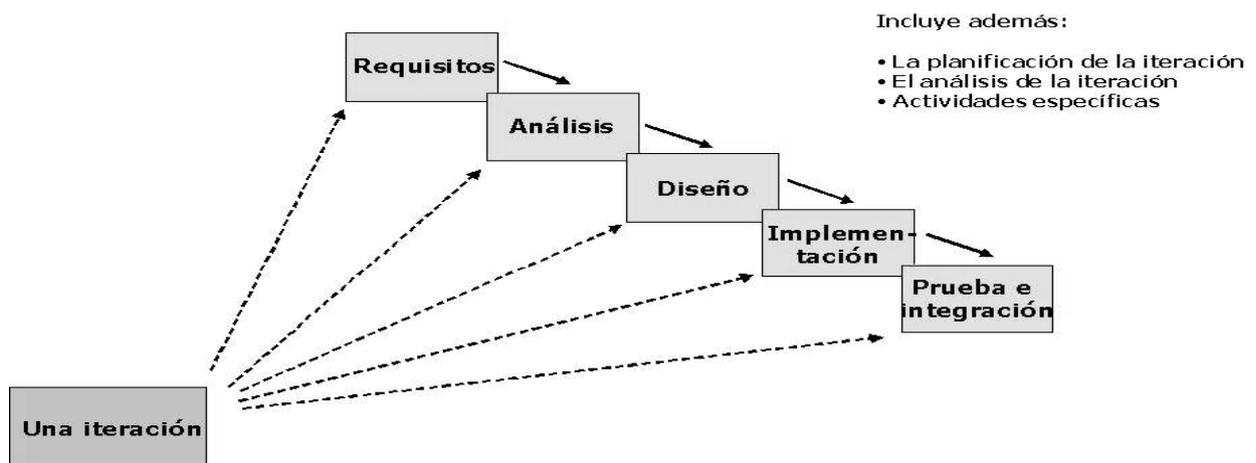


Figura 1. Iteración de RUP.

“Un proceso de desarrollo de software define quién hace qué, cómo y cuándo. RUP define cuatro

Capítulo 1: Fundamentación Teórica

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?”[7]

Hasta el momento ha sido utilizado este proceso de desarrollo para el progreso del LIMS Control Calidad del CIGB.

Este proceso de desarrollo se caracteriza por estar dividido en 4 fases fundamentales:

- Inicio: el objetivo en esta fase es determinar la visión del proyecto.
- Elaboración: en esta fase el objetivo es determinar la arquitectura óptima.
- Construcción: en esta fase el objetivo es llegar a obtener la capacidad operacional inicial.
- Transición: el objetivo es llegar a obtener la liberación del proyecto.

En cada fase se ejecutarán una o varias iteraciones (de tamaño variable según la complejidad del proyecto), y dentro de cada una de ellas se seguirá un modelo de cascada en los flujos de trabajo que lo requieran. (Ver Figura. 2)

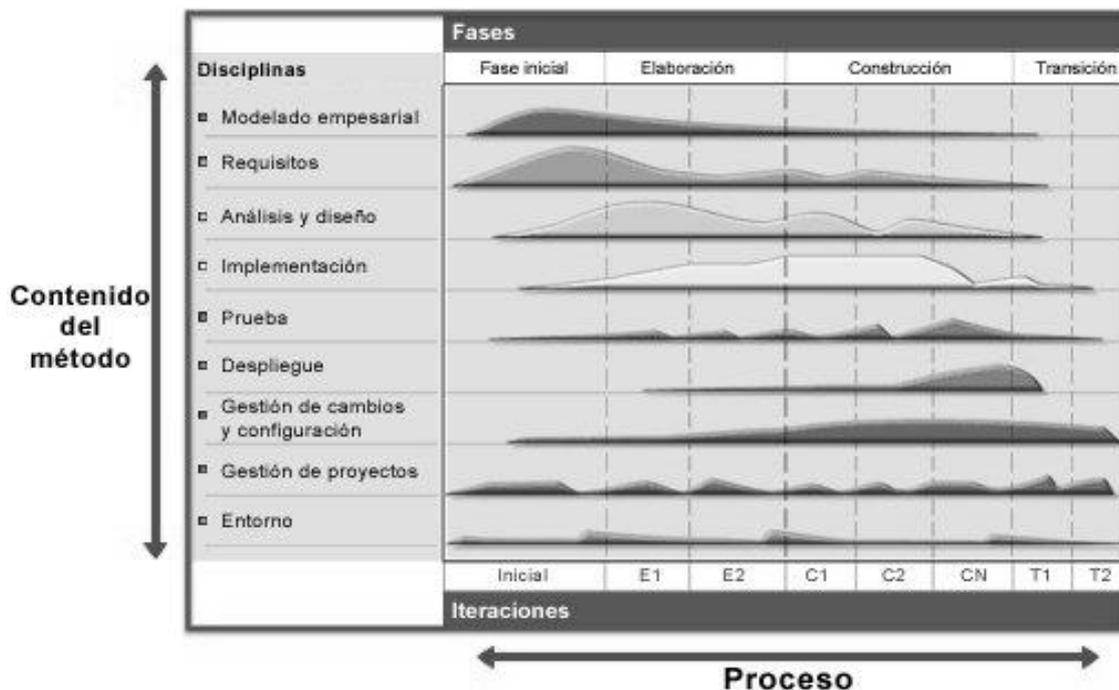


Figura 2. Fases e Iteraciones de la Metodología RUP.

Cada fase del RUP concluye con un hito bien definido, punto en el cual se deben tomar ciertas decisiones y alcanzar las metas clave 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 Figura. 3).

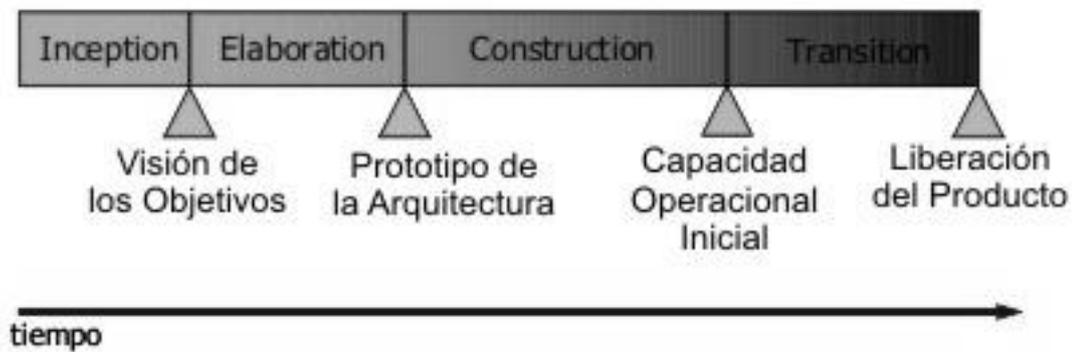


Figura 3. Fases e Hitos en RUP.

Un rol define las responsabilidades de un individuo, o de un grupo de individuos de un equipo de trabajo. Una persona puede desempeñar diversos roles y un rol puede ser representado por varias personas. Los roles definidos por RUP son: analistas, desarrolladores, administradores, soporte y producción, especialistas de pruebas y roles adicionales.

Una actividad es una unidad de trabajo que una persona que desempeña un rol puede realizar. Las actividades tienen un objetivo concreto, crear o actualizar algún producto.

Un artefacto es un fragmento de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los artefactos son los resultados tangibles del proyecto que se crean y usan hasta obtener el producto final.

Un flujo de trabajo es una relación de actividades que producen resultados observables dado por una secuencia de actividades realizadas por los diferentes roles.

Flujo de trabajo de Implementación.

En este flujo de trabajo tiene como actividades fundamentales: la implementación de clases y objetos en ficheros fuente, binarios, 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, se asignan componentes ejecutables a nodos en el diagrama de despliegue. Implementar las clases y subsistemas encontrados durante el diseño.

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. El implementador también es responsable para desarrollar y probar los componentes de la prueba y los subsistemas correspondientes. (Ver Figura. 1)

Artefactos a realizar.

Modelo de Implementación:

El Modelo de Implementación representa la composición física de la implementación en términos de subsistemas de implementación, y elementos de implementación (directorios y ficheros, incluyendo código fuente, datos y archivos ejecutables).

Subsistemas de Implementación:

Un subsistema de implementación es un conjunto de elementos de Implementación.

Un subsistema de Implementación estructura el Modelo de Implementación al dividirlo en partes más pequeñas que puedan ser integrados y probados por separado. (Ver Anexo. 2)

Diagrama de Componentes:

¿Qué es un componente?

En RUP, es usado el término "componente" para significar una parte encapsulada de un sistema, con suerte una parte no-trivial, casi independiente, y reemplazable de un sistema que cumple una función clara en el contexto de una arquitectura bien definida.

Los diagramas de componentes muestran la estructura de los componentes, incluidos los clasificadores que especifique 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 concreto:

- Subsistemas de implementación y sus dependencias de importación.
- Los subsistemas de implementación organizados en capas.

Lenguaje Unificado de Modelado:

RUP utiliza como lenguaje de notación el Lenguaje Unificado de Modelado (UML, del inglés Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema de software orientado a objetos. Es uno de los lenguajes de modelado más

conocidos y utilizados en la actualidad.

Beneficios de UML

- Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos. [8]

1.2.2 Herramienta CASE

Visual Paradigm (versión 6.1).

Es una herramienta CASE (Ingeniería de Software Asistida por Ordenados, por sus siglas en inglés Computer Aided Software Engineering) que utiliza UML (Lenguaje Unificado de Software): como lenguaje de modelaje 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 más 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 con la siguiente herramienta Java:

- Eclipse/IBM WebSphere [9]

Características:

- Modelación de procesos del negocio.
- Administración de requerimientos.
- Generación de la capa Objeto-Relacional.
- Generación de código e ingeniería inversa, incluye 10 lenguajes de programación, entre ellos: Java, C++, .NET, PHP y XML.
- Generación de reportes PDF, MS Word, HTML.
- Permite importar y exportar archivos XML.
- Permite importar archivos desarrollados con Rational Rose.
- Posee una interfaz de usuario amigable.
- Es multiplataforma, disponible para los Sistemas Operativos Linux, Windows, y Mac OS [10]

Beneficios:

- Navegación intuitiva entre el modelo visual y el código.
- Poderosa herramienta de generación de PDF/HTML a partir de diagramas UML.
- Sincronización entre el código fuente y el modelo en tiempo real o bajo demanda.
- Entorno visual de modelado superior.
- Soporte para toda la notación UML.
- Sofisticados y automáticos diagramas de capas.
- Análisis de textos.
- 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 (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas. [11]

Para el desarrollo del módulo se emplea la metodología de desarrollo RUP la cual utiliza UML, y se utiliza Visual Paradigm como herramienta CASE, dicha herramienta 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, en este caso en particular permitirá el desarrollo de los diagramas de componentes.

1.2.3 Lenguajes de programación

PHP (Lenguaje de Programación Interpretado) (Versión: 5.2.5).

PHP es un lenguaje script para el desarrollo de páginas web dinámicas del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación, cuyos fragmentos de código se intercalan fácilmente en páginas HTML.

La última versión es PHP5, que utiliza el motor Zend-2 y presenta mejoras significativas y un entorno de programación orientado a objetos mucho más completo, que permite que el PHP proporcione un alto rendimiento a las aplicaciones Web empresariales a nivel de las plataformas J2EE y .NET.

El principal objetivo de PHP5 ha sido mejorar los mecanismos de POO (Programación Orientada a

Objeto) para solucionar las carencias de las anteriores versiones. [12]

Características:

- Posee una sintaxis semejante de la de C.
- Dispone de una alta conectividad con la mayoría de Sistemas de Gestión de Bases de Datos.
- Es Open-Source y de obtención gratuita.
- Portable y multiplataforma (W95, 98, 2000, XP, NT, Unix, Linux...); lo cual permite su desarrollo desde S.O de base heterogéa.
- Proporciona soporte para la mayoría de protocolos de comunicación de Internet (HTTP, IMAP, FTP, LDAP, INMP...).
- Desde su versión 4, incorpora la posibilidad de desarrollar según la metodología de orientada a objetos, si bien es desde la versión 5 cuando todos los conceptos de este paradigma están disponibles. [13]

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas:

- Se necesita instalar un servidor web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.

- Dificulta la modularización.
- Dificulta la organización por capas de la aplicación. [14]

Javascript.

Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript podemos crear diferentes efectos e interactuar con nuestros usuarios.

Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros. Javascript nació con la necesidad de permitir a los autores de sitio web crear páginas que permitan intercambiar con los usuarios, porque se necesitaba crear webs de mayor complejidad. Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web.

¿Es compatible con navegadores?

Javascript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox.

Ventajas:

- Lenguaje de scripting seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código Javascript se ejecuta en el cliente.

Desventajas:

- Código visible por cualquier usuario.
- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS). [15]

Anteriormente se expusieron detalles de los lenguajes de programación previstos con anterioridad por el proyecto para la implementación del módulo, a través de sus prestaciones se demuestra la utilidad de los mismos.

1.2.4 Framework.

Para el desarrollo de software, un framework es una estructura definida de soporte en la cual otro proyecto de software puede ser constituido y desarrollado. Propiamente puede incluir soporte de programas, bibliotecas y un lenguaje desarrollado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Un framework reduce el desarrollo de una aplicación a través de la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además de proporcionar una estructura al código fuente, exigiendo al desarrollador crear un código más legible y más fácil de conservar. Por último, un framework facilita la programación de aplicaciones, pues encapsula operaciones complicadas en instrucciones sencillas.

Symfony (Versión: 1.0.11).

Symfony es un framework diseñado para perfeccionar, debido a las gentilezas que ofrecen sus características, el desarrollo de las aplicaciones web. En un primer momento, separa la lógica de negocio, la lógica del servidor y la presentación de la aplicación web. Facilita numerosas herramientas y clases enfocadas a reducir el tiempo de desarrollo de una aplicación web complicada. Además de automatizar las tareas más frecuentes, admitiendo al desarrollador dedicarse a cabalidad a los aspectos más específicos de cada aplicación.

Symfony está desarrollado completamente con PHP 5. Tiene compatibilidad con la mayoría de gestores de bases de datos, tales como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Es compatible con plataformas como: Unix, Linux, Windows, etc. A continuación se aluden algunas de sus características más significativas.

Características de Symfony:

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos.

- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos. Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- 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. [16]

Con anterioridad se expusieron las características principales que brinda el framework Symfony, el cual mediante las funcionalidades que ofrece permitirá el posterior desarrollo del módulo a implementar.

1.2.5 Sistema Gestor de Bases de Datos PostgreSQL (Versión: 8.2).

Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle. [17].

Gracias a las características que ofrece el gestor de bases de datos definido, se conseguirá almacenar gran cantidad de información de forma organizada, para el posterior acceso a los mismos, además de funcionar en múltiples plataformas.

1.2.6 Servidor Web Apache (Versión: 2.2.6).

Apache, servidor Web, manejable, rápido y eficaz. Su código fuente es abierto, de continua actualización y adaptado a los nuevos protocolos (HTTP 1.1). Posee disponibilidad con diferentes plataformas como: FreeBSD, NetBSD, OpenBSD, GNU/Linux, Mac OS X Server, Netware, Solaris, Windows, entre otras. Proporciona numerosos módulos de soporte y de conjunto con la API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) de programación de módulos, es de fácil adaptación a diferentes necesidades y entornos. Debido a su modular se han desplegado disímiles extensiones, dentro de ellas PHP, siendo la más destacada.

Por medio de las características y bondades que ofrece el Apache como servidor web permitirá atender y responder a las diferentes peticiones de los navegadores, proporcionándole la transferencia de

hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios y botones.

1.2.7 Entorno de Desarrollo Integrado (IDE) Eclipse (Versión: 3.3.1.1).

Eclipse es una plataforma de software de código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar un entorno integrado de desarrollo. Este IDE es más general para el desarrollo de aplicaciones en Java, en la que se ha especializado, también permite programar en PHP, aunque con muy pocas opciones de revisión y corrección de errores y sin completamiento de código. Es una herramienta que necesita de mucha ayuda del hardware para realizar la compilación del código fuente escrito, además de ser multiplataforma. [18]

El IDE seleccionado permitirá el desarrollo del software, porque cuenta con un editor de código, un compilador o intérprete y un depurador, esto facilita la posterior implementación del módulo, mediante el lenguaje de programación seleccionado.

Conclusiones.

- En el concluido capítulo se hizo alusión a las principales actividades y artefactos a desarrollar por el programador, rol enmarcado en flujo de Trabajo Implementación, el cual corresponde desempeñar, según la metodología de desarrollo de software previamente definida.
- De igual forma se realiza una breve explicación de las características principales de las herramientas y tecnologías a utilizar, dígase el framework Symfony, SGBD PortgreSQL y como lenguaje de programación PHP; para llegar a desarrollar una aplicación que cumpla a cabalidad con los requerimientos exigidos por el cliente.

CAPÍTULO 2: IMPLEMENTACIÓN DEL SISTEMA

En el presente capítulo, se listan los requisitos funcionales, de conjunto con la descripción de uno de los casos uso, a partir de los cuales se realizó la implementación del módulo. De igual forma se muestran los artefactos del Rol de Programador según RUP, dígame diagramas de componentes, además de mostrar parte del código fuente, en especial aquellas funciones que son más importantes para el desarrollo del módulo, conformado por 12 SIC (Sistema de Información y Control). Dicho código se realizó haciendo uso del framework Symfony, esta herramienta mediante su funcionamiento generó 50 clases peer y la misma cantidad no peer, así como también 50 clases base peer, e igual cantidad de clases base no peer.

A este trabajo le precedió una tesis de diploma donde se listaron todas las funcionalidades que debe cumplir el sistema en cuestión. A continuación se muestra un listado de dichas funcionalidades, las cuales se implementan en su totalidad.

2.1 Requisitos Funcionales:

1. Gestionar Ensayo Control de Proceso:
 - 1.1. Crear nuevo registro.
 - 1.2. Buscar y visualizar registro.
 - 1.2.1. Imprimir registro.
 - 1.3. Modificar datos de un registro.
 - 1.3.1. Registrar traza.
 - 1.4. Generar reporte.
 - 1.4.1. Imprimir reporte.
2. Gestionar Chequeo de Viabilidad:
 - 2.1. Crear nuevo registro.
 - 2.2. Buscar y visualizar registro.
 - 2.2.1. Imprimir registro
 - 2.3. Modificar datos de un registro.
 - 2.3.1. Registrar traza.
 - 2.4. Generar reporte.
 - 2.4.1. Imprimir reporte.
3. Gestionar Control Microbiológico de las Aguas:
 - 3.1. Crear nuevo registro.
 - 3.2. Buscar y visualizar registro.
 - 3.2.1. Imprimir registro.
 - 3.3. Modificar datos de un registro.
 - 3.3.1. Registrar traza.
 - 3.4. Generar reporte.
 - 3.4.1. Imprimir reporte.

4. Gestionar Control Ambiental mediante Placa Expuesta:
 - 4.1. Crear nuevo registro.
 - 4.2. Buscar y visualizar registro.
 - 4.2.1. Imprimir registro.
 - 4.3. Modificar datos de un registro.
 - 4.3.1. Registrar traza.
 - 4.4. Generar reporte.
 - 4.4.1. Imprimir reporte.
5. Gestionar Chequeo Inicial de Viabilidad:
 - 5.1. Crear nuevo registro.
 - 5.2. Buscar y visualizar registro.
 - 5.2.1. Imprimir registro.
 - 5.3. Modificar datos de un registro.
 - 5.3.1. Registrar traza.
 - 5.4. Generar reporte.
 - 5.4.1. Imprimir reporte.
6. Gestionar Ensayo Efectividad de Preservos Antimicrobianos:
 - 6.1. Crear nuevo registro.
 - 6.2. Buscar y visualizar registro.
 - 6.2.1. Imprimir registro.
 - 6.3. Modificar datos de un registro.
 - 6.3.1. Registrar traza.
 - 6.4. Generar reporte.
 - 6.4.1. Imprimir reporte.
7. Gestionar Ensayo Límite Microbiano:
 - 7.1. Crear nuevo registro.
 - 7.2. Buscar y visualizar registro.
 - 7.2.1. Imprimir registro
 - 7.3. Modificar datos de un registro.
 - 7.3.1. Registrar traza.
 - 7.4. Generar reporte.
 - 7.4.1. Imprimir reporte.
8. Gestionar Control Microbiológico de las Manos Enguantadas:
 - 8.1. Crear nuevo registro.
 - 8.2. Buscar y visualizar registro.
 - 8.2.1. Imprimir registro.
 - 8.3. Modificar datos de un registro.
 - 8.3.1. Registrar traza.
 - 8.4. Generar reporte.
 - 8.4.1. Imprimir reporte.
9. Gestionar Ensayo de Preparación de Soluciones:
 - 9.1. Crear nuevo registro.
 - 9.2. Buscar y visualizar registro.

