

Universidad de las Ciencias Informáticas

Facultad 6



**Título: “LIMS de Calidad del Centro de Ingeniería
Genética y Biotecnología: “Implementación del módulo Biología
Molecular.”**

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

Autores:

René Fuente Mora.

Yosvani Quiala Sánchez.

Tutor:

Ing. Niusvel Acosta Mendoza

Cotutor:

Lic. José Alberto Ponca Pérez

La Habana, junio 2003

“Año 50 de la Revolución”

005.12

Fue

↓

Fes 19-02-09

Nr: TD-1730-08

“Lo que hace crecer el mundo no es el descubrir cómo está hecho sino el esfuerzo que cada uno hace para descubrirlo”.

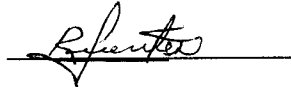
José Martí

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.

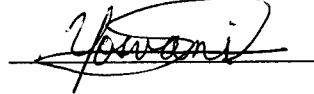
Para que así conste firmo la presente a los 8 días del mes de julio del año 2008.

René Fuente Mora



Firma del Autor

Yosvani Quiala Sánchez



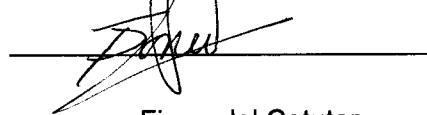
Firma del Autor

Ing. Niusvel Acosta Mendoza



Firma del Tutor

Lic. José Alberto Ponce Pérez



Firma del Cotutor

DATOS DE CONTACTO

Tutor:

Ing. Niusvel Acosta Mendoza. Profesor graduado de Ingeniería en Ciencias Informáticas el año 2007. Desde su segundo año de la carrera forma parte del movimiento de alumnos ayudantes, ha impartido asignaturas como Introducción a la Programación, Programación 1, Programación 2, Programación 3, Gráfico por Computadora y Práctica Profesional. Se desempeña como profesor en la Facultad 6 y actualmente es sublíder del proyecto productivo LIMS de Calidad, donde además desempeña otros roles como: arquitecto, líder de desarrollo, gestor de configuración y diseñador de base de datos.

Email: nacosta@uci.cu

Co-Tutor:

Lic. José Alberto Ponce Pérez. Profesor graduado en el año 1995 en el Instituto Pedagógico Enrique José Varona. Ha impartido asignaturas como Ciencia Tecnología y Sociedad, Filosofía y Economía Política. Se desempeña como profesor en la Facultad 6 y actualmente está cursando el postgrado de Metodología de la Investigación.

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: japonce@uci.cu

DEDICATORIA

A mis padres René y Sonia por ser el mejor regalo que me ha dado la vida, de verdad que los quiero mucho. Gracias de todo corazón por su apoyo y por su confianza en mí.

A mi sister por su confianza y su amistad y por darme la sobrinita más linda del mundo.

A Saday por ser la niña linda de mis ojos.

A alguien muy especial (El Yiyi) donde quieras que te encuentres te lo dedico.

A mi familia, mis tíos y mis tías, gracias por confiar en mí.

A mis abuelas Manga y Tete.

A mis panas José Antonio (El Gallego), Al Dupo y a Gabriel (El Peugeot), esto también es de ustedes.

A mi brother El Bolaiche, a Yunaisi (La Flaca), a Maday y a Dianela y al Tommy y a esos momentos que pasamos juntos en la school y que perdurarán por el resto de los días.

René Fuentes Mora

A mi madre Ania Sánchez García y a Julio Cutiño Oro quien a sido un padre para mí, por todo el amor que siempre me han dado. Se que siempre me han llevado en su corazón.

A Gonzalo Quiala Torres mi papá.

A Diana por todo su cariño y su amor, por todo lo que representa para mi vida.

A mis amigos de la infancia Carlos Roja y Misael Hastie.

A Jorge Samé que es como un hermano para mí.

A mis abuelos y mis tíos, que siempre han estado brindándome su ayuda.

A todas las personas que se interesaron siempre por mi vida.

A Aquel que me ha dado la vida, por darme todo aun cuando no merecía nada, por ser mi Consejero y estar siempre conmigo, gracias por tu amor. A Ti de manera especial con toda mi fuerza, con toda mi alma, con todo mi corazón.

Yosvani Quiala Sánchez

AGRADECIMIENTOS

Agradecemos a todas aquellas personas que han colaborado con nuestra formación en estos 5 años.

A nuestros padres, que nos impulsan a ser cada día mejores.

A todos aquellos profesores y compañeros de estudio, que durante nuestra vida estudiantil nos han ayudado a convertirnos en lo que somos.

A Yunaisi Renté, Ricardo Rodríguez, Franklin Cantillo, Jommy A. Barbán, José Rafael, Manuel Marrero, Denis Medina, Richard Díaz y José Ramón Hermsilla por soportar nuestras preguntas y brindarnos respuestas siempre que estuvo a su alcance.

En fin, a todos los que han contribuido a convertir este sueño en una realidad.

RESUMEN

El Centro de Ingeniería Genética y Biotecnología (CIGB) tiene un papel integrador en la esfera de la Biotecnología Cubana, con alta capacidad científico técnica, es una institución de desarrollo dinámico que le ha permitido alcanzar un alto nivel en la investigación, desarrollo, producción y comercialización de productos biológicos obtenidos a través de los métodos de la biotecnología moderna.

En la Dirección de Calidad del (CIGB), se gestiona a diario un gran volumen de información relacionada con los ensayos que allí se realizan, es de vital importancia el control y aseguramiento de la calidad de las producciones del centro. Debido a la necesidad de gestionar y controlar la información que se genera, se está trabajando en la realización de un Sistema de Gestión de Información de los Laboratorios (LIMS, del inglés *Laboratory Information Management System*) para este centro.

El presente trabajo tiene como objetivo implementar el módulo del Laboratorio de Biología Molecular, para el sistema de gestión de la información de los laboratorios de la Dirección de Calidad del CIGB.

Palabras Clave:

Biología Molecular, Calidad, CIGB, ensayos, laboratorio, LIMS.

ÍNDICE

INTRODUCCIÓN 1

FUNDAMENTACIÓN TEÓRICA..... 6

1.1 Sistemas automatizados existentes vinculados al campo de acción..... 6

1.2 Tendencias, Tecnologías y Herramientas utilizadas 8

 1.2.1 Sistemas de Gestión de Bases de Datos (SGBD).....8

 1.2.2 Internet. Funcionamiento10

 1.2.3 Aplicaciones Web.....11

 1.2.4 Entorno Distribuido. Modelo Cliente Servidor11

 1.2.5 Servidor Web Apache12

 1.2.6 Lenguajes de Programación Web.13

 1.2.7 Lenguajes de programación Web a utilizar.14

 1.2.8 Entorno Integrado de desarrollo (IDE).16

 1.2.9 Patrones de Arquitectura18

 1.2.10 Framework utilizado18

 1.2.11 Metodología de desarrollo de software.....21

 1.2.11.1 Flujo de Trabajo: Implementación23

 1.2.11.2 Objetivos del Flujo de Trabajo de Implementación24

 1.2.11.3 Relación con los demás flujos de trabajo del Proceso Unificado de Desarrollo (RUP)...24

 1.2.11.4 Roles, artefactos y actividades.25

 1.2.11.5 Rol Implementador de Sistema26

 1.2.11.6 Artefactos y actividades del rol Implementador de Sistemas26

 1.2.12 Lenguaje de Modelado.....29

 1.2.13 Herramienta CASE.....30

IMPLEMENTACIÓN DEL SISTEMA. 33

2.1 Requerimientos Funcionales del módulo Biología Molecular 33

2.2 Descripciones textuales de los Casos de Uso 35

2.3 Diagramas de Componentes 52

Figura 13. Diagrama de Componentes Gestionar Registro de Desviaciones de Ensayo Analítico (SIC-0028)..... 63

2.4 Descripción de los Métodos 63

2.5 Fragmentos de código 66

 2.5.1 Validación y manejo de errores.72

CONCLUSIONES 82

RECOMENDACIONES 83

REFERENCIAS BIBLIOGRÁFICAS.....	84
BIBLIOGRAFÍA	86
ANEXO 1: Diagrama de clases del diseño Gestionar Solicitud de destrucción de bancos de células	88
(SIC-0840).....	88
ANEXO 2: Diagrama de clases del diseño Gestionar registro de Destrucción de Banco de Células	89
(SIC-0841).....	89
ANEXO 3: Diagrama de clases del diseño Gestionar registro de Temperatura (SIC-0125R).....	90
ANEXO 4: Diagrama de clases del diseño Gestionar Gestionar Cronograma de chequeo de bancos de células (SIC-0130).....	91
ANEXO 5: Diagrama de clases del diseño Gestionar Certificado de Liberación de los bancos de Células transformados (SIC-0229).....	92
ANEXO 6: Diagrama de clases del diseño Gestionar registro de Descargo para Banco de Células	93
(SIC-0190).....	93
ANEXO 7: Diagrama de clases del diseño Gestionar Informe de análisis de los bancos de células	94
(SIC-0950).....	94
ANEXO 8: Interfaz visual del Crear Sic0840.	94
ANEXO 9: Interfaz visual del Visualizar Sic0840.....	95
GLOSARIO	97

INTRODUCCIÓN

La Informatización de la Sociedad lleva consigo un proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones (TIC) en la vida cotidiana, esto tiene como objetivo satisfacer las necesidades de todas las esferas de la sociedad, en su esfuerzo por lograr cada vez más calidad en todos los procesos y por consiguiente mayor generación de riqueza y aumento en el nivel de vida de los ciudadanos. Una sociedad que aplique la informatización en todas sus esferas será más competitiva. Es evidente que para los países subdesarrollados resulta un reto el logro de este propósito.

Cuba ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las TIC y lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría a la sociedad cubana acercarse más hacia el objetivo de un desarrollo sostenible.

El Centro de Ingeniería Genética y Biotecnología (CIGB) de La Habana, es una institución de desarrollo dinámico que ha logrado alcanzar un alto nivel en la investigación, desarrollo, producción y comercialización de productos biológicos obtenidos a través de los métodos de la biotecnología moderna. Este centro investiga con firmes pasos en presencia de una cultura de producción regida por los conceptos más avanzados de la calidad y su aseguramiento.

El CIGB tiene un papel integrador en la esfera de la biotecnología cubana, con alta capacidad científico-técnica. Asume la responsabilidad de contribuir directamente en el desarrollo económico y social del país. Su impacto está destinado a la salud humana, la producción agropecuaria, acuícola, y al medio ambiente.

En el CIGB existe la Dirección de Calidad, y en ella se encuentran los departamentos de Control de la Calidad y Aseguramiento de la Calidad.

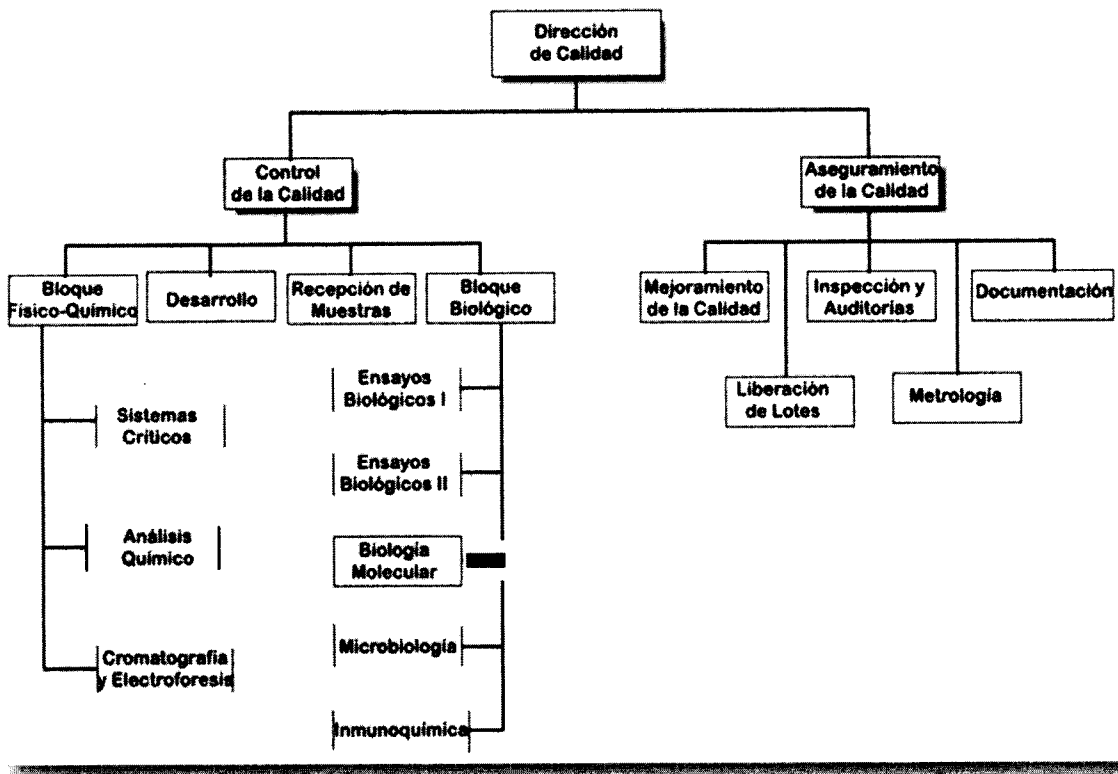


Figura 1. Estructura jerárquica de la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología.

El departamento de Aseguramiento de la Calidad tiene procedimientos para evitar que se produzcan bienes defectuosos, garantiza que se lleven a cabo las acciones planificadas y sistemáticas que son necesarias para proporcionar la confianza de que los productos que allí se elaboran satisfacen al cliente y los requisitos de calidad establecidos, logrando reducir los costes, prevenir errores y ser más competitivo.

El Departamento de Control de la Calidad tiene entre sus funciones fundamentales las relacionadas con el muestreo, las especificaciones, los ensayos y la evaluación de la calidad de los productos que se generan en la empresa. Para acometer estas funciones, el departamento está compuesto por dos secciones y dos grupos de trabajo: el grupo de desarrollo y el grupo de recepción de muestras y manipulación de expedientes.

Las secciones en las que está dividida son:

La sección biológica que está compuesta por cinco laboratorios

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

Y la sección físico-química que está compuesta por tres laboratorios:

- Laboratorio Análisis Químico.
- Laboratorio de Cromatografía y Electroforesis.
- Laboratorio de Sistemas Críticos.

En el laboratorio de Biología Molecular se realizan los ensayos de determinación de ADN contaminante en muestras del ingrediente farmacéutico activo (IFA) y muestras de proceso de cada uno de los productos del centro. ¹

Actualmente en el laboratorio experimentan un gran crecimiento de la información que generan los diferentes ensayos. Es entonces que se dificulta el correcto manejo de este cúmulo de datos para poder establecer un estricto control que avale la calidad requerida en cada producto.

A través de un Sistema de Gestión de Información del Laboratorio (LIMS, del inglés *Laboratory Information Management System*), es posible procesar, almacenar y distribuir información (datos organizados, estructurados) que ayudan a tomar decisiones estratégicas, analizar problemas y controlar una organización. Los LIMS proveen importantes beneficios a la gestión de la información de los laboratorios, algunos de estos son:

- Mejor organización de la información.
- Integridad de la información adquirida en los ensayos.
- Aseguramiento de la Calidad.
- Obtención y validación de resultados.
- Transferencia automática de la información.
- Rapidez en la gestión de informes.

¹ Centro de Ingeniería Genética y Biotecnología. *Departamento de Control de la Calidad* 2003a. [2006]. Disponible en: <http://www.cigb.edu.cu/pages/ccalidad.htm>.

Estas son las razones que conllevaron a que a mediados del año 2006 se comenzara, como parte de un proyecto productivo, a desarrollar un Sistema de Gestión de la Información de los Laboratorios de la dirección de Calidad del CIGB por un grupo de estudiantes de la facultad 6 de la Universidad de las Ciencias Informáticas, la cual implementa en el Plan de Estudios de sus estudiantes, un Segundo Perfil relacionado con la Bioinformática. Hasta el mes de Julio del año 2008 se estuvo realizando el flujo de trabajo de Análisis y Diseño definido por la metodología de desarrollo: Proceso Unificado de Desarrollo (RUP, del inglés *Rational Unified Process*), y del cual se generó la investigación precedente a este trabajo titulada "Sistema para la Gestión de la Información de los Laboratorios de la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis y Diseño del módulo Biología Molecular". Con la realización de este trabajo se pretende dar continuidad a este proceso previamente iniciado.

Por la necesidad que tiene este centro de contar con un Sistema de Gestión del laboratorio se plantea el siguiente **problema a resolver**: ¿Cómo obtener un producto funcional a partir de las clases diseñadas para el módulo "Laboratorio Biología Molecular" del LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología?

El **objeto de estudio** se enmarca en el proceso de desarrollo de los Sistemas de Gestión de la Información de los Laboratorios.

El **campo de acción** se centra en la implementación de los sistemas de gestión de la información de los laboratorios.

Proponiéndose como **objetivo general** (desarrollar el módulo del Laboratorio de Biología Molecular, para el Sistema de Gestión de la Información de los Laboratorios de la Dirección de Calidad del CIGB.)

Para alcanzar con éxito el objetivo general se debe dar seguimiento a las **tareas de la investigación**:

- Estudio del modelado del negocio.
- Estudio del modelo del sistema.
- Estudio del diseño propuesto en la investigación precedente.
- Confección de los diagramas de componentes e implementación del sistema.
- Implementación de los componentes necesarios para el funcionamiento del módulo.

La investigación realizada se estructurará en dos capítulos como sigue:

Capítulo 1. FUNDAMENTACIÓN TEÓRICA: En este capítulo se expone todo lo relacionado con el estado del arte del objeto de estudio. Se mencionan algunos sistemas vinculados al campo de acción, se realiza una descripción del flujo de trabajo de implementación así como sus objetivos. Se explica en términos conceptuales que es un artefacto, una actividad, y un rol, también se realiza una valoración de los procesos de desarrollo de software, de los artefactos y actividades que intervienen en el rol de implementador del proceso unificado de desarrollo de software. Así como las tendencias, metodologías y tecnologías actuales relacionadas con el objeto de estudio.

Capítulo 2. IMPLEMENTACION DEL SISTEMA: Se exponen los artefactos fundamentales que se obtienen como resultado del flujo de trabajo de implementación. Para la obtención de los diagramas de componentes documentados, se tiene como base las descripciones de los Casos de Uso y los diagramas de clases del diseño, que se obtuvieron en la investigación de análisis y diseño precedente a este trabajo. Los fragmentos de código fuente que se presentan en este capítulo son significativos para comprender las funcionalidades de la aplicación.

Por último se tienen las recomendaciones, referencias bibliográficas, bibliografías y anexos.

Capítulo 1

FUNDAMENTACIÓN TEÓRICA

En este capítulo se expone todo lo relacionado con el estado del arte del objeto de estudio. Se mencionan algunos sistemas vinculados al campo de acción, se realiza una descripción del flujo de trabajo de implementación que propone RUP (del inglés *Rational Unified Process*) así como sus objetivos. Se explica en términos conceptuales que es un artefacto, una actividad, y un rol, también se realiza una valoración de los procesos de desarrollo de software, de los artefactos y actividades que intervienen en el rol de implementador del proceso unificado de desarrollo de software. Así como las tendencias, metodologías y tecnologías actuales relacionadas con el objeto de estudio.

1.1 Sistemas automatizados existentes vinculados al campo de acción

Las aplicaciones LIMS (*Laboratory Information Management System*, o Sistema de Gestión de la Información del Laboratorio) son herramientas esenciales para la reducción de costes y aumento de la eficiencia y productividad en las actividades realizadas en los laboratorios. Dado el creciente interés de las empresas hacia la calidad como factor clave para favorecer la productividad, la eficacia y la imagen de los productos o servicios suministrados, las aplicaciones LIMS representan la solución imprescindible para una gestión moderna del control y aseguramiento de la calidad.²

Entre las aplicaciones LIMS existentes se pueden citar:

MediLab Interfaces

MediLab Interfaces es un sistema de información que permite extraer los datos de los resultados de los equipos de análisis y almacenarlos en el LIMS de su laboratorio. Al implantar una interfaz entre sus equipos de análisis y su LIMS se evitan la transcripción de resultados, trayendo estos como

² *Matrix LIMS, la herramienta de productividad para cualquier laboratorio.* Disponible en: <http://www.softwarecientifico.net/productos.asp?pid=472>

consecuencia la desaparición de los errores de transcripción, disminución del tiempo de reporte de resultados y la reducción de los costos del procesamiento de muestras.

MediLab Interfaces también puede integrarse a otros LIMS, en este caso es necesario crear un programa de interpretación para que permita colocar los resultados en su LIMS, para ello se debe contar con acceso a la base de datos de su LIMS. ³

Funcionalidades:

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

LabWare LIMS

LabWare LIMS es un sistema de información de laboratorio (LIMS) con acceso equivalente cliente/servidor y Web, que se integra sin costuras en cualquier entorno informático. Gracias a su completa funcionalidad para seguimiento de muestras, certificación de usuarios, gestión de instrumentos, auditoría, y planificación de muestras e informes, LabWare LIMS cumple las buenas prácticas de Laboratorio y los requisitos respecto a almacenamiento y firma electrónica.

El primer objetivo de LabWare es el éxito de sus clientes. LabWare concentra sus recursos únicamente en LIMS, y sus clientes lo reconocen con distinciones tales como el *Reader's Choice Award* de la revista *Scientific Computing and Instrumentation*. Otro reconocimiento es el de Calidad del Producto de *Frost&Sullivan*, un estudio que compara diferentes productos de acuerdo con su facilidad de operación e integración, fiabilidad del sistema y calidad del soporte. ⁴

Funcionalidades

³ *MediLab Interfaces*. Disponible en: <http://medilabit.com/interfaces.htm>

⁴ *LabWare LIMS - Sistema de Información de Laboratorio*. Disponible en: <http://www.chemie.de/products/es/52345/>

- LabWare LIMS puede leer y escribir las bases de datos de aplicaciones externas, así que en muchos casos no es necesario hacer mucha configuración sobre las otras aplicaciones.
- Los eventos programados, como informes, el registro de entrada de muestras y las rutinas en LIMS Basic pueden estar pre-configuradas y luego activarse basándose en una planificación.⁵

Para la adquisición de un LIMS es necesario realizar un conjunto de estudios previos, empezando por la necesidad de visitar laboratorios que trabajen con este sistema, evaluar el equipo computacional necesario para el soporte, definir los instrumentos para que estén Online al LIMS, modificación del LIMS para una personalización del laboratorio (Reportes), estudiar el sistema de Facturación requerido y el alcance de posteriores *Updates* y conexión de nuevos instrumentos.

También es importante considerar los costos en cuanto a la capacitación del personal de laboratorio, manejo básico de computadoras, instalación, mantenimiento y soporte del software. Debido a que cada laboratorio es muy personalizado, es muy difícil la elección de un LIMS comercial, debido a que estas herramientas para ser adaptadas a una determinada organización deben tener un alto nivel de configuración según las características de la entidad. Es por eso que se decidió desarrollar un sistema propio, en vista de que las funcionalidades y configuraciones de estos sistemas no se adecuan a las necesidades del CIGB.

Otra de las características de estas herramientas es que no son herramientas libres, han sido desarrolladas bajo la plataforma .NET para las diferentes versiones de Microsoft Windows, lo que implicaría altos gastos para las entidades con necesidades de adquirir un LIMS.

