

Universidad de las Ciencias Informáticas

Facultad 6



Título: “Sistema para la Gestión de la Información de los Laboratorios de la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis y Diseño del módulo Biología Molecular”

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

Autores: Eslavy Pedraza González

Pedro Manuel Puig Díaz

Tutora: Ing. Adisley Reyes Crespo

Ing. Elennis Díaz Laurencio

La Habana, junio de 2008

“Año del 50 Aniversario de la Revolución”

DECLARACIÓN DE AUTORÍA

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

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

Eslavy Pedraza González

Autor

Pedro Manuel Puig Díaz

Autor

Ing. Adisley Reyes Crespo
Tutora

Ing. Elennis Díaz Laurencio
Tutora

DATOS DE CONTACTO

Ing. Adisley Reyes Crespo: Facultad 6 de la Universidad de Ciencias Informáticas, Ciudad de la Habana, Cuba. areyesc@uci.cu

Ing. Elennis Díaz Laurencio: Facultad 6 de la Universidad de Ciencias Informáticas, Ciudad de la Habana, Cuba. ediaz@uci.cu

AGRADECIMIENTOS

Agradecemos a Fidel y a la Revolución por darnos la oportunidad de realizar este sueño.

Agradecemos a nuestros familiares por la incondicionalidad y el apoyo de siempre, en especial a nuestros queridos padres por el apoyo y por creer en nosotros.

A nuestra tutora Adisley y a Elennis que se convirtió en nuestra tutora también por aguantarnos todo el tiempo y ayudarnos en todo, gracias Titi y Chuchi de todo corazón.

A los profesores de la Facultad por todo lo que nos han enseñado en estos 5 cursos.

A nuestros compañeros y profesores del proyecto.

A Karina, Lien, Pepe, Norlen, los Reynaldos, Yadira, Yalina, a todos los que nos decían: “¿y la tesis?”, a todos los que de una forma u otra nos ayudaron, muchas gracias.

A nuestros compañeros de la UCI, los que nos soportan, nos quieren, nos aguantan y sinceramente nos estiman, muchas gracias.

DEDICATORIA

Pedro:

A mami y papi, quienes por su ejemplo, esfuerzo y dedicación les debo lo que soy y a quienes no podría defraudar jamás; por todo lo que significan para mí, ustedes son mi todo. Si volviese a nacer y me permitieran elegir, sin dudas, desearía que fuesen mis padres. Los adoro.

A mis familiares que han estado al tanto de mí y me han dado su apoyo, especialmente a mi tía.

A "El Tierno", que más que mi amigo es mi hermano.

A Lien por ser tan especial y soportar todas mis malcriadeces.

A todos mis amigos que han estado siempre en el momento oportuno sin importar para lo que sea: Adonis, Gabriel, Carlos, Ariel, en fin, a todos.

A los excelentes amigos que encontré aquí en la UCI, Rosado, Azulito y Peña. Ustedes son especiales.

A mi compañero de tesis, Eslavy, sabes que estamos ahí para lo que sea.

A Elennis, (Chuchi), por su apoyo incondicional y por no reparar en día ni hora para ayudarme.

A la familia de la "Coca" que tanto cariño me ha brindado.

A la familia de Bauta que tanto se preocupa por mí.

A la FEU por ratificarme que el sacrificio y la entrega sin esperar retribuciones en una mejor persona.

A todos lo que me han dado un consejo oportuno y han formado valores en mí.

Eslavy:

A mis excelentes padres por todo lo que han hecho por mi en la vida, por ser mi razón de ser, por educarme y formarme como soy, por ser demasiado especiales, mami y pacho esto es por ustedes, es lo mínimo que merecen de mí.

A mi abuelita Eva por amarme como lo hace, por desearlo y rezarlo tanto que me volvía loco, no te preocupes viejuca que mi primer sueldo no te lo quita nadie.

A mis demás familiares, a Marti, Yole y familia por todo el apoyo de siempre.

A mi pelusa linda y matancera por ser lo mejor que me ha pasado en la vida, por estar siempre a mi lado y por quererme como soy.

De forma especial a mi hermano Pedruco por estar a mi lado desde el primer día en la UCI y todo el apoyo que siempre me dio, este coronel nunca te olvidará hermano.

A chuchi (Elennis) por su constancia y ayuda descomunal.

A Adisley por ayudarme y estar siempre dispuesta.

A Rey, Rosado y Peña, esos amigos inigualables, insoportables, maravillosos, súper especiales, de las buenas y las malas, a ese piquete que seguro no conoce nadie...han dejado una huella imborrable y siempre los llevaré presente.

A todos los que de una forma u otra tuvieron que ver con este sueño.

RESUMEN

En la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología (CIGB), se manipulan grandes volúmenes de información relacionada con los procesos que allí se realizan, lo cual es de gran importancia en el control y aseguramiento de la calidad de las producciones del Centro. Toda esta información se almacena de forma manual, por lo que se dificulta su acceso y provoca pérdida de tiempo considerable al ser revisada, supervisada y aprobada por las personas con cargos superiores.

Debido a la necesidad de contribuir a la gestión y control de la información que se genera, se está realizando un Sistema de Gestión de Información de los Laboratorios (LIMS, del inglés Laboratory Information Management System) para esta área, en conjunto con la facultad 6 de la Universidad de las Ciencias Informáticas (UCI). El presente trabajo describe el análisis y diseño del módulo de Biología Molecular (BM), laboratorio perteneciente al departamento de Control de la Calidad de la Dirección de Calidad del CIGB.