- 9.2.1. Imprimir registro.
- 9.3. Modificar datos de un registro.
 - 9.3.1. Registrar traza.
- 9.4. Generar reporte.
 - 9.4.1. Imprimir reporte.
- 10. Gestionar Control Muestreo de Superficies:
 - 10.1. Crear nuevo registro.
 - 10.2. Buscar y visualizar registro.
 - 10.2.1. Imprimir registro.
 - 10.3. Modificar datos de un registro.
 - 10.3.1. Registrar traza.
 - 10.4. Generar reporte.
 - 10.4.1. Imprimir reporte.
- 11. Gestionar Control Microbiológico de Vapor Puro:
 - 11.1. Crear nuevo registro.
 - 11.2. Buscar y visualizar registro.
 - 11.2.1. Imprimir registro.
 - 11.3. Modificar datos de un registro.
 - 11.3.1. Registrar traza.
 - 11.4. Generar reporte.
 - 11.4.1. Imprimir reporte.
- 12. Gestionar Control Muestreo con el Analizador Ambiental:
 - 12.1. Crear nuevo registro.
 - 12.2. Buscar y visualizar registro.
 - 12.2.1. Imprimir registro.
 - 12.3. Modificar datos de un registro.
 - 12.3.1. Registrar traza.
 - 12.4. Generar reporte.
 - 12.4.1. Imprimir reporte.

2.2 Descripción del caso de uso.

Para lograr una implementación que se corresponda con las descripciones de casos de uso del sistema realizadas con anterioridad, cumpliéndose a cabalidad las mismas. Se muestra a continuación la descripción extendida del CU Gestionar Ensayo de Preparación de Soluciones. Las demás descripciones se pueden consultar con mayor precisión en el Expediente del Proyecto.

Capítulo 2: Implementación del Sistema

Descripción del caso de uso Gestionar Ensayo de Preparación de Soluciones.

Nombre del Caso de Uso	Gestionar Ensayo de Preparación de Soluciones.	
Actores	Técnico MC (inicia).	
Propósito	Crear un nuevo registro, modificar sus datos, buscar, visualizar e imprimir el registro. Generar e imprimir reporte del registro.	
Resumen	<p>El caso de uso se inicia cuando el técnico MC va a realizar alguna de las siguientes operaciones relacionadas con el ensayo:</p> <ul style="list-style-type: none"> • Crear nuevo registro • Buscar y visualizar registro. • Generar reporte. <p>El sistema muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las operaciones solicitadas.</p>	
Referencias	10.1, 10.2, 10.2.1, 10.3, 10.3.1, 10.4, 10.4.1	
Precondiciones	Que el técnico MC se haya autenticado en la aplicación.	
Poscondiciones	Registro creado, buscado y visualizado, impreso, modificado, reporte generado e impreso.	
Requerimientos especiales		
Flujo normal de los eventos		
Acción del actor	Respuesta del Sistema	
1. El técnico MC quiere realizar una de las siguientes operaciones: <ul style="list-style-type: none"> • Crear nuevo registro. • Buscar y visualizar registro. • Generar reporte. 	2. En dependencia de la operación solicitada, hace lo siguiente: <ul style="list-style-type: none"> • Si decide crear nuevo registro, ir a la Sección “Crear nuevo registro”. • Si decide buscar y visualizar registro, ir a la Sección “Buscar y visualizar registro”. • Si decide generar reportes, ir a la sección “Generar reporte”. 	
Sección “ Crear nuevo registro ”		
Acción del actor	Respuesta del Sistema	
	1. Muestra la interfaz correspondiente para crear un nuevo registro.	

Capítulo 2: Implementación del Sistema

<p>2. El técnico MC provee los datos a los campos siguientes:</p> <ul style="list-style-type: none">• Folio.• No. Parte.• No. Lote.• VT.• Fecha preparación.	<p>3. Genera los campos Departamento y Área con Analítica y Calidad respectivamente.</p>
	<p>4. Debe darle la posibilidad de seleccionar los reactivos y permitirle agregar otros.</p>
<p>5. El técnico MC provee los datos:</p> <ul style="list-style-type: none">• Np.• Cantidad a usar.• Cantidad utilizada.• Lote.• Nombre.• Código.• Fecha de venc.• Calibración.• Solvente.• pH.• Conductividad.• Solución de ajuste de pH.• NP.• No. de lote.• pH final de la solución.• conductividad de la solución.• Vapor saturado.• el tiempo.• No. de frascos de almacenamientos.• volumen por frasco.• Temperatura de almacenamiento de la solución.• fecha de vencimiento.	
	<p>6. Debe dar la posibilidad de seleccionar si hubo o no Esterilización, así como que si la filtración fue por Membrana u otros.</p>
	<p>7. Debe llenar los campos de Pruebas de aceptación con: Promoción del crecimiento y Esterilidad.</p>

Capítulo 2: Implementación del Sistema

<p>8. El técnico MC entra por teclado los valores de los campos: Límite y valor obtenido. Además llena los campos: nombre y apellidos, así como la fecha de realización.</p>	
<p>9. El técnico MC provee los datos a los siguientes campos:</p> <ul style="list-style-type: none"> • Realizado por (Nombre(s) y Apellidos) • Recibido por (Nombre(s) y Apellidos) • Fecha (Para cada uno de los campos anteriores) 	
<p>10. El técnico MC provee los datos al campo siguiente:</p> <ul style="list-style-type: none"> • Terminado <p>Solicita crear registro.</p>	<p>11. Si el campo Terminado fue seleccionado, asigna al campo Estado el valor: Terminado.</p>
	<p>12. Crea un nuevo Registro.</p>
	<p>13. Muestra el registro y brinda la posibilidad de:</p> <ul style="list-style-type: none"> • Imprimir • Modificar
<p>14. El técnico MC solicita una de las opciones siguientes:</p> <ul style="list-style-type: none"> • Imprimir • Modificar 	<p>15. Según lo solicitado por el analista, se realiza una de las siguientes operaciones:</p> <ul style="list-style-type: none"> • Si seleccionó la opción Modificar: ir a Sección “Modificar datos de registro”. • Si seleccionó la opción Imprimir: imprime el registro.
Flujos alternos Sección “ Crear nuevo registro”	
<p>2.1 El técnico MC no provee datos y sale de la sección.</p>	
<p>10.1 El técnico MC no solicita crear el registro sale de la sección.</p>	
	<p>11.1 Asigna al campo Estado el valor: No Terminado.</p>
Sección “Buscar y visualizar registro”	
Acción del actor	Respuesta del Sistema
	<p>1. Muestra la interfaz correspondiente a la búsqueda de registros.</p>

Capítulo 2: Implementación del Sistema

<p>2. El técnico MC provee el o los parámetros para la búsqueda:</p> <ul style="list-style-type: none"> • Departamento • Realizado por • Fecha de comienzo del ensayo (Selecciona un rango de fecha) <p>Solicita realizar la búsqueda.</p>	<p>3. Verifica que al menos uno de estos campos tenga valor.</p>
	<p>4. Busca los registros que cumplan con los parámetros introducidos.</p>
	<p>5. Muestra los resultados de la búsqueda en una tabla con los siguientes campos:</p> <ul style="list-style-type: none"> • Folio • Nombre de la solución. • No. Lote • F. Recepción • No. de Parte <p>Brinda la posibilidad de visualizar los resultados de la búsqueda.</p>
<p>6. El técnico MC selecciona el registro que desea visualizar.</p>	<p>7. Visualiza los datos del registro seleccionado en una nueva interfaz y brinda la posibilidad de:</p> <ul style="list-style-type: none"> • Imprimir • Modificar
<p>8. Selecciona una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Imprimir • Modificar 	<p>9. En dependencia de la opción solicitada, el sistema realiza una de las operaciones:</p> <ul style="list-style-type: none"> • Si decide imprimir el registro, imprime el registro. • Si decide modificar el registro, ir a la Sección “Modificar datos de un registro”.
Flujos alternos Sección “Buscar y visualizar registro”	
<p>2.1 El técnico MC no solicita realizar la búsqueda y sale de la sección.</p>	
	<p>3.1. Si no existe al menos un parámetro con valor, emite un mensaje de error, “Introducir al menos un parámetro para realizar la búsqueda”.</p>
	<p>4.1 Si no encuentra ningún registro que cumpla con los parámetros indicados,</p>

Capítulo 2: Implementación del Sistema

	muestra un mensaje, “No existe el registro solicitado”.
6.1 El técnico MC no desea visualizar ninguna de las solicitudes mostradas y sale de la sección.	
8.1 El analista no desea realizar ninguna de las opciones y sale de la sección.	
Sección “Modificar datos de un registro”	
Acción del actor	Respuesta del Sistema
	1. Muestra el registro listo para modificar.
2. El técnico MC provee los datos a modificar y solicita modificar registro.	3. Actualiza los cambios realizados en la base de datos.
	4. Si el campo Terminado fue seleccionado, se asigna al campo Estado el valor: Terminado .
	5. Registra la actualización en la Base de datos.
	6. Actualiza el registro de trazas.
	7. Muestra el registro con los datos modificados y brinda la posibilidad de: <ul style="list-style-type: none"> • Imprimir • Modificar
8. El técnico MC solicita una de las opciones siguientes en el registro mostrado: <ul style="list-style-type: none"> • Imprimir • Modificar 	9. En dependencia de la opción solicitada, el sistema realiza una de las operaciones: <ul style="list-style-type: none"> • Si decide imprimir el registro, el imprime el registro. • Si decide modificar el registro, ir a la acción 1.
Flujos alternos Sección “Modificar datos de un registro”	
2.1 El técnico MC No provee los datos y no desea modificar el registro, sale de la sección.	
	4.1 Asigna al campo Estado el valor: No Terminado .
8.1 El analista no solicita ninguna de las opciones y sale de la sección.	
Sección “Generar reporte”	
Acción del actor	Respuesta del Sistema
	1. Muestra la interfaz correspondiente para generar un reporte.
2. El analista selecciona el o los parámetros de búsqueda: <ul style="list-style-type: none"> • Nombre de la solución 	3. Verifica que al menos un campo tenga valor.

Capítulo 2: Implementación del Sistema

<ul style="list-style-type: none"> • Fecha (Selecciona un rango de fecha) <p>El analista selecciona de los parámetros siguientes, los que desea visualizar en el reporte:</p> <ul style="list-style-type: none"> • Departamento • Área • Nombre de la solución • No. Parte • No. lote • VT • Fecha prep. • Solvente • PH • Conductividad • Ajuste de pH a la solución • PH deseado • Solución de ajuste de PH • PH final de la solución • Realizado por • Revisado por <p>Solicita generar reporte.</p>	
	4. Genera el reporte según el criterio de búsqueda indicada.
	5. Visualiza el reporte en una tabla. La tabla contiene los parámetros seleccionados por el analista. Brinda la posibilidad de imprimir el reporte.
6. Solicita imprimir reporte.	7. Imprime reporte.
Flujos alternos Sección “ Generar reporte ”	
2.1 El analista no solicita generar reporte y sale de la sección.	
	3.1 Si no existe al menos un parámetro con valor, emite un mensaje de error, “Introducir al menos un parámetro”.
6.1 El analista no solicita imprimir el reporte y sale de la sección.	
Prototipo	

2.3 Realización de Diagramas de Componentes.

Los diagramas de componentes se utilizan para estructurar el modelo de implementación en términos de subsistemas de implementación y exponer las relaciones entre los elementos de implementación.

Su uso más significativo es mostrar la estructura de alto nivel del modelo de implementación, los mismos especifican:

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

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. Lo que lo diferencia 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. Habitualmente los diagramas de componentes se utilizan para modelar código fuente, versiones ejecutables y bases de datos físicas [19].

A continuación se muestran los diagramas de componentes desarrollados para la implementación del módulo Microbiología, cada diagrama está compuesto por un paquete View, dentro del cual se encuentran todas páginas success.php con las cuales interactúa el usuario, el Layout.php que no es más que la parte global en toda la aplicación, es decir, todo el código HTML(Hypertext Markup Language) común para todas las páginas; la relación existente entre las páginas success y el layout, demuestra lo anterior expuesto. También está el componente Java Script dentro del cual se localiza todo el código java script utilizado. Otro componente utilizado es el actions.class.php donde se encuentra toda la implementación de la capa control, de la cual esta forma parte. Dentro del subsistema Model se hallan todas las clases de la capa modelo, es decir, todas las clases peer, no peer, base peer y base no peer. Otro subsistema representado es del componente Symfony, dentro del cual se encuentran 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 Security.yml y Validate.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 Microbiología, que no es más que la base de datos del módulo. Todas las relaciones son de dependencia, esto demuestra la estrecha relación existente entre cada uno de los componentes representados.

Diagramas de Componentes.

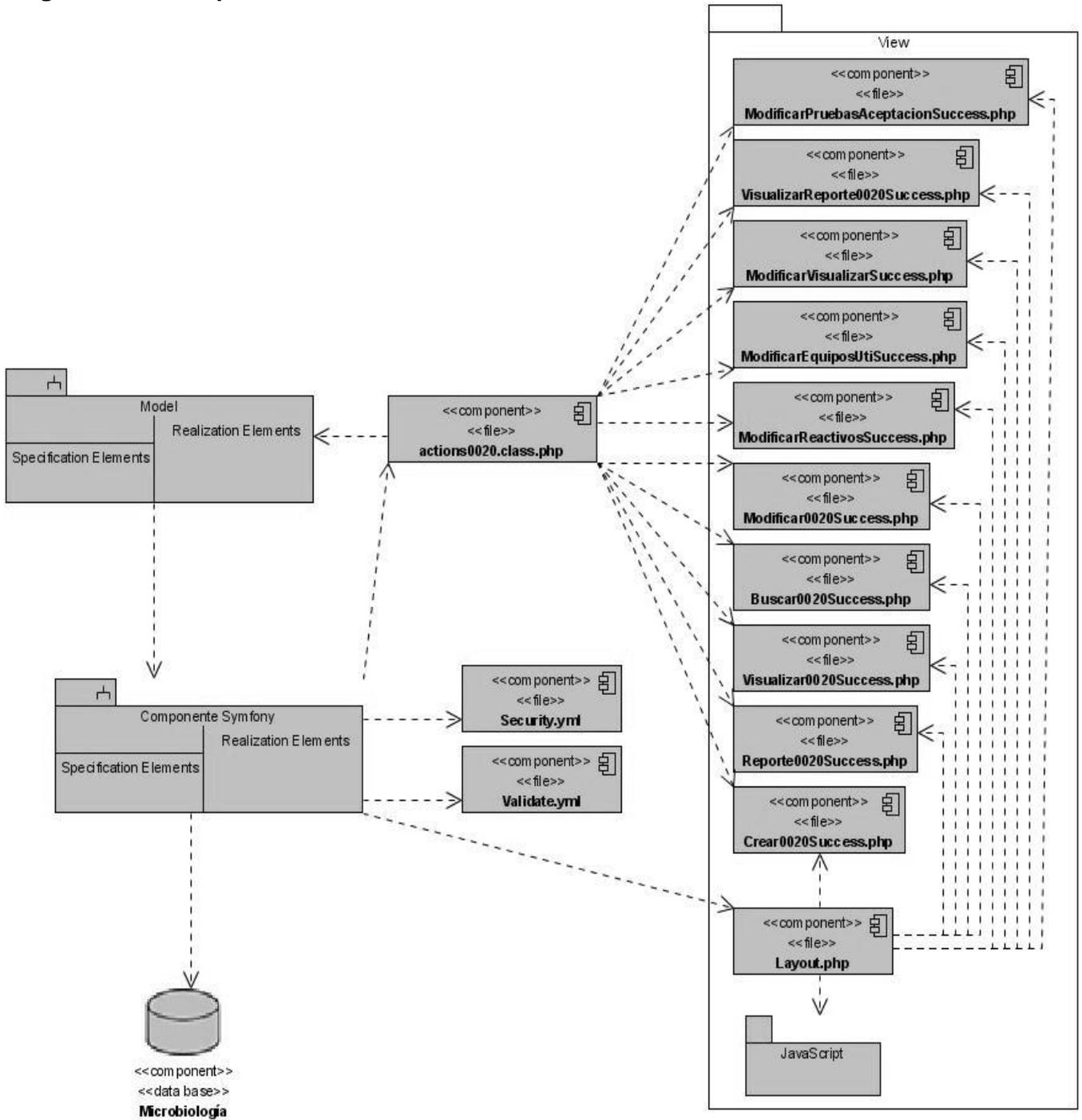


Figura 4. Diagrama de componentes CU Gestionar SIC_0020.

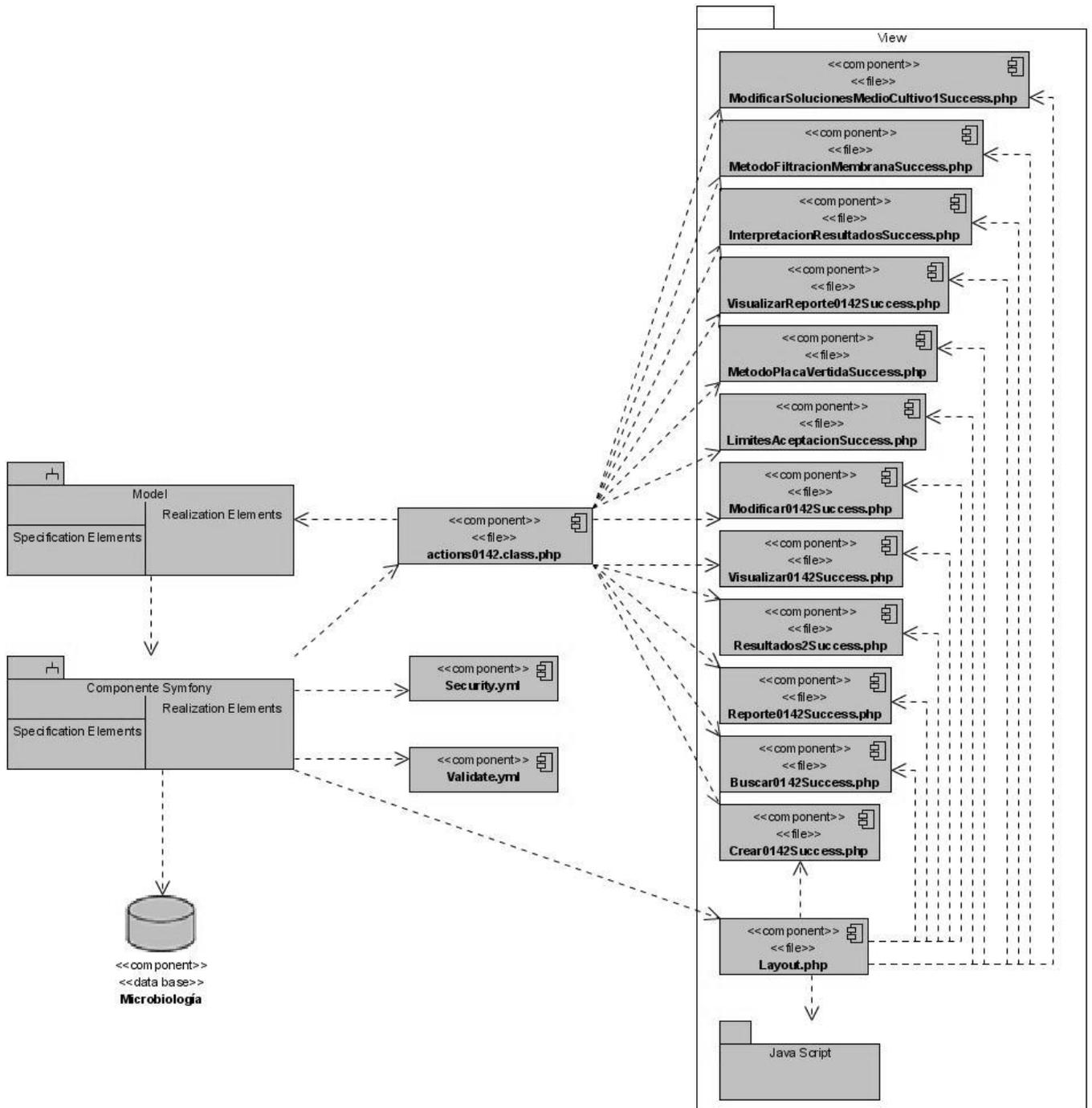


Figura 5. Diagrama de componentes CU Gestionar SIC_0142.