Las características del CIGB impiden configurar un LIMS para su uso particular, porque se realizan distintos ensayos a diferentes productos y eso provoca que se necesite un sistema adaptable a las necesidades específicas de la Dirección de Calidad del CIGB.

1.2 Tendencias, Tecnologías y Herramientas utilizadas

1.2.1 Sistemas de Gestión de Bases de Datos (SGBD)

⁵ Idem Referencia 4.

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.⁶

Para plasmar los tres niveles en el enfoque o modelo de datos seleccionado, es necesaria una aplicación que actúe de interfaz entre el usuario, los modelos y el sistema físico. Esta es la función que desempeñan los SGBD, y que pueden definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

Un SGBD tiene los siguientes objetivos específicos:

- Independencia de los datos y los programas de aplicación.
- Minimización de la redundancia.
- Integración y sincronización de las bases de datos.
- Integridad de los datos.
- Seguridad y protección de los datos.
- Facilidad de manipulación de la información.

PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (*ORDBMS*, del inglés *Object-Relational Database Management Systems*) basado en el proyecto POSTGRES, de la universidad de Berkeley, que ha sido desarrollado de varias formas desde 1977. El proyecto PostgreSQL tiene actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de

⁶ Matos García Rosa María. "Diseño de Bases de Datos". octubre 1999. Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=10192&subdir=/Bibliografia>

desarrolladores y contribuidores de código abierto. Incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.⁷

PostgreSQL es ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Cuenta con una gama de servicios online para su desarrollo, además de una amplia comunidad online.

Ventajas:

- Soporta transacciones y desde la versión 7.0, llaves foráneas (integridad referencial).
- Soporta un subconjunto de SQL92 mayor que el que soporta MySQL.

Desventajas:

- Consume bastantes recursos y carga más el sistema.

Por las características y ventajas expuestas anteriormente se decide utilizar este Sistema Gestor de Base de Datos, además de que es *Open Source* bajo licencia BSD, es decir, sus potencialidades están en constante perfeccionamiento, permitiendo su uso y distribución sin costo. Continuamente se eliminan y mejoran cualquier hueco de seguridad que pueda aparecer. En caso que se esté trabajando con sistemas serios, la integridad referencial de los datos es muy buena, se pueden crear funciones complejas para validar datos. Es también utilizado porque es un SGBD potente y multiplataforma.

1.2.2 Internet. Funcionamiento

Internet permite interconectar a millones de computadoras para formar un nuevo mundo. Estas son capaces de compartir información a través de un conjunto de protocolos TCP/IP.

⁷ Domínguez Vaillant Arodys E. y Miranda Gutiérrez Duniel. "Sistema de Manejo de Datos de Ensayos Clínicos: Diseño e implementación de la Base de Datos." p 23.

Lógicamente para lograr esta interconexión es necesario un canal de comunicación. Básicamente hoy día uno de los protocolos de mayor uso es HTTP (protocolo de transferencia de hipertexto). Es usado para el envío y recepción de páginas web, además que es posible la transmisión de ficheros, fotos y respuestas a programas.

1.2.3 Aplicaciones Web.

Las aplicaciones Web permiten a los usuarios del sistema interactuar con estas, logrando así una interacción dinámica. Los sitios web son un conjunto de ficheros alojados en un servidor web, generalmente no se necesita de arquitecturas complejas para lograr su buen funcionamiento. Ambos necesitan en común un servidor web por ejemplo: Hoy día ha aumentando el uso de Apache (Software Libre) como preferencia, un lenguaje de programación como por ejemplo: PHP, JSP, Perl, Python, ASP/ASP.Net; aunque ciertamente ASP no es un lenguaje de programación, sino una arquitectura de desarrollo web en la que se pueden usar por debajo distintos lenguajes (por ejemplo VB.NET o C# para ASP.NET, o VBScript/JScript para ASP), un Sistema Gestor de Base de Datos (DBMS, del inglés *DataBase Management Systems*), los más utilizados son: Microsoft SQL Server, Oracle, IBM Informix, IBM DB2 todas estas versiones comerciales; también se tienen otro conjunto importante de versiones libres como: PostgreSQL, MySQL, y SQLite, que han aumentado su uso a nivel mundial a través del tiempo.

Para que estas aplicaciones sean usadas por los usuarios es necesario tener solamente un navegador Web en el lado del cliente que realice peticiones HTTP al servidor e interprete código HTML para dar formato a la información.

1.2.4 Entorno Distribuido. Modelo Cliente Servidor.

La arquitectura Cliente-Servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.⁸

⁸ *Arquitectura Cliente Servidor*. Disponible en: <http://www.csi.map.es/csi/silice/Global71.html>

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

El modelo Cliente Servidor ofrece una serie de ventajas que facilitan la calidad de los sistemas:

- El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. El cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.
- El sistema es fácil de escalar.

1.2.5 Servidor Web Apache

El servidor Web Apache es uno de los servidores Web más usados en el mundo. Más del 60% de los servidores Web del mundo lo usan, existen muchas características que hacen a Apache ser el mejor servidor Web, estas son:

- Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Cualquiera que posea una experiencia en la programación de C o Perl puede escribir un módulo para realizar una función determinada. Esto significa que hay una gran cantidad de módulos Apache disponibles para su utilización.
- Apache es una tecnología gratuita de código abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto.
- Apache trabaja con Perl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y Apache utiliza su parte de Perl tanto con soporte CGI como con soporte modperl.
- Apache funciona en Linux y en otros sistemas de Unix, además de que también funciona en Windows.
- Soporte para CGI (*Common Gateway Interface*): Apache soporta CGI utilizando los módulos mod_cgi y mmod_cgid. Es compatible con CGI y aporta características extendidas como personalización de las variables de entorno y soporte de reparación de errores o debugging, que son difíciles de encontrar en otros servidores Web.
- Soporte de host virtuales: Apache es además uno de los primeros servidores Web en soportar tanto host basados en IP como host virtuales.

- Soporte de autenticación HTTP: Apache soporta autenticación básica basada en la Web. Está también preparado para autenticación basada en la digestión de mensajes, que es algo que los navegadores Web populares ya han implementado. Apache puede implementar autenticación básica utilizando tanto archivos estándar de contraseña como los DBM, llamadas a SQL o llamadas a programas externos de autenticación.
- Soporte de scripts PHP: Este lenguaje de script ha comenzado a ser muy utilizado y Apache ofrece un amplio soporte de PHP utilizando el módulo `mod_php`.⁹

1.2.6 Lenguajes de Programación Web.

La programación Web, parte de las siglas WWW, que significan *World Wide Web* o telaraña mundial. Para realizar una página con la programación Web, se deben tener claros, tres conceptos fundamentales los cuales son, el URL (*Uniform Resource Locators*), es un sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una página Web, un servicio o cualquier otra cosa. En resumen el URL no es más que un nombre, que identifica una computadora, dentro de esa computadora un archivo que indica el camino al recurso que se solicita. El siguiente concepto dentro de la programación Web, es el protocolo encargado de llevar la información que contiene una página Web por toda la red de Internet, como es el HTTP (*Hypertext Transfer Protocol*). Con el comienzo de Internet y la programación Web, se desfilaron los diseños gráficos tradicionales, con lo que se empezaron a diseñar interfaces concretas para este medio, buscando ficheros pequeños para facilitar la carga de los mismos. La programación Web se orientaba a un diseño muy cargado interactuando con el usuario, mientras que al empezar a competir con millones de Webs se ha optado más por el diseño sencillo y de fácil comprensión.

Entre los lenguajes que trabajan del lado del servidor se pueden citar algunos, que se destacan por ser los más sobresalientes como son PERL, ASP, PHP, Java y JSP. Dentro de los lenguajes que trabajan del lado del cliente se encuentran el JavaScript, XSLT y el Visual Basic Script, estos dos últimos al combinarse con el HTML forman lo que se conoce como DHTML, es decir, salida estándar dinámica o HTML dinámico.

⁹ Mohammed J. Kabir. *La biblia Servidor Apache 2*, p. 41-43.

1.2.7 Lenguajes de programación Web a utilizar.

Personal Home Page5 (PHP5).

PHP es un lenguaje de programación usado normalmente para la creación de páginas web dinámicas. PHP es un acrónimo recursivo que significa "*PHP Hypertext Pre-processor*" (inicialmente *PHP Tools*, o, *Personal Home Page Tools*), y se trata de un lenguaje interpretado. Últimamente también puede ser utilizado para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica.¹⁰

Fue creado primeramente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. 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 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 Programación Orientada a Objetos para solucionar las carencias de las anteriores versiones.

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 de código abierto y de obtención gratuita.
- Portable y multiplataforma (W95, 98, 2000, XP, NT, Unix, Linux...) lo cual permite su desarrollo desde Sistemas Operativos de base heterogéneos.
- Proporciona soporte para la mayoría de protocolos de comunicación de Internet (HTTP, IMAP, FTP, LDAP, INMP).

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.

¹⁰ Sæther Bakken Stig, Aulbach Alexander, Schmid Egon, Colectivo de Autores. *Manual de PHP*.

- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- 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 de bajo nivel.

Luego de realizado un análisis por la dirección del proyecto, se propuso la utilización de PHP5 debido principalmente a que esta tecnología de desarrollo de aplicaciones generalmente combinada con una base de datos PostgreSQL, reduce los costos, ambos son distribuidos mediante licencia GNU, lo cual no implican un licenciamiento comercial. La portabilidad de PHP5 y PostgreSQL le permiten migrar un sistema sin problemas y sin cambios desde Linux o Unix a plataforma Windows, y viceversa en caso de ser necesario.

JavaScript

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.¹¹

Es un lenguaje muy rico en posibilidades, pues permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos mucho más complejas como así lo requiera el problema a resolver, etc. Toda esta potencia que caracteriza a Javascript se pone a disposición del programador y este será el único que deberá aprovecharla, para así convertirse en el verdadero dueño y controlador de cada cosa que ocurre en la página.

Este lenguaje posee varias características, es un lenguaje basado en acciones que posee menos restricciones. Javascript nació con la necesidad de permitir a los autores de sitios Web crear páginas que permitan intercambiar con los usuarios, ya que se necesitaba crear Webs de mayor complejidad.

¹¹ Eguilus Pérez Javier. Introducción a JavaScript. Disponible en:
<http://www.librosweb.es/javascript/capitulo1.html>

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa *Netscape Communications*. Utilizado principalmente en páginas Web. Es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox.

Los programas JavaScript tienden a ser pequeños y compactos (en comparación con los applets de Java), no requieren mucha memoria ni tiempo adicional de transmisión. Además, al incluirse dentro de las mismas páginas HTML se reduce el número de accesos independientes a la red.

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).

La dirección del proyecto determinó la utilización de JavaScript como lenguaje de programación del lado del cliente por todas las características y ventajas que presenta.

1.2.8 Entorno Integrado de desarrollo (IDE).

Eclipse

Un IDE permite hacer el trabajo mucho más sencillo, sobretodo si el desarrollo ya va manejando un buen número de clases. Estos entornos facilitan más versatilidad para depurar los programas puesto que tienen *debuggers* mucho más avanzados.

Eclipse es una poderosa herramienta que permite integrar diferentes aplicaciones para construir un entorno integrado de desarrollo (IDE). Es un armazón sobre el que se pueden montar herramientas de

desarrollo para cualquier lenguaje, mediante la implementación de los plugins adecuados. La arquitectura de plugins de Eclipse permite, además de integrar diversos lenguajes sobre un mismo IDE, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo como: herramientas UML, editores visuales de interfaces, ayuda en línea para librerías.

Características:

- Editor de textos.
- Resaltado de sintaxis.
- Compilación en tiempo real.
- Pruebas unitarias con JUnit.
- Control de versiones con CVS.
- Asistentes (wizards) para creación de proyectos, clases, test, etc.
- A través de plugins libremente disponibles es posible añadir:
- Control de versiones con Subversión.
- Integración con Hibernate.

Ventajas:

- Es una herramienta open-source.
- Soporta la construcción de una variedad de herramientas para el desarrollo de aplicaciones.
- Soporta herramientas que manipulan diferentes tipos de archivos como por ejemplo java, C, C++, HTML, GIF.
- Corre en una gran cantidad de sistemas operativos, incluyendo Linux y Windows.
- Provee a los desarrolladores herramientas que facilitan la creación de plugins.
- Mediante *Java Development Toolkit (JDT)* facilita la creación de aplicaciones programadas en java.

Eclipse fue desarrollado originalmente por IBM como sucesor de su familia de herramientas para *VisualAge*. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos, complementarios, capacidades y servicios.¹²

¹² *Plataforma Comunidad en Español de Eclipse IDE*. Disponible en: <http://plataformaclipse.com/>

1.2.9 Patrones de Arquitectura

Los patrones de arquitectura expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos.¹³

Patrón Modelo-Vista-Controlador

El Modelo-Vista-Controlador (*Model-View-Controller*, en adelante MVC) fue introducido inicialmente en la comunidad de desarrolladores de Smalltalk-80. MVC divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

- **Modelo (*Model*):** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- **Vista (*View*):** Muestra la información al usuario. Obtiene los datos del modelo. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- **Controlador (*Controller*):** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio ("*service requests*" en el texto original) para el modelo o la vista. El usuario interactúa con el sistema a través de los controladores.¹⁴

Este patrón es muy popular y ha sido portado a una gran cantidad de entornos y frameworks entre los que se encuentran WinForms, ASP.Net y Symfony.

1.2.10 Framework utilizado

¹³ LEON WELICKI. *Patrones y Antipatrones: una Introducción - Parte II*. Disponible en:
http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317/default.aspx

¹⁴ Idem Referencia 13.

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes.¹⁵ Proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener, encapsulando operaciones complejas en instrucciones sencillas.

Los Frameworks facilitan el desarrollo de software, proporcionan una estructura definida la cual ayuda a crear aplicaciones y brindarle mantenimiento con mayor rapidez gracias a la organización durante el desarrollo de la aplicación.

Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones Web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web.

¹⁵ Fabien Potencier, François Zaninotto. *Symfony la guía definitiva*, 2008. Disponible en: <http://www.librosweb.es/symfony> , p 13.

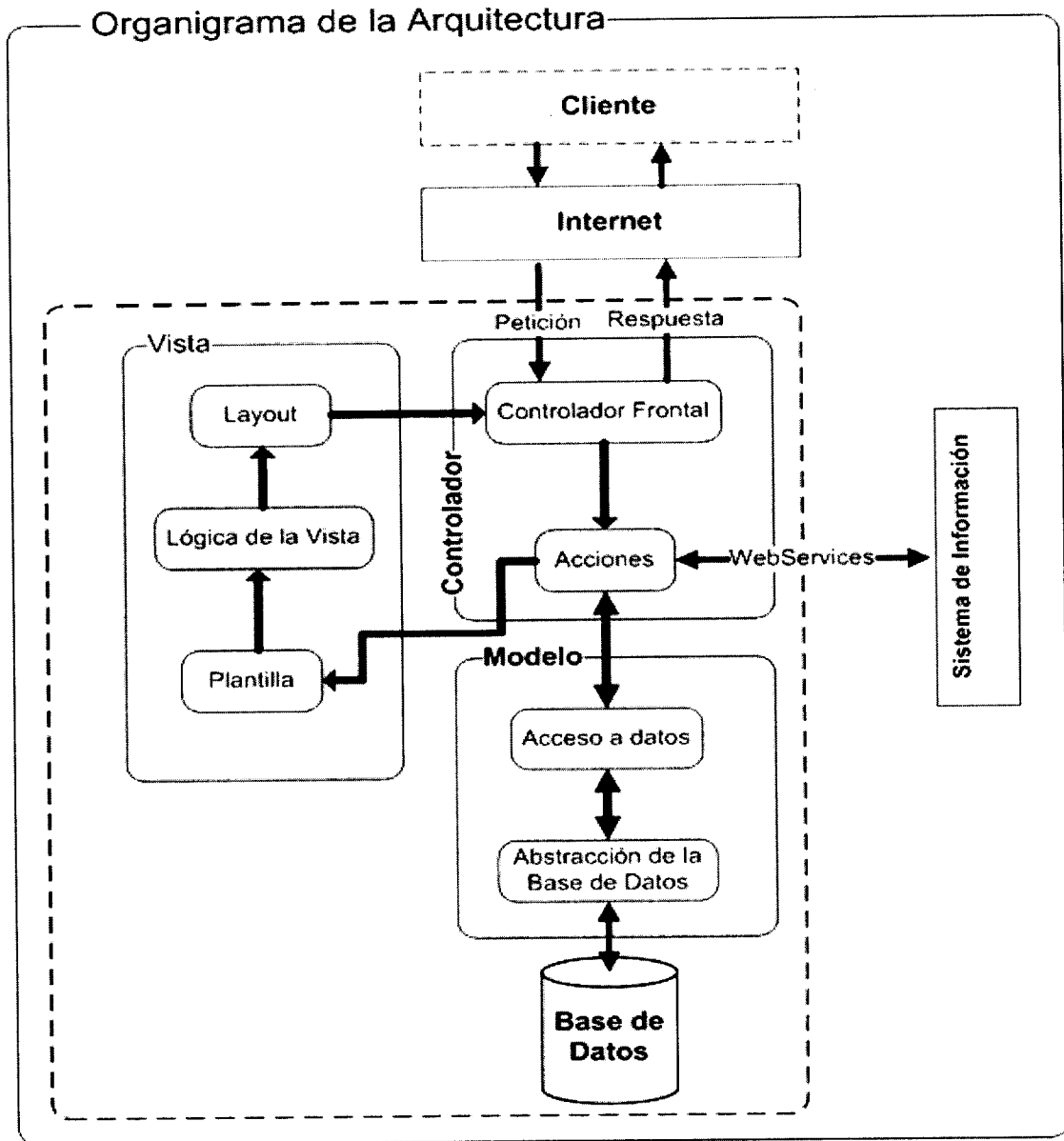


Figura 2. Organigrama de Arquitectura en Symfony.

Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.¹⁶

¹⁶ Ídem Referencia 15.

Symfony es compatible con la mayoría de gestores de base de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft.

Características de Symfony

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

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares)
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- 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.

El principal argumento que influyó en la decisión de utilizar Symfony, es debido primeramente a que la realización de esta investigación es parte de un proyecto y en el proyecto existen una gran cantidad de módulos que necesitan que sean implementados.

Por otra parte en los diferentes módulos se desarrollan aplicaciones web complejas con mucha lógica de negocio, no es recomendable utilizar solo PHP. Para asegurar el mantenimiento y las ampliaciones futuras del LIMS de Calidad del CIGB, es necesario que el código sea ligero, legible y efectivo permitiendo incorporar los últimos avances en interacción con usuarios (como por ejemplo Ajax), también se hace necesario escribir cientos de líneas de JavaScript para validaciones que son precisas, para todos estos casos, Symfony es un framework que se ajusta a estas necesidades.

1.2.11 Metodología de desarrollo de software

Cada día la producción de software busca adecuarse más a las necesidades del usuario, esto trae como consecuencia que aumente en tamaño y complejidad. Para lograr la productividad del software se necesita un proceso que integre las múltiples facetas del desarrollo del mismo.

Entre las metodologías de desarrollo de software más reconocidas a nivel mundial se destaca el Proceso Unificado de Desarrollo (RUP), que junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

RUP es un proceso para el desarrollo de un proyecto de software, que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Como tres características esenciales está dirigido por los casos de uso: que orientan el proyecto a la importancia para el usuario y lo que este quiere, está centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en que orden y es iterativo e incremental: donde divide el proyecto en mini proyectos, donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

17

Entre las características fundamentales de RUP se pueden citar:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo interactivo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

RUP está basado en componentes. Utiliza el lenguaje unificado de modelado UML para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial de RUP, sus desarrollos fueron paralelos. No obstante los verdaderos aspectos definitorios del proceso unificado se

¹⁷ IVAR JACOBSON, G. B., JAMES RUMBAUGH. El Proceso Unificado de Desarrollo de Software., 2000. Pág 4.

resumen en tres fases claves: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. ¹⁸

Todo proceso de desarrollo de software es riesgoso y difícil de controlar, de ahí la necesidad de tener una metodología que garantice cumplir con dos cosas fundamentales: con los planes de producción del software y la satisfacción del cliente. La metodología RUP, por su adaptabilidad y las condiciones de desarrollo que ofrece, garantizan los cumplimientos de las metas trazadas.

1.2.11.1 Flujo de Trabajo: Implementación

La metodología RUP está compuesta por 9 flujos de trabajo, entre ellos el flujo de trabajo de Implementación, al cual se le prestará mayor atención durante el desarrollo de esta investigación.

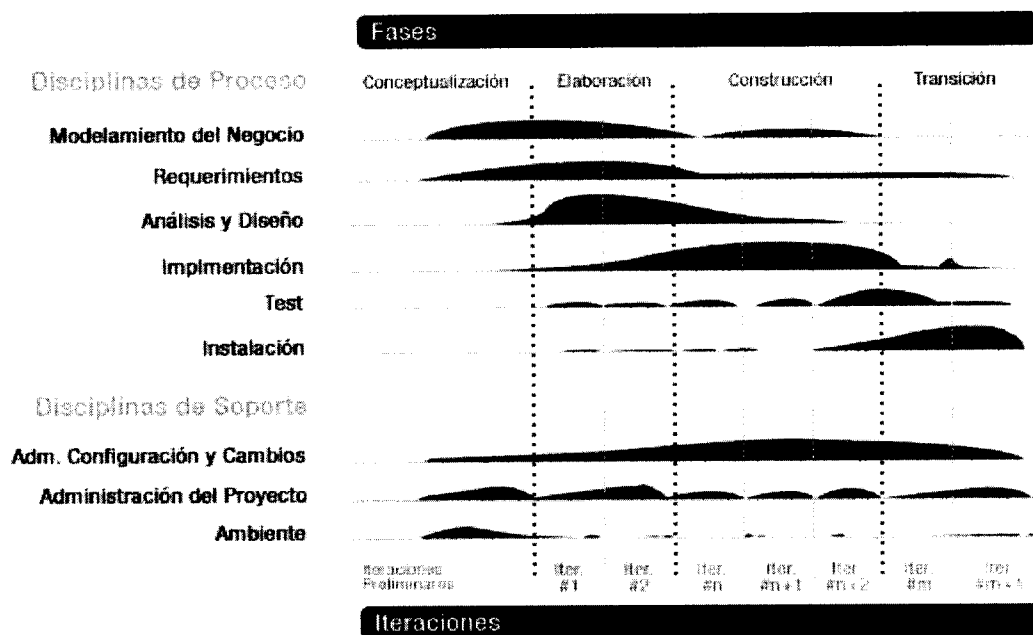


Figura 3. Flujos de Trabajo de RUP

Un flujo de trabajo es una realización de un caso de uso del negocio o parte de él. Puede describirse en términos de diagramas de actividad, que incluyen a los trabajadores participantes, las actividades que realizan y los artefactos que producen. ¹⁹

¹⁸ Ídem referencia 17

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

1.2.11.2 Objetivos del Flujo de Trabajo de Implementación

El flujo de trabajo de Implementación tiene los siguientes objetivos:

- Planificar las integraciones de sistema necesarias en cada iteración. Se sigue para ello un enfoque incremental, lo que da lugar a un sistema que se implementa en una sucesión de pasos pequeños y manejables.
- Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue. Esto se basa fundamentalmente en las clases activas encontradas durante el diseño.
- Implementar las clases y subsistemas encontrados durante el diseño. En particular, las clases se implementan como componentes de fichero que contienen código fuente.
- Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones de sistema.²⁰

1.2.11.3 Relación con los demás flujos de trabajo del Proceso Unificado de Desarrollo (RUP)

El flujo de trabajo de implementación se relaciona estrechamente con el resto de los flujos de trabajo que propone el proceso unificado de desarrollo. En el flujo de requerimientos, se capturan los requisitos funcionales y no funcionales y los casos de usos que se van a implementar durante el flujo de trabajo de implementación. En el flujo de trabajo de análisis y diseño se describe cómo desarrollar el modelo de diseño. Este modelo de diseño representa la primera entrada para el flujo de trabajo de implementación.