PALABRAS CLAVE: CIGB, LIMS, Gestión de la Información, Control de la Calidad.

ÍNDICE

AGRADECIMIENTOS	I
DEDICATORIA	II
RESUMEN	III
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Sistemas de Gestión de la Información	6
1.1.1 La Web como sistema de información	7
1.1.2 Tipos y Usos de los Sistemas de Gestión de la Información	7
1.1.3 Sistemas de Gestión de la Información de Laboratorio (LIMS: Laboratory Information Management System)	9
1.2 Herramientas y Metodologías utilizadas en la aplicación.	12
1.2.1 Metodología de desarrollo de software.	12
1.2.2 Flujos de Trabajo a realizar: Requerimientos y Análisis y Diseño.	14
1.2.2.1 Roles y Artefactos.	15
1.2.3 Herramientas	20
1.2.3.1 Herramienta CASE.....	20
1.2.4 Lenguaje de Modelado.	21
1.2.5 Herramienta para la Gestión de Requerimientos.....	22
1.2.6 Lenguaje de Programación.....	25
1.2.7 Framework de Desarrollo.....	25
1.3 Patrones.	26
1.3.1 Patrones de Casos de Usos	26
1.3.2 El patrón MVC	27
1.3.3 Patrones GRASP	29
1.4 Vistas Arquitectónicas.....	31
1.5 Conclusiones	32
CAPÍTULO 2: REQUERIMIENTOS	33
2.1 Actores del sistema a automatizar: Laboratorio de Biología Molecular (LBM).....	33
2.2 Requerimientos Funcionales y No Funcionales del Sistema.....	33
2.2.1 Requerimientos funcionales: módulo de BM.....	33
2.2.2 Requerimientos no funcionales.	36
2.3 Modelo de Casos de Uso del Sistema	38
2.4 Descripción de los casos de uso del módulo: BM.	39
2.5 Conclusiones.....	53
CAPÍTULO 3: DISEÑO DEL SISTEMA	54

3.1 Propósitos del diseño.....	54
3.2 Principios de diseño.	55
3.3 Referencia a la Arquitectura del Sistema.	55
3.3.1 Vista lógica.....	55
3.3.2 Vista de despliegue.....	60
3.4 Mapa de Navegación.....	61
3.5 Diagramas de Clases del Diseño.	66
3.6 Diagrama de Secuencia.	76
3.7 Evaluación de la calidad del diseño	76
3.8 Conclusiones.....	81
CONCLUSIONES.....	82
RECOMENDACIONES	83
REFERENCIA BIBLIOGRÁFICA	84
BIBLIOGRAFÍA	86
ANEXOS	88
Anexo 1. Organigrama del Departamento de Calidad del CIGB.....	88
Anexo 2. Fases e Iteraciones de RUP.....	89
Anexo 3. Fases e hitos en RUP.	89
Anexo 4. Artefactos a realizar según los roles a desempeñar.	90
Anexo 5. Representación gráfica del Lenguaje de Modelado.....	90
Anexo 6. Prototipos no funcionales del CUS.....	91
GLOSARIO DE TÉRMINOS.....	94

ÍNDICE DE FIGURAS

<i>Fig 1. Actividades que realiza un Sistema de Información.</i>	8
<i>Fig 2. Tipos y usos de los Sistemas de Información.</i>	9
<i>Fig 3. Diagrama de Gestión de Requerimientos.</i>	23
<i>Fig 4. Funcionamiento del patrón MVC.</i>	28
<i>Fig 5. Diagrama de Casos de Uso del Sistema: LBM.</i>	39
<i>Fig 6. Vista Lógica: Paquete Caracterización y Chequeo de Bancos de Células.</i>	57
<i>Fig 7. Realización del CU Gestionar Solicitud de destrucción de bancos de células (SIC_0840).</i>	58
<i>Fig 8. Gestionar Solicitud de Destrucción de Bancos de Células (SIC-0840).</i>	59
<i>Fig 9. Vista de Despliegue.</i>	60
<i>Fig 10. Mapa de navegación: Diagrama General.</i>	62
<i>Fig 11. Mapa de Navegación: Diagrama # 1: Técnicas de Hibridación.</i>	63
<i>Fig 12. Mapa de Navegación: Diagrama # 2: Caracterización y Chequeo de Bancos de Células.</i>	64
<i>Fig 13. Mapa de Navegación: Diagrama # 3: Técnicas de Biología Molecular. Reactivos Biológicos.</i>	65
<i>Fig 14. Gestionar registro de Destrucción de Banco de Células (SIC-0841).</i>	67
<i>Fig 15. Gestionar Registro de Temperatura (SIC-0125R).</i>	68
<i>Fig 16. Gestionar Cronograma de Chequeo de Bancos de Células (SIC-0130).</i>	69
<i>Fig 17. Gestionar registro de Determinación de la Estabilidad Plasmídica (SIC-0149).</i>	70
<i>Fig 18. Gestionar registro de Determinación del Marcador de Selección (SIC-0152).</i>	71
<i>Fig 19. Gestionar registro de Descargo para Banco de Células (SIC-0190).</i>	72
<i>Fig 20. Gestionar Certificado de Liberación de los Bancos de Células transformados (SIC-0229).</i>	73
<i>Fig 21. Gestionar Informe de análisis de los bancos de células (SIC-0950).</i>	74
<i>Fig 22. Gestionar registro de Muestreo de los bancos de Células (SIC-0951).</i>	75
<i>Fig 23. Escenario Crear SIC_0840: Gestionar Solicitud de Destrucción de Bancos de Células.</i>	78
<i>Fig 24. Escenario Buscar-Visualizar SIC_0840: Gestionar Solicitud de Destrucción de Bancos de Células.</i>	79
<i>Fig 25. Escenario Modificar SIC_0840: Gestionar Solicitud de Destrucción de Bancos de Células.</i>	80