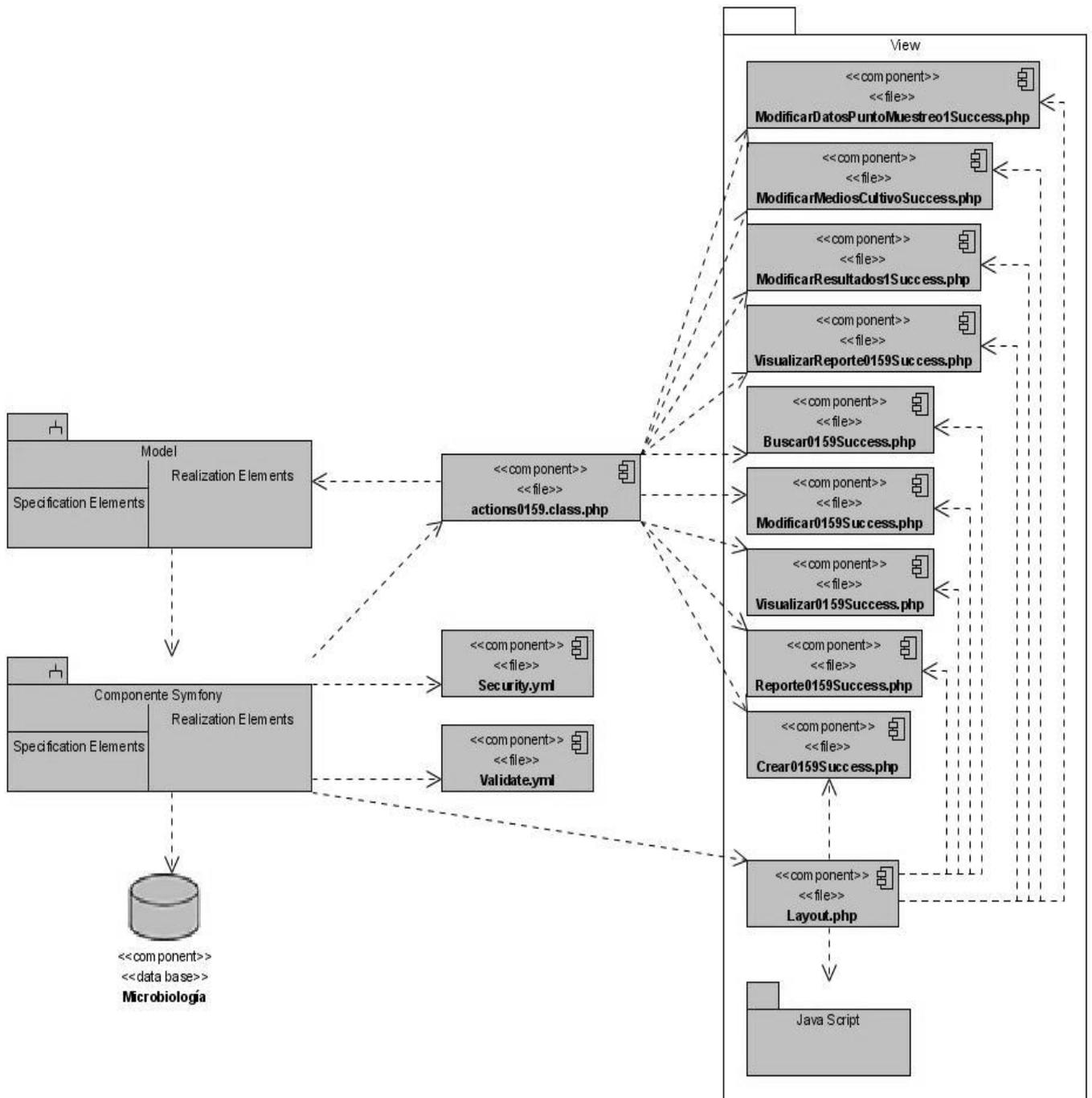


Figura 6. Diagrama de componentes CU Gestionar SIC_0159.

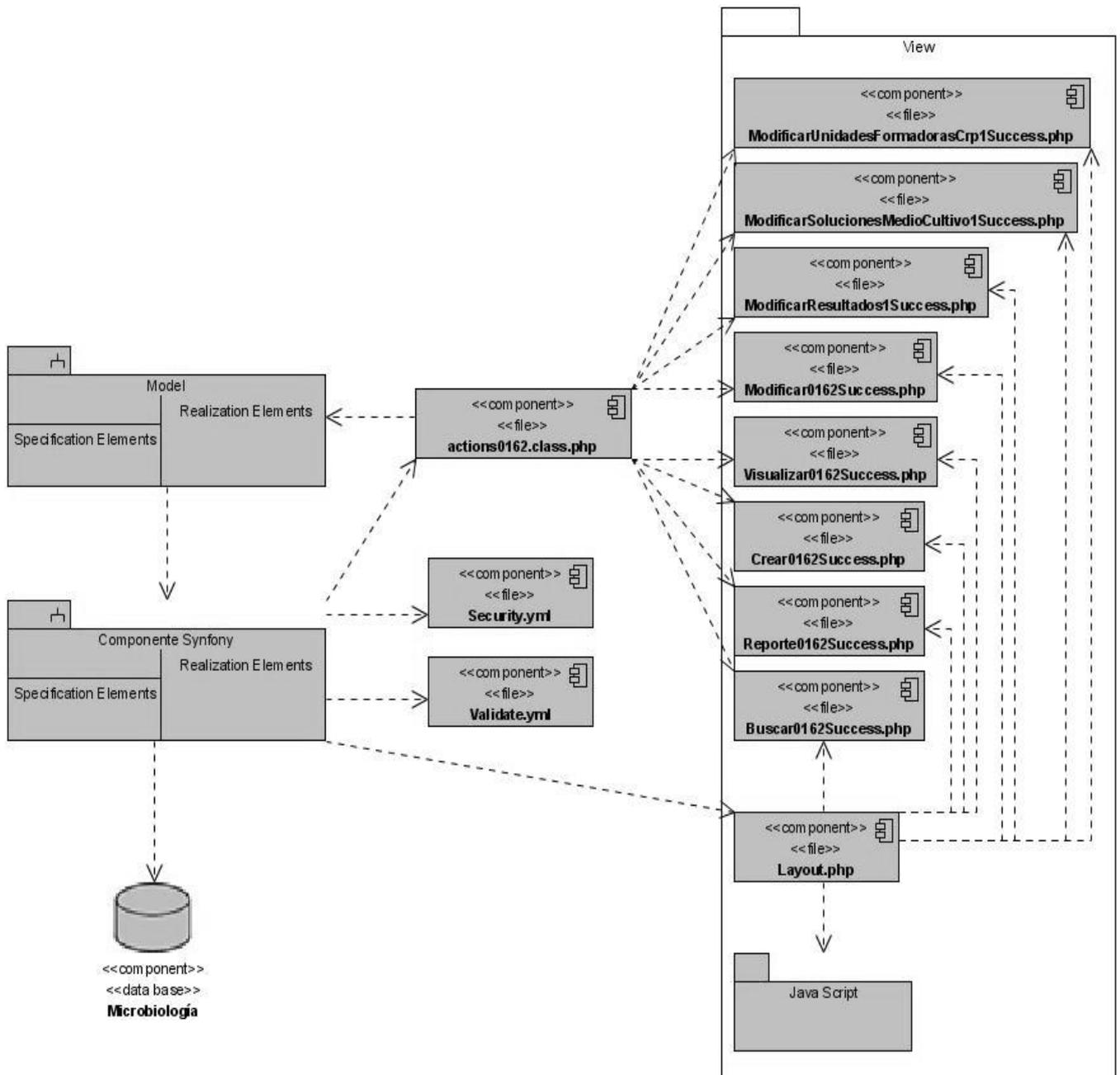


Figura 7. Diagrama de componentes CU Gestionar SIC_0162.

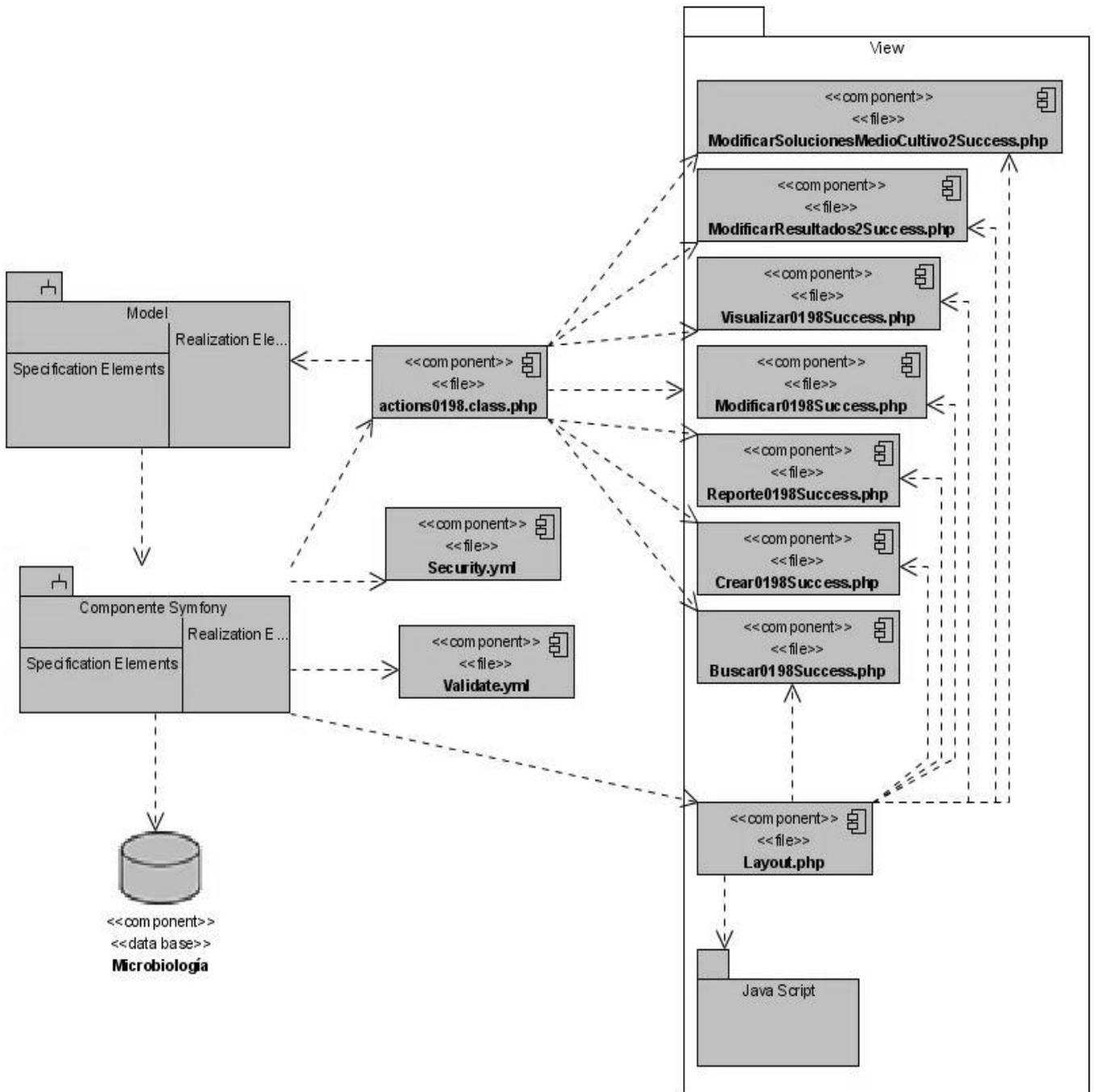


Figura 8. Diagrama de componentes CU Gestionar SIC_0198.

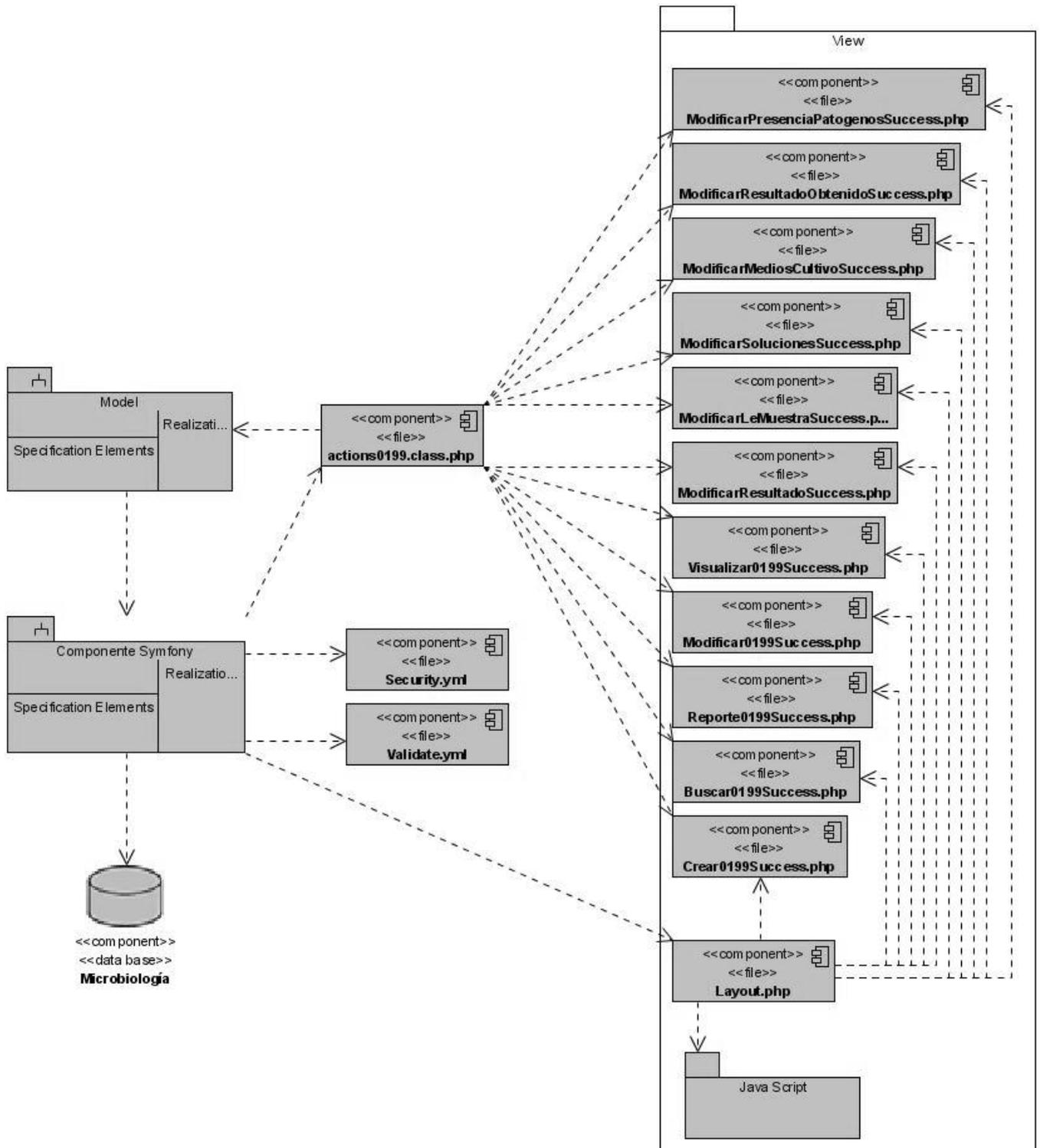


Figura 9. Diagrama de componentes CU Gestionar SIC_0199.

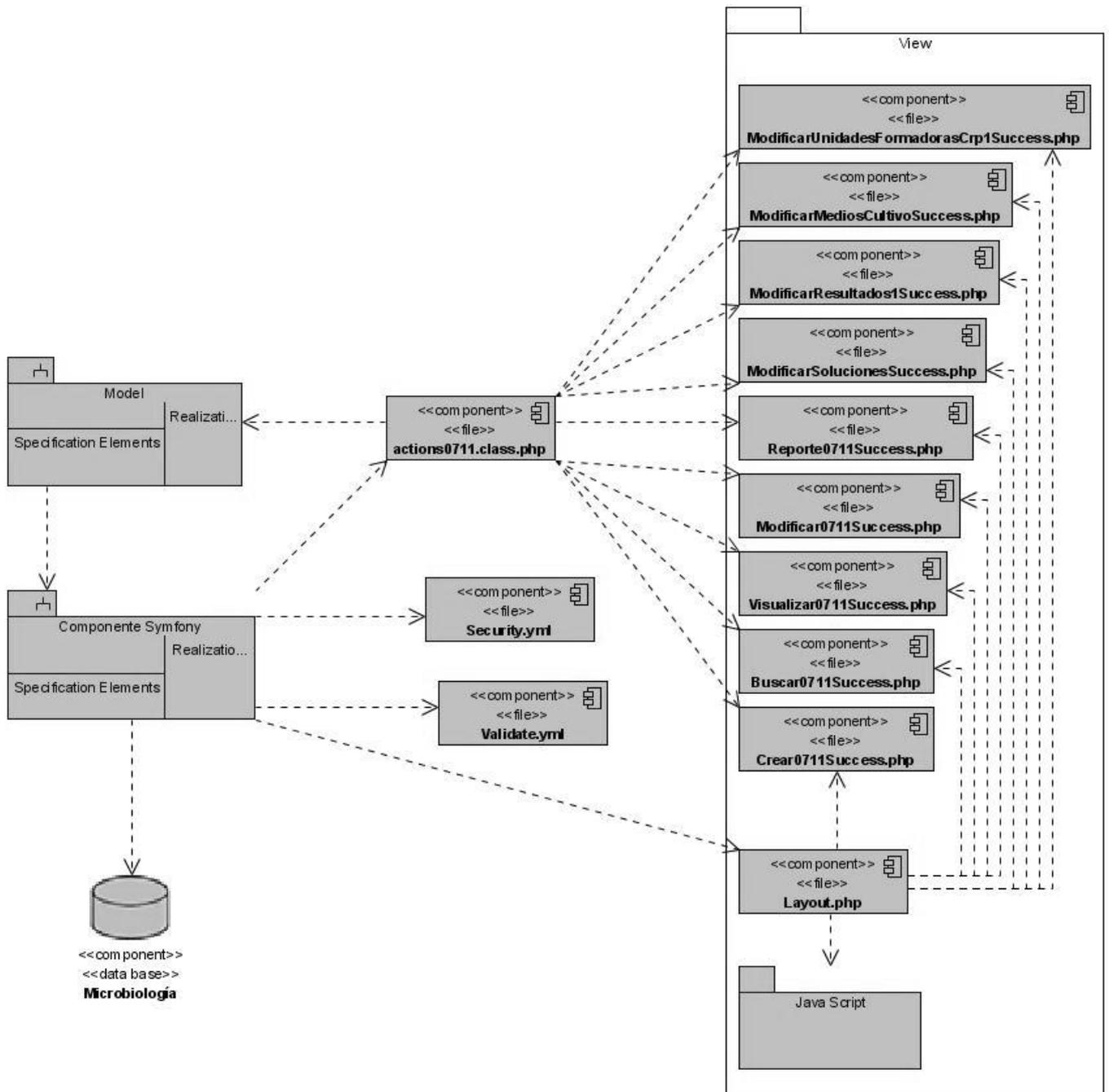


Figura 10. Diagrama de componentes CU Gestionar SIC_0711.

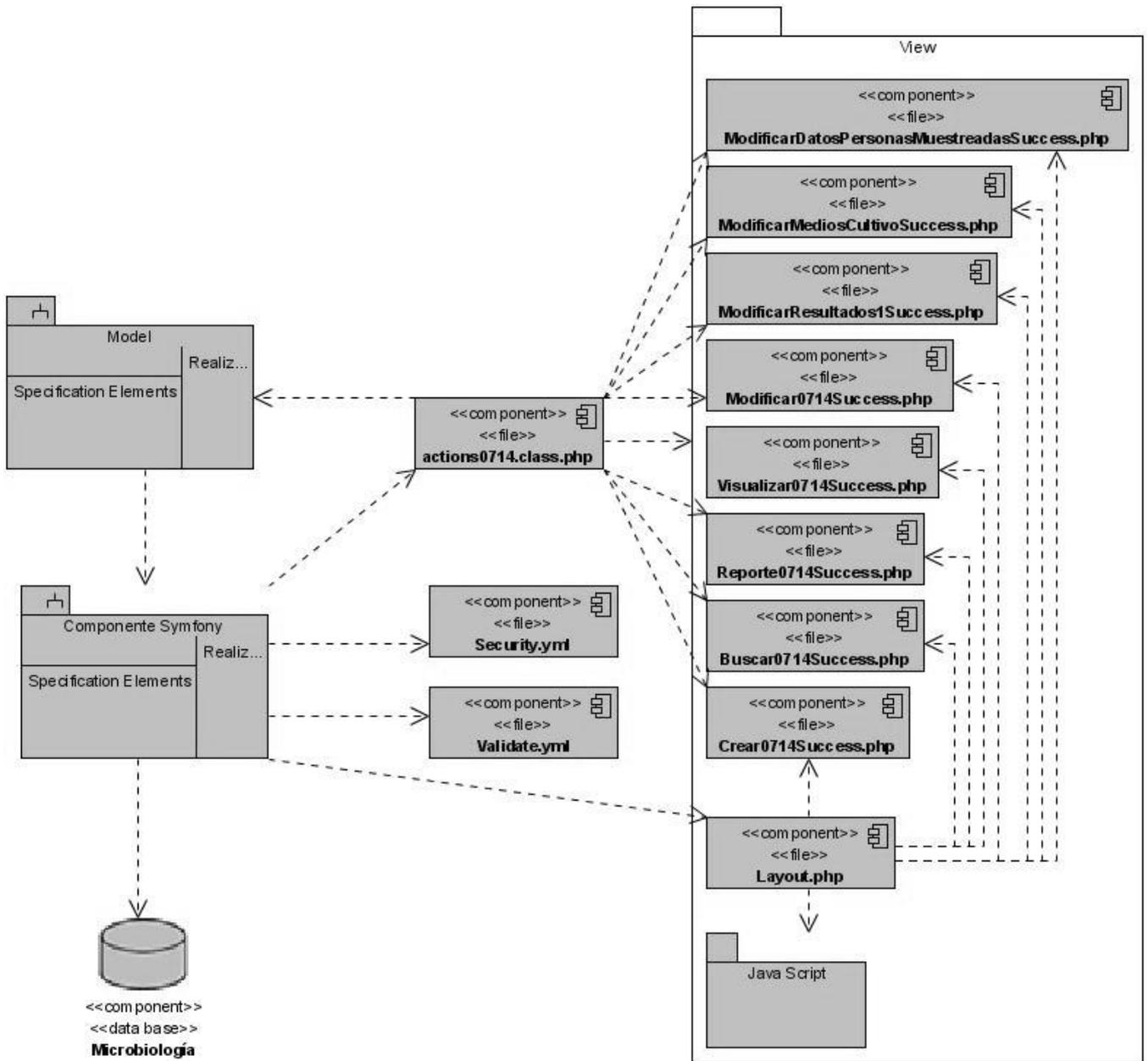


Figura 11. Diagrama de componentes CU Gestionar SIC_0714.