¹⁹ Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 4.

²⁰ Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 255.

En el flujo de trabajo de prueba se describe cómo hacer las pruebas de integración a cada una de las versiones operacionales del sistema, durante la integración del mismo, describiendo también como realizar las pruebas al sistema con el objetivo de verificar que todos los requerimientos hayan sido implementados, determinando los defectos del sistema en caso de tenerlos.

En el flujo de trabajo de Entorno se describe cómo desarrollar y mantener artefactos de soporte que se usan durante la implementación como la descripción del proceso, las pautas del diseño y las pautas de la programación.

En el flujo de trabajo de despliegue se describe cómo usar al modelo de implementación para producir y entregar el código al cliente. En el flujo de trabajo de gestión de proyecto se describe cómo mejorar el plan de proyecto. ²¹

1.2.11.4 Roles, artefactos y actividades.

El flujo de trabajo de implementación está compuesto por varios elementos entre los cuales se pueden destacar los roles, artefactos y actividades.

Un rol o trabajador es aquel que define el comportamiento y las responsabilidades de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de los elementos resultantes de dichas actividades. ²²

Una actividad es una tarea con un propósito claro, crear o actualizar algún elemento. Es realizada por un rol o trabajador y manipula elementos. ²³

Un artefacto es una pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades, representa un área de responsabilidad y es candidata a ser tenida

²¹ Conferencia UCI "Fase de Elaboración. Flujo de trabajo de implementación", 2008. Disponible en: [http://internos.uci.cu/Teleclases/tc_conferencias.asp?id_asig=ingenieria de software 2](http://internos.uci.cu/Teleclases/tc_conferencias.asp?id_asig=ingenieria_de_software_2)

²² Ídem referencia 21

²³ Ídem referencia 21

en cuenta para el control de la configuración. Un artefacto puede ser un modelo, un elemento de un modelo o un documento. ²⁴

1.2.11.5 Rol Implementador de Sistema

Dentro de los roles propuestos por el Proceso Unificado de Desarrollo se prestará atención al rol de Implementador dentro de un proceso de desarrollo de software.

Un implementador juega un papel fundamental en el ciclo de vida de desarrollo del software, pues tiene como premisa fundamental la de implementar el sistema, también debe de implementar elementos del diseño, elementos de pruebas de diseño y analizar el comportamiento del sistema en tiempo de ejecución y es el encargado además de crear los instaladores del sistema.

El rol de implementador tiene objetivos específicos dentro del proceso de desarrollo de software, entre ellos:

- Planificar las integraciones de sistema necesarias en cada iteración.
- Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue.
- Implementar las clases y subsistemas encontrados durante el diseño.

1.2.11.6 Artefactos y actividades del rol Implementador de Sistemas

En el flujo de trabajo de Implementación se generan una serie de artefactos a través de las actividades desarrolladas por los diferentes roles que intervienen. En la elaboración de este trabajo se generan los artefactos que se mencionan a continuación:

Modelo de Implementación.

Uno de los artefactos que se generan en el flujo de trabajo de implementación es el modelo de implementación que describe cómo las clases del diseño, se implementan en términos de componentes, ficheros de código fuente, ejecutables. ²⁵

²⁴ Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 426

Este modelo puede describir cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización, disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes uno de otro.

Componente

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. Algunos estereotipos estándares de componentes son las librerías, ejecutables, ficheros, tablas, documentos, entre otros.²⁶

Los componentes mantienen relaciones de trazas con los elementos de modelo que implementan. Por ejemplo con las clases del diseño que fueron establecidas.

Diagramas de Componentes.

Estos diagramas son utilizados en mayor parte para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

Se representan como un grafo de componentes de software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten.

Se utilizan para modelar la vista estática de un sistema, la organización y las dependencias lógicas entre un conjunto de componentes de software, sean éstos componentes de código fuente, librerías, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

²⁵ Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 257.

²⁶ Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 258.

Subsistema de implementación.

La importancia de los subsistemas de implementación es que proporcionan una forma de organizar los artefactos del modelo de implementación en trozos más manejables. Un subsistema puede estar formado por componentes, interfaces y otros subsistemas recursivamente.

Además, un subsistema puede implementar y así proporcionar las interfaces que representan la funcionalidad que exportan, en forma de operaciones. Los subsistemas de implementación deberían seguir la traza uno a uno de sus subsistemas de diseño correspondiente.²⁷

En el flujo de trabajo de implementación se desarrollan varias actividades por los roles involucrados. En el desarrollo de este trabajo se realizan las que se describen a continuación:

Implementación de la Arquitectura

El resultado de esta actividad parte de la arquitectura definida en la fase de elaboración. El propósito de la Implementación de la Arquitectura es esbozar el modelo de implementación y su arquitectura mediante la identificación de componentes significativos arquitectónicamente, tales como componentes ejecutables y la asignación de componentes a los nodos en las configuraciones de redes relevantes.²⁸

Recordar que durante el diseño de la arquitectura se esbozan los subsistemas de diseño, sus contenidos e interfaces. Durante la implementación utilizamos subsistemas de implementación que siguen la traza uno a uno a estos sistemas de diseño y proporcionan las mismas interfaces.

Implementar un Subsistema

El propósito de implementar un subsistema es el de asegurar que un subsistema cumple su papel en cada construcción. Se asegura que los requisitos implementados en la construcción son implementados correctamente por componentes o por otros subsistemas dentro del sistema.

Implementar una clase

²⁷ Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 260.

²⁸ Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 268.

El propósito de la implementación de una clase es implementar una clase de diseño en un componente fichero.²⁹

Para la elaboración de este trabajo se determinó que se utilizaría la metodología RUP como metodología de desarrollo de software por su adaptabilidad al proyecto y las condiciones de desarrollo que ofrece. Se define el rol de implementador como el rol a desempeñar, así como los artefactos que son generados: Modelo de Implementación, Subsistema de Implementación, Componente y Diagrama de Componentes, además de las actividades que son realizadas y que se citan a continuación: Implementación de la Arquitectura, Implementación de los subsistemas e Implementación de las clases.

1.2.12 Lenguaje de Modelado

UML (Lenguaje Unificado de Modelado) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.³⁰

Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios.

Hoy día, UML está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código, pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. El UML esta compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

²⁹ Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 282.

³⁰ Largman Craig. *Uml y Patrones. Introducción al análisis y diseño orientado a objetos*, 2004.

En otros términos, así como en la construcción de un edificio se realizan planos previo a su construcción, en Software se deben realizar diseños en UML previa codificación de un sistema, ahora bien, aunque UML es un lenguaje, éste posee más características visuales que programáticas, mismas que facilitan a integrantes de un equipo multidisciplinario participar e intercomunicarse fácilmente, estos integrantes siendo los analistas, diseñadores, especialistas de área y desde luego los programadores.

1.2.13 Herramienta CASE

Herramientas CASE, (del inglés *Computer Aided Software Engineering*) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores.³¹

Objetivos:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Mejorar el tiempo, coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto.
- Aumentar la biblioteca de conocimientos informáticos de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar, desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.

Lista de aplicaciones CASE:

- Database Designer for MySQL

³¹ Herramientas Case. Disponible en:

http://209.85.215.104/search?q=cache:7UN1dYGFZGoJ:www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r19670.DOC+Herramientas+CASE:+son+diversas+aplicaciones+inform%C3%A1ticas+destinadas+a+aumentar+la+productividad&hl=es&ct=clnk&cd=5&gl=cu&lr=lang_es

- Eclipse
- Embarcadero ER/Studio
- Enterprise Architect
- MetaCASE
- Oracle Designer
- Rational Rose
- System Architect
- Umbrello
- Visible Enterprise Products
- Visual Paradigm for UML

Dentro de la lista de aplicaciones CASE que se mencionan anteriormente, la que se utiliza para la realización de este trabajo es Visual Paradigm.

Visual Paradigm

Es una herramienta CASE que utiliza UML como lenguaje de modelado.³² Está diseñada para distintos usuarios entre los que se incluyen ingenieros de software, analistas de sistemas, analistas de negocio, arquitectos y desarrolladores. Facilita la interoperabilidad con otras herramientas CASE y se integra con las siguientes herramientas: NetBeans, JBuilder, Oracle JDeveloper, Eclipse entre otras.

Está orientada a la creación de diseños usando el paradigma de programación orientada a objetos. Visual Paradigm incluye una herramienta llamada Visual Architect que permite la generación de código para el manejo de la base de datos. Con esta herramienta se puede generar código para los lenguajes PHP, JAVA y C# y para los gestores de base de datos DB2, Informix, SQL Server, MYSQL, Oracle y PostgreSQL. Esta disponible en varias ediciones: Professional, Enterprise, Standard, Modeler y Personal. También Visual Parading ofrece:

- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.

³² Almenteros Yuniesky, Sánchez Yusdenis. "Software para la Simulación de Sistemas Biológicos: Módulo de modelación gráfica de Sistemas Biológicos" p 19

- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

Conclusiones

En este capítulo se mencionan los objetivos del flujo de trabajo de implementación, los artefactos generados y las actividades desarrolladas en dicho flujo de trabajo, así como el papel fundamental que juega el implementador en el ciclo de vida de desarrollo del software. También se da una panorámica del estudio de las tendencias actuales de las herramientas, tecnologías y metodologías de desarrollo de software que serán utilizadas a lo largo del desarrollo del sistema propuesto, y se fundamentaron las elecciones del lenguaje usado (PHP), el sistema gestor de bases de datos (PostgreSQL). Symfony como framework a utilizar y como metodología a utilizar (RUP).

Capítulo IMPLEMENTACIÓN DEL SISTEMA.

2

En el presente capítulo se exponen los artefactos fundamentales que se obtienen como resultado del flujo de trabajo de implementación. Para la obtención de los diagramas de componentes documentados, se tiene como base las descripciones de los CU y los diagramas de clases del diseño, que se obtuvieron en la investigación de análisis y diseño precedente a este trabajo. Los fragmentos de código fuente que se presentan en este capítulo son significativos para comprender las funcionalidades de la aplicación.

2.1 Requerimientos Funcionales del módulo Biología Molecular

En la investigación precedente se realizó el análisis y diseño del módulo Biología Molecular, obteniéndose los siguientes Requisitos Funcionales:

CU Gestionar Solicitud de destrucción de bancos de células (SIC-0840)

RF 1- Crear nueva Solicitud de destrucción de bancos de células.

RF 2- Modificar datos del registro de Solicitud de destrucción de bancos de células.

2.1 - Registrar traza

RF 3- Buscar y visualizar el registro de Solicitud de destrucción de bancos de células.

RF 4- Imprimir Solicitud de destrucción de bancos de células.

CU Gestionar registro de Destrucción de Banco de Células (SIC-0841)

RF 5- Crear nuevo registro de Destrucción de Banco de Células

RF 6- Modificar datos del registro de Destrucción de Banco de Células

6.1- Registrar traza

RF 7- Buscar y visualizar el registro de Destrucción de Banco de Células

RF 8- Imprimir el registro de Destrucción de Banco de Células

CU Gestionar registro de Temperatura (SIC-0125R)

RF 9- Crear nuevo registro de Temperatura

RF 10- Modificar datos del registro de Temperatura

10.1- Registrar traza

RF 11- Buscar y visualizar el registro de Temperatura

RF 12- Imprimir el registro de Temperatura.

CU Gestionar Cronograma de chequeo de bancos de células (SIC-0130)

RF 13- Crear nuevo Cronograma de chequeo de bancos de células.

RF 14- Modificar datos del Cronograma de chequeo de bancos de células

14.1-Registrar traza

RF 15- Buscar y visualizar el Cronograma de chequeo de bancos de células

RF 16- Imprimir el Cronograma de chequeo de bancos de células.

CU Gestionar Certificado de Liberación de los bancos de Células transformados (SIC-0229)

RF 17- Crear nuevo Certificado de Liberación de los bancos de Células transformados

RF 18- Modificar datos del Certificado de Liberación de los bancos de Células transformados

18.1- Registrar traza.

RF 19- Buscar y visualizar el Certificado de Liberación de los bancos de Células transformados.

RF 20- Imprimir el Certificado de Liberación de los bancos de Células transformados.

CU Gestionar registro de Descargo para Banco de Células (SIC-0190)

RF 21- Crear nuevo registro de Descargo para Banco de Células.

RF 22- Modificar datos del registro de Descargo para Banco de Células.

22.1- Registrar traza

RF 23- Buscar y visualizar el registro de Descargo para Banco de Células.

RF 24- Imprimir el registro de Descargo para Banco de Células.

CU Gestionar Informe de análisis de los bancos de células (SIC-0950)

RF 25- Crear nuevo Informe de análisis de los bancos de células

RF 26- Modificar datos del Informe de análisis de los bancos de células

26.1- Registrar traza

RF 27- Buscar y visualizar el Informe de análisis de los bancos de células

RF 28- Imprimir el Informe de análisis de los bancos de células.

CU Gestionar Registro de preparación de soluciones (SIC-0020)

RF 29- Crear nuevo Registro de preparación de soluciones.

RF 30- Modificar Registro de preparación de soluciones.

30.1 -Registrar traza.

RF 31- Buscar y visualizar Registro de preparación de soluciones.

RF 32- Imprimir Registro de preparación de soluciones.

CU Gestionar Libro de entrada de muestras de producción

RF 33- Crear libro de entrada de muestras de producción.

RF 34- Registrar datos en el Libro de entrada de muestras de producción.

RF 35 - Modificar datos en el Libro de entrada de muestras de producción.

35.1- Registrar traza.

RF 36- Buscar y visualizar Libro de entrada de muestras de producción.

RF 37- Imprimir datos del Libro de entrada de muestras de producción.

CU Gestionar Registro de Desviaciones de Ensayo Analítico (SIC-0028)

RF 38- Crear nuevo Registro de Desviaciones de Ensayo Analítico.

RF 39 - Buscar y Visualizar Registro de Desviaciones de Ensayo Analítico.

RF 40- Imprimir Registro de Desviaciones de Ensayo Analítico.

2.2 Descripciones textuales de los Casos de Uso

Con el objetivo de entender plenamente las funcionalidades del sistema, se presentan las descripciones de los casos de uso. Estas descripciones son artefactos generados en la investigación precedente de análisis y diseño del módulo Biología Molecular del LIMS de Calidad del CIGB.³³

A continuación se presentan las descripciones textuales resumidas de los casos de uso y la descripción textual extendida de los casos de usos "Gestionar Solicitud de destrucción de bancos de

³³ Pedraza Eslavy, Puig Pedro Manuel. 2008. "Sistema para la Gestión de la Información de los Laboratorios de la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis y Diseño del módulo Biología Molecular"

células” y “Gestionar Registro de destrucción de Bancos de Células”, a los que se les dará seguimiento en el flujo de trabajo de implementación.

Nombre del Caso de Uso	Gestionar Solicitud de destrucción de bancos de células (SIC-0840)	
Actores	Analista (Inicia)	
Propósito	Registrar los datos de Solicitud de destrucción de bancos de células (SIC-0840), buscar y visualizar sus datos, modificar dichos datos e imprimir los mismos.	
Resumen	<p>El caso de uso se inicia cuando el analista va a realizar alguna de las siguientes operaciones relacionadas con la gestión de la Solicitud de destrucción de bancos de células:</p> <p style="padding-left: 40px;">Crear nueva Solicitud de destrucción de bancos de células.</p> <p style="padding-left: 40px;">Modificar la Solicitud de destrucción de bancos de células.</p> <p style="padding-left: 40px;">Buscar y visualizar la Solicitud de destrucción de bancos de células.</p> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p> <p>El caso de uso se inicia cuando el Analista va a realizar alguna de las siguientes operaciones concernientes a la gestión.</p>	
Referencias	R1.1, R1.2, R1.2.1, R1.3, R1.4	
Precondiciones	Que el Analista se haya autenticado en el sistema.	
Poscondiciones	El sistema crea una nueva Solicitud de destrucción de bancos de células (SIC-0840), modifica, imprime, busca y visualiza datos según los criterios indicados por el analista.	
Requerimientos especiales	-	
Flujo normal de los eventos		
Acción del actor	Respuesta del Sistema	
1. El Analista, quiere realizar una de las siguientes operaciones: Crear una nueva Solicitud de	2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente: Si el Analista realiza (a), crea una nueva	

<p>destrucción de bancos de células (SIC-0840). Buscar y visualizar la Solicitud de destrucción de bancos de células (SIC-0840). Modificar datos del SIC-0840.</p>	<p>Solicitud de destrucción de bancos de células (SIC-0840) ir a la Sección "Crear nuevo registro" Si el Analista realiza (b), busca y visualiza la Solicitud de destrucción de bancos de células (SIC-0840) ir a la Sección "Buscar y visualizar registro". Si el Analista realiza (c), modificar datos de la Solicitud de destrucción de bancos de células (SIC-0840) ir a la Sección "Modificar datos de un registro".</p>
<p>Sección "Crear nuevo registro"</p>	
<p>Acción del actor</p>	<p>Respuesta del Sistema</p>
	<p>1. El sistema muestra la interfaz correspondiente a la creación de la nueva registro de Solicitud de destrucción de bancos de células (SIC-0840).</p>
	<p>2. El sistema genera el campo Folio consecutivamente.</p>
	<p>3. El sistema genera automáticamente los campos: Laboratorio (con las siglas: BM). Área (con la palabra: Calidad)</p>
<p>4. El Analista edita los siguientes campos: Código del banco a destruir Motivo de destrucción Número de viales destruir Fecha destrucción Hora de destrucción Cargo autoriza Cargo responsable Fecha responsable destrucción Responsable destrucción</p>	
	<p>5. El sistema genera automáticamente el campo: Autoriza destrucción Con el nombre y los apellidos de la persona</p>

	autenticada.
	6. El sistema actualiza el campo y fecha autoriza destrucción con la fecha actual.
	7. El sistema verifica que todos los datos introducidos sean válidos.
	8. El sistema crea el registro.
	9. El sistema muestra el registro creado y brinda la posibilidad de imprimirlo y/o modificarlo.
10. El analista indica cuál de las siguientes operaciones realizar: Imprimir registro. Modificar registro.	11. El sistema en correspondencia a lo que haya escogido el analista realiza lo siguientes: Si decide imprimir, el sistema imprime el registro brindando la posibilidad de hacerlo en formato Word. Si decide modificar, va a la sección "Modificar datos de un Registro"
Flujo alternativo Sección "Crear nuevo registro"	
Acción del actor	Respuesta del Sistema
	7.1. Si alguno de los datos no es válido el sistema muestra el siguiente mensaje de error "Los datos están incorrectos"
10.1. Si el analista no escoge ninguna de las opciones brindadas, sale de la sección.	
Prototipo	
Sección "Modificar datos de un registro"	
Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz correspondiente a la modificación de los datos en el registro.
2. El analista realiza los cambios necesarios y/o actualizaciones pertinentes.	3. El sistema verifica que todos los campos tengan al menos un valor.
	4. El sistema actualiza la información en la BD.
	5. Se actualiza en el registro de trazas.

	6. El sistema muestra el registro modificado y brinda la posibilidad de imprimirlo y/o modificarlo
7. El analista indica la operación que desea realizar: Imprimir registro. Modificar registro.	8. El sistema en correspondencia con la operación que escogió el analista realiza lo siguiente: Si decide imprimir, el sistema imprime el registro dando la posibilidad de hacerlo en formato Word. Si decide modificar, va a la acción 1.
Flujo alternativo Sección "Modificar datos de un registro"	
Acción del actor	Respuesta del Sistema
2.1. Si el analista no desea modificar los datos, sale de la sección.	3.1. El sistema muestra un mensaje de error, para que sean llenados todos los campos.
7.1. Si el analista no desea realizar ninguna de las opciones brindadas, sale de la sección.	
Prototipo	
Sección "Buscar y visualizar registro"	
Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz correspondiente a la búsqueda y visualización del registro (SIC-0840)
2. El analista introduce el o los criterios para realizar la búsqueda: Código del banco a destruir No viales destruir Indica realizar la búsqueda.	3. El sistema verifica que ha sido introducido al menos un criterio para realizar la búsqueda.
	4. El sistema realiza la búsqueda.
	5. El sistema muestra un listado con el resultado de la búsqueda en una tabla que contiene los siguientes campos. Código del banco a destruir No viales destruir Brinda la posibilidad de visualizar los registros

	encontrados.
6. El analista selecciona el documento que desea visualizar.	7. El sistema muestra el registro seleccionado en una nueva interfaz y brinda la posibilidad de imprimirlo y/o modificarlo.
8. El analista selecciona cuáles de las siguientes operaciones realizar: <p style="margin-left: 40px;">Imprimir registro. Modificar registro.</p>	9. El sistema en correspondencia con la operación escogida por el analista realiza lo siguiente: <p style="margin-left: 40px;">Si decide imprimir, el sistema imprime el registro y brinda la posibilidad de hacerlo en formato Word. Si decide modificar, va a la sección "Modificar datos de un Registro"</p>
Flujo alternativo Sección "Buscar visualizar registro"	
Acción del actor	Respuesta del Sistema
2.1. Si el analista no selecciona o inserta los parámetros para realizar la búsqueda, sale de la sección.	3.1. Si no ha sido introducido al menos un parámetro para realizar la búsqueda, el sistema muestra el siguiente mensaje de error "Debe introducir al menos un parámetro para realizar la búsqueda"
6.1. Si el analista no desea visualizar ninguno de los registros mostrados, sale de la sección.	
7.1. Si el analista no desea realizar ninguna de las opciones brindadas, sale de la sección.	
Prototipo	

Tabla 1. Descripción Textual Extendida del CU Gestionar Solicitud de destrucción de bancos de células.

Nombre del Caso de Uso	Gestionar registro de Destrucción de Banco de Células (SIC-0841)
Actores	Analista
Propósito	Registrar los datos del registro de Destrucción de Banco de Células (SIC-0841), buscar y visualizar sus datos, modificar dichos datos e imprimir los

	mismos.	
Resumen	<p>El caso de uso se inicia cuando el analista va a realizar alguna de las siguientes operaciones relacionadas con la gestión del registro de Destrucción de Banco de Células:</p> <p style="padding-left: 40px;">Crear nuevo registro de Destrucción de Banco de Células.</p> <p style="padding-left: 40px;">Modificar registro de Destrucción de Banco de Células</p> <p style="padding-left: 40px;">Buscar y visualizar el registro de Destrucción de Banco de Células</p> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p> <p>El caso de uso se inicia cuando el Analista va a realizar alguna de las siguientes operaciones concernientes a la gestión.</p>	
Referencias	R2.1, R2.2, R2.2.1, R2.3, R2.4	
Precondiciones	Que el Analista se haya autenticado en el sistema.	
Poscondiciones	El sistema crea un nuevo registro Destrucción de Banco de Células (SIC-0841), modifica, imprime, busca y visualiza datos según los criterios indicados por el analista.	
Requerimientos especiales	-	
Flujo normal de los eventos		
Acción del actor	Respuesta del Sistema	
<p>1. El Analista, quiere realizar una de las siguientes operaciones:</p> <p style="padding-left: 40px;">Crear un nuevo registro de Destrucción de Banco de Células (SIC-0841).</p> <p style="padding-left: 40px;">Buscar y visualizar el registro de Destrucción de Banco de Células (SIC-0841) en el sistema.</p> <p style="padding-left: 40px;">Modificar datos del SIC-0841.</p>	<p>2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente:</p> <p style="padding-left: 40px;">Si el Analista realiza (a), crea un nuevo registro de Destrucción de Banco de Células (SIC-0841) ir a la Sección "Crear nuevo registro"</p> <p style="padding-left: 40px;">Si el Analista realiza (b), busca y visualiza el registro Destrucción de Banco de Células (SIC-0841) ir a la Sección "Buscar y visualizar registro".</p> <p style="padding-left: 40px;">Si el Analista realiza (c), modificar datos del registro de Destrucción de Banco de Células (SIC-0841) ir a la Sección "Modificar datos de un registro".</p>	

Sección "Crear nuevo registro"	
Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz correspondiente a la creación del nuevo registro de Destrucción de Banco de Células (SIC-0841).
	2. El sistema genera el campo Folio consecutivamente.
	3. El sistema genera automáticamente los campos: Laboratorio (con las siglas: BM) Área (con la palabra: Calidad) No (consecutivamente) Fecha (con la fecha actual)
4. El Analista selecciona el tipo de banco, que puede ser: BCT BCP	
5. El analista llena los campos: Código Número de viales	
6. El analista selecciona el motivo de la destrucción que puede ser: RFE NV	
	7. El sistema verifica que todos los datos introducidos sean válidos.
	8. El sistema crea el registro.
	9. El sistema muestra el registro creado y brinda la posibilidad de imprimirlo y/o modificarlo.