ÍNDICE DE TABLAS

<i>Tabla 1. Descripción de Actores del Sistema</i>	33
<i>Tabla 2. Gestionar Solicitud de destrucción de bancos de células (SIC-0840).</i>	44
<i>Tabla 3. Gestionar registro de Determinación del marcador de selección (SIC-0152).</i>	45
<i>Tabla 4. Gestionar registro de Descargo para Banco de Células (SIC-0190).</i>	46
<i>Tabla 5. Gestionar registro de Temperatura (SIC-0125R).</i>	47
<i>Tabla 6. Gestionar registro de Muestreo de los bancos de Células (SIC-0951).</i>	48
<i>Tabla 7. Gestionar Cronograma de chequeo de bancos de células (SIC-0130).</i>	49
<i>Tabla 8. Gestionar Certificado de Liberación de los bancos de Células transformados (SIC-0229).</i>	50
<i>Tabla 9. Gestionar registro de Determinación de la Estabilidad Plasmídica (SIC-0149).</i>	51
<i>Tabla 10. Gestionar registro de Destrucción de Banco de Células (SIC-0841).</i>	52
<i>Tabla 11. Gestionar Informe de Análisis de los bancos de células (SIC-0950).</i>	52

INTRODUCCIÓN

Como parte del Programa para Informatizar la Sociedad Cubana, el Estado Socialista Cubano, ha venido experimentando un incremento en la utilización de las tecnologías de la información en los últimos años. Con el fin de alcanzar un sustancial mejoramiento en la infraestructura tecnológica que permita satisfacer las necesidades de acceso, almacenamiento y utilización de la información tanto en la esfera socioeconómica como política; elevándose así la calidad de vida, la eficiencia y eficacia de los servicios prestados, la preparación del capital humano y la competitividad del país; como una vía para alcanzar una posición de vanguardia a nivel mundial en la utilización de estas tecnologías. Este proyecto se va realizando de manera acelerada y alcanzando resultados satisfactorios en áreas esenciales como la Educación, la Salud y la Investigación.

A importantes centros del país, que se informatizan exitosamente, han llegado los efectos de esta labor, logrando grandes avances en el procesamiento de la información que generan. Entre ellos está el Centro de Ingeniería Genética y Biotecnología (CIGB) con 21 años de experiencia en la investigación científica y la obtención de productos reconocidos a nivel mundial. Este Centro, junto a otros, ha situado a Cuba entre los países subdesarrollados más destacados en este ámbito. [1]

Gran impacto en la biomedicina, salud animal, mejoramiento vegetal y la bioindustria ha tenido el CIGB, propiciado por la excelente labor que ha desplegado el personal científico que labora en este centro; ha desarrollado nuevas vacunas y fármacos para la salud humana que se encuentran actualmente en uso dentro del sistema de salud cubano, así como en diferentes países.

Los productos desarrollados y elaborados en este Centro se caracterizan por su eficacia y seguridad para con sus clientes, por tal motivo, en la estructura del centro existe la Dirección de Calidad, compuesta por el Departamento de Aseguramiento de la Calidad y el Departamento de Control de las Calidad. [1] (Ver Anexo 1).

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

- Grupo de Desarrollo.
- Grupo de Liberación Analítica.
- Sección Físico-Química compuesta por tres laboratorios:
 - Laboratorio Análisis Químico.
 - Laboratorio de Cromatografía y Electroforesis.
 - Laboratorio de Sistemas Críticos.
- Sección Biológica compuesta por cinco laboratorios:
 - Laboratorio de Microbiología.
 - **Laboratorio de Biología Molecular (LBM).**
 - Laboratorio de Ensayos Biológicos I.
 - Laboratorio de Ensayos Biológicos II.
 - Laboratorio de Inmunoquímica.

La sección biológica se encarga de ejecutar las técnicas analíticas relacionadas con la determinación de la actividad biológica in vivo e in vitro, entre otras. Entre sus laboratorios se encuentra el de **Biología Molecular (BM)** que se encarga básicamente de: cuidar los bancos de células y bancos primarios, para esto se realizan experimentos y pruebas con diferentes muestras de células, se aplican diferentes técnicas biológicas a los reactivos: Técnicas de caracterización y chequeo de Bancos de Células, Caracterización y control de productos ADN, Técnicas de Hibridación de ácidos nucleicos (Dot-Blot y Southern Blot), Técnicas de preparación de reactivos biológicos (preparación de sondas, purificación de ADN Plasmídico y Purificación de ADN Cromosomal) y las Técnicas Generales (Procedimientos normativos que permiten el cumplimiento de las Buenas Prácticas del Laboratorio).