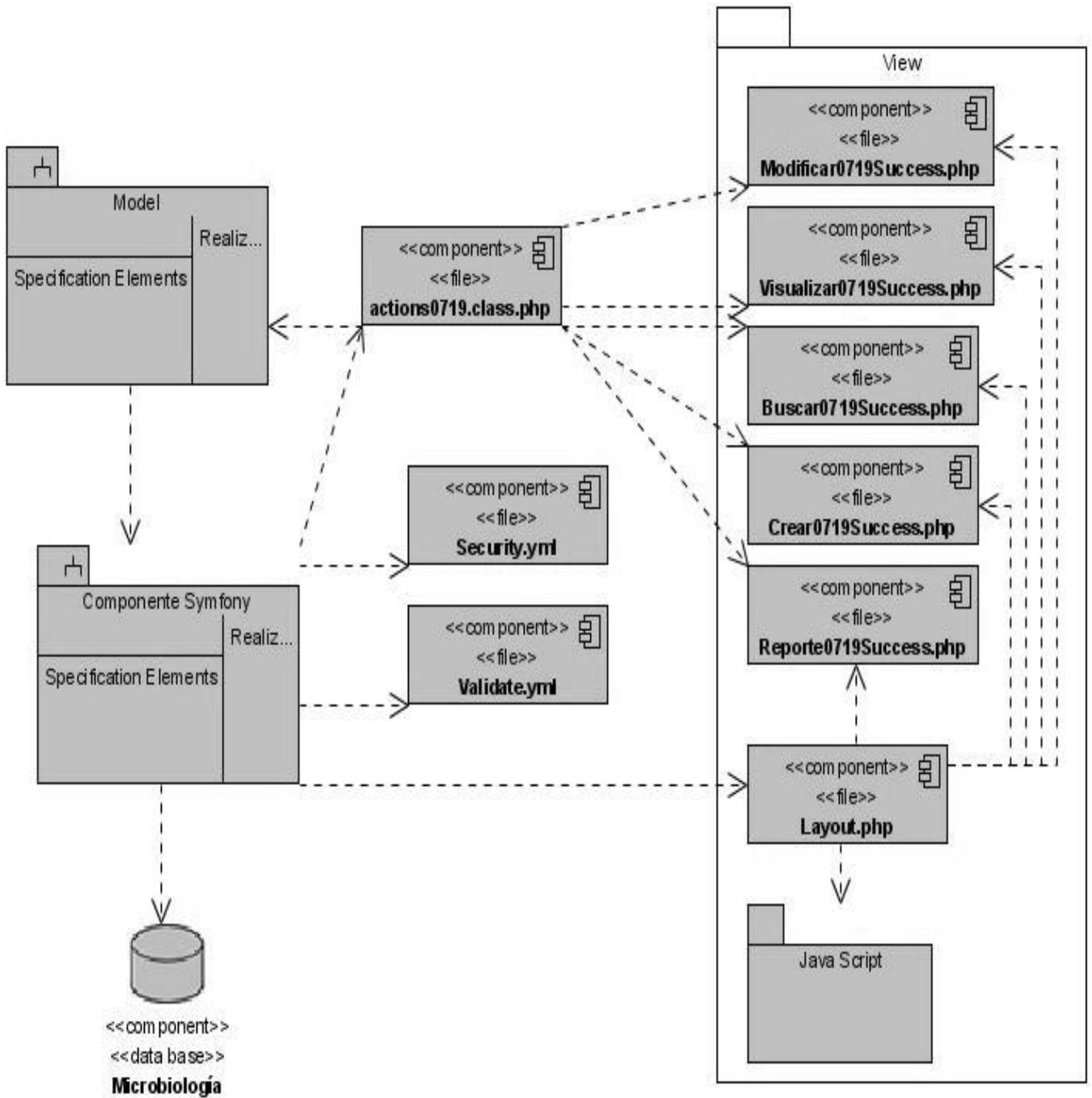


Figura 12. Diagrama de componentes CU Gestionar SIC_0719.

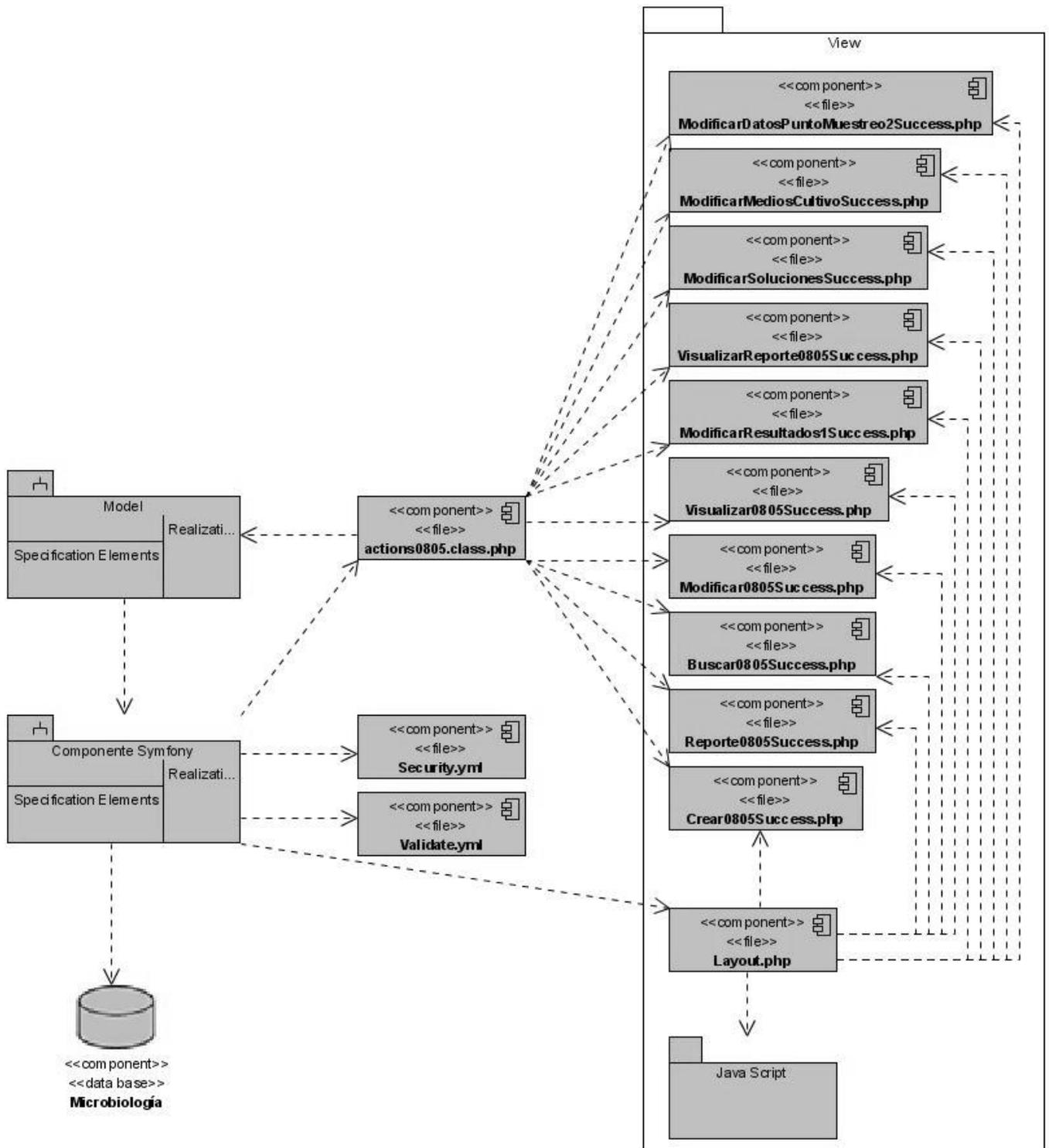


Figura 13. Diagrama de componentes CU Gestionar SIC_0805.

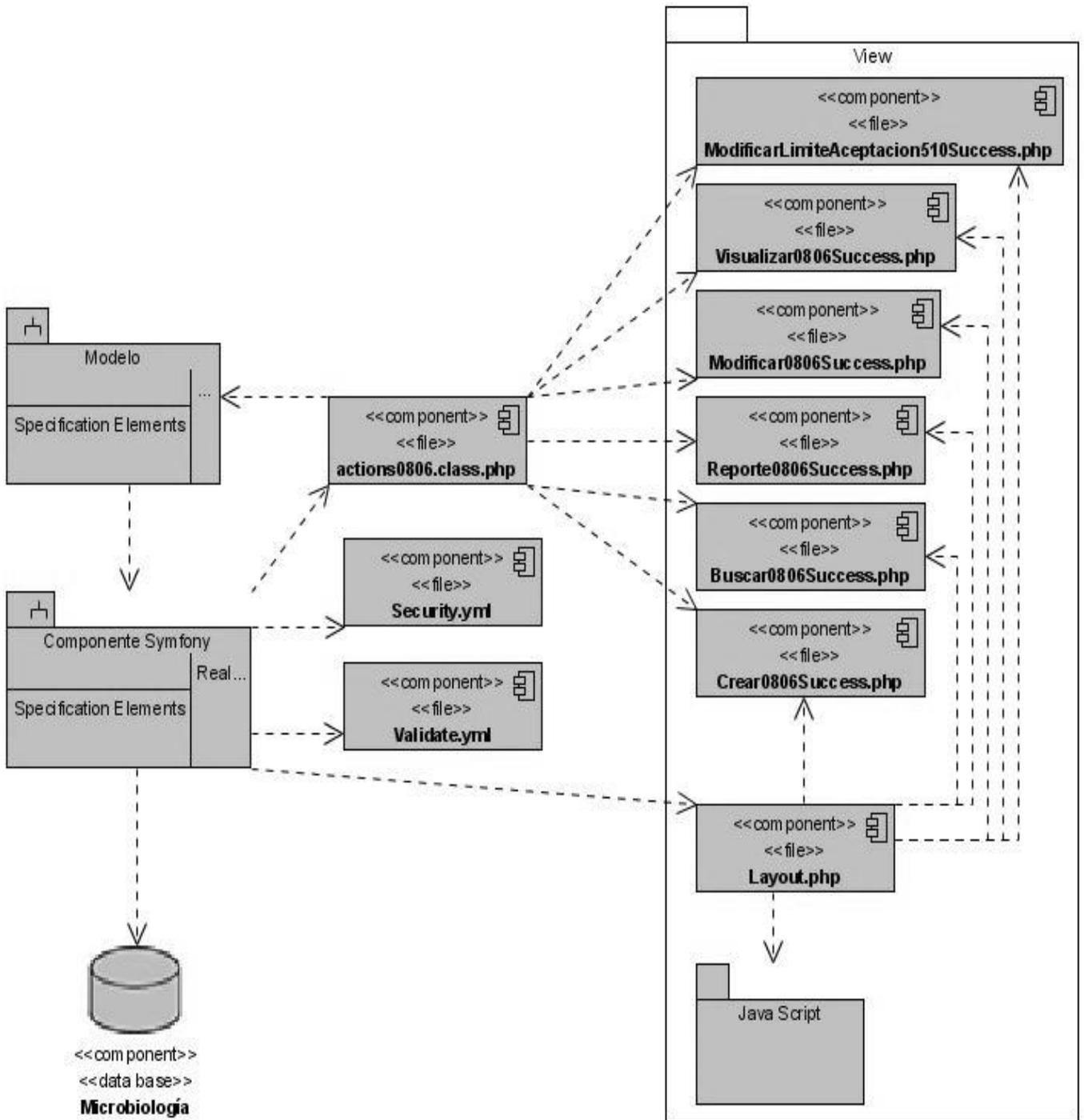


Figura 14. Diagrama de componentes CU Gestionar SIC_0806.

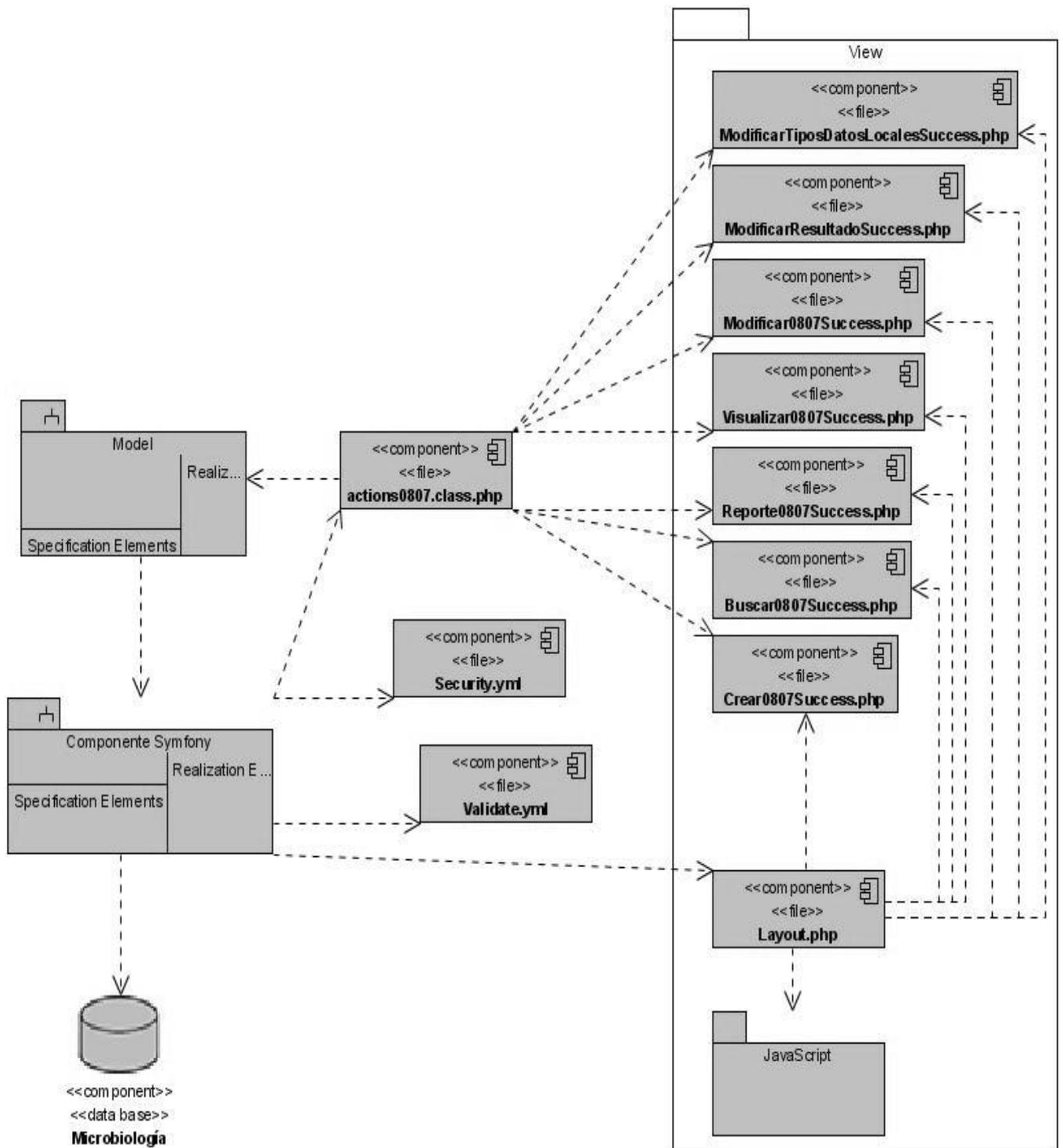


Figura 15. Diagrama de componentes CU Gestionar SIC_0807.

2.4 Estructura de la Arquitectura de Symfony

El framework Symfony, utilizado para la realización del módulo está basado en un patrón clásico del diseño web conocido como arquitectura MVC (modelo-vista-controlador), 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 Figura. 16). [16].

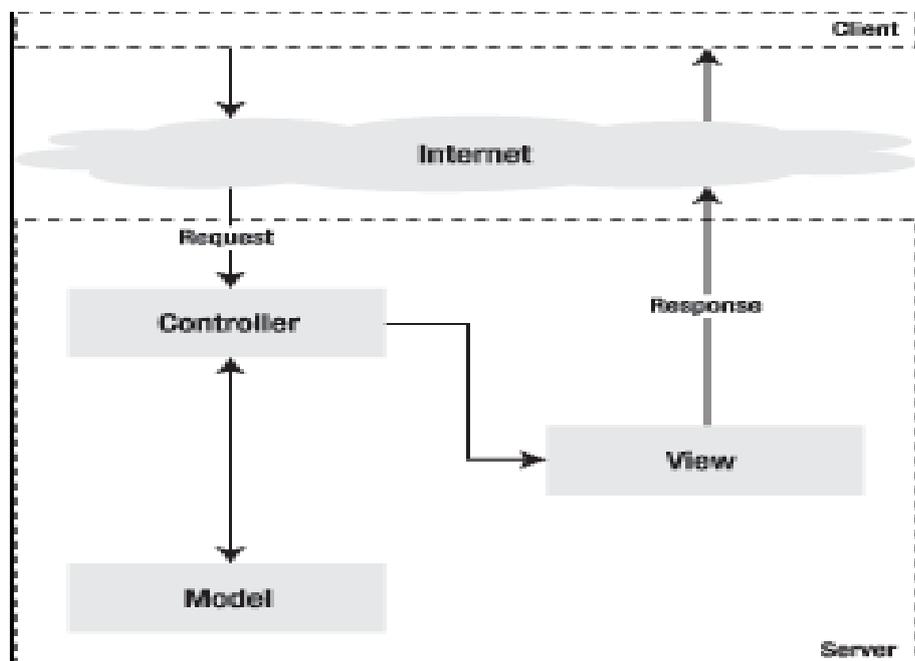


Figura 16. Patrón MVC.

A partir de la descripción de caso de uso mostrada al inicio del capítulo, referente al caso de uso Gestionar Ensayo de Preparación de Soluciones, a continuación se muestra el código fuente de varias de las funciones empleadas para el desarrollo del mismo.

2.5 Código Fuente.

2.5.1 Capa Control.

Las actions son el núcleo de la aplicación, puesto que contienen toda la lógica de la aplicación. Verifican la integridad de las peticiones y preparan los datos requeridos por la capa de presentación.

Capítulo 2: Implementación del Sistema

Estas utilizan el modelo y definen variables para la vista. La clase que representa las actions de un módulo se encuentra en el archivo **actions.class.php**, en el directorio **actions/** del módulo.

Clase actions

action.class.php: En esta clase se implementan métodos a utilizar en la realización del CU Gestionar SIC 0020.

```
<?php

/* SIC0020 actions.
 *
 * @package Tesis
 * @subpackage SIC0020
 * @author Your name here
 * @version SVN: $Id: actions.class.php 2692 2006-11-15 21:03:55Z fabien $
 */
class SIC0020Actions extends sfActions
{
    public function executeModificar0020()
    public function executeModificarReactivos()
    public function executeModificarEquiposUti()
    public function executeModificarPruebasAceptacion()
    public function executeReporte0020()
    public function executeVisualizar0020()
    public function executeCrear0020()
    public function executeBuscar0020()
    public function executeVisualizarBuscar0020()
    public function executeVisualizarReporte0020()

}
```

En lo adelante se muestra la implementación de las funciones que forman parte del actions.class.