<p>10. El analista indica cuál de las siguientes operaciones realizar: Imprimir registro. Modificar registro.</p>	<p>11. El sistema en correspondencia a lo que haya escogido el analista realiza lo siguientes: Si decide imprimir, el sistema imprime el registro brindando la posibilidad de hacerlo en formato Word. Si decide modificar, va a la sección "Modificar datos de un Registro"</p>
<p>Flujo alternativo Sección "Crear nuevo registro"</p>	
<p>Acción del actor</p>	<p>Respuesta del Sistema</p>
	<p>7.1. Si alguno de los datos no es válido el sistema muestra el siguiente mensaje de error "Los datos están incorrectos"</p>
<p>10.1. Si el analista no escoge ninguna de las opciones brindadas, sale de la sección.</p>	
<p>Sección "Modificar datos de un registro"</p>	
<p>Acción del actor</p>	<p>Respuesta del Sistema</p>
	<p>1. El sistema muestra la interfaz correspondiente a la modificación de los datos en el registro.</p>
<p>2. El analista realiza los cambios necesarios y/o actualizaciones pertinentes.</p>	<p>3. El sistema verifica que todos los campos tengan al menos un valor.</p>
	<p>4. El sistema actualiza la información en la BD.</p>
	<p>5. Se actualiza en el registro de trazas.</p>
	<p>6. El sistema muestra el registro modificado y brinda la posibilidad de imprimirlo y/o modificarlo</p>
<p>7. El analista indica la operación que desea realizar: Imprimir registro. Modificar registro.</p>	<p>8. El sistema en correspondencia con la operación que escogió el analista realiza lo siguiente: Si decide imprimir, el sistema imprime el registro dando la posibilidad de hacerlo en formato Word.</p>

	Si decide modificar, va a la acción 1.
Flujo alternativo Sección "Modificar datos de un registro"	
Acción del actor	Respuesta del Sistema
2.1. Si el analista no desea modificar los datos, sale de la sección.	3.1. El sistema muestra un mensaje de error, para que sean llenados todos los campos.
7.1. Si el analista no desea realizar ninguna de las opciones brindadas, sale de la sección.	
Sección "Buscar y visualizar registro"	
Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz correspondiente a la búsqueda y visualización del registro (SIC-0841)
2. El analista introduce el o los criterios para realizar la búsqueda: Código Número de viales Indica realizar la búsqueda.	3. El sistema verifica que ha sido introducido al menos un criterio para realizar la búsqueda.
	4. El sistema realiza la búsqueda.
	5. El sistema muestra un listado con el resultado de la búsqueda en una tabla que contiene los siguientes campos. Código Número de viales Brinda la posibilidad de visualizar los registros encontrados.
6. El analista selecciona el documento que desea visualizar.	7. El sistema muestra el registro seleccionado en una nueva interfaz y brinda la posibilidad de imprimirlo y/o modificarlo.
8. El analista selecciona cuáles de las siguientes operaciones realizar: Imprimir registro. Modificar registro.	9. El sistema en correspondencia con la operación escogida por el analista realiza lo siguiente: Si decide imprimir, el sistema imprime el registro y brinda la posibilidad de hacerlo en formato

	Word. Si decide modificar, va a la sección "Modificar datos de un Registro"
Prototipo	
Flujo alternativo Sección "Buscar visualizar registro"	
Acción del actor	Respuesta del Sistema
2.1. Si el analista no selecciona o inserta los parámetros para realizar la búsqueda, sale de la sección.	3.1. Si no ha sido introducido al menos un parámetro para realizar la búsqueda, el sistema muestra el siguiente mensaje de error "Debe introducir al menos un parámetro para realizar la búsqueda"
6.1. Si el analista no desea visualizar ninguno de los registros mostrados, sale de la sección.	
7.1. Si el analista no desea realizar ninguna de las opciones brindadas, sale de la sección.	
Prototipo	

Tabla 2. Descripción Textual Extendida del CU Gestionar registro de Destrucción de Bancos de Células.

Nombre del Caso de Uso	Gestionar registro de Temperatura (SIC-0125R)
Actores	Analista
Propósito	Registrar los datos del registro de Temperatura (SIC-0125R), buscar y visualizar sus datos, modificar dichos datos e imprimir los mismos.
Resumen	<p>El caso de uso se inicia cuando el analista va a realizar alguna de las siguientes operaciones relacionadas con la gestión del registro de Temperatura:</p> <ul style="list-style-type: none"> Crear nuevo registro de Temperatura Modificar el registro de Temperatura Buscar y visualizar el registro de Temperatura <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta</p>

	<p>las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p> <p>El caso de uso se inicia cuando el Analista va a realizar alguna de las siguientes operaciones concernientes a la gestión.</p>
Referencias	R3.1, R3.2, R3.2.1, R3.3, R3.4
Precondiciones	Que el Analista se haya autenticado en el sistema.
Poscondiciones	El sistema crea un nuevo registro de Temperatura (SIC-0125R), modifica, imprime, busca y visualiza datos según los criterios indicados por el analista
Requerimientos especiales	-

Tabla 3. Descripción Textual Resumida del CU Gestionar registro de Temperatura.

Nombre del Caso de Uso	Gestionar Cronograma de chequeo de bancos de células (SIC-0130)
Actores	Analista
Propósito	Registrar los datos del Cronograma de chequeo de bancos de células (SIC-0130), buscar y visualizar sus datos, modificar dichos datos e imprimir los mismos.
Resumen	<p>El caso de uso se inicia cuando el analista va a realizar alguna de las siguientes operaciones relacionadas con la gestión del Cronograma de chequeo de bancos de células:</p> <ul style="list-style-type: none"> Crear nuevo Cronograma de chequeo de bancos de células. Modificar Cronograma de chequeo de bancos de células. Buscar y visualizar Cronograma de chequeo de bancos de células. <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p> <p>El caso de uso se inicia cuando el Analista va a realizar alguna de las siguientes operaciones concernientes a la gestión.</p>

Referencias	R4.1, R4.2, R4.2.1, R4.3, R4.4
Precondiciones	Que el Analista se haya autenticado en el sistema.
Poscondiciones	El sistema crea un nuevo Cronograma de chequeo de bancos de células (SIC-0130), inserta, modifica, imprime, busca y visualiza datos según el interés que tenga el analista.
Requerimientos especiales	-

Tabla 4. Descripción Textual Resumida del CU Gestionar Cronograma de chequeo de bancos de células.

Nombre del Caso de Uso	Gestionar Certificado de Liberación de los bancos de Células transformados (SIC-0229)
Actores	Analista
Propósito	Registrar los datos del Certificado de Liberación de los bancos de Células transformados (SIC-0229), buscar y visualizar sus datos, modificar dichos datos e imprimir los mismos.
Resumen	<p>El caso de uso se inicia cuando el analista va a realizar alguna de las siguientes operaciones relacionadas con la gestión del Certificado de Liberación de los bancos de Células transformados</p> <ul style="list-style-type: none"> Crear nuevo Certificado de Liberación de los bancos de Células transformados Modificar Certificado de Liberación de los bancos de Células transformado Buscar y visualizar Certificado de Liberación de los bancos de Células transformados <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p> <p>El caso de uso se inicia cuando el Analista va a realizar alguna de las siguientes operaciones concernientes a la gestión.</p>
Referencias	R5.1, R5.2, R5.2.1, R5.3, R5.4

Precondiciones	Que el Analista se haya autenticado en el sistema.
Poscondiciones	El sistema crea un nuevo Certificado de Liberación de los bancos de Células transformados (SIC-0229), modifica, imprime, busca y visualiza datos según el interés que tenga el analista.
Requerimientos especiales	-

Tabla 5. Descripción Textual Resumida del CU Gestionar Certificado de Liberación de bancos de células transformados.

Nombre del Caso de Uso	Gestionar registro de Descargo para Banco de Células (SIC-0190)
Actores	Analista
Propósito	Registrar los datos del registro de Descargo para Banco de Células (SIC-0190), buscar y visualizar sus datos, modificar dichos datos e imprimir los mismos.
Resumen	<p>El caso de uso se inicia cuando el analista va a realizar alguna de las siguientes operaciones relacionadas con la gestión del registro de Descargo para Banco de Células:</p> <ul style="list-style-type: none"> Crear nuevo registro de Descargo para Banco de Células Modificar el registro de Descargo para Banco de Células Buscar y visualizar el registro de Descargo para Banco de Células <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p> <p>El caso de uso se inicia cuando el Analista va a realizar alguna de las siguientes operaciones concernientes a la gestión.</p>
Referencias	R6.1, R6.2, R6.2.1, R6.3, R6.4
Precondiciones	Que el Analista se haya autenticado en el sistema.
Poscondiciones	El sistema crea un nuevo registro de Descargo para Banco de Células (SIC-0190), modifica, imprime, busca y visualiza datos según los criterios indicados por el analista.
Requerimientos especiales	-

Tabla 6. Descripción Textual Resumida del CU Gestionar Registro de Descargo para Bancos de Células.

Nombre del Caso de Uso	Gestionar Informe de Análisis de los bancos de células (SIC-0950)
Actores	Analista
Propósito	Registrar los datos del Informe de Análisis de los bancos de células (SIC-0950), buscar y visualizar sus datos, modificar dichos datos e imprimir los mismos.
Resumen	<p>El caso de uso se inicia cuando el analista va a realizar alguna de las siguientes operaciones relacionadas con la gestión del Informe de Análisis de los bancos de células:</p> <ul style="list-style-type: none"> Crear nuevo Informe de Análisis de los bancos de células. Modificar Informe de Análisis de los bancos de células. Buscar y visualizar Informe de Análisis de los bancos de células. <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se ejecuta alguna de las acciones solicitadas.</p> <p>El caso de uso se inicia cuando el Analista va a realizar alguna de las siguientes operaciones concernientes a la gestión.</p>
Referencias	R7.1, R7.2, R7.2.1, R7.3, R7.4
Precondiciones	Que el Analista se haya autenticado en el sistema.
Poscondiciones	El sistema crea un nuevo Informe de Análisis de los bancos de células (SIC-0950), modifica, imprime, busca y visualiza datos según el interés que tenga el analista.
Requerimientos especiales	-

Tabla 7. Descripción Textual Resumida del CU Gestionar Informe de Análisis de los bancos de células.

Nombre del Caso de Uso	Gestionar SIC-0028
Actores	Jefe de Grupo
Propósito	Registrar los datos en <i>Registro de Desviaciones de Ensayo Analítico</i> ,

	buscar SIC-0028 y visualizar sus datos, modificar dichos datos e imprimirlo tanto vacío como lleno.
Resumen	<p>El caso de uso se inicia cuando el Analista ha culminado de llenar algún SIC-0027 (Reporte Desviación de Ensayo Analítico) y entonces el Jefe de Grupo debe actualizar el Registro de Desviaciones de Ensayo Analítico, aquí realiza alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"> Crear un nuevo registro en un SIC-0028 Buscar y visualizar un SIC-0028 en el sistema Imprimir registro (SIC-0028) <p>El sistema le muestra la interfaz correspondiente y el Jefe de Grupo ejecuta las acciones necesarias. El caso de uso finaliza cuando el Jefe de Grupo termina el registro de la desviación referida de algún SIC-0027.</p>
Referencias	
Precondiciones	<p>Que el Jefe de Grupo se haya autenticado en la aplicación.</p> <p>Que se haya registrado alguna desviación en el SIC-0027.</p>
Poscondiciones	El sistema registra una nueva desviación referida de algún SIC-0027 (Reporte Desviación de Ensayo Analítico)
Requerimientos especiales	-

Tabla 8. Descripción Textual Resumida del CU Gestionar Registro de Desviaciones de Ensayo

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

	acciones solicitadas.
Referencias	32.1, 32.2, 32.3, 32.3.1, 32.4.
Precondiciones	- Que el Analista se haya autenticado en la aplicación.
Poscondiciones	Libro creado, datos insertados en el libro, libro buscado, libro modificado.
Requerimientos especiales	-

Tabla 9. Descripción Textual Resumida del CU Gestionar Libro de entrada de muestras de producción.

Nombre del Caso de Uso	Gestionar Registro de preparación de soluciones (SIC-0020).
Actores	Analista (Inicia).
Propósito	Crear un nuevo SIC-0020, modificar el SIC-0020; buscar y visualizar el SIC-0020; imprimir SIC-0020.
Resumen	<p>El caso de uso se inicia cuando el Analista va a realizar alguna de las siguientes acciones relacionadas con el Registro de Preparación de Soluciones :</p> <ul style="list-style-type: none"> - Crear nuevo SIC-0020. - Insertar datos en el SIC-0020. - Buscar y visualizar SIC-0020. <p>El sistema muestra la interfaz correspondiente según la solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.</p>
Referencias	1.1, 1.2, 1.3, 1.3.1, 1.4.
Precondiciones	-Que el Analista se haya autenticado en la aplicación.
Poscondiciones	SIC-0020 creado, SIC-0020 buscado y visualizado, SIC-0020 modificado, SIC-0020 impreso.
Requerimientos especiales	

Tabla 10. Descripción Textual Resumida del CU Gestionar Registro de preparación de soluciones (SIC-0020).

2.3 Diagramas de Componentes

El uso más importante de los diagramas de componentes es mostrar la estructura de alto nivel del modelo de implementación, especificando los subsistemas de implementación y sus dependencias a la hora de importar código y organizar los subsistemas de implementación en capas.

Algunos estereotipos estándares de componentes son los siguientes:

- Ejecutable: Es un programa que se puede ejecutar en un nodo.
- Tabla: Es una tabla de una base de datos.
- Archivo: Es un fichero que contiene código fuente o datos.
- Biblioteca: Es una biblioteca de objetos estática o dinámica.
- Documento: Es un documento.
- Página Web: Es una página que se obtiene de la ejecución del sistema.

A continuación se muestran los diagramas de componentes que se generan para este flujo de trabajo, a partir de los diagramas de clases del diseño. Los diagramas de clases del diseño fueron generados en la investigación precedente de análisis y diseño. [\[Ver anexos\]](#)

Cada diagrama está compuesto por un paquete (View), en el cual se encuentran todas las páginas Success.php con las cuáles interactúa el usuario, el Layout.php es la parte global en toda la aplicación, es decir, todo el código HTML común para todas las páginas. También dentro de la vista se encuentra el paquete JavaScript el cual tiene la funcionalidad de contener todo el código javascript utilizado en la aplicación. Otro componente utilizado es el index SMC.php que es el controlador, este se relaciona con las páginas Success.php y la actions.class.php.

Es en el actions.class.php donde se encuentra toda la implementación de la capa control, la actions.class.php será controlada por la sfFrontWebController.class.php o controlador frontal, pues a través de este se van a ejecutar todas las peticiones de los usuarios. La action.class.php está relacionada con todos los archivos Success.php de la vista y contiene todos los métodos y operaciones a realizar que serán mostrados en la vista, también tiene relación con los archivos Security.yml y Validate.yml, pues es la encargada de la seguridad y las validaciones de la aplicación. De igual forma trabaja sobre el modelo que es la capa que contiene la persistencia de los datos. En el componente

Validate.yml están las validaciones de las acciones o sea hay un archivo yml por cada acción que se quiera validar con el nombre validateNombreAccion ().

Además se tiene otro componente el subsistema Model, que en él se encuentran todas las clases de la capa modelo, es decir todas las clases, clases peer, base y base peer.

Otro subsistema representado es el componente Subsistema Propel de Symfony, a través de este se realiza la conexión a la base de datos. Por último se encuentra el componente Base de Datos, que no es más que la base de datos del módulo. Todas las relaciones son de dependencia, demostrando esto la estrecha relación existente entre cada uno de los componentes representados.

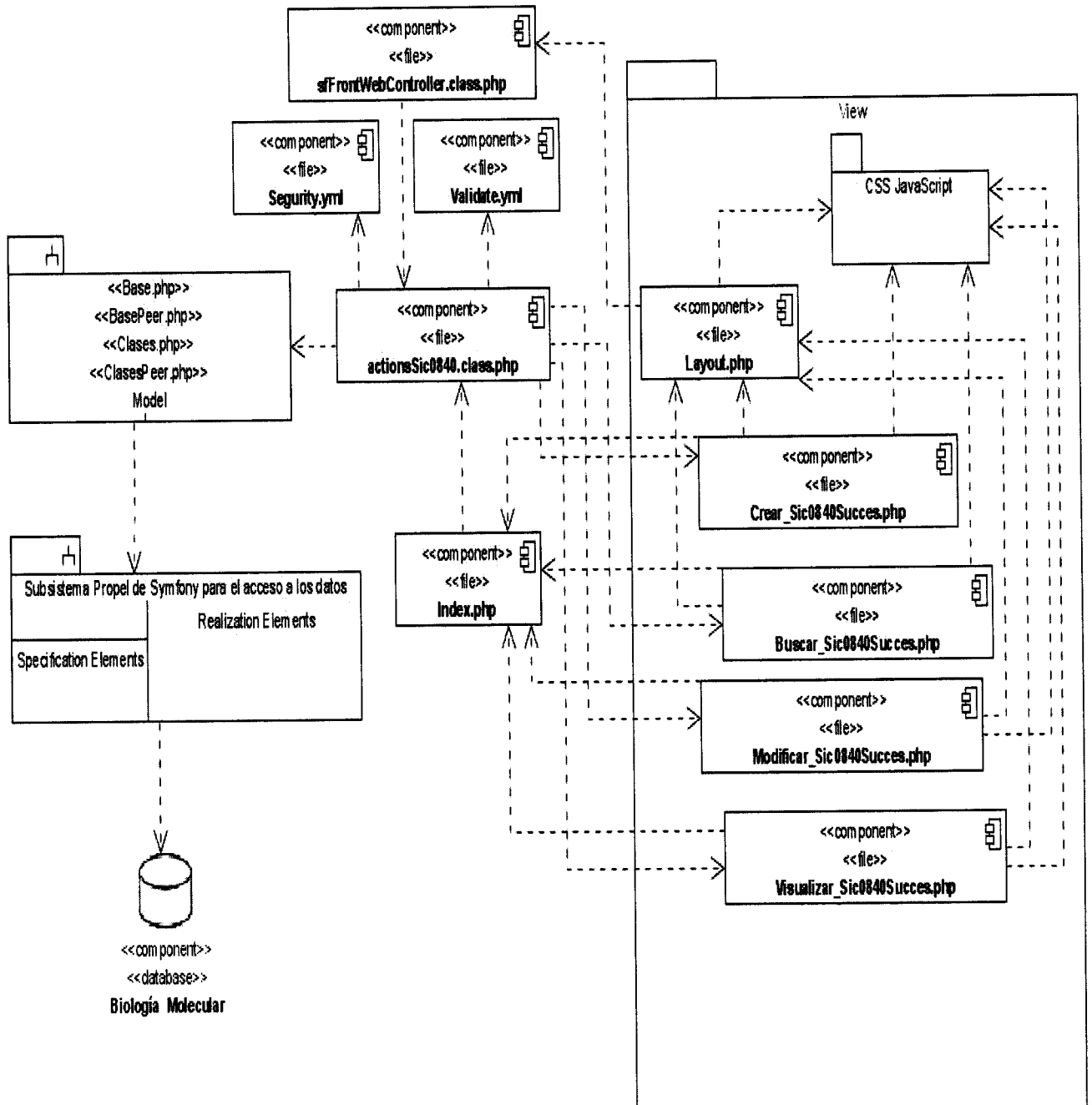


Figura 4. Diagrama de Componentes Gestionar Solicitud de destrucción de bancos de células. (Sic0840).

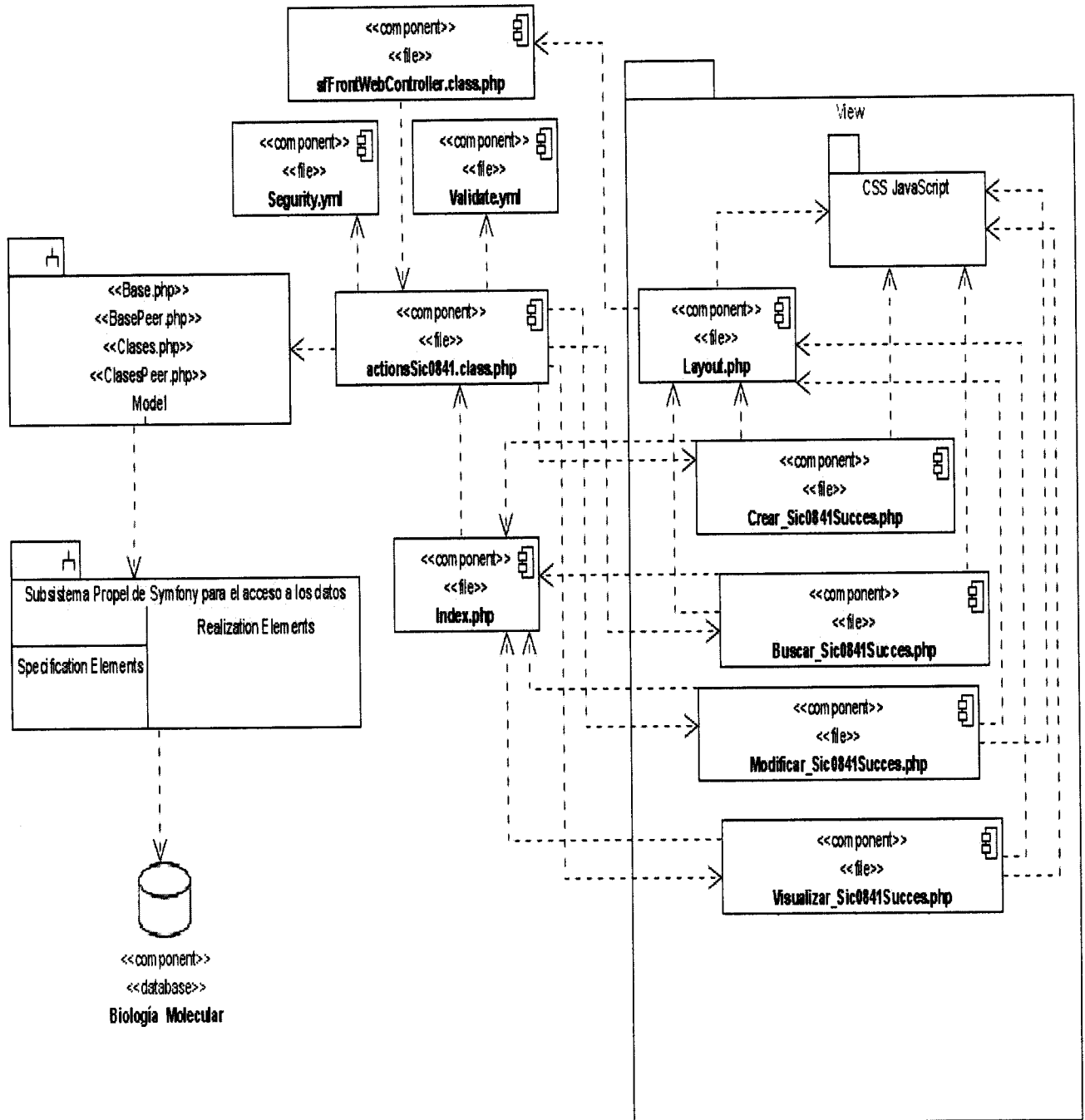


Figura 5. Diagrama de Componentes Gestionar Registro de destrucción de bancos de células (Sic0841).