Las **Técnicas de Hibridación de ácidos nucleicos (Dot-Blot y Southern Blot)**: permiten detectar mediante un ensayo semi-cuantitativo la presencia de trazas de ADN contaminantes en productos farmacéuticos obtenidos por la tecnología del ADN recombinante. Estos ensayos se realizan a IFA (Ingrediente Farmacéutico Activo) de investigación y producción y a Productos Intermedios que forman parte de la validación de sistemas y caracterización del proceso. El Southern-Blot se emplea en la caracterización y chequeo de Bancos de Células.

Técnicas que se emplean para la obtención de Reactivos Biológicos:

Purificación de ADN Cromosomal: para el control de los Bancos de Células mediante Southern-Blot y en las curvas patrones de los Dot-Blot.

Purificación de ADN Plasmídico: para la Preparación de Sondas, como Control en los Análisis de Restricción y para el Chequeo de los Bancos de Células.

Preparación de sondas: para las técnicas de hibridación pues ayuda en el reconocimiento del ADN que se quiere detectar ya sea por Dot-Blot o Southern-Blot.

De forma general, el laboratorio de Biología Molecular necesita mejorar la gestión de toda la información que genera el desarrollo de estas técnicas. Todos los datos no están integrados en una plataforma capaz de organizar y centralizar todo el flujo de información que actualmente es recogida en documentos; que además, en la mayoría de los casos, es revisada, supervisada y aprobada por las personas con rangos superiores, lo que en ocasiones puede provocar pérdida de tiempo considerable cuando se necesita de una respuesta rápida para la toma de decisiones.

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 2007 un grupo de analistas de este proyecto estuvieron realizando el flujo de trabajo de Modelación del Negocio, para el módulo de Biología Molecular, definido por la metodología de desarrollo: Proceso Unificado de Desarrollo (RUP, de sus siglas en inglés Rational Unified Process), con este trabajo se pretende dar continuidad a este proceso previamente iniciado. De lo anterior se puede concluir que el proceso de Gestión de la Información del laboratorio de BM no fluye todavía de manera eficiente, segura y rápida. Por lo que se identifica como **problema científico**: ¿Cómo transformar los procesos identificados para el módulo Biología Molecular del Sistema de Gestión de la Información de los Laboratorios en elementos que puedan ser implementados?

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

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

Para dar solución al problema se define como **objetivo general**: Desarrollar el análisis y el diseño del módulo Biología Molecular para el Sistema de Gestión de la Información de los Laboratorios de la Dirección de Calidad del CIGB; de donde se derivaron los siguientes **objetivos específicos**:

- Obtener los requerimientos y las características del sistema para el módulo de BM.
- Realizar el diseño del módulo.

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

- Investigación sobre los roles de analista y diseñador.
- Estudio del Modelo del Negocio realizado en la investigación precedente.
- Identificación de los requerimientos.
- Identificación de actores y casos de uso.
- Realización del modelo de casos de uso.
- Especificación de los casos de uso.
- Realización de los Casos de Uso del Diseño.
 - Diagrama de clases del diseño.
 - Diagramas de interacción del diseño.
 - Prototipos de interfaz de usuario.
- Elaboración de la Vista Lógica.
- Elaboración del Mapa de Navegación.
- Elaboración del Diagrama de Clases Persistentes.
- Realización del Diagrama de Despliegue.

El presente trabajo consta de Introducción, 3 capítulos, conclusiones, recomendaciones, bibliografías y anexos.

En el Capítulo 1 “**FUNDAMENTACIÓN TEÓRICA**”: se abordan diferentes elementos que brindan la base teórico conceptual para el desarrollo del análisis y diseño del módulo de BM del LIMS de la Dirección de Calidad del CIGB. Se brinda información acerca del estudio realizado de los Sistemas de Gestión de la Información en el mundo así como del desarrollo de estos sistemas en nuestro país. Además se realiza un análisis de las técnicas, tecnologías, metodologías y herramientas de software sobre las cuales se realizará el proceso de desarrollo.

En el Capítulo 2 “**REQUERIMIENTOS**”: se definen los requisitos funcionales, los no funcionales, los actores del sistema y el diagrama de casos de uso del sistema, la descripción textual de los casos de uso del sistema aprobados y los prototipos no funcionales de los mismos.

En el Capítulo 3 “**DISEÑO DEL SISTEMA**”: se realiza el diseño las clases que se implementarán, se representan los diagramas de las clases diseñadas con sus relaciones, agrupados en paquetes de diseño. Se realiza además el mapa de navegación, se hace referencia a la arquitectura del sistema a través de las Vista Lógica y de Despliegue y se realizan los diagramas de secuencia, así como la validación del diseño.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El presente capítulo tiene como objetivo abordar los diferentes elementos que brindan la base teórico conceptual para el desarrollo del análisis y diseño del módulo de BM del LIMS de la Dirección de Calidad del CIGB. Se brinda información acerca del estudio realizado de los Sistemas de Gestión de la Información en el mundo así como del desarrollo de estos sistemas en nuestro país. Además se realiza un análisis de las técnicas, tecnologías, metodologías y herramientas de software sobre las cuales se realizará el proceso de desarrollo.