Función Crear0020: Esta función permite crear el SIC0020, mediante la entrada de los parámetros que contiene el mismo para su conformación.

```
public function executeCrear0020()
{
    if ($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $this->mensaje="";

        return sfView::SUCCESS;
    }
}
```

```
    }
    else
    {
$fol=$this->getRequestParameter('crear_folio');
$departamento=$this->getRequestParameter('crear_departamento');
$area=$this->getRequestParameter('crear_area');
$nombre_solucion=$this->getRequestParameter('nombre_solucion');
$No_parte=$this->getRequestParameter('crear_No_parte');
$lote=$this->getRequestParameter('crear_lote');
$VT=$this->getRequestParameter('crear_VT');
$fecha_prep=$this->getRequestParameter('fecha_prep');
$solvente=$this->getRequestParameter('crear_solvente');
$ph=$this->getRequestParameter('crear_ph');
$conductividad=$this->getRequestParameter('crear_conductividad');
$ajuste_ph=$this->getRequestParameter('ajuste_ph');
$ph_deseado=$this->getRequestParameter('crear_ph_deseado');
$ph_solucion=$this->getRequestParameter('crear_solucion');
$NP2=$this->getRequestParameter('crear_NP2');
$No_lote=$this->getRequestParameter('crear_No_lote');
$solucion_final=$this->getRequestParameter('crear_solucion_final');
$esterilizacion=$this->getRequestParameter('esterilizacion');
$filtracion=$this->getRequestParameter('filtracion');
$vapor_saturado=$this->getRequestParameter('crear_vapor_saturado');
$min=$this->getRequestParameter('crear_min');
$No_frascos=$this->getRequestParameter('crear_No_frascos');
$VI_frasco=$this->getRequestParameter('crear_VI_frasco');
$almacenar=$this->getRequestParameter('crear_almacenar');
$fecha1=$this->getRequestParameter('crear_fecha');
$observaciones=$this->getRequestParameter('observaciones');
$nombre_apellido=$this->getRequestParameter('crear_nombre_apellido');
$fecha=$this->getRequestParameter('crear_fecha');
$terminado=$this->getRequestParameter('crear_terminado');

$id=Sic0020GepsolucionesPeer::Crear($fol,$No_parte,$lote,$VT,$fecha_prep,$departamento,$area,
$nombre_solucion,$solvente,$ph,$conductividad,$ajuste_ph,$ph_deseado,$ph_solucion,$NP2,
$No_lote,$solucion_final,$esterilizacion,$vapor_saturado,$min,$No_frascos,$VI_frasco,$filtracion,
$almacenar,$fecha1,$observaciones,$nombre_apellido,$fecha,$terminado);

$reactivo=$this->getRequestParameter('reactivo');
$arr_reactivo= explode('*', $reactivo);

$Np=$this->getRequestParameter('nnp');
$arr_Np= explode('*', $Np);
```

Capítulo 2: Implementación del Sistema

```
$cant_usar=$this->getRequestParameter('usar');
$arr_cant_usar= explode('*', $cant_usar);

$cant_utilizada=$this->getRequestParameter('utilizada');
$arr_cant_utilizada= explode('*', $cant_utilizada);

$lote1=$this->getRequestParameter('lote');
$arr_lote1= explode('*', $lote1);

        for($i=0; $i< count($arr_reactivo)-1; $i++)

ReactivosPeer::CrearReactivos($id,$arr_reactivo[$i],$arr_Np[$i],$arr_cant_usar[$i],$arr_cant_utilizada[
$i],$arr_lote1[$i],0);

$fol=$this->getRequestParameter('crear_folio');

$nombre=$this->getRequestParameter('nombre');
$arr_nombre= explode('*', $nombre);

$cod=$this->getRequestParameter('codigo');
$arr_cod= explode('*', $cod);

$fech=$this->getRequestParameter('fecha');
$arr_fech= explode('*', $fech);

        for($i=0; $i< count($arr_nombre)-1; $i++)

EquiposUtilizadosPeer::CrearEquiposUtilizados($id,$arr_nombre[$i],$arr_cod[$i],$arr_fech[$i],0);

$pru=$this->getRequestParameter('prueba');
$arr_pru= explode('*', $pru);

$li=$this->getRequestParameter('limite');
$arr_li= explode('*', $li);

$val=$this->getRequestParameter('valor');
$arr_val= explode('*', $val);

        for($i=0; $i< count($arr_pru)-1; $i++)

PruebasAceptacionPeer::CrearPruebasAceptacion($id,$arr_pru[$i],$arr_li[$i],$arr_val[$i],0);
```

Capítulo 2: Implementación del Sistema

```

ReactivosPeer::InsertarFolioReactivos($id);
EquiposUtilizadosPeer::InsertarFolioEquiposUtilizados($id);
PruebasAceptacionPeer::InsertarFolioPruebasAceptacion($id);
    $this->redirect('SIC0020/Visualizar0020?enviar='.$id);
}
} (Ver Figura. 17).
    
```

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA SIC-0020 PPO 4.09.148.04
DIRECCIÓN DE CALIDAD Edición 03
REGISTRO DE PREPARACIÓN DE SOLUCIONES FOLIO: 101

Departamento: Matemáticas Área: Personal

Nombre de la Solución: Solucion

No. Parte: 1 No. lote: 2 VT: 3 Fecha prep.: 1970-08-21

Reactivos	Np.	Cantidad a usar	Cantidad utilizada	Lote
111	111	111	111	111
3333	333	333	333	333
999999999999999	9999999	9999999	9999999	9999999

Modificar Add

EQUIPOS UTILIZADOS

Nombre	Código	Fecha de venc. calibración
222	222	2002-02-05
4444	4444	2005-02-04
999999999999999	99999999	2005-02-04

Modificar Add

Solvente: 4 pH: 5 Conductividad: 6

Ajuste de pH a la solución: Seleccionar... pH deseado: 7

Solución de ajuste de pH: 8

NP: 9 No de lote: 10

pH final de la solución: 11 Esterilización: Seleccionar...

Filtración: Otros

Vapor saturado: 12 °C: 13 min

Nombre	Código	Fecha de venc. calibración
222	222	2002-02-05
4444	4444	2005-02-04
999999999999999	99999999	2005-02-04

Modificar Add

Solvente: 4 pH: 5 Conductividad: 6

Ajuste de pH a la solución: Seleccionar... pH deseado: 7

Solución de ajuste de pH: 8

NP: 9 No de lote: 10

pH final de la solución: 11 Esterilización: Seleccionar...

Filtración: Otros

Vapor saturado: 12 °C: 13 min

Almacenamiento:

No. de frascos: 14 Volumen por frasco: 16

Almacenar la solución a: 15 Fecha de vencimiento: 1970-10-20

PRUEBAS DE ACEPTACIÓN	LÍMITE	VALOR OBTENIDO
333	333	333
666	666	666
999999999	9999999	9999999

Modificar Add

OBSERVACIONES:

Realizado por	NOMBRE Y APELLIDOS	FIRMA	FECHA
	Microbiología		20020305

Terminado Imprimir Modificar

Figura 17. Interfaz visual de la función Crear del CU Gestionar SIC_0020.

Capítulo 2: Implementación del Sistema

Función Modificar0020: Mediante esta función se le permite al usuario modificar datos previamente entrados a la base de datos.

```
public function executeModificar0020()
    {
        if ($this->getRequest()->getMethod() != sfRequest::POST)
        {
            $id=$this->getRequestParameter('enviar');
            $this->ver=$id;
            $var=ReactivosPeer::BuscarUltFolioReactivos($id);
            $var1=EquiposUtilizadosPeer::BuscarUltFolioEquiposUtilizados($id);
            $var2=PruebasAceptacionPeer::BuscarUltFolioPruebasAceptacion($id);

            $this->dev=ReactivosPeer::BuscarArregloReactivos($var);
            $this->dev2=EquiposUtilizadosPeer::BuscarArregloEquiposUtilizados($var1);
            $this->dev3=PruebasAceptacionPeer::BuscarArregloPruebasAceptacion($var2);
            $this->dev1=Sic0020GepsolucionesPeer::BuscarPorFolio($id);
            return sfView::SUCCESS;
        }
        else
        {
            $fol=$this->getRequestParameter('crear_folio');

            $reactivo=$this->getRequestParameter('reactivo');
            $arr_reactivo= explode('*', $reactivo);
            $Np=$this->getRequestParameter('nnp');
            $arr_Np= explode('*', $Np);
            $cant_usar=$this->getRequestParameter('usar');
            $arr_cant_usar= explode('*', $cant_usar);
            $cant_utilizada=$this->getRequestParameter('utilizada');
            $arr_cant_utilizada= explode('*', $cant_utilizada);
            $lote1=$this->getRequestParameter('lote');
            $arr_lote1= explode('*', $lote1);

            for($i=0; $i< count($arr_reactivo)-1; $i++)

            ReactivosPeer::CrearReactivos($fol,$arr_reactivo[$i],$arr_Np[$i],$arr_cant_usar[$i],
            $arr_cant_utilizada[$i],$arr_lote1[$i],1);

            $fol=$this->getRequestParameter('crear_folio');
            $nombre=$this->getRequestParameter('nombre');
            $arr_nombre= explode('*', $nombre);
```

Capítulo 2: Implementación del Sistema

```
$cod=$this->getRequestParameter('codigo');
$arr_cod= explode('*', $cod);
$fech=$this->getRequestParameter('fecha');
$arr_fech= explode('*', $fech);
        for($i=0; $i< count($arr_reactivo)-1; $i++)
```

```
EquiposUtilizadosPeer::CrearEquiposUtilizados($fol,$arr_nombre[$i],$arr_cod[$i],$arr_fech[$i],1);
$fol=$this->getRequestParameter('crear_folio');
$pru=$this->getRequestParameter('prueba');
$arr_pru= explode('*', $pru);
$li=$this->getRequestParameter('limite');
$arr_li= explode('*', $li);
$val=$this->getRequestParameter('valor');
$arr_val= explode('*', $val);
        for($i=0; $i< count($arr_reactivo)-1; $i++)
```

```
PruebasAceptacionPeer::CrearPruebasAceptacion($fol,$arr_pru[$i],$arr_li[$i],$arr_val[$i],1);
```

```
$departamento=$this->getRequestParameter('crear_departamento');
$area=$this->getRequestParameter('crear_area');
$nombre_solucion=$this->getRequestParameter('nombre_solucion');
$No_parte=$this->getRequestParameter('crear_No_parte');
$lote=$this->getRequestParameter('crear_lote');
$VT=$this->getRequestParameter('crear_VT');
$fecha_prep=$this->getRequestParameter('fecha_prep');
$solvente=$this->getRequestParameter('crear_solvente');
$ph=$this->getRequestParameter('crear_ph');
$conductividad=$this->getRequestParameter('crear_conductividad');
$ajuste_ph=$this->getRequestParameter('ajuste_ph');
$ph_deseado=$this->getRequestParameter('crear_ph_deseado');
$ph_solucion=$this->getRequestParameter('crear_solucion');
$NP2=$this->getRequestParameter('crear_NP2');
$No_lote=$this->getRequestParameter('crear_No_lote');
$solucion_final=$this->getRequestParameter('crear_solucion_final');
$esterilizacion=$this->getRequestParameter('esterilizacion');
$filtracion=$this->getRequestParameter('filtracion');
$vapor_saturado=$this->getRequestParameter('crear_vapor_saturado');
$min=$this->getRequestParameter('crear_min');
$No_frascos=$this->getRequestParameter('crear_No_frascos');
$VI_frasco=$this->getRequestParameter('crear_VI_frasco');
$almacenar=$this->getRequestParameter('crear_almacenar');
$fecha1=$this->getRequestParameter('crear_fecha');
$observaciones=$this->getRequestParameter('observaciones');
$nombre_apellido=$this->getRequestParameter('crear_nombre_apellido');
```

Capítulo 2: Implementación del Sistema

```
$fecha=$this->getRequestParameter('crear_fecha');
$terminado=$this->getRequestParameter('crear_terminado');

$SIC0020=Sic0020GepsolucionesPeer::retrieveByPK($folio);

$SIC0020->setNoparte($No_parte);
$SIC0020->setNoLote($lote);
$SIC0020->setVt($VT);
$SIC0020->setFechapreparacion($fecha_prep);
$SIC0020->setDepartamento($departamento);
$SIC0020->setArea($area);
$SIC0020->setNombresolucion($nombre_solucion);
$SIC0020->setSolvente($solvente);
$SIC0020->setPH($ph);
$SIC0020->setConductividad($conductividad);
$SIC0020->setAjustepHsol($ajuste_ph);
$SIC0020->setPHdeseado($ph_deseado);
$SIC0020->setSolajustepH($ph_solucion);
$SIC0020->setNp($NP2);
$SIC0020->setNoDeLote($No_lote);
$SIC0020->setPHfinalsol($solucion_final);
$SIC0020->setEsterilizacion($esterilizacion);
$SIC0020->setVaporsaturado($vapor_saturado);
$SIC0020->setTiempo($min);
$SIC0020->setNofrascosalmacen($No_frascos);
$SIC0020->setVolumenfrasco($VI_frasco);
$SIC0020->setFiltracion($filtracion);
$SIC0020->setAlmacenarsol($almacenar);
$SIC0020->setFechavencimiento($fecha1);
$SIC0020->setObservaciones($observaciones);
$SIC0020->setRealizadoPor($nombre_apellido);
$SIC0020->setFecharealizado($fecha);
$SIC0020->save();

        if($this->getRequestParameter('crear_terminado')!=1)
            $this->mensaje="Esta Terminado";
        else
            $this->mensaje="No esta Terminado";
            $this->redirect('SIC0020/Visualizar0020?enviar='.$ver);
    }
} (Ver Figura. 18).
```

Capítulo 2: Implementación del Sistema

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA		SIC-0020	PPO 4.09.148.04
DIRECCIÓN DE CALIDAD		Edición 03	
REGISTRO DE PREPARACIÓN DE SOLUCIONES			FOLIO: 101
Departamento: Matemáticas		Área: Personal	
Nombre de la Solución: Solución			
No. Parte: 1	No. lote: 2	VT: 3	Fecha prep.: 1970-03-21
Reactivos	Np	Cantidad a usar	Cantidad utilizada
111	111	111	111
333	333	333	333
99999999999999	99999999	99999999	99999999
: Modificar :			
Add			
EQUIPOS UTILIZADOS			
Nombre	Código	Fecha de venc. calibración	
222	2222	2002-02-05	
4444	4444	2005-02-04	
99999999999999	99999999	2005-02-04	
: Modificar :			
Add			
Solvente: 4	pH: 5	Conductividad: 8	
Ajuste de pH a la solución: Seleccionar... pH deseado: 7			
Solución de ajuste de pH: 8			
Np: 9	No de lote: 10		
pH final de la solución: 11 Esterilización: Seleccionar...			
Filtración: Otros			
Vapor saturado: 12 °C : 13 min			
222	2222	2002-02-05	
4444	4444	2005-02-04	
99999999999999	99999999	2005-02-04	
: Modificar :			
Add			
Solvente: 4	pH: 5	Conductividad: 6	
Ajuste de pH a la solución: Seleccionar... pH deseado: 7			
Solución de ajuste de pH: 8			
Np: 9	No de lote: 10		
pH final de la solución: 11 Esterilización: Seleccionar...			
Filtración: Otros			
Vapor saturado: 12 °C : 13 min			
Almacenamiento:			
No. de frascos: 4	Volumen por frasco: 16		
Almacenar la solución a: 15 Fecha de vencimiento: 1970-06-20			
PRUEBAS DE ACEPTACIÓN	LÍMITE	VALOR OBTENIDO	
333	333	333	
666	666	666	
99999999999999	99999999	99999999	
: Modificar :			
Add			
OBSERVACIONES:			
Realizado por	NOMBRE Y APELLIDOS	FIRMA	FECHA
	Microbiología		20020325
Terminado <input checked="" type="checkbox"/>			
			Imprimir Modificar

Figura 18. Interfaz visual de la función Modificar del CU Gestionar SIC_0020.

Capítulo 2: Implementación del Sistema

Función Reporte0020: Esta función emite un reporte mediante una de serie de parámetros que el usuario selecciona.

```
public function executeReporte0020()
{
    $nombre_solucion=array('Select...');
    $lista_nombres=Sic0020GepsolucionesPeer::BuscarNombreSolucion();

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

    array_push($nombre_solucion,$lista_nombres[$i]);
    $this->lista_nomb=$nombre_solucion;
    $nueva=$this->getRequestParameter('Nombre');

    if($nueva==0)
    {
        $this->mensaje='Usted Debe Seleccionar el Nombre de la Solucion';
    }
    else
    {
        $c= new Criteria();
        $this->mensaje="";
    }
}
} (Ver Figura. 19 y 20).
```

sf 1.0.11 vars & config logs & msgg 1 5957.1 KB 3565 ms X

CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA
DIRECCIÓN DE CALIDAD
REGISTRO DE PREPARACIÓN DE SOLUCIONES

Parámetros de Búsqueda
Seleccionar nombre de la solución: Solucion (dropdown) Usted Debe Seleccionar el Nombre de la Solucion
Seleccionar rango de fecha: Desde 2002-05-04 hasta 2002-05-04

Seleccione los parámetros que desea visualizar en el Reporte

<input checked="" type="checkbox"/> Departamento	<input checked="" type="checkbox"/> No. lote	<input checked="" type="checkbox"/> pH	<input checked="" type="checkbox"/> Solución de ajuste de pH
<input checked="" type="checkbox"/> Área	<input checked="" type="checkbox"/> VT	<input checked="" type="checkbox"/> Conductividad	<input checked="" type="checkbox"/> pH final de la solución
<input checked="" type="checkbox"/> Nombre de la Solución	<input checked="" type="checkbox"/> Fecha prep.	<input checked="" type="checkbox"/> Ajuste de pH a la solución	<input checked="" type="checkbox"/> Realizado por
<input checked="" type="checkbox"/> No. Parte	<input checked="" type="checkbox"/> Solvente	<input checked="" type="checkbox"/> pH deseado	

Cancelar Generar Reporte

Figura 19. Interfaz visual de la función Reporte del CU Gestionar SIC_0020.

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA							
DIRECCIÓN DE CALIDAD							
REGISTRO DE PREPARACIÓN DE SOLUCIONES							
Nombre_Solucion:Solucion							
Dpto	Area	Nombre	#Parte	#Lote	VT	Fecha prep:	Solvente
Matematicas	Personal	Solucion	1	2	3	20080710	4
PH	Conductividad	Ajuste	PH Deseado	Solucion	PH Final	Realizado por:	
5	6	1	7	8	11	Microbiologia	

Figura 20. Interfaz visual de la función Visualizar Reporte del CU Gestionar SIC_0020.

Función Visualizar0020: Esta función visualiza los campos entrados previamente en la función Crear0020 expuesta con anterioridad.

```

public function executeVisualizar0020()
{
    $id=$this->getRequestParameter('enviar');
    $this->pepe=$id;
    $arreglo=Sic0020GepsolucionesPeer::retrieveByPK($id);
    if(count($arreglo)>0)
    $this->ver=Sic0020GepsolucionesPeer::BuscarArregPrincipal($arreglo);
    else
    $this->ver=$id;

    $arre=ReactivosPeer::BuscarUltFolioReactivos($id);
    $arre1=EquiposUtilizadosPeer::BuscarUltFolioEquiposUtilizados($id);
    $arre2=PruebasAceptacionPeer::BuscarUltFolioPruebasAceptacion($id);
    $this->verr=ReactivosPeer::BuscarArregloReactivos($arre);
    $this->ver1=EquiposUtilizadosPeer::BuscarArregloEquiposUtilizados($arre1);
    $this->ver2=PruebasAceptacionPeer::BuscarArregloPruebasAceptacion($arre2);

    if($this->getRequestParameter('crear_terminado')!=1)

    $this->mensaje="Esta Terminado";
    else

    $this->mensaje="No esta Terminado";

} (Ver Figura. 21).
    
```

Capítulo 2: Implementación del Sistema

DIRECCION DE CALIDAD		Edición 03	
REGISTRO DE PREPARACIÓN DE SOLUCIONES:			FOLIO: 101
Departamento: Matemáticas		Area: Personal	
Nombre de la Solución: Solución			
No. Parte: 1	No. lote: 2	VT: 3	Fecha prep. 2008-07-10
Reactivos	Np	Cantidad a usar	Cantidad utilizada
111	111	111	111
3333	333	333	333
9999999999999999	99999999	99999999	999999
EQUIPOS UTILIZADOS			
Nombre	Código	Fecha de venc. calibración	
222	2222	2002-02-05	
4444	4444	2005-02-04	
999999999999	99999999	2005-02-04	
Solvente 4 pH No Conductividad 6 Ajuste de pH a la solución: 5l pH deseado: 7 Solución de ajuste de pH: 8 NP: 9 No de lote: 10 pH final de la solución: 11 Esterilización: 5l Filtración: Membrana 0.22 Vapor saturado 13 °C: 14 min Almacenamiento: No. de frascos: 16 Volumen por frasco: 1 Almacenar la solución a: 15 Fecha de vencimiento: 20020205			
PRUEBAS DE ACEPTACION	LIMITE	VALOR OBTENIDO	
333	333	333	
666	666	666	
9999999999	99999999	99999999	
OBSERVACIONES: Esto es una Prueba para que el tribunal Evalúe. Gracias. (Técnicos: Manuel Marrero y Pedro Luis).			
Realizado por	NOMBRE Y APELLIDOS	FIRMA	FECHA
	Microbiología		2002-02-05
Terminado <input checked="" type="checkbox"/> Esta Terminado			
		Imprimir	Modificar

Figura 21. Interfaz visual de la función Visualizar del CU Gestionar SIC_0020.