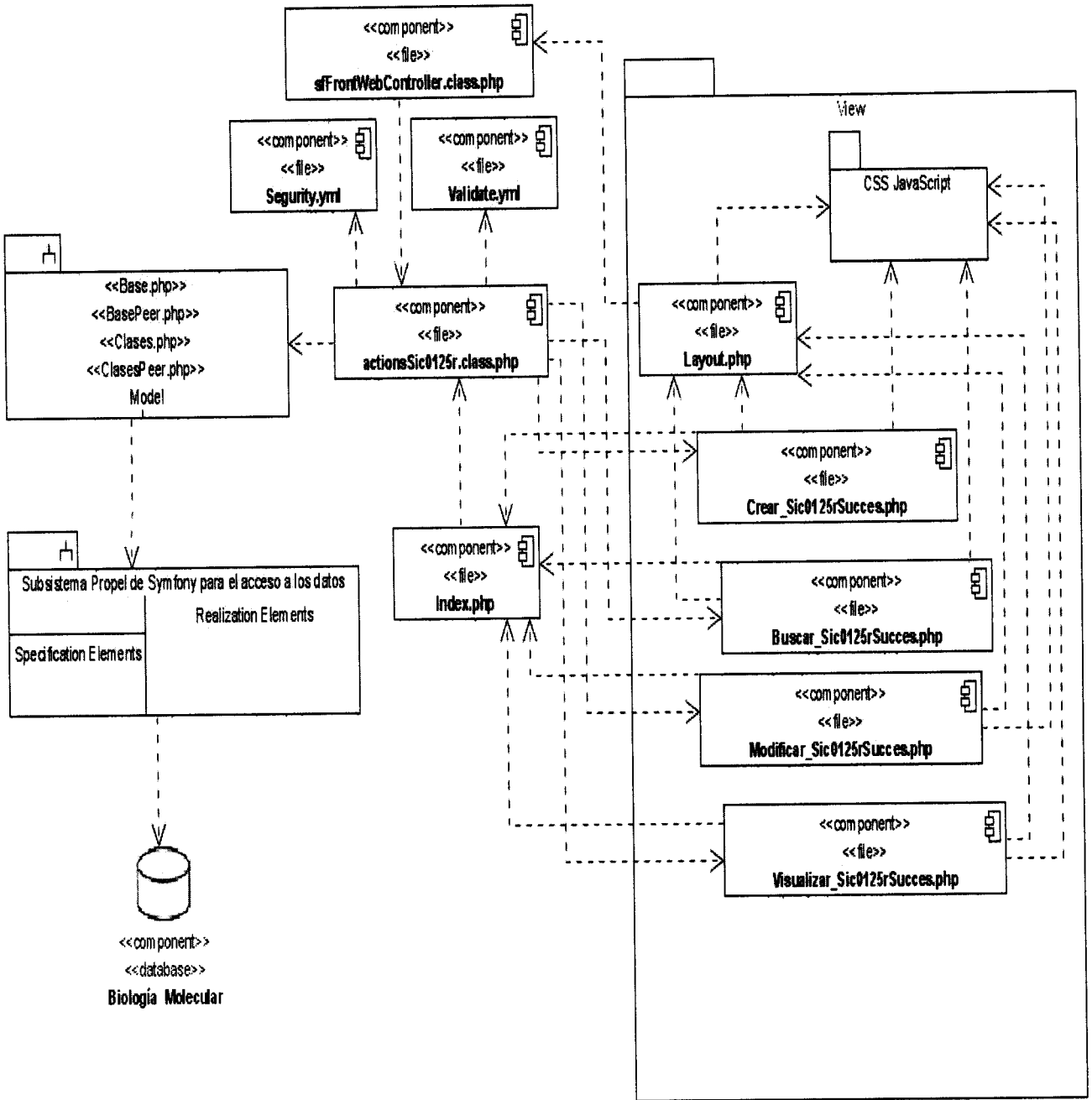


Figura 6. Diagrama de Componentes Gestionar registro de Temperatura (SIC-0125R).

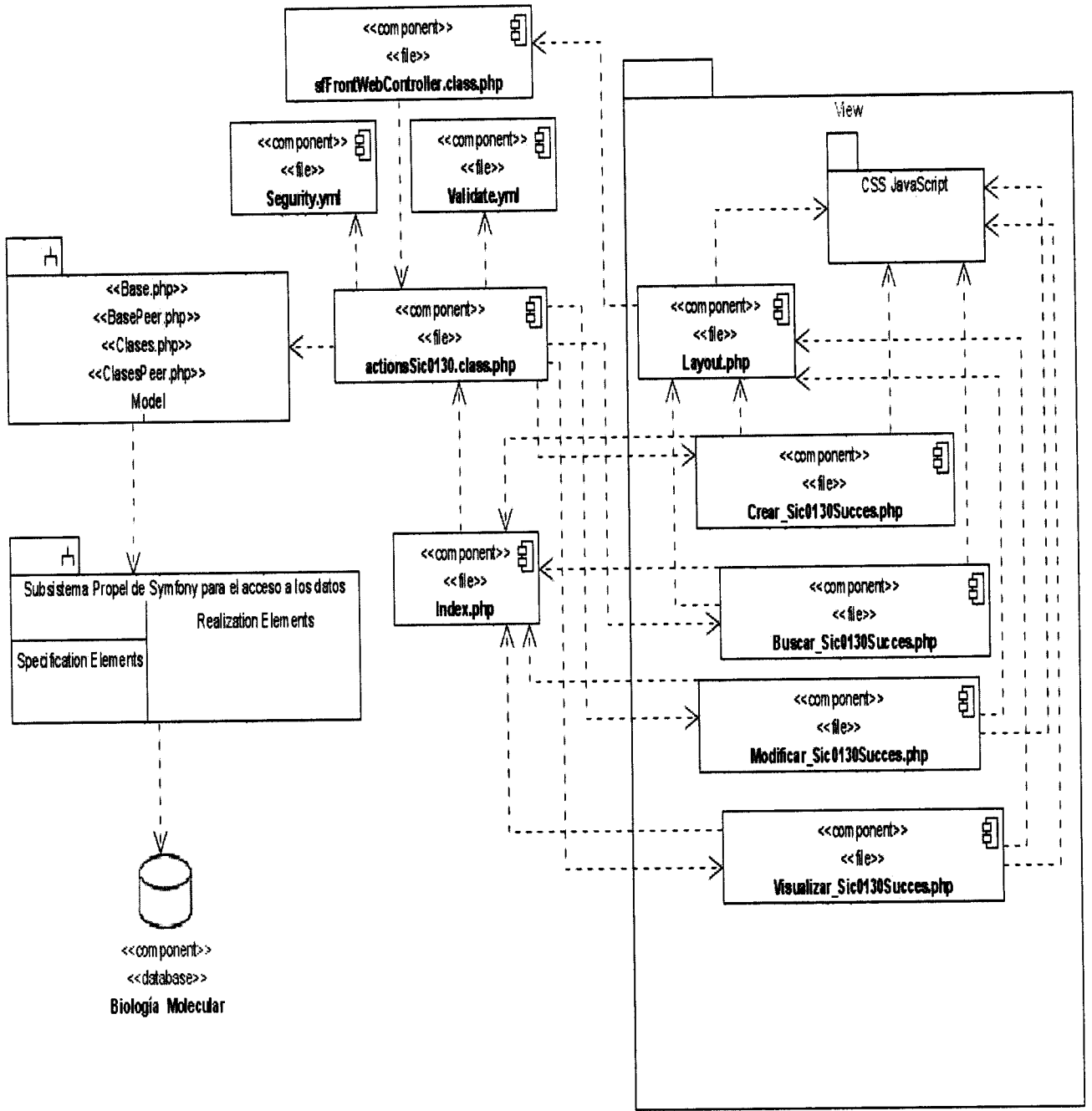


Figura 7. Diagrama de Componentes Gestionar Cronograma de chequeo de bancos de células (Sic0130).

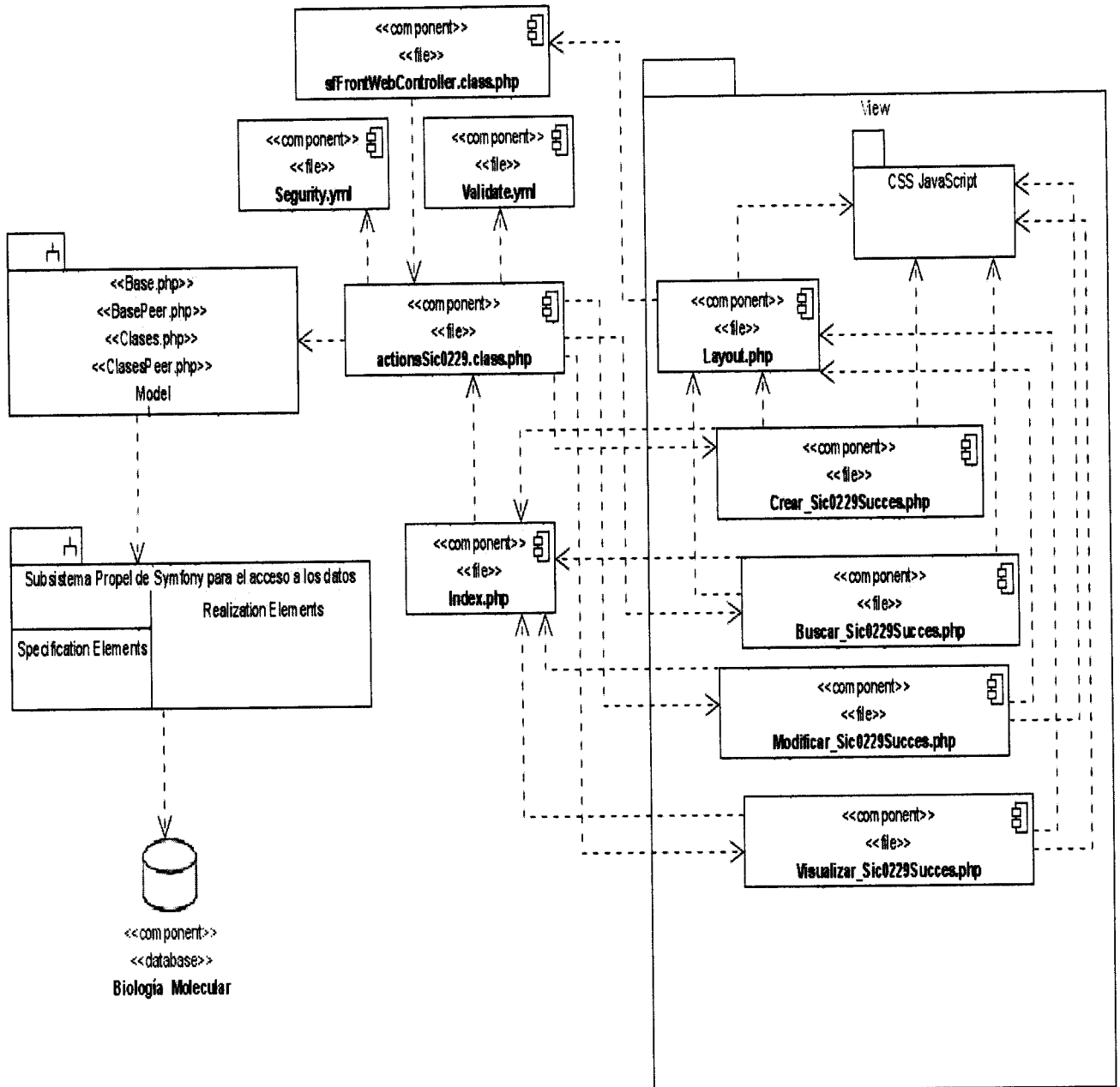


Figura 8. Diagrama de Componentes Gestionar Certificado de Liberación de los bancos de Células transformados (Sic0229).

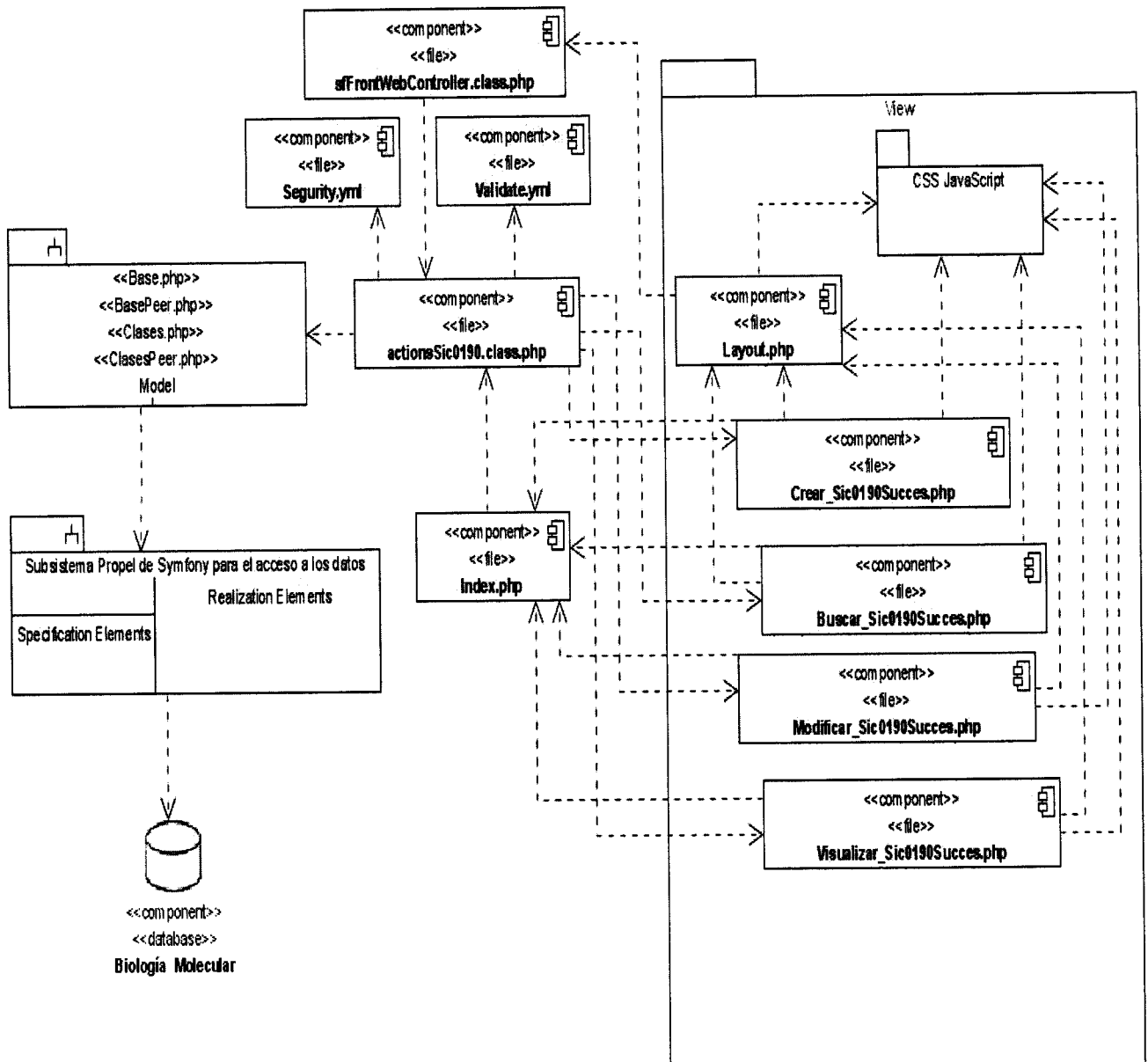


Figura 9. Diagrama de Componentes Gestionar Certificado de Liberación de los bancos de Células transformados (Sic0190).

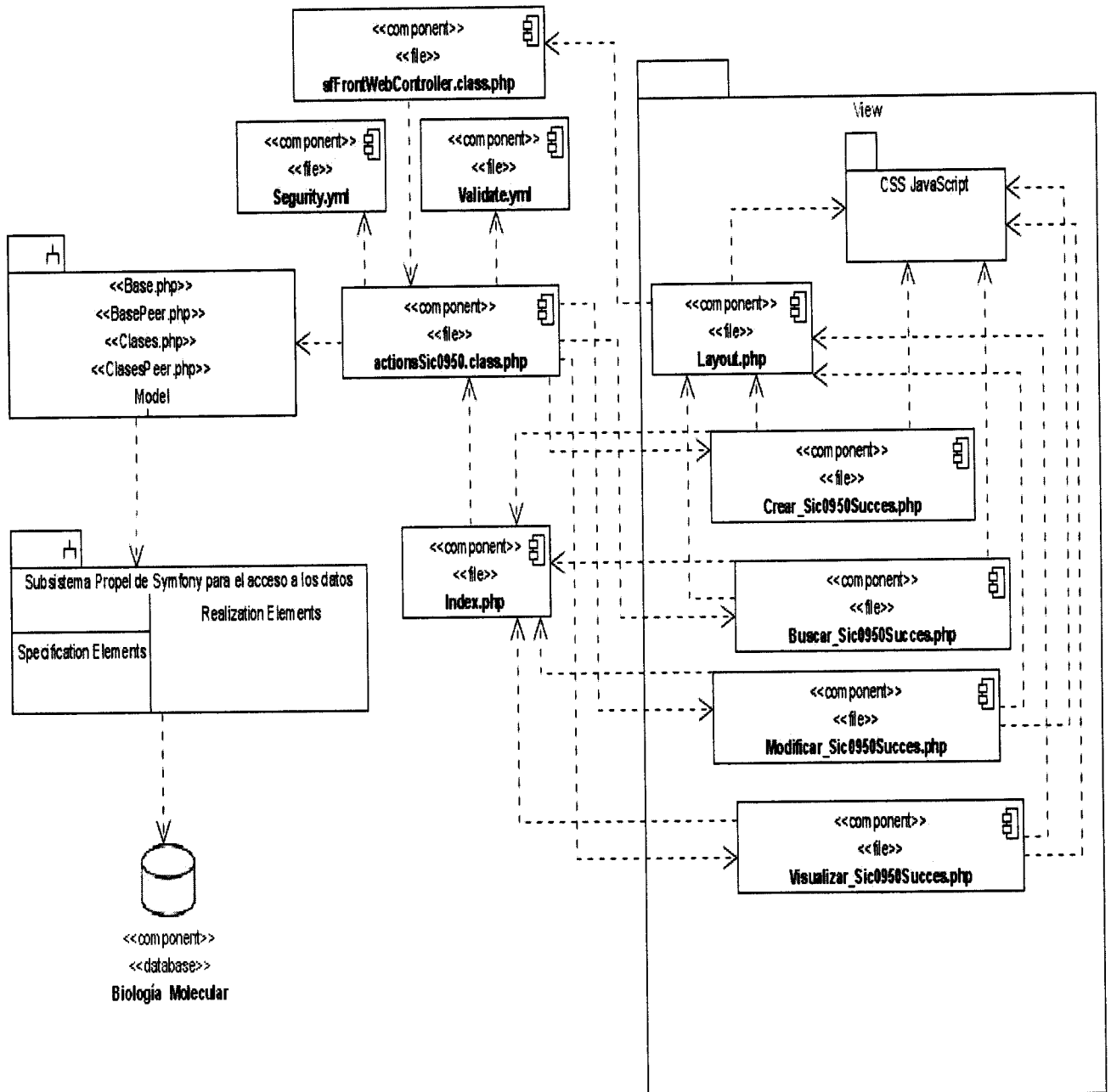


Figura 10. Diagrama de Componentes Gestionar Informe de Análisis de los bancos de células (Sic0950).

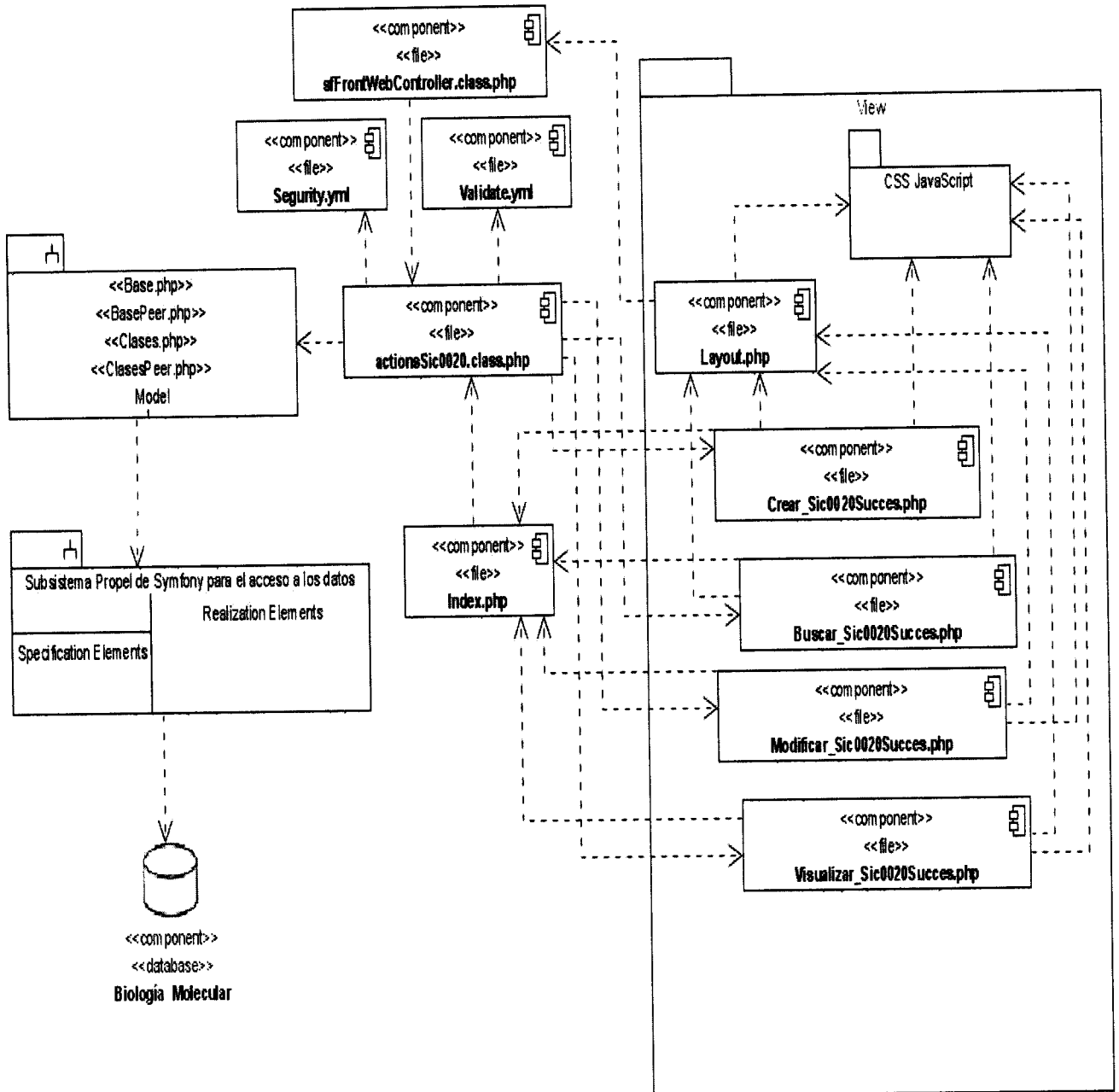


Figura 11. Diagrama de Componentes Gestionar Registro de preparación de soluciones (SIC-0020).