1.1 Sistemas de Gestión de la Información

La información se ha convertido en el activo principal de las empresas, representando en la mayoría de los casos su principal ventaja estratégica. Es por ello, que el desarrollo de sistemas de información se ve sometido actualmente a grandes exigencias en cuanto a productividad y calidad, y se hace necesaria la aplicación de un nuevo enfoque en la producción del software, más cercano a una disciplina de ingeniería que a los hábitos y modos artesanales que, desafortunadamente, se han venido aplicando en más de una ocasión. [3]

La gestión de la información se vincula con la generación y la aplicación de estrategias, el establecimiento de políticas, así como con el desarrollo de una cultura organizacional y social dirigida al uso racional, efectivo y eficiente de la información en función de los objetivos y metas trazadas en materia de desempeño y de calidad.

Los Sistemas de Gestión de Información son un conjunto de elementos relacionados y ordenados, según ciertas reglas que aporta al sistema objeto, es decir, a la organización a la que sirve y que marca sus directrices de funcionamiento así como la información necesaria para el cumplimiento de sus fines; para ello, debe recoger, procesar y almacenar datos procedentes tanto de la organización como de fuentes externas, con el propósito de facilitar su recuperación, elaboración y presentación. Un sistema de gestión de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

1.1.1 La Web como sistema de información

Durante los últimos años, en el mundo, las empresas han estado sometidas a fuertes presiones y cambios en la apreciación de su efectividad y nivel de competitividad en el mercado. Se están implantando sistemas rigurosos de Aseguramiento de la Calidad para obtener relevancia en cuanto a todo lo referente en el mercado, a la Calidad de los productos desarrollados por cada compañía. Nuestro país no está ajeno al desarrollo de estos sistemas; cada vez son más las instituciones u organizaciones que eligen incorporar aplicaciones de gestión de la información, logrando así una mayor dinámica en sus procesos de negocio, mejor calidad y eficiencia; además de mantener líneas de trabajo, planes y acciones trazadas en cuanto a incrementar y mejorar el desarrollo de estos sistemas.

Específicamente, se percibe un auge del desarrollo de la Web como Sistema de Gestión de la Información en la sociedad cubana. La evolución de Internet como red de comunicación global y el surgimiento y desarrollo de la Web como servicio imprescindible para compartir información, creó un excelente espacio para la interacción del hombre con la información hipertextual, a la vez que sentó las bases para el desarrollo de una herramienta integradora de los servicios existentes en Internet. Los sitios Web, como expresión de sistemas de información, deben poseer los siguientes componentes:

- Usuarios.
- Mecanismos de entrada y salida de la información.
- Almacenes de datos, información y conocimiento.
- Mecanismos de recuperación de información.[4]

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

1.1.2 Tipos y Usos de los Sistemas de Gestión de la Información

Los Sistemas de Gestión de Información cumplen tres objetivos básicos dentro de las organizaciones:

- Automatización de procesos operativos.
- Proporcionar información que sirva de apoyo al proceso de toma de decisiones.
- Lograr ventajas competitivas a través de su implantación y uso.

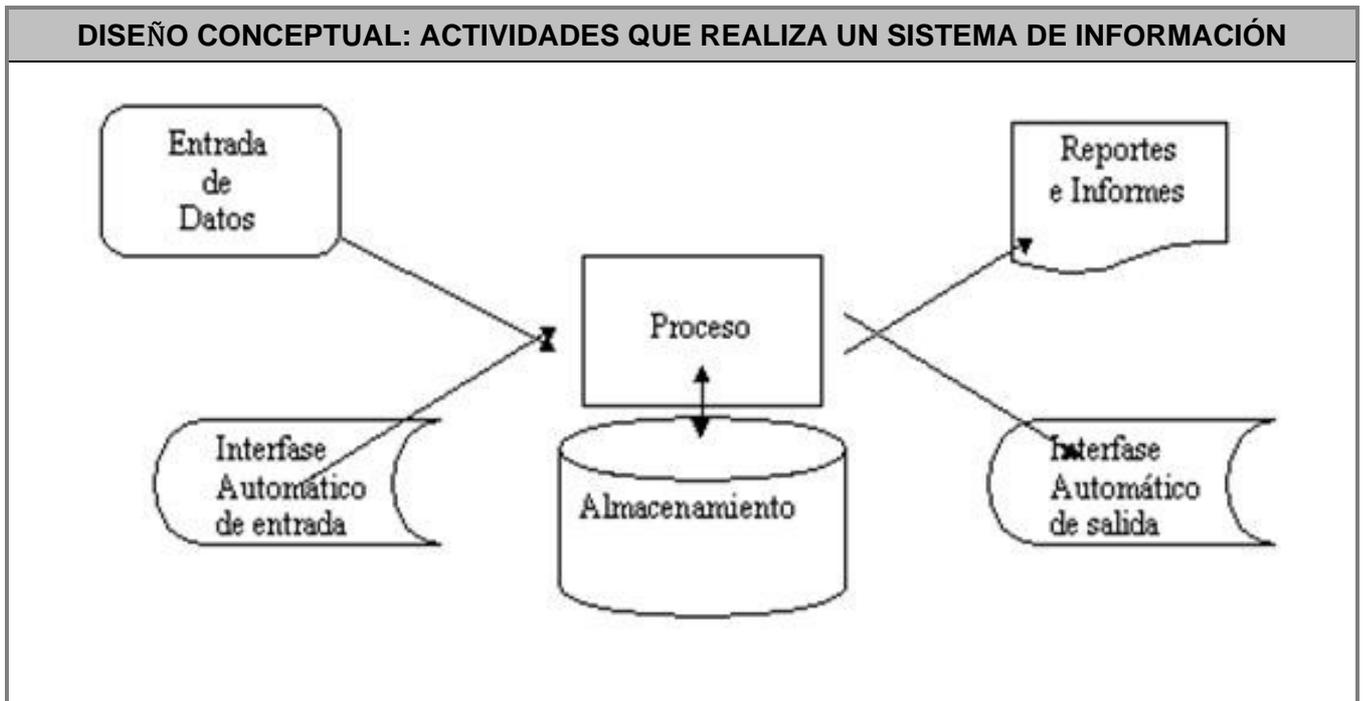


Fig. 1. Actividades que realiza un Sistema de Información.