Función Buscar0020: A través de los parámetros de búsqueda para cada SIC, se realiza una búsqueda minuciosa de lo que se desea.

```

public function executeBuscar0020()
{
$verdep=array('Select...');
$verrealizado=array('Select...');
$lista_dep=Sic0020GepsolucionesPeer::BuscarPorDepartamento();
$lista_realpor=Sic0020GepsolucionesPeer::BuscarRealizadoPor();

for($i=0;$i<count($lista_dep);$i++)
array_push($verdep,$lista_dep[$i]);

```

Capítulo 2: Implementación del Sistema

```
        for($i=0;$i<count($lista_realpor);$i++)
            array_push($verrealizado,$lista_realpor[$i]);
$this->select_dep=$verdep;
$this->select_real=$verrealizado;
$dep=$this->getRequestParameter('crear_departamento');
$real=$this->getRequestParameter('realizado_por');
$f_i = $this->getRequestParameter('fecha_inicial');
$f_f= $this->getRequestParameter('fecha_final');
    if($dep==0 && $real==0)
    {
        $this->mensaje='Entre un Criterio de Busqueda';
        $this->ver=-1;
    }
    else
    {
        $dev=array();
        $dev_arreglo=array();
        $dev1=array();
        $c= new Criteria();
        $this->mensaje="";
        if($depl=0)
        {
            $depart=$verdep[$dep];
            $c->add(Sic0020GepsolucionesPeer::DEPARTAMENTO,$depart);
        }
        if($real!=0)
        {
            $realiza=$verrealizado[$real];
            $c->add(Sic0020GepsolucionesPeer::REALIZADO_POR,$realiza);
        }
        $sic0020=Sic0020GepsolucionesPeer::doSelect($c);
        $dev1=Sic0020GepsolucionesPeer::BuscarArregPrincipal1($sic0020);

        if($f_f!=" && $f_i!=")
        {
            $fecha_i=str_replace('-',",$f_i);
            $fecha_f=str_replace('-',",$f_f);
            $fecha=array();

            for($i=0;$i<count($dev1);$i++)
            {
                $arr_aux=array();
```

```
        $fecha=$dev1[$i]->getFecha();
        if($fecha>=$fecha_i && $fecha<=$fecha_f)
            array_push($arr_aux,$dev1[$i]);
    }
    $dev_arreglo=$arr_aux;
}
else
    $dev_arreglo=$dev1;
$this->mensaje="";
$this->ver=$dev_arreglo;
}
} (Ver Figura. 22).
```

FOLIO	Nombre de la Solución	No. LOTE	F. Preparación	No. Parte	
101	Solucion	2	20020205	1	Visualizar

Figura 22. Interfaz visual de la función Buscar del CU Gestionar SIC_0020.

2.5.2 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 ORM (Object Relational Mapping), 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 ese mapeo o 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. También 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, dígase 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 relacionada. [16].

A continuación se muestra el código una clase peer y más adelante el de una clase base peer.

Clase Peer:

class Sic0020GepsolucionesPeer: En esta clase se implementan métodos a utilizar por la clase actions.class.php del caso de uso Gestionar SIC 0020, además de las funcionalidades que brinda Symfony.

```
abstract class BaseSic0020Gepsoluciones extends BaseObject implements Persistent
{
protected static $peer;
protected $sic_20_id;
protected $folio;
protected $no_parte;
protected $no_lote;
protected $terminado;

protected $collEquiposUtilizadoss;
protected $lastEquiposUtilizadosCriteria = null;
protected $collPruebasAceptacions;
protected $lastPruebasAceptacionCriteria = null;
protected $collReactivoss;
protected $lastReactivosCriteria = null;
protected $alreadyInSave = false;
protected $alreadyInValidation = false;

public function getSic20Id()
    {
        return $this->sic_20_id;
    }
public function getFolio()
    {
        return $this->folio;
    }
public function getNoParte()
    {
```

```
        return $this->no_parte;
    }
    public function getNoLote()
    {
        return $this->no_lote;
    }
    public function getTerminado()
    {
        return $this->terminado;
    }

    public function setSic20Id($v)
    {
        if ($v !== null && !is_string($v)) {
            $v = (string) $v;
        }
        if ($this->sic_20_id !== $v) {
            $this->sic_20_id = $v;
            $this->modifiedColumns[] = Sic0020GepsolucionesPeer::SIC_20_ID;
        }
    }
    public function setFolio($v)
    {
        if ($v !== null && !is_int($v) && is_numeric($v)) {
            $v = (int) $v;
        }
        if ($this->folio !== $v) {
            $this->folio = $v;
            $this->modifiedColumns[] = Sic0020GepsolucionesPeer::FOLIO;
        }
    }
    public function setNoParte($v)
    {
        if ($v !== null && !is_string($v)) {
            $v = (string) $v;
        }
        if ($this->no_parte !== $v) {
            $this->no_parte = $v;
            $this->modifiedColumns[] = Sic0020GepsolucionesPeer::NO_PARTE;
        }
    }
    public function setNoLote($v)
    {
        if ($v !== null && !is_string($v)) {
```

Capítulo 2: Implementación del Sistema

```
        $v = (string) $v;
    }
    if ($this->no_lote !== $v) {
        $this->no_lote = $v;
        $this->modifiedColumns[] = Sic0020GepsolucionesPeer::NO_LOTE;
    }
}

public function setTerminado($v)
{
    if ($this->terminado !== $v) {
        $this->terminado = $v;
        $this->modifiedColumns[] = Sic0020GepsolucionesPeer::TERMINADO;
    }
}

public function hydrate(ResultSet $rs, $startcol = 1)
{
    try {

        $this->sic_20_id = $rs->getString($startcol + 0);
        $this->folio = $rs->getInt($startcol + 1);
        $this->no_parte = $rs->getString($startcol + 2);
        $this->no_lote = $rs->getString($startcol + 3);
        $this->terminado = $rs->getBoolean($startcol + 29);
        $this->resetModified();

        $this->setNew(false);

        return $startcol + 30;
    } catch (Exception $e) {
        throw new PropelException("Error populating Sic0020Gepsoluciones object", $e);
    }
}

public function delete($con = null)
{
    if ($this->isDeleted()) {
        throw new PropelException("This object has already been deleted.");
    }
    if ($con === null) {
        $con = Propel::getConnection(Sic0020GepsolucionesPeer::DATABASE_NAME);
    }
    try {
        $con->begin();
    }
}
```

Capítulo 2: Implementación del Sistema

```
        Sic0020GepsolucionesPeer::doDelete($this, $con);
        $this->setDeleted(true);
        $con->commit();
    } catch (PropelException $e) {
        $con->rollback();
        throw $e;
    }
}

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(Sic0020GepsolucionesPeer::DATABASE_NAME);
    }
    try {
        $con->begin();
        $affectedRows = $this->doSave($con);
        $con->commit();
        return $affectedRows;
    } catch (PropelException $e) {
        $con->rollback();
        throw $e;
    }
}

protected function doSave($con)
{
    $affectedRows = 0;
    if (!$this->alreadyInSave)
    {
        $this->alreadyInSave = true;

        if ($this->isModified()) {
            if ($this->isNew()) {
                $pk = Sic0020GepsolucionesPeer::doInsert($this, $con);
                $affectedRows += 1;

                $this->setSic20Id($pk);
                $this->setNew(false);
            }
        }
    }
    else
```

Capítulo 2: Implementación del Sistema

```
{
    $affectedRows += Sic0020GepsolucionesPeer::doUpdate($this, $con);
}
$this->resetModified();
}

if ($this->collEquiposUtilizadoss !== null) {
    foreach($this->collEquiposUtilizadoss as $referrerFK) {
        if (!$referrerFK->isDeleted()) {
            $affectedRows += $referrerFK->save($con);
        }
    }
}

if ($this->collPruebasAceptacions !== null) {
    foreach($this->collPruebasAceptacions as $referrerFK) {
        if (!$referrerFK->isDeleted()) {
            $affectedRows += $referrerFK->save($con);
        }
    }
}

if ($this->collReactivoss !== null) {
    foreach($this->collReactivoss as $referrerFK) {
        if (!$referrerFK->isDeleted()) {
            $affectedRows += $referrerFK->save($con);
        }
    }
}

$this->alreadyInSave = false;
}
return $affectedRows;
}

protected $validationFailures = array();

public function getValidationFailures()
{
    return $this->validationFailures;
}

public function validate($columns = null)
{

```

Capítulo 2: Implementación del Sistema

```
$res = $this->doValidate($columns);
if ($res === true) {
    $this->validationFailures = array();
    return true;
} else {
    $this->validationFailures = $res;
    return false;
}
}
protected function doValidate($columns = null)
{
    if (!$this->alreadyInValidation) {
        $this->alreadyInValidation = true;
        $retval = null;

        $failureMap = array();

        if (($retval = Sic0020GepsolucionesPeer::doValidate($this, $columns)) !== true) {
            $failureMap = array_merge($failureMap, $retval);
        }

        if ($this->collEquiposUtilizadoss !== null) {
            foreach($this->collEquiposUtilizadoss as $referrerFK) {
                if (!$referrerFK->validate($columns)) {
                    $failureMap = array_merge($failureMap,
                    $referrerFK->getValidationFailures());
                }
            }
        }

        if ($this->collPruebasAceptacions !== null) {
            foreach($this->collPruebasAceptacions as $referrerFK) {
                if (!$referrerFK->validate($columns)) {
                    $failureMap = array_merge($failureMap, $referrerFK->getValidationFailures());
                }
            }
        }

        if ($this->collReactivoss !== null) {
            foreach($this->collReactivoss as $referrerFK) {
                if (!$referrerFK->validate($columns)) {
                    $failureMap = array_merge($failureMap, $referrerFK->getValidationFailures());
                }
            }
        }
    }
}
```

Capítulo 2: Implementación del Sistema

```
        $this->alreadyInValidation = false;
    }

    return (!empty($failureMap) ? $failureMap : true);
}

public function getByName($name, $type = BasePeer::TYPE_PHPNAME)
{
    $pos = Sic0020GepsolucionesPeer::translateFieldName($name,$type, BasePeer::TYPE_NUM);

    return $this->getPosition($pos);
}

public function getPosition($pos)
{
    switch($pos) {
        case 0:
            return $this->getSic20Id();
            break;
        case 1:
            return $this->getFolio();
            break;
        case 2:
            return $this->getNoParte();
            break;
        case 3:
            return $this->getNoLote();
            break;
        case 4:
            return $this->getVt();
            break;
        default:
            return null;
            break;
    }
}

public function toArray($keyType = BasePeer::TYPE_PHPNAME)
{
    $keys = Sic0020GepsolucionesPeer::getFieldNames($keyType);
    $result = array(
        $keys[0] => $this->getSic20Id(),
        $keys[1] => $this->getFolio(),
        $keys[2] => $this->getNoParte(),
        $keys[3] => $this->getNoLote(),
    );
}
```

Capítulo 2: Implementación del Sistema

```
        $keys[29] => $this->getTerminado(),
    );
    return $result;
}
```

```
public function setName($name, $value, $type = BasePeer::TYPE_PHPNAME)
{
    $pos = Sic0020GepsolucionesPeer::translateFieldName($name, $type, BasePeer::TYPE_NUM);

    return $this->setByPosition($pos, $value);
}
```

```
public function setByPosition($pos, $value)
{
    switch($pos) {
        case 0:
            $this->setSic20Id($value);
            break;
        case 1:
            $this->setFolio($value);
            break;
        case 2:
            $this->setNoParte($value);
            break;
        case 3:
            $this->setNoLote($value);
            break;
        case 4:
            $this->setVt($value);
            break;
        case 29:
            $this->setTerminado($value);
            break;
    }
}
```

```
public function fromArray($arr, $keyType = BasePeer::TYPE_PHPNAME)
{
    $keys = Sic0020GepsolucionesPeer::getFieldNames($keyType);

    if (array_key_exists($keys[0], $arr)) $this->setSic20Id($arr[$keys[0]]);
    if (array_key_exists($keys[1], $arr)) $this->setFolio($arr[$keys[1]]);
    if (array_key_exists($keys[2], $arr)) $this->setNoParte($arr[$keys[2]]);
    if (array_key_exists($keys[3], $arr)) $this->setNoLote($arr[$keys[3]]);
}
```

Capítulo 2: Implementación del Sistema

```
        if (array_key_exists($keys[29], $arr)) $this->setTerminado($arr[$keys[29]]);
    }

    public function buildCriteria()
    {
        $criteria = new Criteria(Sic0020GepsolucionesPeer::DATABASE_NAME);

        if ($this->isColumnModified(Sic0020GepsolucionesPeer::SIC_20_ID))
            $criteria->add(Sic0020GepsolucionesPeer::SIC_20_ID, $this->sic_20_id);

        if ($this->isColumnModified(Sic0020GepsolucionesPeer::FOLIO))
            $criteria->add(Sic0020GepsolucionesPeer::FOLIO, $this->folio);

        if ($this->isColumnModified(Sic0020GepsolucionesPeer::NO_PARTE))
            $criteria->add(Sic0020GepsolucionesPeer::NO_PARTE, $this->no_parte);

        if ($this->isColumnModified(Sic0020GepsolucionesPeer::NO_LOTE))
            $criteria->add(Sic0020GepsolucionesPeer::NO_LOTE, $this->no_lote);

        if ($this->isColumnModified(Sic0020GepsolucionesPeer::TERMINADO))
            $criteria->add(Sic0020GepsolucionesPeer::TERMINADO, $this->terminado);

        return $criteria;
    }

    public function buildPkeyCriteria()
    {
        $criteria = new Criteria(Sic0020GepsolucionesPeer::DATABASE_NAME);
        $criteria->add(Sic0020GepsolucionesPeer::SIC_20_ID, $this->sic_20_id);
        return $criteria;
    }

    public function getPrimaryKey()
    {
        return $this->getSic20Id();
    }

    public function setPrimaryKey($key)
    {
        $this->setSic20Id($key);
    }

    public function copyInto($copyObj, $deepCopy = false)
    {
```

Capítulo 2: Implementación del Sistema

```
$copyObj->setFolio($this->folio);
$copyObj->setNoParte($this->no_parte);
$copyObj->setNoLote($this->no_lote);
$copyObj->setTerminado($this->terminado);

if ($deepCopy) {
    $copyObj->setNew(false);

    foreach($this->getEquiposUtilizadoss() as $relObj) {
        $copyObj->addEquiposUtilizados($relObj->copy($deepCopy));
    }
    foreach($this->getPruebasAceptacions() as $relObj) {
        $copyObj->addPruebasAceptacion($relObj->copy($deepCopy));
    }
    foreach($this->getReactivoss() as $relObj) {
        $copyObj->addReactivos($relObj->copy($deepCopy));
    }
}

$copyObj->setNew(true);
$copyObj->setSic20Id(NULL);
}
```

```
public function copy($deepCopy = false)
{
    $clazz = get_class($this);
    $copyObj = new $clazz();
    $this->copyInto($copyObj, $deepCopy);
    return $copyObj;
}
```

```
public function getPeer()
{
    if (self::$peer === null)
    {
        self::$peer = new Sic0020GepsolucionesPeer();
    }
    return self::$peer;
}
```

```
public function initEquiposUtilizadoss()
{
    if ($this->collEquiposUtilizadoss === null)
```

Capítulo 2: Implementación del Sistema

```
    {
        $this->colEquiposUtilizadoss = array();
    }
}
```

```
public function getEquiposUtilizadoss($criteria = null, $con = null)
```

```
{
    include_once 'lib/model/om/BaseEquiposUtilizadosPeer.php';
    if ($criteria === null)
    {
        $criteria = new Criteria();
    }
    elseif ($criteria instanceof Criteria)
    {
        $criteria = clone $criteria;
    }
}
```

```
if ($this->colEquiposUtilizadoss === null)
{
    if ($this->isNew())
    {
        $this->colEquiposUtilizadoss = array();
    }
    else
    {
```

```
$criteria->add(EquiposUtilizadosPeer::SIC_0020_GEPSOLUCIONESSIC_20_ID, $this->getSic20Id());
EquiposUtilizadosPeer::addSelectColumns($criteria);
$this->colEquiposUtilizadoss = EquiposUtilizadosPeer::doSelect($criteria, $con);
```

```
    }
}
else
{
    if (!$this->isNew())
    {
```

```
$criteria->add(EquiposUtilizadosPeer::SIC_0020_GEPSOLUCIONESSIC_20_ID, $this->getSic20Id());
EquiposUtilizadosPeer::addSelectColumns($criteria);
```

```
if (!isset($this->lastEquiposUtilizadosCriteria) || !$this->lastEquiposUtilizadosCriteria->equals($criteria))
{
    $this->colEquiposUtilizadoss = EquiposUtilizadosPeer::doSelect($criteria, $con);
}
}
```

Capítulo 2: Implementación del Sistema

```
    }
    $this->lastEquiposUtilizadosCriteria = $criteria;
    return $this->collEquiposUtilizados;
}

public function countEquiposUtilizados($criteria = null, $distinct = false, $con = null)
{
    include_once 'lib/model/om/BaseEquiposUtilizadosPeer.php';
    if ($criteria === null)
    {
        $criteria = new Criteria();
    }
    elseif ($criteria instanceof Criteria)
    {
        $criteria = clone $criteria;
    }
    $criteria->add(EquiposUtilizadosPeer::SIC_0020_GEPSOLUCIONESSIC_20_ID, $this->getSic20Id());
    return EquiposUtilizadosPeer::doCount($criteria, $distinct, $con);
}

public function addEquiposUtilizados(EquiposUtilizados $I)
{
    $this->collEquiposUtilizados[] = $I;
    $I->setSic0020Gepsoluciones($this);
}

public function initPruebasAceptacions()
{
    if ($this->collPruebasAceptacions === null)
    {
        $this->collPruebasAceptacions = array();
    }
}

public function getPruebasAceptacions($criteria = null, $con = null)
{
    include_once 'lib/model/om/BasePruebasAceptacionPeer.php';
    if ($criteria === null)
    {
        $criteria = new Criteria();
    }
    elseif ($criteria instanceof Criteria)
    {
        $criteria = clone $criteria;
    }
}
```

Capítulo 2: Implementación del Sistema

```
    }
    if ($this->collPruebasAceptacions === null)
    {
        if ($this->isNew())
        {
            $this->collPruebasAceptacions = array();
        }
        else
        {
            $criteria->add(PruebasAceptacionPeer::SIC_0020_GEPSOLUCIONESSIC_20_ID,
            $this->getSic20Id());
            PruebasAceptacionPeer::addSelectColumns($criteria);
            $this->collPruebasAceptacions = PruebasAceptacionPeer::doSelect($criteria, $con);
        }
    } else {
        if (!$this->isNew())
        {
            $criteria->add(PruebasAceptacionPeer::SIC_0020_GEPSOLUCIONESSIC_20_ID,$this->getSic20Id());
            PruebasAceptacionPeer::addSelectColumns($criteria);
            if (!isset($this->lastPruebasAceptacionCriteria) ||
            !$this->lastPruebasAceptacionCriteria->equals($criteria)) {
                $this->collPruebasAceptacions = PruebasAceptacionPeer::doSelect($criteria, $con);
            }
        }
        $this->lastPruebasAceptacionCriteria = $criteria;
        return $this->collPruebasAceptacions;
    }
}

public function countPruebasAceptacions($criteria = null, $distinct = false, $con = null)
{
    include_once 'lib/model/om/BasePruebasAceptacionPeer.php';
    if ($criteria === null)
    {
        $criteria = new Criteria();
    }
    elseif ($criteria instanceof Criteria)
    {
        $criteria = clone $criteria;
    }

    $criteria->add(PruebasAceptacionPeer::SIC_0020_GEPSOLUCIONESSIC_20_ID,$this->getSic20Id());
```

Capítulo 2: Implementación del Sistema

```
        return PruebasAceptacionPeer::doCount($criteria, $distinct, $con);
    }

public function addPruebasAceptacion(PruebasAceptacion $I)
    {
        $this->collPruebasAceptaciones[] = $I;
        $I->setSic0020Gepsoluciones($this);
    }

public function initReactivoss()
    {
        if ($this->collReactivoss === null)
            {
                $this->collReactivoss = array();
            }
    }

public function getReactivoss($criteria = null, $con = null)
    {
        include_once 'lib/model/om/BaseReactivosPeer.php';
        if ($criteria === null)
            {
                $criteria = new Criteria();
            }
        elseif ($criteria instanceof Criteria)
            {
                $criteria = clone $criteria;
            }

        if ($this->collReactivoss === null)
            {
                if ($this->isNew()) {
                    $this->collReactivoss = array();
                } else {

$criteria->add(ReactivosPeer::SIC_0020_GEPSOLUCIONESSIC_20_ID, $this->getSic20Id());

ReactivosPeer::addSelectColumns($criteria);
$this->collReactivoss = ReactivosPeer::doSelect($criteria, $con);
                }
            } else {
                if (!$this->isNew())
```

Capítulo 2: Implementación del Sistema

```
{

$criteria->add(ReactivosPeer::SIC_0020_GEPSOLUCIONESSIC_20_ID,
$this->getSic20Id());

ReactivosPeer::addSelectColumns($criteria);
if (!isset($this->lastReactivosCriteria) || $this->lastReactivosCriteria->equals($criteria))
{
$this->collReactivoss = ReactivosPeer::doSelect($criteria, $con);
    }
    }
    }
    $this->lastReactivosCriteria = $criteria;
    return $this->collReactivoss;
}

public function countReactivoss($criteria = null, $distinct = false, $con = null)
{
    include_once 'lib/model/om/BaseReactivosPeer.php';
    if ($criteria === null) {
        $criteria = new Criteria();
    }
    elseif ($criteria instanceof Criteria)
    {
        $criteria = clone $criteria;
    }

    $criteria->add(ReactivosPeer::SIC_0020_GEPSOLUCIONESSIC_20_ID, $this->getSic20Id());
    return ReactivosPeer::doCount($criteria, $distinct, $con);
}

public function addReactivos(Reactivos $I)
{
    $this->collReactivoss[] = $I;
    $I->setSic0020Gepsoluciones($this);
}

}
```

Clase Base Peer.

class BaseSic0020GepsolucionesPeer: Esta es la clase base peer correspondiente a la clase peer anterior, relacionada con el caso de uso Gestionar SIC 0020.