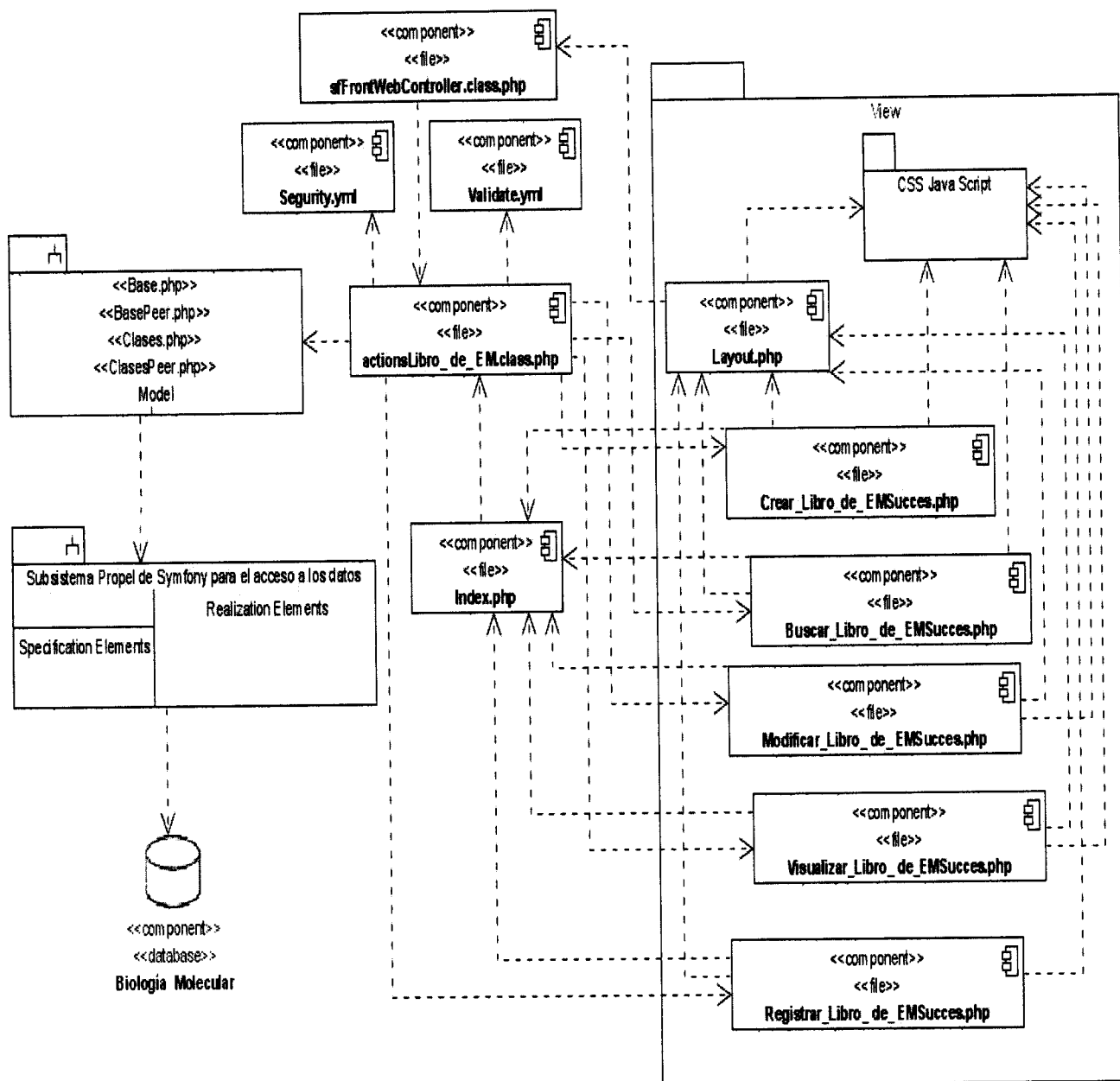


Figura 12. Diagrama de Componentes Gestionar Libro de Entrada de Muestras.

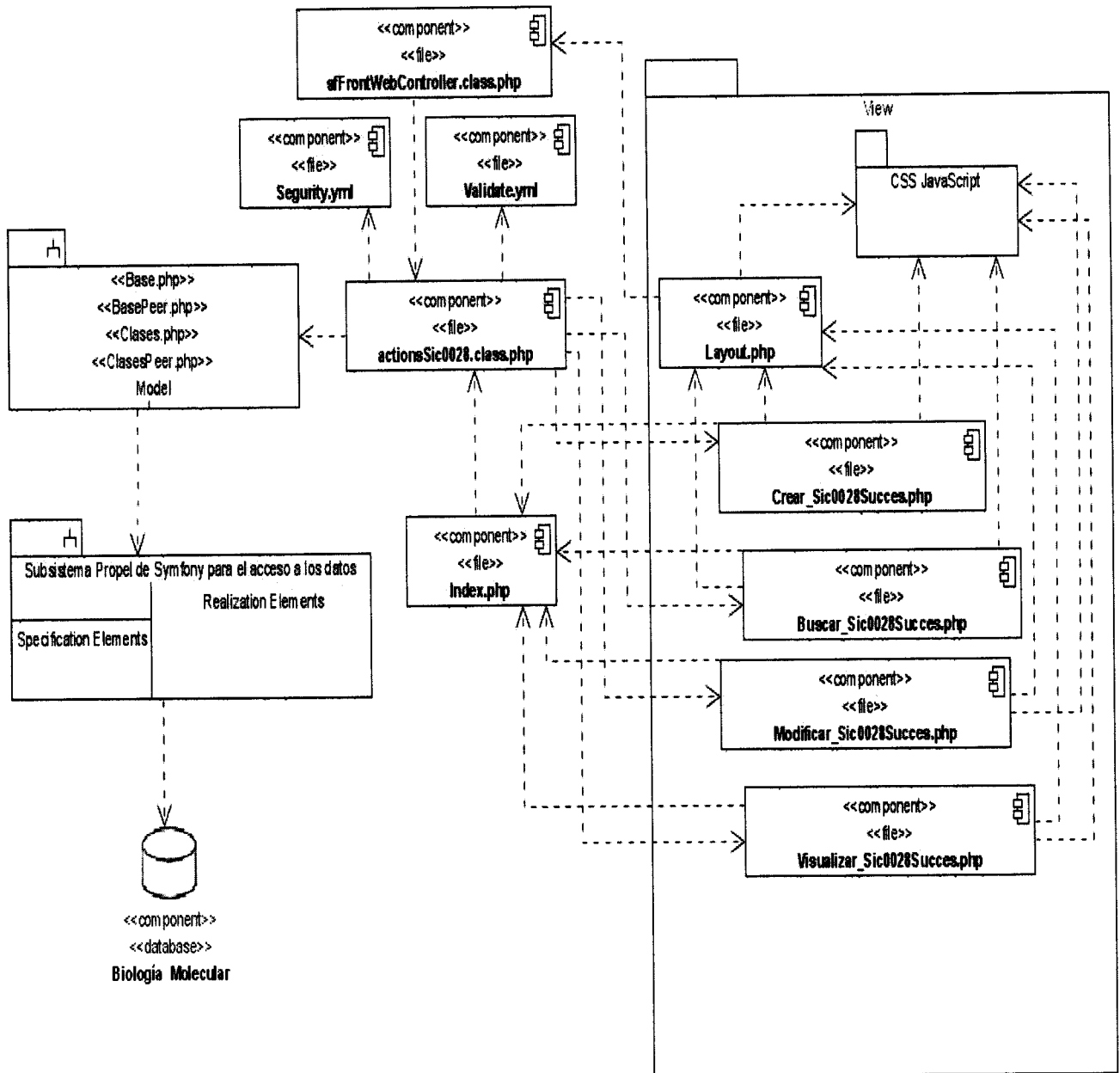


Figura 13. Diagrama de Componentes Gestionar Registro de Desviaciones de Ensayo Analítico (SIC-0028).

2.4 Descripción de los Métodos

A continuación se describen los principales métodos del caso de uso Gestionar Solicitud de destrucción de bancos de células (SIC-0840), debido a la gran similitud que presenta con el resto de los casos de uso.

Nombre	executeCrear()
Caso de uso	Gestionar Solicitud de destrucción de bancos de células (SIC-0840)
Parámetros de Entrada	
Nombre	Tipo
Código del banco a destruir	String
Motivo de destrucción	String
Número de viales destruir	Int
Fecha destrucción	Date
Hora de destrucción	Date
Cargo autoriza	String
Cargo responsable	String
Fecha responsable destrucción	Date
Responsable destrucción	String
Autoriza destrucción	String
Fecha autoriza destrucción	Date
Descripción General	
Su función es crear una nueva solicitud de banco de células. Esta función se ejecuta en el actions.class.php.	

Tabla 11. Descripción del método executeCrear()

Nombre	executeBuscar()
Caso de uso	Gestionar Solicitud de destrucción de bancos de células (SIC-0840)
Parámetros de Entrada	
Nombre	Tipo

Código del banco a destruir	String
Número de viales destruir	Int
Descripción General	
Su función es buscar una solicitud de banco de células. Esta función se ejecuta en el actions.class.php y lanza un mensaje de error cuando no se le pasa ningún parámetro.	

Tabla 12. Descripción del método executeBuscar()

Nombre	executeModificar()
Caso de uso	Gestionar Solicitud de destrucción de bancos de células (SIC-0840)
Parámetros de Entrada	
Nombre	Tipo
Código del banco a destruir	String
Motivo de destrucción	String
Número de viales destruir	Int
Fecha destrucción	Date
Hora de destrucción	Date
Cargo autoriza	String
Cargo responsable	String
Fecha responsable destrucción	Date
Responsable destrucción	String
Autoriza destrucción	String
Fecha autoriza destrucción	Date
Descripción General	
Su función es modificar una solicitud de banco de células previamente creada. Esta función se	

ejecuta en el actions.class.php y asegura que los parámetros para la actualización no sean nulos.

Tabla 13. Descripción del método executeModificar()

2.5 Fragmentos de código

La presentación de estos fragmentos de código, que son significativos en la aplicación, ayudarán a comprender su funcionamiento. La implementación que realiza Symfony de la arquitectura MVC incluye varias clases, aquí se muestran fragmentos de código de dichas clases.

Este fragmento de código es de una de las clases de objetos propias (ej. Sic0840) que están en el directorio lib/model, las clases de este tipo heredan de las clases con nombre Base. Estas clases no se modifican cuando se ejecuta la tarea propel-build-model, por lo que son las clases en las que se añaden los métodos propios. Son clases objeto que representan un registro de la base de datos. Permiten acceder a las columnas de un registro y a los registros relacionados.

```
<?php
@/**
 * Subclass for representing a row from the 'sic_0840' table.
 *
 *
 *
 * @package lib.model
 */
class Sic0840 extends BaseSic0840
(
)
```

Las clases de tipo Peer que también están en el directorio lib/model, en este caso se muestra un fragmento de código del (ej. Sic0840Peer) 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.

```
public static function CrearSic0840($codigo_banco_destruir,$no_viales_destruir,  
$motivo_destruccion,$fecha_destruccion, $hora_destruccion,$responsable_destruccion,$carga_responsable,  
$autoriza_destruccion,$carga_autoriza,$fecha_responsable_destruccion,$fecha_autoriza_destruccion)  
{  
    $sic0840= new Sic0840();  
    $sic0840->setCodigoBancoDestruir($codigo_banco_destruir);  
    $sic0840->setNoVialesDestruir($no_viales_destruir);  
    $sic0840->setMotivoDestruccion($motivo_destruccion);  
    $sic0840->setFechaDestruccion($fecha_destruccion);  
    $sic0840->setHoraDestruccion($hora_destruccion);  
    $sic0840->setResponsableDestruccion($responsable_destruccion);  
    $sic0840->setCargaResponsable($carga_responsable);  
    $sic0840->setAutorizaDestruccion($autoriza_destruccion);  
    $sic0840->setCargaAutoriza($carga_autoriza);  
    $sic0840->setFechaResponsableDestruccion($fecha_responsable_destruccion);  
    $sic0840->setFechaAutorizaDestruccion($fecha_autoriza_destruccion);  
    $sic0840->save();  
    return $sic0840->getFolio();  
}
```

Figura 14. Método CrearSic0840

Las acciones son el corazón de la aplicación, puesto que contienen toda la lógica de la aplicación. Las acciones utilizan el modelo y definen variables para la vista. Las acciones son métodos con el nombre `executeNombreAccion` de una clase llamada `nombreModuloActions` que hereda de la clase `SfActions` y se encuentran agrupadas por módulos. La clase que representa las acciones de un módulo se encuentra en el archivo `actions.class.php`, en el directorio `actions/` del módulo. A continuación se presenta el código fuente del Modificar del Sic0841. También se presenta el código fuente del buscar en el `actions.class.php` del Sic0841. La búsqueda se realiza según los criterios, que en este caso son el código y el número de viales. Si el usuario no entra al menos un parámetro para realizar la búsqueda, el sistema lanza un mensaje de error, alertando al usuario que debe escoger al menos un criterio para la búsqueda. Se muestra la interfaz correspondiente para esta situación con el objetivo de entender el funcionamiento del sistema.

```
public function executeModificar()
{
    try
    {
        $this->id=$this->getRequestParameter('id');
        if ($this->getRequest()->getMethod() == sfRequest::POST )
        {
            $cant = $this->getRequestParameter('cant');
            $c = new Criterias();
            for ($j = 0; $j < $cant; $j++)
            {
                $c->clear();
                $c->add(DestruccionPeer::ID_DESTRUCCION,$this->getRequestParameter('id_destruccion'.$j));
                $obj = DestruccionPeer::doSelectOne($c);
                if ($obj instanceof Destruccion )
                {
                    $obj->setFecha($this->getRequestParameter('fecha'.$j));
                    $obj->setTipoBanco($this->getRequestParameter('getTipo'.$j));
                    $obj->setCodigo($this->getRequestParameter('getCodigo'.$j));
                    $obj->setMotivoDestruccion($this->getRequestParameter('getMotivoDestruccion'.$j));
                    $obj->setNoVialesDestruidos($this->getRequestParameter('getNoVialesDestruidos'.$j));
                    $obj->setInspeccionPor($this->getRequestParameter('getInspeccionPor'.$j));
                    $obj->setDestruidoPor($this->getRequestParameter('getDestruidoPor'.$j));
                    $obj->save();
                }
            }
        }
        $this->redirect('sic0841/buscar');
```

Figura 15. Método executeModificar

```

public function executeBuscar()
{
    $this->viales = DestruccionPeer::doSelect(new Criteria());
    if ($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $this->resultado = array();
    }
    else
    {
        $via = $this->getRequestParameter('viales');
        $cod = $this->getRequestParameter('codigo');

        if (($via == '') && ($cod == ''))
        {
            $this->getRequest()->setError('error','Debe escoger los criterios de busqueda');
            $this->resultado = array();
            return sfView::SUCCESS;
        }

        $c = new Criteria();
        if ($via != '')
        $c->add(DestruccionPeer::NO_VIALES_DESTRUIDOS, $via);
        if ($cod != '')
        $c->add(DestruccionPeer::CODIGO, $cod);

        $this->resultado = DestruccionPeer::doSelect($c);
    }
}
}

```

Figura 16. Método executeBuscar

Sf 1.0.10 vars & config logs & msgs 1 4594.2 KB

CENTRO DE INGENIERIA GENETICA Y BIOTECNOLOGIA SUBDIRECCIÓN DE CALIDAD REGISTRO DE DESTRUCCIÓN DE BANCOS DE CÉLULAS	
Número de Viales a Destruir <<Seleccione>> ▼	Código: <<Seleccione>> ▼

! Debe escoger los criterios de busqueda !

Cancelar
Buscar

Figura 17. Interfaz de Búsqueda de Sic0841.

En las plantillas su contenido está formado por código HTML y algo de código PHP sencillo, normalmente llamadas a las variables definidas en la acción (mediante la instrucción `$this->nombre_variable = 'valor';`) y algunos *helpers*. Los *helpers* son funciones de PHP que devuelven código HTML y que se utilizan en las plantillas. En este caso se muestra el código fuente en el template del `buscarSucces.php` del `Sic0841` que permite en caso de que el usuario escoja un criterio para la búsqueda, que el sistema muestre un listado con el resultado de la búsqueda en una tabla que contiene los siguientes campos:

- Código
- Número de viales

También el sistema brinda la posibilidad de visualizar los registros encontrados.

```
<table align="center" border="1" width="750" bordercolor="#000000" cellpadding="0" cellspacing="0">
  <tr>
    <td width="234"><div align="center" class="style4">Codigo</div></td>
    <td width="282"><div align="center" class="style4">Numero de Viales destruidos </div></td>
    <td width="226"><div align="center" class="style4"></div></td>
  </tr>

  <?php foreach ($resultado as $p): ?>
  <tr>
    <td><div align="center" class="style4"><?php echo $p->getCodigo() ?</div></td>
    <td><div align="center" class="style4"><?php echo $p->getNoVialesDestruidos() ?</div></td>
    <td><div align="center" class="style4"><?php echo link_to('Visualizar', 'Sic0841/visualizar?id='.$p->getSic0841idSic0841()) ?>
  </tr>
  <?php endforeach; ?>
</table>
```

Figura 18. Código Fuente en el template Buscar del Sic0841

Sf 1.0.10 vars & config logs & msgs 2 4598.9 KB

CENTRO DE INGENIERIA GENÉTICA Y BIOTECNOLOGÍA		
SUBDIRECCIÓN DE CALIDAD		
REGISTRO DE DESTRUCCIÓN DE BANCOS DE CÉLULAS		
Número de Viales a Destruir	<<Selecione>> ▼	Código: <<Selecione>> ▼
		<input type="button" value="Cancelar"/> <input type="button" value="Buscar"/>

Codigo	Numero de Viales destruidos	
2	77	Visualizar

Figura 19. Interfaz de Buscar Sic0841.

Código fuente en la actions.class.php referente a la acción Visualizar del Sic0841, en el que primeramente se realiza una consulta a la base de datos para determinar el folio que se quiere visualizar. Seguidamente se realiza una consulta a la tabla de muchos para obtener los objetos asociados al determinado folio obtenido de la primera consulta.

```

public function executeVisualizar()
{
    $this->id=$this->getRequestParameter('id');
    $c=new Criteria();
    $c->add(Sic0841Peer::FOLIO,$this->getRequestParameter('id'));
    $this->resultadocambio =Sic0841Peer::doSelectOne($c);

    $d=new Criteria();
    $d->add(DestruccionPeer::SIC_0841ID_SIC_0841,$this->getRequestParameter('id'));
    $this->resultado1=DestruccionPeer::doSelect($d);
    return sfView::SUCCESS;
}

```

Figura 20. Método executeVisualizar

En este fragmento de código que se encuentra en el template visualizarSuces.php, recibe un id, que ese id no es más que el id que se guarda en la variable resultadocambio de la acción visualizar en el actions.class.php, acto seguido se muestran los objetos de la tabla de muchos guardados en la variable resultado1, este proceso se hace dentro de un foreach para poder mostrar todos los objetos.

```

<?php use_helper('Object')?>
<?php echo input_hidden_tag('id', $id) ?>
<?php echo form_tag('Sic0841/modificar?id='.$resultadocambio->getFolio(), 'method=get')?>

<?php foreach ($resultadol as $p): ?>
<tr>
    <td width="36" align="center" valign="top"><?php echo $p->getNo();?></td>
    <td width="59" align="center" valign="top"><?php echo $p->getFecha(); ?>&nbsp;  </td>
    <td width="66" align="center" valign="top"><?php echo $p->getTipoBanco();?></td>
    <td width="66" align="center" valign="top"><?php echo $p->getCodigo();?></td>
    <td width="123" valign="top"><?php echo $p->getMotivoDestruccion();?></td>
    <td width="104" valign="top"><?php echo $p->getNoVialesDestruidos();?></td>
    <td width="104" valign="top"><?php echo $p->getDestruidoPor();?></td>
    <td width="104" valign="top"><?php echo $p->getInspeccionPor();?></td>
</tr>
<?php endforeach; ?>
    
```

Figura 21. Fragmentos de Código

Sf 1.0.10 [vars & config](#) [logs & msgs](#) 2

CENTRO DE INGENIERIA GENÉTICA Y BIOTECNOLOGÍA SUBDIRECCIÓN DE CALIDAD REGISTRO DE DESTRUCCIÓN DE BANCOS DE CÉLULAS					SIC-0841	PPO 4.09.118.99	
					Edición 01		
					FOLIO: 1		
Laboratorio BCP				Area: calidad			
Ilo.	Fecha	Tipo de Banco	Código	Motivos de la destrucción	Ilo de viales destruidos	Destruídos por	Inspeccionado por
1	1	1	1	1	1	1	1
4	46	66	2	64	77	68	474

Figura 22. Interfaz de Visualizar Sic0841.

2.5.1 Validación y manejo de errores.

En Symfony cuando un usuario realiza una petición a `miAccion`, este siempre busca primero un método llamado `validateMiAccion()`. De ser encontrado este procedimiento lo ejecuta. El valor de retorno de esta validación determina el siguiente método a realizar: si devuelve `true`, entonces se hace el método `executeMiAccion()`; en caso de que se devuelva `false`, se ejecuta `handleErrorMiAccion()`. En el caso de que `handleErrorMiAccion()` no exista, Symfony busca un método genérico llamado `handleError()`. Si tampoco existe el `handleError()`, simplemente devuelve el valor `sfView::ERROR` para producir la plantilla `miAccionError.php`.

La clave para un correcto funcionamiento de la validación es respetar la convención de nombres para los métodos de la acción:

- `validateNombreAccion`: es el método de validación, que devuelve `true` o `false`. Se trata del primer método buscado cuando se solicita la acción `NombreAccion`. Si no existe, la acción se ejecuta directamente.
- `handleErrorNombreAccion`: es el método llamado cuando el método de validación falla. Si no existe, entonces se muestra la plantilla `Error`.
- `executeNombreAccion`: es el método de la acción. Debe existir para todas las acciones.

Seguidamente se presentará un fragmento del archivo `Crear.yml` que se encuentra en la carpeta `validate` del módulo `Sic0020`.