Los Sistemas de Información que logran la automatización de procesos operativos dentro de una organización, son llamados frecuentemente Sistemas Transaccionales, su función primordial consiste en procesar transacciones tales como pagos, cobros, pólizas, entradas, salidas, entre otras. Los Sistemas de Información que apoyan el proceso de toma de decisiones son los Sistemas de Soporte a la Toma de Decisiones, Sistemas para la Toma de Decisión de Grupo, Sistemas Expertos de Soporte a la Toma de Decisiones y Sistema de Información para Ejecutivos. Los Sistemas Estratégicos se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información.

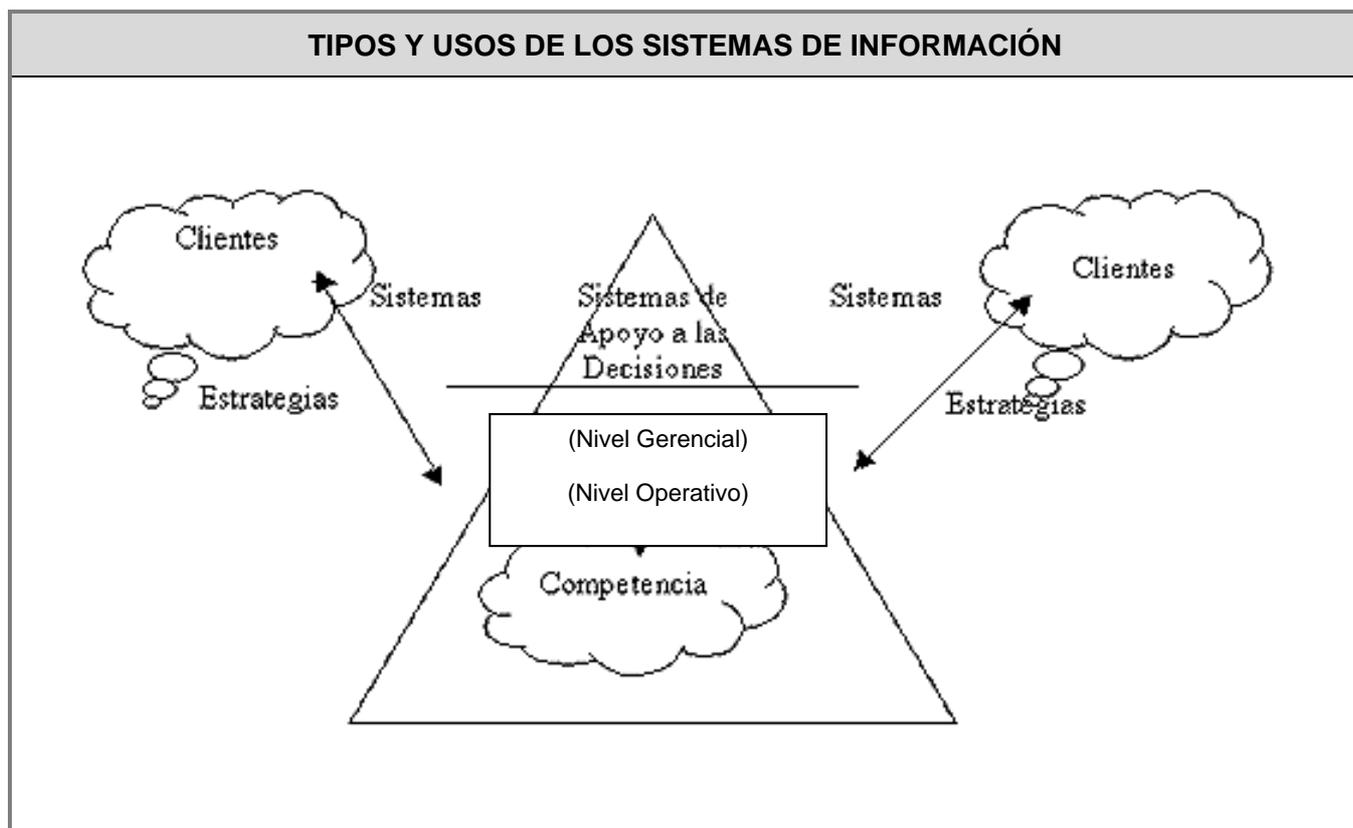


Fig 2. Tipos y usos de los Sistemas de Información.

1.1.3 Sistemas de Gestión de la Información de Laboratorio (LIMS: Laboratory Information Management System)

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

Un LIMS: es un programa de gestión de laboratorios que permite recoger, almacenar, calcular y gestionar datos en una amplia variedad de formas. Los LIMS representan una importante herramienta para la gestión global de un laboratorio en un entorno de calidad, agilizando temas de registro de datos primarios, archivo, trazabilidad, entre otros, y minimizando los errores provocados por la transferencia de información.[5] Proporcionan un conjunto de herramientas basadas en Sistemas Informáticos que

permiten la aplicación de técnicas de adquisición y gestión avanzada de la información producida en el laboratorio.