Capítulo 2: Implementación del Sistema

```
abstract class BaseSic0020GepsolucionesPeer {

    const DATABASE_NAME = 'propel';
    const TABLE_NAME = 'sic_0020_gepsoluciones';
    const CLASS_DEFAULT = 'lib.model.Sic0020Gepsoluciones';
    const NUM_COLUMNS = 30;
    const NUM_LAZY_LOAD_COLUMNS = 0;
    const SIC_20_ID = 'sic_0020_gepsoluciones.SIC_20_ID';
    const FOLIO = 'sic_0020_gepsoluciones.FOLIO';
    const NO_PARTE = 'sic_0020_gepsoluciones.NO_PARTE';
    const NO_LOTE = 'sic_0020_gepsoluciones.NO_LOTE';
    const TERMINADO = 'sic_0020_gepsoluciones.TERMINADO';

    private static $phpNameMap = null;

    private static $fieldNames = array (BasePeer::TYPE_PHPNAME => array ('Sic20Id', 'Folio', 'NoParte',
    'NoLote', 'Terminado'), BasePeer::TYPE_COLNAME => array (Sic0020GepsolucionesPeer::SIC_20_ID,
    Sic0020GepsolucionesPeer::FOLIO,
    Sic0020GepsolucionesPeer::NO_PARTE, Sic0020GepsolucionesPeer::NO_LOTE,
    Sic0020GepsolucionesPeer::VT,
    Sic0020GepsolucionesPeer::FECHA_PREPARACION,
    Sic0020GepsolucionesPeer::DEPARTAMENTO,
    Sic0020GepsolucionesPeer::AREA,
    Sic0020GepsolucionesPeer::NOMBRE_SOLUCION,
    Sic0020GepsolucionesPeer::SOLVENTE,
    Sic0020GepsolucionesPeer::PH,
    Sic0020GepsolucionesPeer::CONDUCTIVIDAD,
    Sic0020GepsolucionesPeer::AJUSTE_PH_SOL,
    Sic0020GepsolucionesPeer::PH_DESEADO,
    Sic0020GepsolucionesPeer::SOL_AJUSTE_PH,
    Sic0020GepsolucionesPeer::NP,
    Sic0020GepsolucionesPeer::NO_DE_LOTE,
    Sic0020GepsolucionesPeer::PH_FINAL_SOL,
    Sic0020GepsolucionesPeer::ESTERILIZACION,
    Sic0020GepsolucionesPeer::VAPOR_SATURADO,
    Sic0020GepsolucionesPeer::TIEMPO,
    Sic0020GepsolucionesPeer::NO_FRASCOS_ALMACEN,
    Sic0020GepsolucionesPeer::VOLUMEN_FRASCO,
    Sic0020GepsolucionesPeer::FILTRACION,
    Sic0020GepsolucionesPeer::ALMACENAR_SOL,
    Sic0020GepsolucionesPeer::FECHA_VENCIMIENTO,
    Sic0020GepsolucionesPeer::OBSERVACIONES,
```

Capítulo 2: Implementación del Sistema

```
Sic0020GepsolucionesPeer::REALIZADO_POR,  
Sic0020GepsolucionesPeer::FECHA_REALIZADO,  
Sic0020GepsolucionesPeer::TERMINADO, ),
```

```
BasePeer::TYPE_FIELDNAME => array ('sic_20_id', 'folio', 'no_parte', 'no_lote', 'terminado', ),  
BasePeer::TYPE_NUM => array (0, 1, 2, 3,29, );
```

```
private static $fieldKeys = array (  
BasePeer::TYPE_PHPNAME => array ('Sic20Id' => 0, 'Folio' => 1, 'NoParte' => 2,  
'NoLote' => 3,'Terminado' => 29, ),  
BasePeer::TYPE_COLNAME => array (Sic0020GepsolucionesPeer::SIC_20_ID => 0,  
Sic0020GepsolucionesPeer::FOLIO => 1,  
Sic0020GepsolucionesPeer::NO_PARTE => 2,  
Sic0020GepsolucionesPeer::NO_LOTE => 3,  
Sic0020GepsolucionesPeer::VT => 4,  
Sic0020GepsolucionesPeer::FECHA_PREPARACION => 5,  
Sic0020GepsolucionesPeer::DEPARTAMENTO => 6,  
Sic0020GepsolucionesPeer::AREA => 7,  
Sic0020GepsolucionesPeer::NOMBRE_SOLUCION => 8,  
Sic0020GepsolucionesPeer::SOLVENTE => 9,  
Sic0020GepsolucionesPeer::PH => 10,  
Sic0020GepsolucionesPeer::CONDUCTIVIDAD => 11,  
Sic0020GepsolucionesPeer::AJUSTE_PH_SOL => 12,  
Sic0020GepsolucionesPeer::PH_DESEADO => 13,  
Sic0020GepsolucionesPeer::SOL_AJUSTE_PH => 14,  
Sic0020GepsolucionesPeer::NP => 15,  
Sic0020GepsolucionesPeer::NO_DE_LOTE => 16,  
Sic0020GepsolucionesPeer::PH_FINAL_SOL => 17,  
Sic0020GepsolucionesPeer::ESTERILIZACION => 18,  
Sic0020GepsolucionesPeer::VAPOR_SATURADO => 19,  
Sic0020GepsolucionesPeer::TIEMPO => 20,  
Sic0020GepsolucionesPeer::NO_FRASCOS_ALMACEN => 21,  
Sic0020GepsolucionesPeer::VOLUMEN_FRASCO => 22,  
Sic0020GepsolucionesPeer::FILTRACION => 23,  
Sic0020GepsolucionesPeer::ALMACENAR_SOL => 24,  
Sic0020GepsolucionesPeer::FECHA_VENCIMIENTO => 25,  
Sic0020GepsolucionesPeer::OBSERVACIONES => 26,  
Sic0020GepsolucionesPeer::REALIZADO_POR => 27,  
Sic0020GepsolucionesPeer::FECHA_REALIZADO => 28,  
Sic0020GepsolucionesPeer::TERMINADO => 29, ),
```

```
BasePeer::TYPE_FIELDNAME => array ('sic_20_id' => 0, 'folio' => 1, 'no_parte' => 2,  
'no_lote' => 3,'terminado' => 29, ),  
BasePeer::TYPE_NUM => array (0, 1, 2, 3,29, )
```

```
);

public static function getMapBuilder()
{
    include_once 'lib/model/map/Sic0020GepsolucionesMapBuilder.php';
    return BasePeer::getMapBuilder('lib.model.map.Sic0020GepsolucionesMapBuilder');
}

public static function getPhpNameMap()
{
    if (self::$phpNameMap === null) {
        $map = Sic0020GepsolucionesPeer::getTableMap();
        $columns = $map->getColumns();
        $nameMap = array();
        foreach ($columns as $column) {
            $nameMap[$column->getPhpName()] = $column->getColumnName();
        }
        self::$phpNameMap = $nameMap;
    }
    return self::$phpNameMap;
}

static public function translateFieldName($name, $fromType, $toType)
{
    $toNames = self::getFieldNames($toType);
    $key = isset(self::$fieldKeys[$fromType][$name]) ? self::$fieldKeys[$fromType][$name] : null;
    if ($key === null) {
        throw new PropelException("'" . $name . "' could not be found in the field names of type '$fromType'.
        These are: " . print_r(self::$fieldKeys[$fromType], true));
    }
    return $toNames[$key];
}

static public function getFieldNames($type = BasePeer::TYPE_PHPNAME)
{
    if (!array_key_exists($type, self::$fieldNames)) {
        throw new PropelException('Method getFieldNames() expects the parameter
        $type to be one of the class constants TYPE_PHPNAME, TYPE_COLNAME, TYPE_FIELDNAME,
        TYPE_NUM. ' . $type . ' was given.');
```

Capítulo 2: Implementación del Sistema

```
        return str_replace(Sic0020GepsolucionesPeer::TABLE_NAME.'!', $alias.'!', $column);
    }

public static function addSelectColumns(Criteria $criteria)
{
    $criteria->addSelectColumn(Sic0020GepsolucionesPeer::SIC_20_ID);
    $criteria->addSelectColumn(Sic0020GepsolucionesPeer::FOLIO);
    $criteria->addSelectColumn(Sic0020GepsolucionesPeer::NO_PARTE);
    $criteria->addSelectColumn(Sic0020GepsolucionesPeer::NO_LOTE);
    $criteria->addSelectColumn(Sic0020GepsolucionesPeer::TERMINADO);
}

const COUNT = 'COUNT(sic_0020_gepsoluciones.SIC_20_ID)';
const COUNT_DISTINCT = 'COUNT(DISTINCT sic_0020_gepsoluciones.SIC_20_ID)';

public static function doCount(Criteria $criteria, $distinct = false, $con = null)
{
    $criteria = clone $criteria;

    $criteria->clearSelectColumns()->clearOrderByColumns();
    if ($distinct || in_array(Criteria::DISTINCT, $criteria->getSelectModifiers())) {
        $criteria->addSelectColumn(Sic0020GepsolucionesPeer::COUNT_DISTINCT);
    } else {
        $criteria->addSelectColumn(Sic0020GepsolucionesPeer::COUNT);
    }

    foreach($criteria->getGroupByColumns() as $column)
    {
        $criteria->addSelectColumn($column);
    }

    $rs = Sic0020GepsolucionesPeer::doSelectRS($criteria, $con);
    if ($rs->next()) {
        return $rs->getInt(1);
    } else {
        return 0;
    }
}

public static function doSelectOne(Criteria $criteria, $con = null)
{
    $critcopy = clone $criteria;
    $critcopy->setLimit(1);
    $objects = Sic0020GepsolucionesPeer::doSelect($critcopy, $con);
    if ($objects) {
```

Capítulo 2: Implementación del Sistema

```
        return $objects[0];
    }
    return null;
}

public static function doSelect(Criteria $criteria, $con = null)
{
    return
    Sic0020GepsolucionesPeer::populateObjects(Sic0020GepsolucionesPeer::doSelectRS($criteria,
    $con));
}

public static function doSelectRS(Criteria $criteria, $con = null)
{
    if ($con === null) {
        $con = Propel::getConnection(self::DATABASE_NAME);
    }

    if (!$criteria->getSelectColumns()) {
        $criteria = clone $criteria;
        Sic0020GepsolucionesPeer::addSelectColumns($criteria);
    }

    $criteria->setDbName(self::DATABASE_NAME);

    return BasePeer::doSelect($criteria, $con);
}

public static function populateObjects(ResultSet $rs)
{
    $results = array();

    $cls = Sic0020GepsolucionesPeer::getOMClass();
    $cls = Propel::import($cls);
    while($rs->next())
    {
        $obj = new $cls();
        $obj->hydrate($rs);
        $results[] = $obj;
    }
    return $results;
}

public static function getTableMap()
{

```

Capítulo 2: Implementación del Sistema

```
        Return
Propel::getDatabaseMap(self::DATABASE_NAME)->getTable(self::TABLE_NAME);
    }

public static function getOMClass()
    {
        return Sic0020GepsolucionesPeer::CLASS_DEFAULT;
    }

public static function doInsert($values, $con = null)
    {
        if ($con === null) {
            $con = Propel::getConnection(self::DATABASE_NAME);
        }

        if ($values instanceof Criteria) {
            $criteria = clone $values;        } else {
            $criteria = $values->buildCriteria();        }

        $criteria->remove(Sic0020GepsolucionesPeer::SIC_20_ID);

        $criteria->setDbName(self::DATABASE_NAME);
        try {
            $con->begin();

            $pk = BasePeer::doInsert($criteria, $con);
            $con->commit();
        } catch(PropelException $e) {
            $con->rollback();
            throw $e;
        }
        return $pk;
    }

public static function doUpdate($values, $con = null)
    {
        if ($con === null) {
            $con = Propel::getConnection(self::DATABASE_NAME);
        }

        $selectCriteria = new Criteria(self::DATABASE_NAME);

        if ($values instanceof Criteria) {
            $criteria = clone $values;
            $comparison = $criteria->getComparison(Sic0020GepsolucionesPeer::SIC_20_ID);
```

```
        $selectCriteria->add(Sic0020GepsolucionesPeer::SIC_20_ID,
        $criteria->remove(Sic0020GepsolucionesPeer::SIC_20_ID), $comparison);
    }
else
    {
        $criteria = $values->buildCriteria();
        $selectCriteria = $values->buildPkeyCriteria();
    }
    $criteria->setDbName(self::DATABASE_NAME);

    return BasePeer::doUpdate($selectCriteria, $criteria, $con);
}
```

```
public static function doDeleteAll($con = null)
```

```
{
    if ($con === null) {
        $con = Propel::getConnection(self::DATABASE_NAME);
    }
    $affectedRows = 0;
    try {
        $con->begin();
        $affectedRows+= BasePeer::doDeleteAll(Sic0020GepsolucionesPeer::TABLE_NAME, $con);
        $con->commit();
        return $affectedRows;
    } catch (PropelException $e) {
        $con->rollback();
        throw $e;
    }
}
```

```
public static function doDelete($values, $con = null)
```

```
{
    if ($con === null) {
        $con = Propel::getConnection(Sic0020GepsolucionesPeer::DATABASE_NAME);
    }
    if ($values instanceof Criteria)
    {
        $criteria = clone $values;
    }
    elseif ($values instanceof Sic0020Gepsoluciones)
    {
        $criteria = $values->buildPkeyCriteria();
    }
    else
```

Capítulo 2: Implementación del Sistema

```
{
$criteria = new Criteria(self::DATABASE_NAME);
$criteria->add(Sic0020GepsolucionesPeer::SIC_20_ID, (array) $values, Criteria::IN);
}
$criteria->setDbName(self::DATABASE_NAME);
$affectedRows = 0;
try
{
    $con->begin();
    $affectedRows += BasePeer::doDelete($criteria, $con);
    $con->commit();
    return $affectedRows;
}
catch (PropelException $e)
{
    $con->rollback();
    throw $e;
} }
```

```
public static function doValidate(Sic0020Gepsoluciones $obj, $cols = null)
{
    $columns = array();
    if ($cols) {
        $dbMap = Propel::getDatabaseMap(Sic0020GepsolucionesPeer::DATABASE_NAME);
        $tableMap = $dbMap->getTable(Sic0020GepsolucionesPeer::TABLE_NAME);

        if (! is_array($cols)) {
            $cols = array($cols);
        }

        foreach($cols as $colName) {
            if ($tableMap->containsColumn($colName))
            {
                $get = 'get' . $tableMap->getColumn($colName)->getPhpName();
                $columns[$colName] = $obj->$get();
            } } }
    else {}
    $res=BasePeer::doValidate(Sic0020GepsolucionesPeer::DATABASE_NAME,
Sic0020GepsolucionesPeer::TABLE_NAME, $columns);
    if ($res !== true) {
        $request = sfContext::getInstance()->getRequest();
        foreach ($res as $failed) {
            $col= Sic0020GepsolucionesPeer::translateFieldname($failed->getColumn(),
BasePeer::TYPE_COLNAME, BasePeer::TYPE_PHPNAME);
```

Capítulo 2: Implementación del Sistema

```
        $request->setError($col, $failed->getMessage());
    } }
    return $res;
}

public static function retrieveByPK($pk, $con = null)
{
    if ($con === null) {
        $con = Propel::getConnection(self::DATABASE_NAME);
    }
    $criteria = new Criteria(Sic0020GepsolucionesPeer::DATABASE_NAME);
    $criteria->add(Sic0020GepsolucionesPeer::SIC_20_ID, $pk);
    $v = Sic0020GepsolucionesPeer::doSelect($criteria, $con);
    return !empty($v) > 0 ? $v[0] : null;
}

public static function retrieveByPKs($pks, $con = null)
{
    if ($con === null) {
        $con = Propel::getConnection(self::DATABASE_NAME);
    }
    $objs = null;
    if (empty($pks)) {
        $objs = array();
    } else {
        $criteria = new Criteria();
        $criteria->add(Sic0020GepsolucionesPeer::SIC_20_ID, $pks, Criteria::IN);
        $objs = Sic0020GepsolucionesPeer::doSelect($criteria, $con);
    }
    return $objs;
}
}}
if (Propel::isInit()) {
    try {
        BaseSic0020GepsolucionesPeer::getMapBuilder();
    } catch (Exception $e) {
        Propel::log('Could not initialize Peer: ' . $e->getMessage(), Propel::LOG_ERR);
    } }
else
{
    require_once 'lib/model/map/Sic0020GepsolucionesMapBuilder.php';
    Propel::registerMapBuilder('lib.model.map.Sic0020GepsolucionesMapBuilder');
}
```

2.5.3 Capa Vista.

La vista se encarga de producir las páginas que se muestran como resultado de las actions. La vista en Symfony está compuesta por diversas partes, cada una de ellas está 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 actions que se ejecuta en el momento) y con el layout, que contiene el código HTML común a todas las páginas y pequeñas representaciones de código PHP, que normalmente son llamadas a los diversos helpers disponibles.

Los programadores centran su trabajo relativo a la vista en los archivos de configuración YAML (Otro Lenguaje de Marcado, del inglés Yet Another Markup Language), los cuales permiten establecer opciones para las propiedades de la respuesta y para otros elementos de la interfaz y en el objeto respuesta. Cuando se trabaja con variables en las plantillas, deben considerarse los posibles riesgos de seguridad de XSS (cross-site scripting) por lo que es necesario aplicar la validación de los caracteres introducidos por los usuarios.

A continuación se muestra parte del código realizado para las validaciones, de los datos suministrados a la aplicación por los usuarios.

Tratamiento de errores:

La aplicación consta con tratamiento de errores a la hora de suministrar los datos el usuario a la aplicación; estos serán validados a través de archivos .yml vinculados directamente a los formularios de la interfaz usuario, del lado del servidor. Detectado un error, dígame datos escritos de forma equívoca o dejar algún campo vacío, automáticamente se lanzará un mensaje de error, informándole al usuario, el tipo de error cometido; posteriormente el usuario tiene la posibilidad de continuar con el llenado del SIC.

Efectuado satisfactoriamente el llenado del SIC con los datos proveídos por el usuario, dichos datos se remitirán directamente a la base de datos. Este tratamiento permite el llenado de la base de datos sin la existencia de errores en la misma.

A continuación se muestran ejemplos de las funciones realizadas en java script para lograr el tratamiento de errores en la aplicación.