`fillin:`

`enabled: true`

`fields:`

`nombre_solucion:`

`required:`

`msg: Dato requerido`

`vt:`

`required:`

`msg: Dato requerido`

`sfNumberValidator:`

`max: 100.0`

max_error: Entre un vol menor que 100.0

min: 0.01

min_error: Entre un vol mayor que 0

type: float

type_error: Dato incorrecto

reactivo:

sfStringValidator:

max: 25

max_error: Entre menos de 26 caracteres

min: 3

min_error: Entre más de 3 caracteres

values: varchar

values_error: Dato incorrecto

np:

sfNumberValidator:

min: 1

min_error: Entre un # mayor que 1

type: integer

type_error: Dato incorrecto

cantidad_utilizada:

sfNumberValidator:

max: 10.0

max_error: Entre un vol menor que 10.0

min: 0.01

min_error: Entre un vol mayor que 0

type: float

type_error: Dato incorrecto

nombre:

sfStringValidator:

max: 15

max_error: Entre menos de 16 caracteres

min: 3

min_error: Entre más de 3 caracteres

values: varchar
values_error: Dato incorrecto
fecha_calib:
sfDateValidator:
date_error: Entre una fecha(yyyy-mm-dd).
solvente:
required:
msg: Dato requerido
sfStringValidator:
max: 15
max_error: Entre menos de 16 caracteres
min: 3
min_error: Entre más de 3 caracteres
values: varchar
values_error: Dato incorrecto
ph:
required:
msg: Dato requerido
sfNumberValidator:
max: 100
max_error: Entre un ph menor
min: 1
min_error: Entre un ph mayor que 0
type: integer
type_error: Dato incorrecto
conductividad:
required:
msg: Dato requerido
sfNumberValidator:
max: 10.0
max_error: Entre menor conductividad que 10
min: 0.01
min_error: Entre mayor conductividad 0

type: float

type_error: Dato incorrecto

silice:

required:

msg: Dato requerido

sfNumberValidator:

max: 10.0

max_error: Entre un vol menor 10.0

min: 0.01

min_error: Entre un vol mayor que 0

type: float

type_error: Dato incorrecto

nomb_solucion:

sfStringValidator:

max: 15

max_error: Entre menos de 16 caracteres

min: 3

min_error: Entre más de 3 caracteres

values: varchar

values_error: Dato incorrecto

phfinal:

sfNumberValidator:

max: 100

max_error: Entre un ph menor

min: 1

min_error: Entre un ph mayor que 0

type: integer

type_error: Dato incorrecto

conductividad_solucion:

sfNumberValidator:

max: 10.0

max_error: Entre menor conductividad que 10

min: 0.01

min_error: Entre mayor conductividad 0

type: float

type_error: Dato incorrecto

filtracion:

required:

msg: Dato requerido

sfStringValidator:

max: 20

max_error: Entre menos de 21 caracteres

min: 3

min_error: Entre más de 3 caracteres

vaporsaturado:

sfNumberValidator:

max: 100.0

max_error: Entre menor vapor que 100

min: 0.01

min_error: Entre mayor vapor 0

type: float

type_error: Dato incorrecto

volumen_frascos:

required:

msg: Dato requerido

sfNumberValidator:

max: 100.0

max_error: Entre un vol menor que 100.0

min: 0.01

min_error: Entre un vol mayor que 0

type: float

type_error: Dato incorrecto

pruebas:

sfStringValidator:

max: 255

max_error: Entre menos de 256 caracteres

min: 3

min_error: Entre más de 3 caracteres

limite:

sfStringValidator:

max: 15

max_error: Entre menos de 16 caracteres

min: 3

min_error: Entre más de 3 caracteres

realizado:

required:

msg: Dato requerido

sfRegexValidator:

pattern: /^[a-z]+\$ /i

match: yes

match_error: Dato incorrecto

revisado:

required:

msg: Dato requerido

sfRegexValidator:

pattern: /^[a-z]+\$ /i

match: yes

match_error: Dato incorrecto

fecha_r:

required:

msg: Dato requerido

sfDateValidator:

date_error: Entre una fecha(yyyy-mm-dd).

terminado:

required:

msg: Dato requerido

YAML es el acrónimo de “*YAML Ain’t Markup Language*” (“YAML no es un Lenguaje de Marcado”) y se pronuncia “*yamel*”. El formato se lleva utilizando desde 2001 y existen utilidades para procesar YAML en una gran variedad de lenguajes de programación. YAML es mucho más rápido de escribir que XML (ya que no hacen falta las etiquetas de cierre y el uso continuo de las comillas) y es mucho más poderoso que los tradicionales archivos .ini (ya que estos últimos no soportan la herencia y las estructuras complejas). Por este motivo, Symfony utiliza el formato YAML como el lenguaje preferido para almacenar su configuración.³⁴

³⁴ Fabien Potencier, François Zaninotto. *Symfony la guía definitiva*, 2008. Disponible en: <http://www.librosweb.es/symfony> .

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA DIRECCIÓN DE CALIDAD		SIC-0020	PPO 4.09.148.04
		Edición 04	PÁGINA 1 DE 1
REGISTRO DE PREPARACIÓN DE SOLUCIONES			FOLIO ↓ Dato requerido ↓ <input type="text"/>
Departamento:	Control de Calidad	Área:	Calidad
Nombre de la Solución: <input type="text"/>			
No. Parte:	Select... ▼	No. lote:	↓ Dato requerido ↓ <input type="text"/>
		VT:	↓ Dato requerido ↓ <input type="text"/>
		Fecha prep:	2008-06-21 ...
Reactivos	HP	Cantidad utilizada	Lote
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
			Fecha Vencimiento: 2009-04-01 ...
<input type="button" value="Add"/>			
EQUIPOS UTILIZADOS			
Nombre		Código	Fecha de venc. calibración
<input type="text"/>		<input type="text"/>	2019-04-01 ...
<input type="button" value="Add"/>			
Solvente:	Seleccionar... ▼ Otros	pH:	↓ Dato requerido ↓ <input type="text"/>
		Conductividad:	↓ Dato requerido ↓ <input type="text"/>
		Silice:	↓ Dato requerido ↓ <input type="text"/>
Requiere ajuste la solución SI <input type="radio"/> No <input type="radio"/>			

Filtración:	<input type="text" value="Seleccionar..."/> <input type="button" value="Otros"/>		
Vapor saturado:	<input type="text"/> °C <input type="text"/> min		
Almacenamiento			
No. de frascos:	↓ Dato requerido ↓ <input type="text"/>	Volumen por frasco:	↓ Dato requerido ↓ <input type="text"/>
Almacenar la solución a:	↓ Dato requerido ↓ <input type="text"/>	Fecha de vencimiento:	2019-04-01 <input type="button" value="..."/>
PRUEBAS DE ACEPTACIÓN	LÍMITE	VALOR OBTENIDO	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
Observaciones:			
<input type="text"/>			
	Nombre y Apellidos	Firma	Fecha
REALIZADO POR	↓ Dato requerido ↓ <input type="text"/>		↓ Dato requerido ↓ <input type="text"/> <input type="button" value="..."/>
REVISADO POR	↓ Dato requerido ↓ <input type="text"/>		

Figura 23. Interfaz de Error en la opción de Crear Sic0020.

Conclusiones

En este capítulo fueron mostrados los artefactos que se generaron en el flujo de trabajo de implementación. Se obtuvieron los diagramas de componentes de cada caso de uso implementado, asociado con el diagrama de clases del diseño correspondiente y las descripciones de cada uno de ellos. Además se muestra un número de funciones desarrolladas para la implementación propuesta, siendo estas las más significativas para el funcionamiento de la misma; así como fragmentos de código para la validación y manejo de errores.

CONCLUSIONES

Los Sistemas para el Manejo de Información de Laboratorios son útiles para controlar los principales procesos de un laboratorio o entidad, es por ello que la realización de este trabajo permitió dar los primeros pasos de la implementación de un LIMS para el CIGB, cumpliendo los objetivos trazados:

- Se realizó el estudio del modelado del negocio.
- Se realizó el estudio del modelo del sistema propuesto.
- Se realizó el estudio del diseño propuesto en la investigación precedente.
- Se confeccionaron los diagramas de componentes e implementación del sistema a partir de las clases del diseño.
- Se implementaron los componentes necesarios para el funcionamiento del módulo Biología Molecular del LIMS de calidad del CIGB mediante los requisitos funcionales, apoyándose en las descripciones de los casos de uso.

RECOMENDACIONES

Después de haber culminado el presente trabajo de diploma es recomendable:

- Dar continuidad a este trabajo y culminar con la implementación de los restantes casos de uso.
- Integrar el módulo Biología Molecular con los demás módulos que formarán parte del LIMS de Calidad del CIGB, después de que se termine con el diseño y la implementación de cada uno de ellos.
- Una vez desarrollado en Cuba un producto de este tipo, pudiera ser extensible a otros centros del Polo Científico y en general a entidades que lo requieran para mejorar y agilizar la calidad de sus productos.
- Realizar las pruebas de unidad una vez que se concluya con la implementación de todos los Casos de Uso.

REFERENCIAS BIBLIOGRÁFICAS

1. Centro de Ingeniería Genética y Biotecnología. *Departamento de Control de la Calidad* 2003a. [2006]. Disponible en: <http://www.cigb.edu.cu/pages/ccalidad.htm>
2. *Matrix LIMS, la herramienta de productividad para cualquier laboratorio*. Disponible en: <http://www.softwarecientifico.net/productos.asp?pid=472>
3. *MediLab Interfaces*. Disponible en: <http://medilabit.com/interfaces.htm>
4. *LabWare LIMS - Sistema de Información de Laboratorio*. Disponible en: <http://www.chemie.de/products/es/52345/>
5. Idem Referencia 4.
6. Matos García Rosa María. "Diseño de Bases de Datos". octubre 1999. Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=10192&subdir=/Bibliografia>
7. Domínguez Vaillant Arodys E. y Miranda Gutiérrez Duniel. "Sistema de Manejo de Datos de Ensayos Clínicos: Diseño e implementación de la Base de Datos." p 23.
8. *Arquitectura Cliente Servidor*. Disponible en: <http://www.csi.map.es/csi/silice/Global71.html>
9. Mohammed J. Kabir. *La biblia Servidor Apache 2*, p. 41-43.
10. Sæther Bakken Stig, Aulbach Alexander, Schmid Egon, Colectivo de Autores. *Manual de PHP*.
11. Eguilus Pérez Javier. Introducción a JavaScript. Disponible en: <http://www.librosweb.es/javascript/capitulo1.html>
12. *Plataforma Comunidad en Español de Eclipse IDE*. Disponible en: <http://plataformaclipse.com/>
13. LEON WELICKI. *Patrones y Antipatrones: una Introducción - Parte II*. Disponible en: http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317/default.aspx
14. Idem Referencia 13.
15. Fabien Potencier, François Zaninotto. *Symfony la guía definitiva*, 2008. Disponible en: <http://www.librosweb.es/symfony> , p 13.
16. Ídem Referencia 15.
17. IVAR JACOBSON, G. B., JAMES RUMBAUGH. *El Proceso Unificado de Desarrollo de Software.*, 2000. Pág 4.
18. Ídem referencia 17
19. Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 4.
20. Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 255.
21. Conferencia UCI "Fase de Elaboración. Flujo de trabajo de implementación", 2008. Disponible en: http://inter-nos.uci.cu/Teleclases/tc_conferencias.asp?id_asig=ingenieria_de_software_2

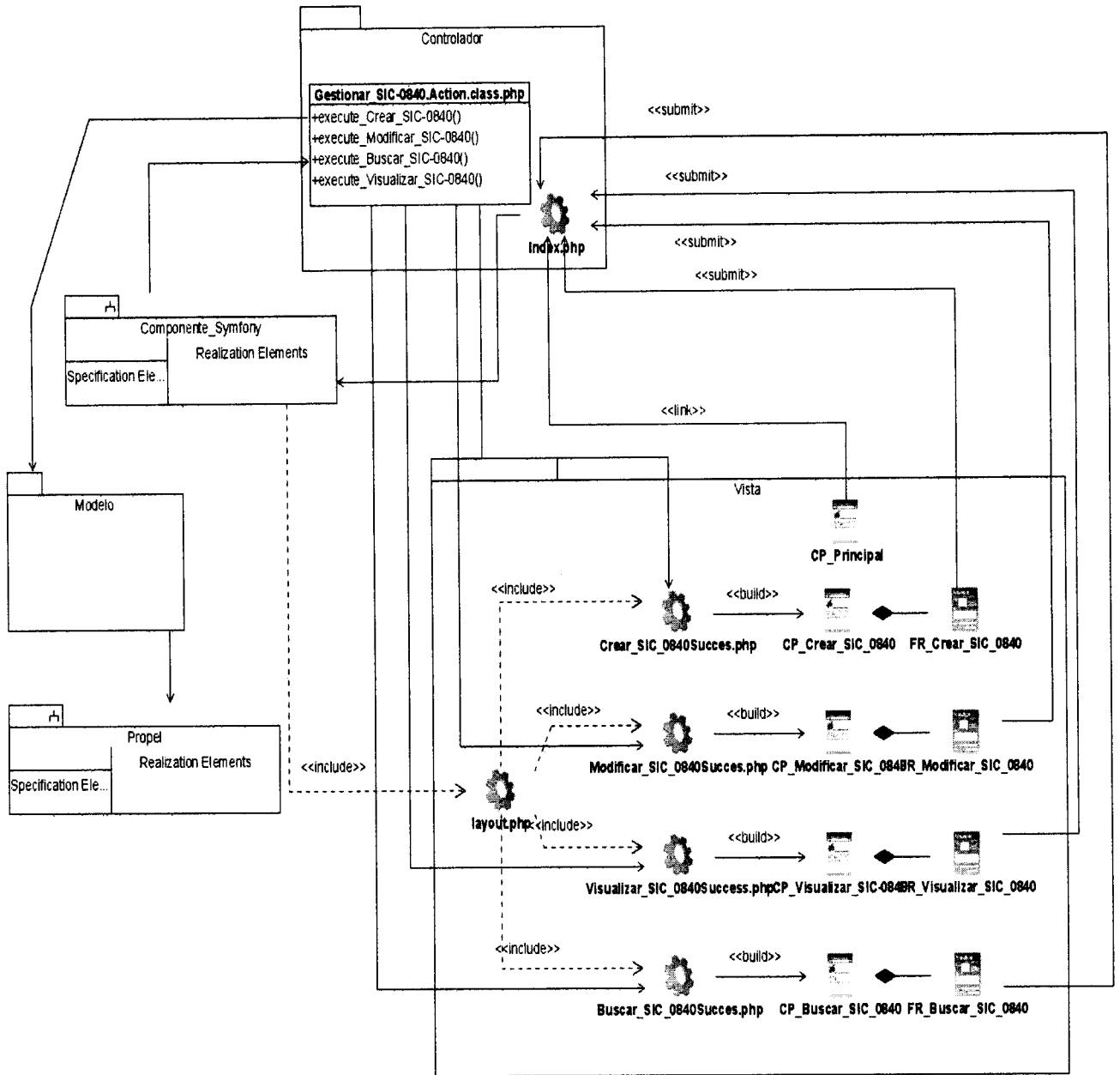
22. Ídem referencia 21
23. Ídem referencia 21
24. Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 426
25. Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 257.
26. Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 258.
27. Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 260.
28. Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 268.
29. Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*, 2000. p 282.
30. LARGMAN CRAIG. *Uml y Patrones. Introducción al análisis y diseño orientado a objetos*, 2004.
31. *Herramientas Case. Disponible en:*
http://209.85.215.104/search?q=cache:7UN1dYGFZGoJ:www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r19670.DOC+Herramientas+CASE:+son+diversas+aplicaciones+inform%C3%A1ticas+destinadas+a+aumentar+la+productividad&hl=es&ct=clnk&cd=5&gl=cu&lr=lang_es
32. Almenteros Yuniesky, Sánchez Yusdenis. "Software para la Simulación de Sistemas Biológicos: Módulo de modelación gráfica de Sistemas Biológicos" p 19
33. Pedraza Eslavy, Puig Pedro Manuel. 2008. "Sistema para la Gestión de la Información de los Laboratorios de la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis y Diseño del módulo Biología Molecular"
34. Fabien Potencier, François Zaninotto. *Symfony la guía definitiva*, 2008. Disponible en:
<http://www.librosweb.es/symfony> .

BIBLIOGRAFÍA

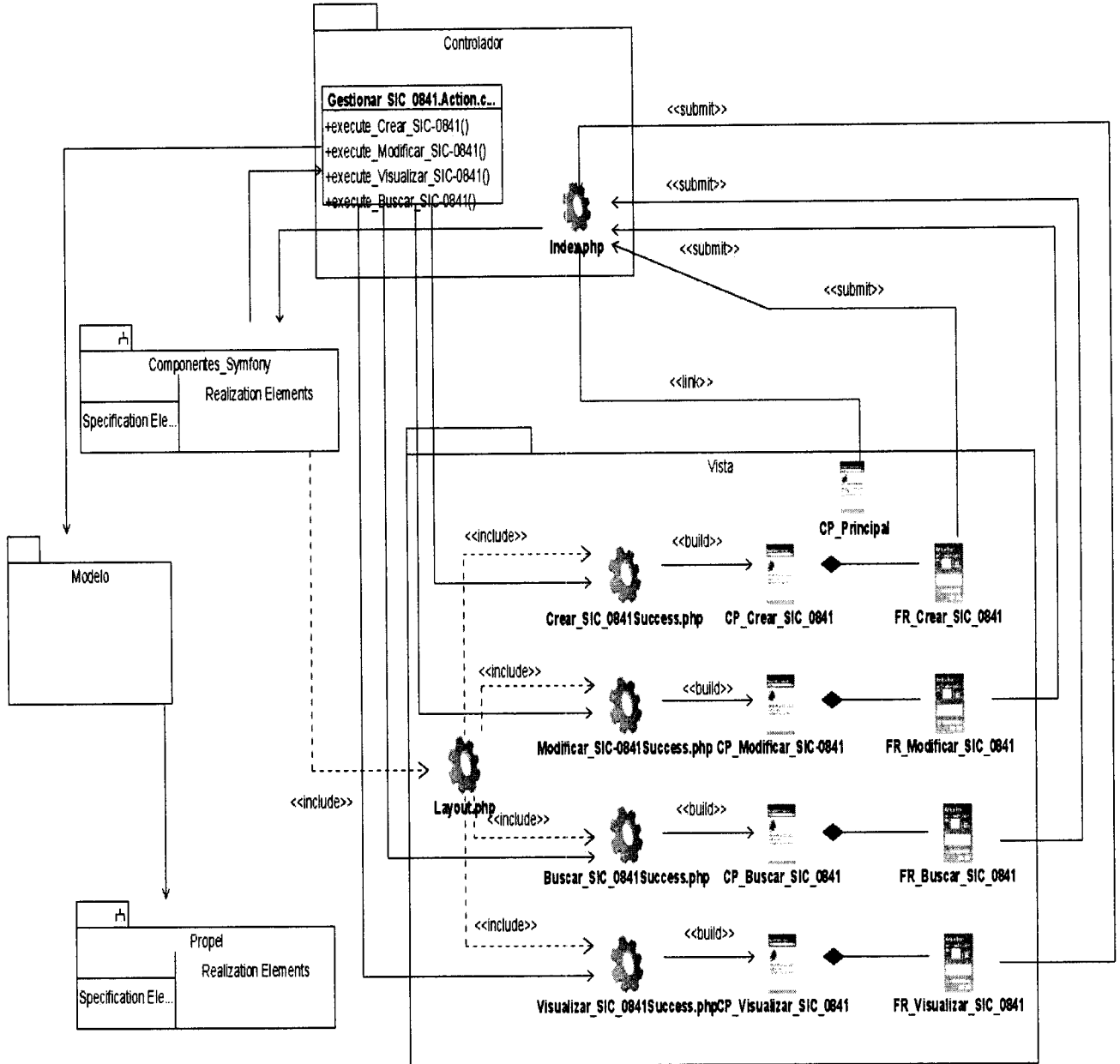
- Almenteros Yuniesky, Sánchez Yusdenis. "Software para la Simulación de Sistemas Biológicos: Módulo de modelación gráfica de Sistemas Biológicos" p 19
- Ayuda Extendida Rational Rose.
- BOOCH, Grady; RUMBAUGH, James, JACOBSON; Ivar; "El lenguaje unificado de modelado. Manual de referencia".2000. Addison Wesley. Capítulos 4-5, 8, 12 Páginas 41-64, 93-102.
- Carlo Batini, Stefano Ceri & Shamkant B. Navathe: "Diseño conceptual de bases de datos" Addison-Wesley / Díaz de Santos, 1991 ISBN 0-201-60120-6
- CIGB. Departamento de Control de la Calidad.2003. Disponible en: <http://www.cigb.edu.cu/pages/ccalidad.htm>
- CIGB. Biología Molecular. .2003. Disponible en: http://www.cigb.edu.cu/pages/bm_calidad.htm
- Conferencia de Implementación, UCI. 2008. Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=22199>
- Domínguez Vaillant Arodys E. y Miranda Gutiérrez Duniel. "Sistema de Manejo de Datos de Ensayos Clínicos: Diseño e implementación de la Base de Datos." p 23.
- Egulus Pérez Javier. Introducción a JavaScript. Disponible en: <http://www.librosweb.es/javascript/capitulo1.html>
- Fabien Potencier, François Zaninotto. Symfony la guía definitiva,2008.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. s.l. : PEARSON EDUCACIÓN S.A, 2000.
- JACOBSON, Ivar; RUMBAUGH, James; BOOCH, Grady, "El proceso unificado de desarrollo".2000. Addison Wesley. capítulo 8 Páginas 165-181, 185-200
- LARMAN, Craig, "UML y Patrones". 2000. Prentice Hall. Capítulos 9, 10, 11, 12.
- Matos García Rosa María. "Diseño de Bases de Datos" . octubre 1999. Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=10192&subdir=/Bibliografia>.
- MENDOZA SANCHEZ, M. A. Metodologías de desarrollo de software, 2004. [2007]. Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
- Mohammed J. Kabir. La biblia Servidor Apache 2, p. 41-43.
- Pedraza Eslavy, Puig Pedro Manuel. 2008. "Sistema para la Gestión de la Información de los Laboratorios de la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis y Diseño del módulo Biología Molecular"

- PRESSMAN, Roger "Ingeniería del Software. Un enfoque práctico". 2002. McGraw-Hill/Interamericana de España. Páginas 184-186 "Técnicas para facilitar las especificaciones de la aplicación.
- Ramez A. Elmasri & Shamkant B. Navathe: "Fundamentos de Sistemas de Bases de Datos" Addison-Wesley, 2002 [3ª ed.] ISBN 84-782-9051-6
- RUMBAUGH, James, JACOBSON, Ivar; BOOCH, Grady, "El lenguaje unificado de modelado. Manual de referencia". 2000. Addison Wesley. Capítulos 4 y 13 Páginas 42-48, 122-125, 131-143, 156, 162-170.
- Sæther Bakken Stig, Aulbach Alexander, Schmid Egon, Colectivo de Autores. Manual de PHP.
- VICE-RECTORÍA DE FORMACIÓN and DEPARTAMENTO DE PRÁCTICA PROFESIONAL. Propuesta de Guía para la presentación del Trabajo de Diploma (Versión para desarrollo de Sistemas Informáticos), 2006. [2006].
- VISUAL PARADIGM. Visual Paradigm for UML, 2005. [2007]. Disponible en: <http://www.visual-paradigm.com/product/vpuml/index.jsp>

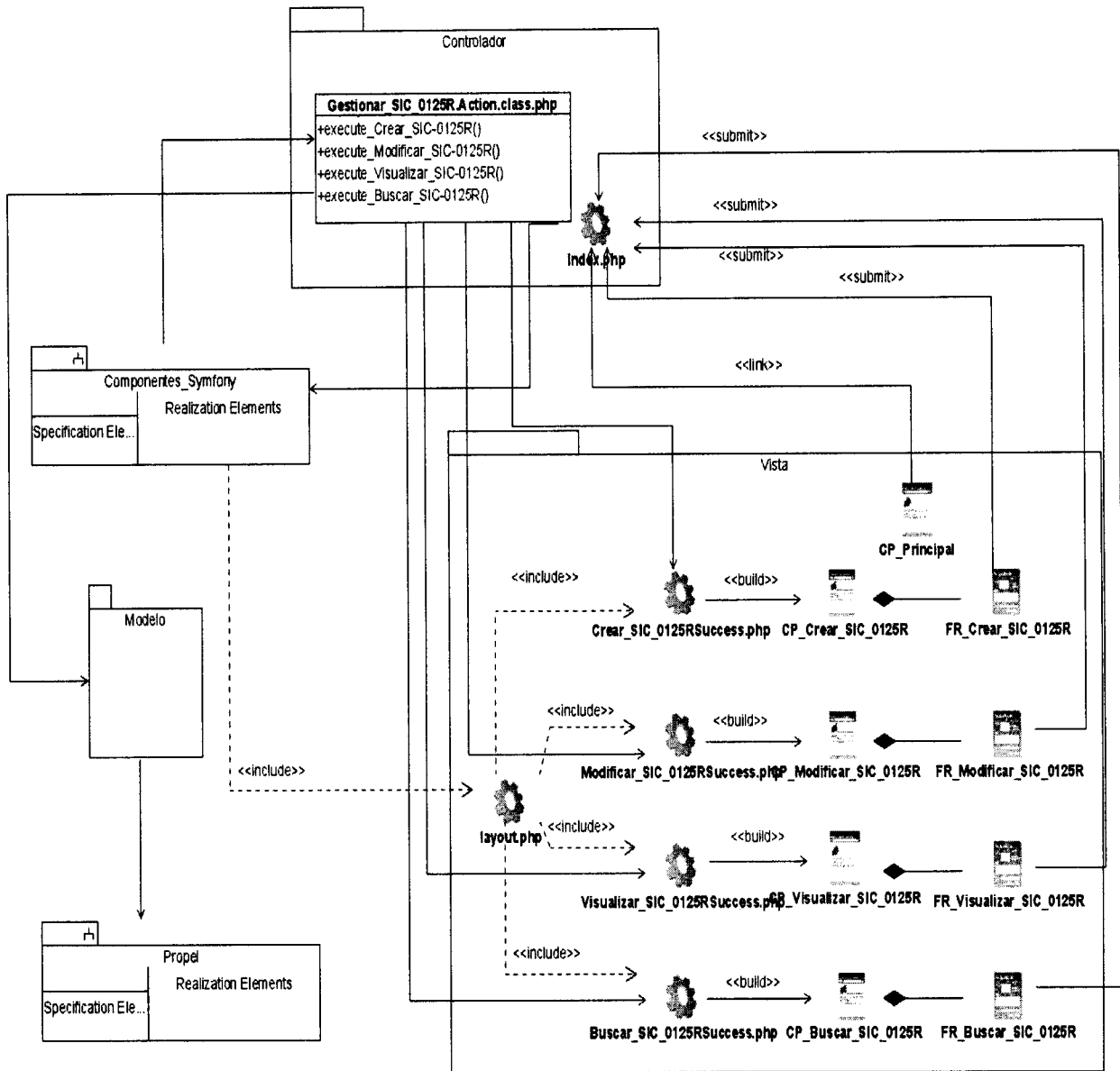
ANEXO 1: Diagrama de clases del diseño Gestionar Solicitud de destrucción de bancos de células (SIC-0840).