Funciones Script: Estas funciones permiten la validación de los campos de los SIC.

```
<script>  
function save()
```

Capítulo 2: Implementación del Sistema

```
{
  if(NoAste('crear_nombre') && NoVacio('crear_nombre') && NoAste('crear_codigo') &&
  NoVacio('crear_codigo') && NoAste('fecha11') && NoVacio('fecha11'))
  {
    var nomb= document.getElementById('crear_nombre');
    var nombre= document.getElementById('nombre');
    nombre.value= nombre.value + nomb.value + '*';
    nomb.value="";
    var cod= document.getElementById('crear_codigo');
    var codigo= document.getElementById('codigo');
    codigo.value= codigo.value + cod.value + '*';
    cod.value="";
    var fe= document.getElementById('fecha11');
    var fecha= document.getElementById('fecha');
    fecha.value= fecha.value + fe.value + '*';
    fe.value='2005-02-04';
  }
  {
    if(!NoAste('crear_nombre'))
    {
      alert('No entre * en el campo "Nombre"');
    }
    if(document.getElementById('crear_nombre')==")
    {
      alert('No puede dejar el campo "Nombre" vacio');
    }
    if(!NoAste('crear_codigo'))
    {
      alert('No entre * en el campo "Codigo"');
    }
    if(document.getElementById('crear_codigo')==")
    {
      alert('No puede dejar el campo "Codigo" vacio');
    }
    if(!NoAste('fecha11'))
    {
      alert('No entre * en el campo "Fecha"');
    }
    if(document.getElementById('fecha11')==")
    {
      alert('No puede dejar el campo "Fecha" vacio');
    }
  }
}
```

- Esta función permite que a determinados campos requeridos, se le suministren solamente números.

```
function SoloNumeros(id)
{
    var elemento = document.getElementById(id);
    var expresion= /^[0-9]+$/;
    if(!expresion.test(elemento.value))
    {
        event.returnValue = false;
        return false;
    }
    else
    return true;
}
```

- Esta función valida que ningún campo permanezca sin parámetros.

```
function NoVacio(id)
{
    var elemento = document.getElementById(id);
    if(elemento.value=="")
    {
        elemento.focus();
        event.returnValue = false; //Para que no se envíe el formulario
        return false;
    }
    else
    return true;
}
```

- Esta función valida que no se entren asteriscos en los campos.

```
function NoAste(id)
{
    var elemento = document.getElementById(id);
    var expresion= /^[*]+$/;
    if(expresion.test(elemento.value))
    {
        elemento.focus();
        event.returnValue = false;
    }
}
```

```
        return false;
    }
    else
        return true;
} }
</script>
```

Fragmentos del código donde se validan los datos.

- {
 \$this->mensaje='Entre un Criterio de Busqueda';

 \$this->ver=-1;
 }
 else
 {..... }
• if(!\$ban)
 {
 \$this->ver_pruebas='Modificados Correctamente';

 \$i=0;

 while(\$this->getRequestParameter('*****'. \$i))
 {

 }
 }
 else
 \$this->ver_pruebas='Ente Todos Los Datos Bien';

• if(\$this->getRequestParameter('crear_terminado')!=1)

 \$this->mensaje="Esta Terminado";

 Else

 \$this->mensaje="No esta Terminado"; (Ver Figura.23).

The screenshot shows a web browser window displaying a data entry form. The form is titled 'CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA' and 'DIRECCIÓN DE CALIDAD'. It contains various input fields for solution preparation, such as 'No. Parte', 'No. lote', 'VT', 'Fecha prep.', 'Reactivos', 'Np', 'Cantidad a usar', 'Cantidad utilizada', 'Lote', 'Solvente', 'Ajuste de pH a la solución', 'pH deseado', 'Solución de ajuste de pH', 'pH final de la solución', 'Filtración', 'Vapor saturado', 'Almacenamiento', 'No. de frascos', 'Volumen por frasco', and 'Fecha de vencimiento'. A table at the bottom is titled 'PRUEBAS DE ACEPTACIÓN' with columns 'LIMITE' and 'VALOR OBTENIDO'. An error dialog box from 'Microsoft Internet Explorer' is overlaid on the form, displaying a warning icon and the message 'No entre * en el campo "Reactivos"', with an 'OK' button.

Figura 23. Interfaz visual de validación de errores.

Conclusiones

- En el concluido capítulo se muestran una serie de artefactos, dígame diagramas de componentes referentes al flujo de trabajo implementación, más específicamente se desarrollaron 12 diagramas de componentes, de los cuales se muestran anteriormente un número de ellos como muestra del trabajo desarrollado.
- Además de mostrar un número de funciones desarrolladas para la implementación propuesta, siendo estas las de mayor importancia para el funcionamiento de los SIC, el código mostrado con anterioridad es el que pertenece al SIC 0020.
- Conjuntamente también se sitúa parte del código Java Script utilizado para la validación de los campos, a la hora de entrar los datos de cada SIC, cada uno con sus particularidades específicas.

CONCLUSIONES GENERALES

- Se realizaron las actividades que están definidas por el rol de Implementador y propuestas por RUP, es decir, se implementaron los métodos necesarios, para darle solución a la problemática propuesta. Del mismo modo se exponen fragmentos del código fuente, de vital importancia para el desarrollo y funcionamiento de la aplicación en cuestión, desarrollada mediante el framework Symfony, el cual nos permitió un mejor desempeño a la hora de la programación, porque facilita herramientas y clases permitiendo reducir el tiempo de desarrollo de la aplicación. Conjuntamente se realizaron doce diagramas de componentes.

RECOMENDACIONES

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

- Se extienda la experiencia del desarrollo del LIMS a los demás centros del Polo Científico y a todos aquellos donde se necesite gestionar información para laboratorios.
- Se realice la integración de éste módulo de Microbiología de conjunto a los restantes módulos que van a componer el LIMS, luego de haber culminado con el diseño e implementación en su totalidad de todos los demás.

REFERENCIAS BIBLIOGRÁFICAS

1. Centro de Ingeniería Genética y Biotecnología. Centro de Ingeniería Genética y Biotecnología. [En línea] 2003. [Citado el: 20 de Noviembre de 2007.] www.cigb.edu.cu/pages/ccalidad.html.
2. Centro de Ingeniería Genética y Biotecnología. Centro de Ingeniería Genética y Biotecnología. [En línea] 2003. [Citado el: 25 de Noviembre de 2007.] www.cigb.edu.cu/pages/micro_calidad.html.
3. Cuevas, Olga. Onlabweb. Onlabweb. [En línea] 6 de 12 de 2007. [Citado el: 20 de Diciembre de 2007.] www.onlabweb.com/lims.htm.
4. Matrix LIMS. Matrix LIMS. [En línea] [Citado el: 15 de Octubre de 2007.] www.automatica.es/imprimir.asp_pid=472.
5. PRNewswire. PRNewswire. [En línea] 2008. [Citado el: 15 de Enero de 2008.] <http://www.prnewswire.co.uk/cgi/news/release?id=125283>.
6. Starlims. Starlims. [En línea] 2007. [Citado el: 10 de Diciembre de 2007.] www.stralims.com/es/default.htm. www.stralims.com/es/default.htm.
7. Universidad de Valencia. Universidad de Valencia. [En línea] [Citado el: 10 de Diciembre de 2007.] www.pid.dsic.upv.es/C1/Materiales/Documentos%20Disponibles/Introducci%20n%20d%20RUP.doc.
8. Díaz, De La Fuente, Lisdany y del Sol, González, Sonia. "LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis del Grupo de Inspección y Auditorías y del Laboratorio Análisis Químico.". La Habana: s.n., 2007. Tesis de diploma.
9. Programación en Castellano. Programación en Castellano. [En línea] [Citado el: 27 de Enero de 2008.] www.programacion.net/noticia/1363/.
10. Manual De Visual Paradigm.
11. Free Download Manager. Free Download Manager. [En línea] [Citado el: 10 de febrero de 2008.] [www.freedownloadmanger.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_1470_P](http://www.freedownloadmanger.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_1470_P).
12. Desarrollo Web. Desarrollo Web. [En línea] [Citado el: 15 de Febrero de 2008.] www.desarrollo.com/articulos/1696.php.
13. IES Camp. IES Camp. [En línea] [Citado el: 5 de Diciembre de 2007.] www.iescamp.es/tutoriales/php/tema1p21.htm.
14. Programación Bases de Datos Y Estructura. Programación Bases de Datos Y Estructura. [En línea] 3 de Mayo de 2008. [Citado el: 2 de Marzo de 2008.] www.bredondo.awardspace.com/?Programaci%20n_web:Lenguaje_PHP.

Referencias Bibliográficas

15. Maestros del Web. Maestros del Web. [En línea] 2 de Noviembre de 2007. [Citado el: 19 de Diciembre de 2007.] www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web.
16. Blog de Programación. Blog de Programación. [En línea] Octubre de 2007. [Citado el: 25 de Enero de 2008.] www.blogdeprogramacion.blogspot.com/2007/10/un_framework_para_php_symfony.html.
17. Pecos Daniel. PostGreSQL vs. MySQL. [Citado el: 20 de Enero de 2008.] www.netpecos.org/docs/mysql_postgres/index.html.
18. Arceo, Claro, Alfonso y Sánchez, Pantoja, Maidolys. "Portal Digital de la Facultad Regional de la UCI en Granma". La Habana: s.n., 2007. Tesis de diploma.
19. Fabien Potencier y Francois Zanonotto. Libros Web. Libros Web. [En línea] Febrero de 2008. [Citado el: 5 de Abril de 2008.] www.librosweb.es.

BIBLIOGRAFÍA

1. EMS SQL Manager for PostgreSQL. EMS SQL Manager for PostgreSQL. [En línea] 2007. [Citado el: 20 de Enero de 2008.] www.sqlmanager.net/product/postgreesql/manager.
2. EMS SQL Manager for PostgreSQL. EMS SQL Manager for PostgreSQL. [En línea] 2007. [Citado el: 20 de enero de 2008.] www.sqlmanager.net/product/postgreesql/datagenetor.
3. EMS SQL Manager for PostgreSQL. EMS SQL Manager for PostgreSQL. [En línea] 2007. [Citado el: 20 de Enero de 2008.] www.sqlmanager.net/product/postgreesql/query.
4. Ana Fernández Vila. [En línea] 20 de Marzo de 2001. [Citado el: 20 de enero de 2008.] <http://www.gris.det.uvigo.es/~avilas/UML/node49.html>.
5. Arceo, Claro, Alfonso y Sánchez, Pantoja, Maidolys. "Portal Digital de la Facultad Regional de la UCI en Granma". La Habana: s.n., 2007. Tesis de diploma
6. Ayanet. Consultoria Integral. Ayanet. Consultoria Integral. [En línea] [Citado el: 15 de Febrero de 2008.] www.ayanet.es. www.ayanet.es.
7. Blog de Programación. Blog de Programación. [En línea] Octubre de 2007. [Citado el: 25 de Enero de 2008.] www.blogdeprogramacion.blogspot.com/2007/10/un_framework_para_php_symfony.html.
8. Centro de Ingeniería Genética y Biotecnología. Centro de Ingeniería Genética y Biotecnología. [En línea] 2003. [Citado el: 20 de Noviembre de 2007.] www.cigb.edu.cu/pages/ccalidad.html.
9. Centro de Ingeniería Genética y Biotecnología. Centro de Ingeniería Genética y Biotecnología. [En línea] 2003. [Citado el: 25 de Noviembre de 2007.] www.cigb.edu.cu/pages/micro_calidad.html.
10. Desarrollo Web. Desarrollo Web. [En línea] [Citado el: 15 de Febrero de 2008.] www.desarrollo.com/articulos/1696.php.
11. Enrique Hernández Orallo. DISCA Lenguaje Unificado de Modelado (UML). DISCA. [En línea] [Citado el: 5 de Febrero de 2008.] www.disca.upv.es.
12. Free Download Manager. Free Download Manager. [En línea] [Citado el: 10 de febrero de 2008.] [www.freedownloadmanger.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_1470_P](http://www.freedownloadmanger.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_1470_P).
13. IES Camp. IES Camp. [En línea] [Citado el: 5 de Diciembre de 2007.] www.iescamp.es/tutoriales/php/tema1p21.htm.
14. Grupo de Investigación. Kybele. Kybele. [En línea] 2007. [Citado el: 19 de Diciembre de 2007.] <http://www.kybele.etsii.urjc.es/ISS>.
15. Maestros del Web. Maestros del Web. [En línea] 2 de Noviembre de 2007. [Citado el: 19 de Diciembre de 2007.] www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web.

16. Manual De Visual Paradigm.
17. Matrix LIMS. Matrix LIMS. [En línea] [Citado el: 15 de Octubre de 2007.] www.automatica.es/imprimir.asp_pid=472.
18. PRNewswire. PRNewswire. [En línea] 2008. [Citado el: 15 de Enero de 2008.] <http://www.prnewswire.co.uk/cgi/news/release?id=125283>.
19. Programación Bases de Datos Y Estructura. Programación Bases de Datos Y Estructura. [En línea] 3 de Mayo de 2008. [Citado el: 2 de Marzo de 2008.] www.bredondo.awardspace.com/?Programaci%F3n_web:Lenguaje_PHP.
20. Programación en Castellano. Programación en Castellano. [En línea] [Citado el: 27 de Enero de 2008.] www.programacion.net/noticia/1363/.
21. Starlims. Starlims. [En línea] 2007. [Citado el: 10 de Diciembre de 2007.] www.stralims.com/es/default.htm.
22. Universidad de Valencia. Universidad de Valencia. [En línea] [Citado el: 10 de Diciembre de 2007.] www.pid.dsic.upv.es/C1/Materiales/Documentos%20Disponibles/Introducci%B3n%20d%20RUP.doc.
23. Visual Paradigm. Visual Paradigm. [En línea] 2005. [Citado el: 25 de Noviembre de 2007.] www.visual-paradigm.com/product/vpuml.
24. Cuevas, Olga. Onlabweb. Onlabweb. [En línea] 6 de 12 de 2007. [Citado el: 20 de Diciembre de 2007.] www.onlabweb.com/lims.htm.
25. Lacoste, Yoendris y Sánchez, Jorge Daniel. LIMS de Calidad del CIGB: Análisis del Laboratorio de Microbiología. Ciudad de La Habana : Universidad de Ciencias Informáticas, 2007.
26. Lago, Ramiro. Proactivo. Proactivo. [En línea] Enero de 2006. [Citado el: 10 de Diciembre de 2007.] www.proactiva-calidad.com/java/herramientas/eclipse/introduccion.html.
27. Markiewies, Marcus E. y P. de Lucena, Carlos J. Crossroad. Crossroad. [En línea] 2004. [Citado el: 15 de Diciembre de 2007.] www.acm.org/crossroad/espanol/xrds7-4/framework.htm.
28. Pecos, Daniel. Post Gres SQL vs. MySQL. Post Gres SQL vs. MySQL. [En línea] 7 de Junio de 2002. [Citado el: 20 de Noviembre de 2007.] www.netpecos.org/doc/mysqlportgres.
29. Santesmases, Prof. José Garcia. Facultad de Informática de Madrid. Facultad de Informática de Madrid. [En línea] [Citado el: 25 de Enero de 2008.] www.fi.upm.es.
30. Sarmiento, Alieski y Cutiño, Elián. LIMS de Calidad del CIGB: Análisis del Grupo de Recepción de Muestra y Manipulación de Expedientes. Ciudad de La Habana : Instituto Superior Politécnico "José Antonio Echevarría", 2006.

31. Zaninotto, Francois y Pontencier, Fabien. Librosweb. Librosweb. [En línea] 1 de Febrero de 2008. [Citado el: 15 de Marzo de 2008.] www.librosweb.es/symfony.

ANEXOS

Anexo1.

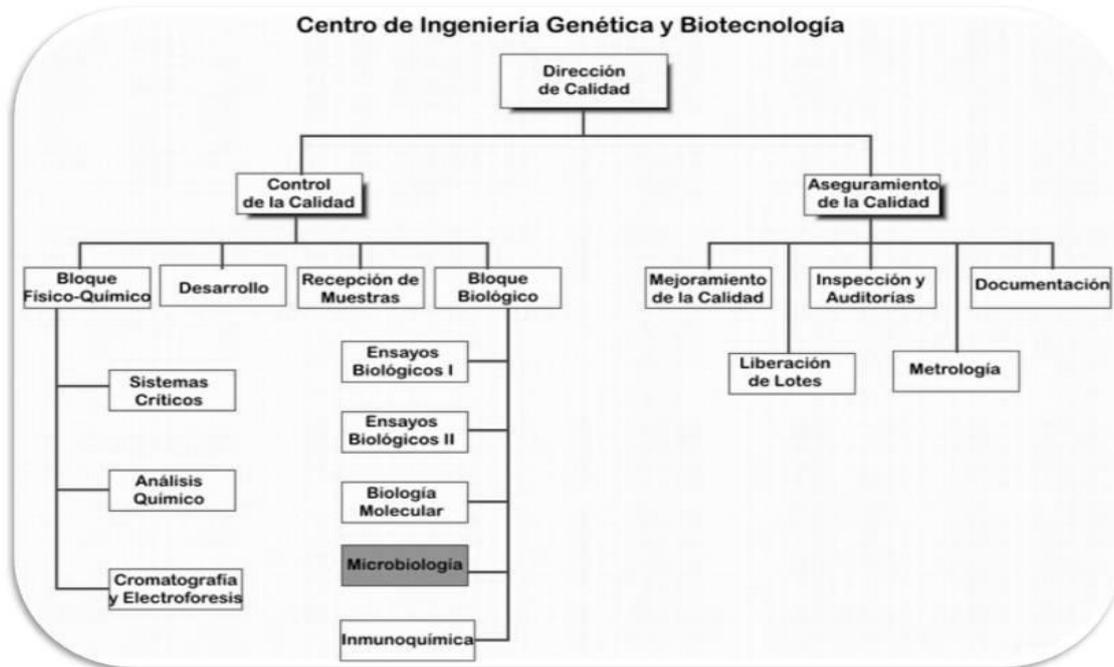


Figura 24. Estructura jerárquica del Área de Calidad del Centro de Ingeniería Genética y Biotecnología.

Anexo 2.

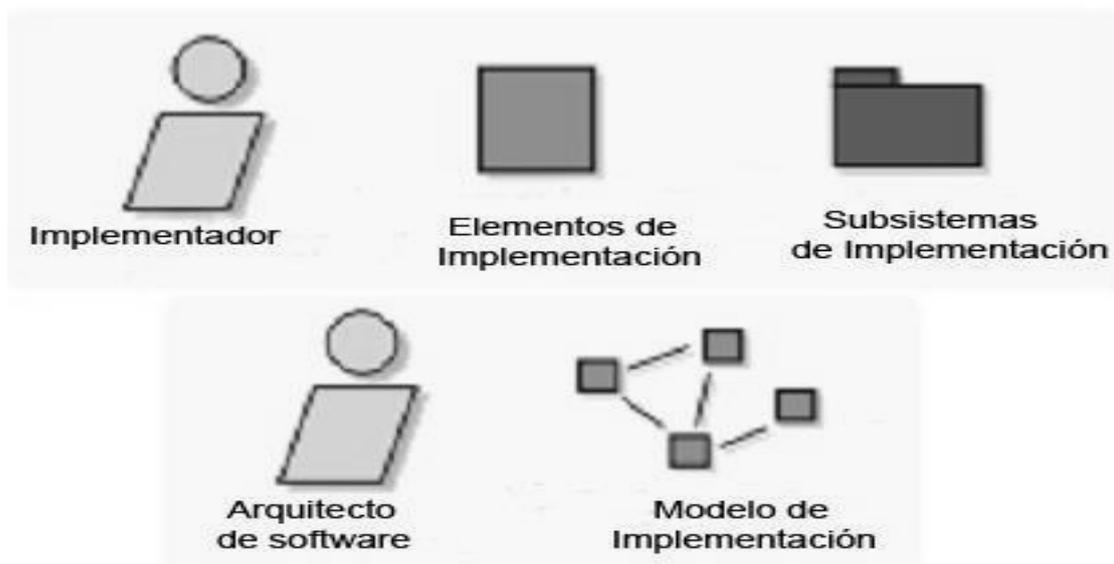


Figura 25. Trabajadores y artefactos.

GLOSARIO DE TÉRMINOS

- **Artefacto:** Pieza de información utilizada o producida por un proceso de desarrollo de software, como un documento externo o el producto de un trabajo. Un artefacto puede ser un modelo, una descripción o un software.
- **Aseguramiento de la Calidad:** Conjunto de acciones planificadas y sistemáticas que son necesarias para proporcionar la confianza de que un producto o servicio satisface los requisitos de calidad preestablecidos.
- **Biomedicina:** medicina clínica basada en los principios de las ciencias naturales.
- **Biotecnología:** conocimientos y avances biológicos aplicados a procesos tecnológicos o de interés industrial.
- **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:** Técnicas y actividades de carácter operativo utilizadas para satisfacer los requisitos de calidad.
- **Ingredientes farmacéuticos activos (IFA) o Producto intermedio:** Sustancia farmacéutica activa o producto intermedio de calidad y pureza establecidas en comparación con estándar primario, que se utiliza como sustancia de referencia para los ensayos rutinarios de laboratorio. [Guía Internacional Armonizada para IFA. Buenas Prácticas de Manufactura. Septiembre 1999]
- **Metodología:** Un sistema de principios y normas generales de organización y estructuración teórico práctica de actividades.
- **Microbiología:** Rama de la biología que estudia los microbios o microorganismos.
- **SIC:** Sistema de Información y Control.