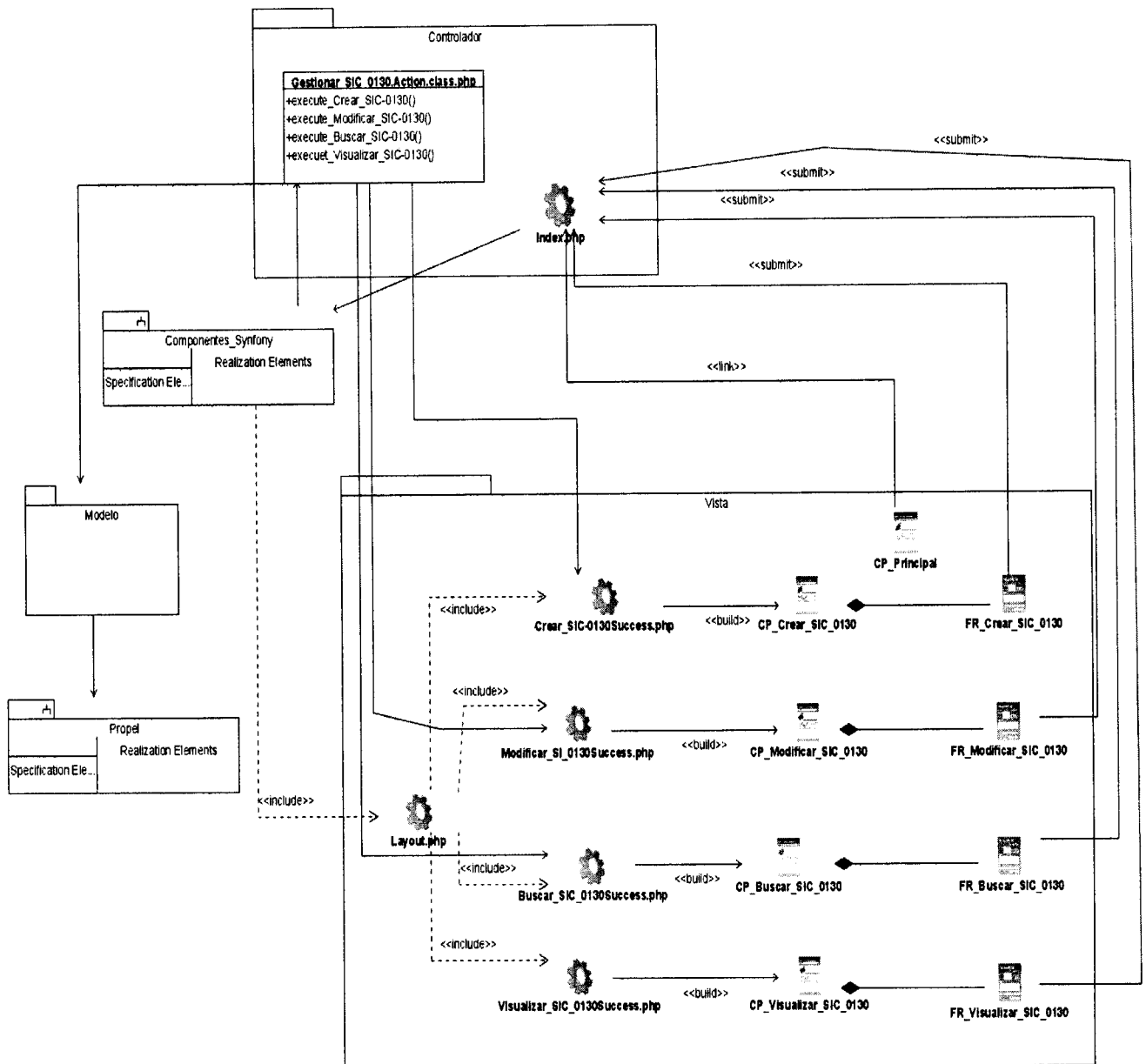
ANEXO 2: Diagrama de clases del diseño Gestionar registro de Destrucción de Banco de Células (SIC-0841).



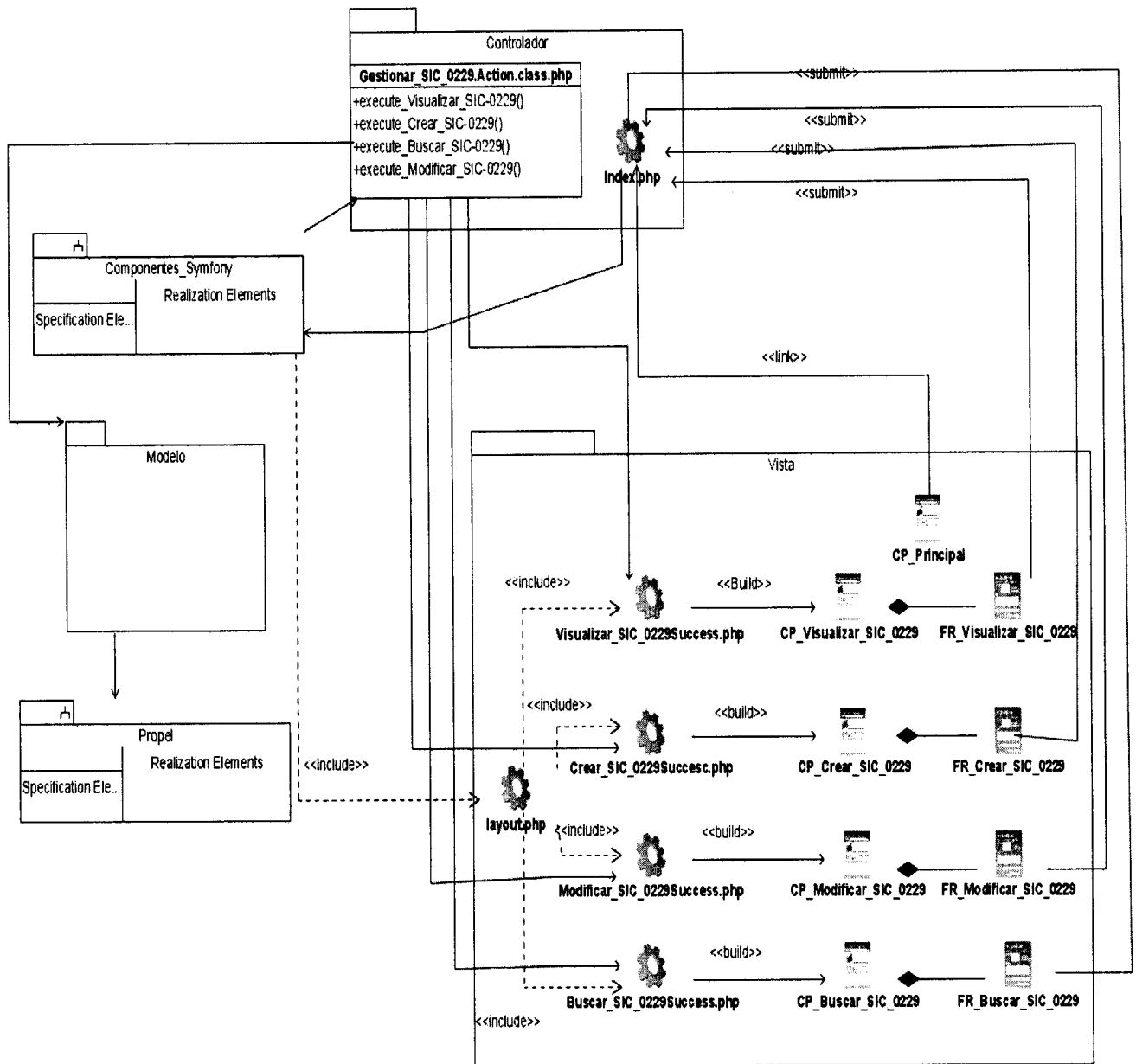
ANEXO 3: Diagrama de clases del diseño Gestionar registro de Temperatura (SIC-0125R)



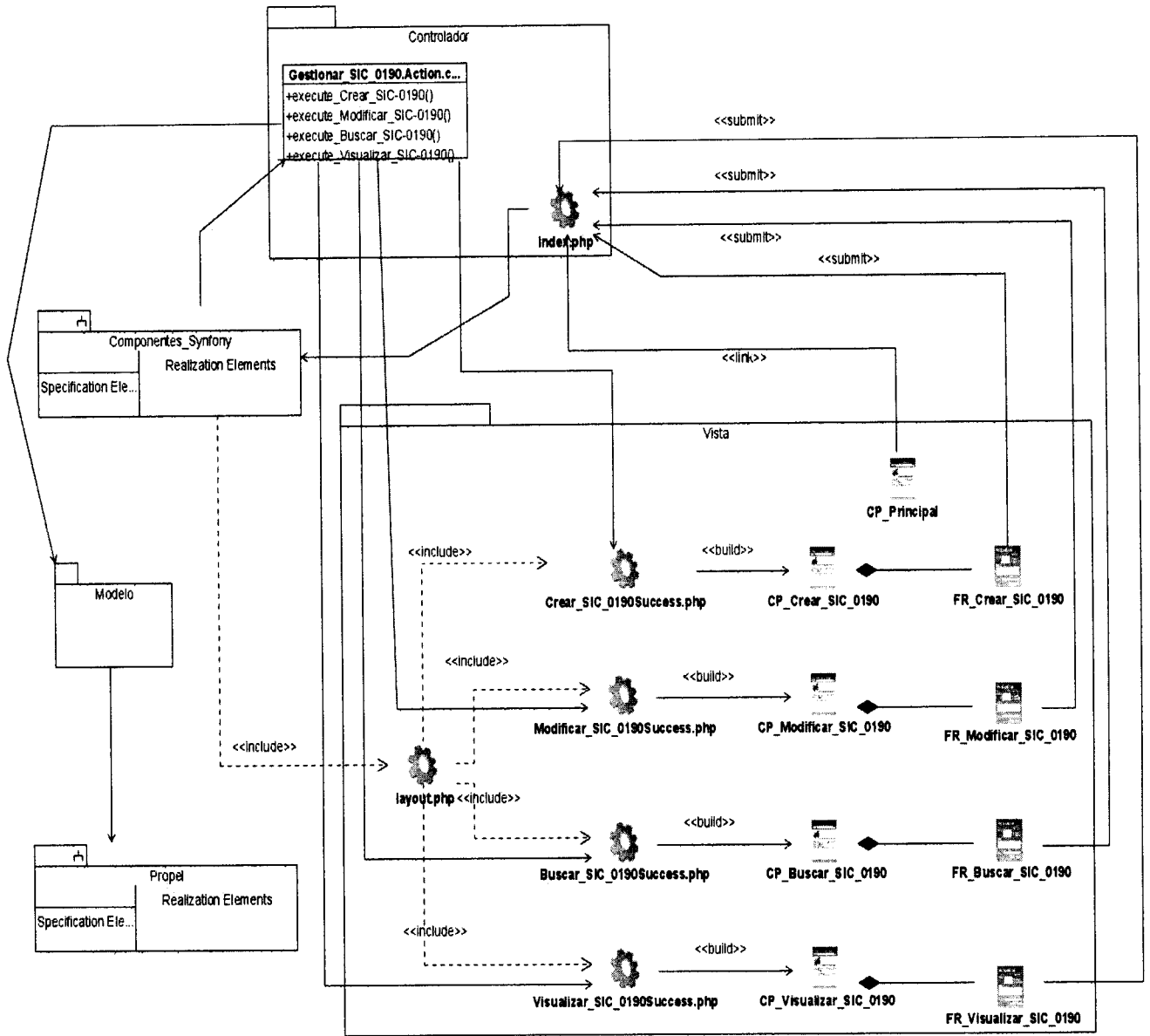
ANEXO 4: Diagrama de clases del diseño Gestionar Gestionar Cronograma de chequeo de bancos de células (SIC-0130).



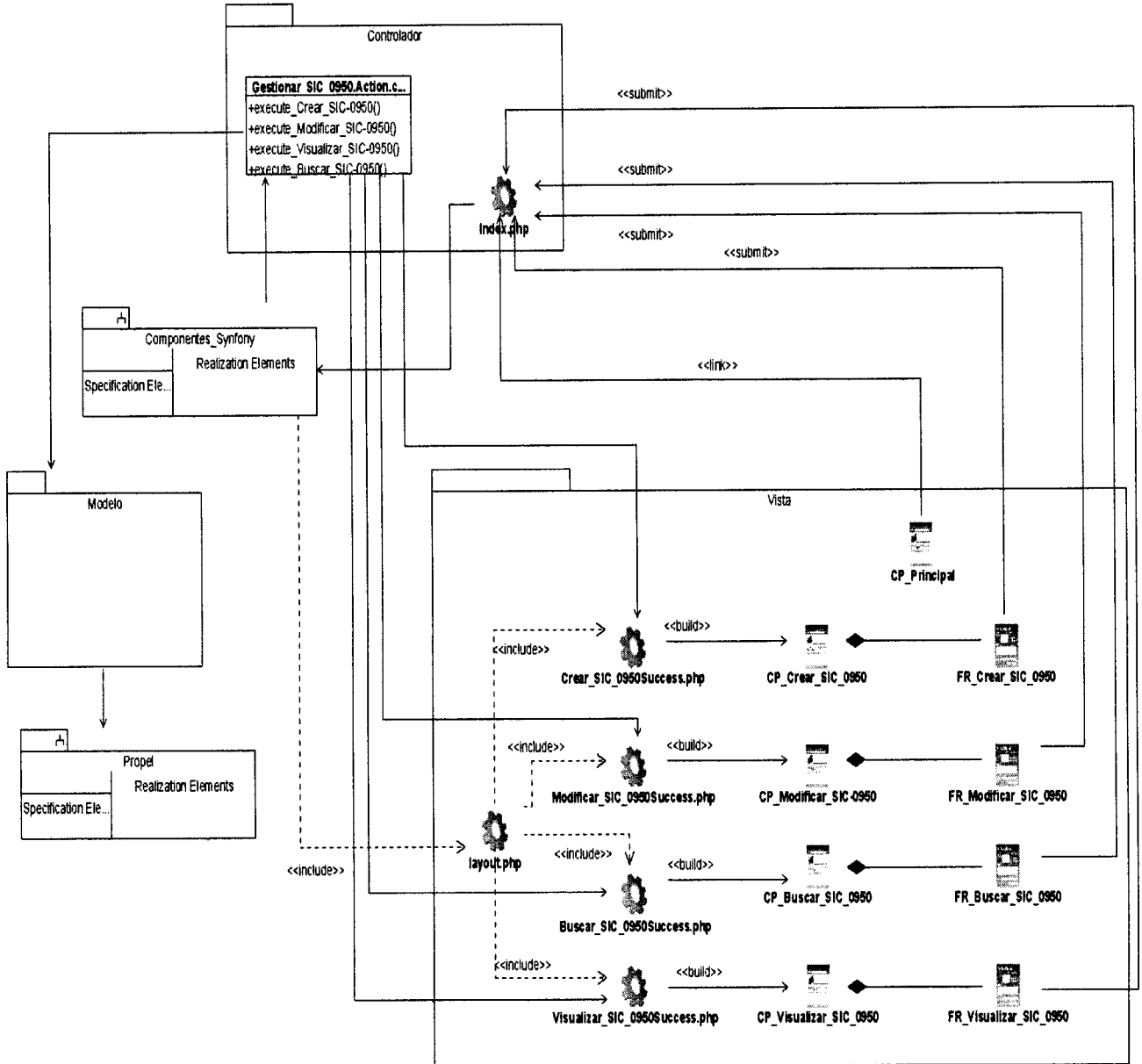
ANEXO 5: Diagrama de clases del diseño Gestionar Certificado de Liberación de los bancos de Células transformados (SIC-0229)



ANEXO 6: Diagrama de clases del diseño Gestionar registro de Descarga para Banco de Células (SIC-0190)



ANEXO 7: Diagrama de clases del diseño Gestionar Informe de análisis de los bancos de células (SIC-0950)



ANEXO 8: Interfaz visual del Crear Sic0840.

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA SUBDIRECCIÓN DE CALIDAD SOLICITUD DE DESTRUCCIÓN DE BANCOS DE CÉLULAS		SIC-0840	PPO 4.09.118.99
		Edición 01	
		FOLIO:	86
Area: Calidad			
Código del Banco a destruir	07A8	Número de viales a destruir:	124
Motivos de la Destrucción: Sic presenta problemas en la estructura molecular.			
Fecha de destrucción:		Hora de la destrucción	
Persona responsable de la destrucción			
Nombre y apellido	René Fuentes Mora	Firma	
Cargo que ocupa	analista	Fecha	2008-05-21 ...
Autorización de la destrucción			
Nombre y apellido	Yosvani Quiala Sánchez	Firma	
Cargo que ocupa	Main	Fecha	2008-06-03 ...

ANEXO 9: Interfaz visual del Visualizar Sic0840.

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA SUBDIRECCIÓN DE CALIDAD SOLICITUD DE DESTRUCCIÓN DE BANCOS DE CÉLULAS		SIC-0840	PPO 4.09.118.99
		Edición 01	
		FOLIO: 87	
Laboratorio BCP		Area: calidad	
Código del Banco a destruir	07A8	Número de viales a destruir : 124	
Motivos de la Destrucción: Sic presenta problemas en la estructura molecular.			
Fecha de destrucción:	2008-07-03	Hora de la destrucción	11:4
Persona responsable de la destrucción			
Nombre y apellido	René Fuentes Mora	Firma	
Cargo que ocupa	analista	Fecha	2008-05-21
Autorización de la destrucción			
Nombre y apellido	Yosvani Quijala Sánchez	Firma	
Cargo que ocupa	Main	Fecha	2008-06-03

GLOSARIO

A

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

B

Bioteología: Conocimientos y avances biológicos aplicados a procesos tecnológicos o de interés industrial.

C

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

Concurrent Versions System (CVS): Es una aplicación informática que implementa un sistema de control de versiones. Mantiene el registro de todo el trabajo y los cambios en los ficheros que forman un proyecto.

E

Electroforesis: La electroforesis es una técnica para la separación de moléculas (proteínas o ácidos nucleicos) según la movilidad de estas en un campo eléctrico a través de una matriz porosa, la cual finalmente las separa por tamaños moleculares y carga eléctrica, dependiendo de la técnica que se use.

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

Entorno de desarrollo integrado: En inglés *Integrated Development Environment ('IDE')* es un programa compuesto por un conjunto de herramientas para un programador.

H

HTML: Sus siglas en inglés son: HyperText Markup Language (Lenguaje de Marcado de Hipertexto) en español, es el lenguaje de marcado predominante para la construcción de páginas web.

Hibernate: Es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

I

IBM: International Business Machines o IBM (NYSE: IBM) (conocida coloquialmente como el Gigante Azul) es una empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática. Tiene su sede en Armonk (Estados Unidos) y está constituida como tal desde el 15 de junio de 1911, pero lleva operando desde 1888.

Ingrediente Farmacéutico Activo (IFA): Cualquier sustancia o mezclas de sustancias destinadas a ser usadas en la fabricación de un producto farmacéutico y que cuando se use, constituye el ingrediente activo del mismo. Tales sustancias están destinadas a ejercer una actividad farmacológica u otro efecto directo en el diagnóstico, cura, mitigación, tratamiento o prevención de enfermedades o a afectar la estructura y función del organismo.

Inmunoquímica: Laboratorio que realiza el control de la calidad a los productos obtenidos por vía recombinante que se producen en el CIGB, empleando las siguientes técnicas inmunoquímicas: ELISA, Inmuno-Dot y Western-Blot.

Interfaz de usuario: Interfaz a través de la cual el usuario interactúa con el sistema.

J

JUnit : Es un conjunto de librerías creadas por Erich Gamma y Kent Beck que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.

L

Lenguaje scripting: Lenguaje de programación que fue diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados.

Licencia BSD: Es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Pertenece al grupo de licencias de software Libre.

M

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

P

Plugin: Es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica.

PhpDocumentor: Es un sistema para crear documentación de aplicaciones creadas con PHP.

PPO: Procedimiento patrón de operación.

S

SIC: Sistema Informativo de Calidad.

X

XSS: Es un ataque basado en explotar vulnerabilidades del sistema de validación de HTML incrustado.