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

- Revisión y visualización de datos más completa, flexible y accesible.
- Generación más rápida y efectiva de informes.
- Centralización de la información en una única base de datos, lo que conlleva a que la accesibilidad a los datos sea más segura, rápida, cómoda, sencilla.
- Mantenimiento de estados ligados a Proyectos, Muestras, pruebas, Resultados, de manera automática por el Sistema.
- Generación automática de recibos de recepción de muestras.
- Planificación del trabajo del Laboratorio y generación automática de Hojas de Trabajo.
- Realización automática de cálculos y gráficos.

Es por ello que muchas empresas se han dedicado a la fabricación de LIMS, entre las que se pueden mencionar:

- **LabWorks LIMS**

LabWare LIMS es un Sistema de Gestión de Información de Laboratorios que opera en diferentes plataformas según la cantidad de usuarios que utilicen el sistema y se adapta a diferentes tipos de empresas. Es modular, por lo que el sistema puede expandirse de acuerdo a las necesidades de trabajo. Posee acceso equivalente cliente/servidor y Web, y se integra en cualquier entorno informático. La aplicación corre sobre Windows.

Gracias a su completa funcionalidad para seguimiento de muestras, certificación de usuarios, gestión de instrumentos, auditoría, y planificación de muestras e informes, así como muchas otras funciones, LabWare LIMS cumple las Buenas Prácticas de Laboratorio. LabWare LIMS fue desarrollado por PerkinElmer, empresa de Estados Unidos que se ha ido consolidando como una de las empresas líderes en el mercado de instrumental analítico de la más alta calidad y en soporte técnico. [1]

- **Matrix LIMS**

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

Como características principales se tienen:

- La flexibilidad de configuración.
- Es adecuado para laboratorios pequeños y grandes organizaciones.
- Incluye herramientas que son necesarias para cumplir los requerimientos de los laboratorios altamente regulados.
- Es compatible con las bases de datos Oracle y Microsoft SQL Server.
- Presenta una interfaz de usuario sencilla.
- Tiene tres opciones de producto: Matrix Express, Matrix Plus y Matrix Enterprise para brindar respuesta a las necesidades de cualquier laboratorio.

Autoscribe Limited, del Reino Unido, es una de las empresas líderes en el desarrollo y provisión de soluciones de software para LIMS y es, a su vez, el fabricante de Matrix LIMS. [1]

¿Por qué desarrollar un LIMS?

En investigaciones anteriores se concluyó que era necesaria la realización de un LIMS con características adaptables al CIGB, pues necesita ser flexible a los cambios que surgen con regularidad en el centro, además de las características que se listan a continuación:

- La mayoría de los proveedores de LIMS son de Estados Unidos.
- Las características del CIGB impiden poder configurar un LIMS para su uso particular, porque se realizan ensayos diferentes a diferentes productos y eso provoca que se necesite un sistema adaptable a las necesidades específicas de la Dirección de Calidad del CIGB.
- Resulta menos costoso desarrollarlo, debido a que los procesos de instalación, mantenimiento y actualización cuestan miles de dólares y el sistema que necesita la Dirección de Calidad del CIGB requiere mantenimiento de por vida por ser tan grande y variable.

- Una vez desarrollado en Cuba puede ser extensible a los centros del Polo Científico y en general a entidades que lo requieran para mejorar y agilizar los procesos relacionados con el control de la calidad de sus producciones.

1.2 Herramientas y Metodologías utilizadas en la aplicación.

Para realizar una aplicación informática siempre se deben definir los tipos de tecnologías a utilizar así como las herramientas y metodologías que serán de mayor utilidad para su implementación. Un proceso de desarrollo de software es un conjunto completo de actividades necesarias para convertir los requisitos de usuarios en un conjunto consistente de artefactos que conforman un producto software y para convertir los cambios sobre esos requisitos en un nuevo conjunto consistente de artefactos.

Después de un estudio de las características y necesidades requeridas para el desarrollo del LIMS, como estrategia del proyecto quedaron definidas las herramientas, así como la metodología a utilizar. Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y finales.

Entre las metodologías de desarrollo más conocidas se tienen: Programación Extrema (Extreme Programming, XP) clasificado como método ligero, Desarrollo Guiado por la Funcionalidad (Feature Driven Development, FDD) clasificado como un proceso de término medio, debido a que puede ser un método ligero o pesado y Proceso Unificado de Desarrollo (Rational Unified Process, RUP) que se clasifica como método pesado; y es la metodología definida para el desarrollo del trabajo.

1.2.1 Metodología de desarrollo de software.

Proceso Unificado de Desarrollo

La metodología RUP es el proceso unificado de desarrollo, que plantea quién hace qué, cuándo y cómo, tiene como objetivos asegurar la producción de software de calidad dentro de plazos y

presupuestos predecibles. Se actualiza constantemente para tener en cuenta las mejores prácticas y utiliza además UML como lenguaje de modelado.

El ciclo de vida de RUP se caracteriza por ser:

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo, por lo que describe los elementos del modelo más importantes para su construcción. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.

Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, unos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. [6]

RUP está compuesto por 4 fases fundamentales: (Ver anexo 2).

- **Inicio:** el objetivo es determinar la visión del proyecto.
- **Elaboración:** el objetivo es determinar la arquitectura óptima.
- **Construcción:** el objetivo es obtener la capacidad operacional inicial.
- **Transición:** el objetivo es obtener el producto del proyecto.

Cada fase concluye con un hito bien definido: (Ver anexo 3).

- **Inicio:** visión de los objetivos.
- **Elaboración:** prototipo de la arquitectura.
- **Construcción:** capacidad operacional inicial.
- **Transición:** liberación del producto.

RUP define los roles a jugar por cada miembro del equipo de desarrollo en cada una de las fases por las que transita el desarrollo de un sistema y facilita la comunicación entre los diferentes miembros del equipo de desarrollo.

Acorde a la metodología RUP, anteriormente se estuvo trabajando en el flujo de trabajo de **Modelación del negocio**, en el presente trabajo va a estar enmarcado en el desarrollo de los flujos de trabajo de: **Requerimientos y Análisis y Diseño**, definido por esta metodología.

1.2.2 Flujos de Trabajo a realizar: Requerimientos y Análisis y Diseño.

Flujo de Trabajo: Requerimientos.

La especificación de los requerimientos de software constituye un elemento de vital importancia para la elaboración de un software de calidad superior. La IEEE (Standard Glossary of Software Engineering Terminology) define a los requerimientos de software como condiciones o capacidades que deben estar presentes en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.[7]

Es necesario hacer énfasis en la precisión con que se debe realizar esta tarea debido a que cumple un papel primordial en el proceso de producción de software, pues enfoca un área fundamental: la definición de lo que se desea producir y permiten describir con mayor claridad, el comportamiento del sistema logrando de esta manera, minimizar los problemas relacionados al desarrollo de sistemas. Éste es uno de los flujos de trabajo más importantes, porque en él se establece qué tiene que hacer exactamente el sistema que se va a construir.

Los trabajadores que participan en este flujo de trabajo son:

- Analista del sistema.
- Especificador de casos de uso.
- Diseñador de interfaz de usuario.
- Arquitecto.

Principales actividades que se realizan:

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

Flujo de Trabajo: Análisis y Diseño.

Su objetivo es traducir los requisitos a una especificación que describe cómo implementar el sistema. El desarrollo de este flujo de trabajo estará centrado en la parte relacionada al Diseño. El diseño tiene

como propósito adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario. Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.

Los trabajadores que participan en este flujo de trabajo son:

- Arquitecto de Software.
- Diseñador.
- Diseñador de Base de Datos.
- Diseñador de Interfaz de Usuario.
- Revisor Técnico.

Principales actividades que se realizan:

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

1.2.2.1 Roles y Artefactos.

”**Rol:** Un rol es una definición abstracta de un conjunto de actividades realizadas y de artefactos obtenidos. Los roles son realizados típicamente por un individuo, o un conjunto de individuos, trabajando juntos en equipo. Un miembro del equipo de proyecto cumple normalmente muchos roles. Los roles describen cómo los individuos se comportan en el negocio y qué responsabilidades tienen”. [8]

- **Analista.**

Roles que desempeña el Analista.

El Analista agrupa los roles que están involucrados fundamentalmente en la captura y gestión de los requisitos del sistema, que pueden estar representados por una o varias personas entre los que se encuentran: **Analista de Procesos del Negocio, Diseñador del Negocio, Analista del Sistema y Especificador de Requerimientos.**

Por las actividades a desarrollar dentro de los flujos de trabajo de Requerimientos-Análisis y Diseño y dando continuidad al trabajo realizado referente a los principales procesos identificados por el Analista de procesos del Negocio en la Modelación del Negocio realizado anteriormente, se cumplirá específicamente con los roles: Analista del Sistema, Especificador de Requerimientos, Diseñador, Diseñador de Interfaz de Usuario y Arquitecto de Software.

- Analista del Sistema.

Es el responsable de dirigir y coordinar el proceso de captura de requisitos y desarrollo del modelo de casos de uso, definiendo las funcionalidades y límites del sistema. Los artefactos que realiza son: plan de gestión de requerimientos, documento visión, modelo de casos de uso, glosario, solicitudes de los stakeholder, storyboard, especificación suplementaria y atributos de requerimientos.

- Especificador de Requerimientos.

Es el encargado de especificar los detalles de una o varias partes de la funcionalidad del sistema, describiendo uno o varios aspectos de los requisitos, además de agrupar los casos de usos en paquetes. Los artefactos que realiza son: casos de uso, paquete de casos de uso, requerimientos de software y especificación de requerimientos del software. [9]

- Diseñador.

Es el responsable de diseñar una parte del sistema, dentro de las limitaciones de los requerimientos, la arquitectura y los procesos de desarrollo del proyecto. Identifica y define las responsabilidades, operaciones y relaciones entre los elementos del diseño. Además asegura que el diseño es consistente con la arquitectura del software y es detallado a un punto donde la implementación puede continuar. Los artefactos que genera son: realización de los casos de uso, clases del análisis, subsistemas del diseño, paquetes del diseño, clases del diseño y clases de prueba.

- Diseñador de interfaz de usuario.

Es el que coordina y elabora el diseño de la interfaz del usuario abarcando en la misma los requerimientos de forma tal que sean cumplidos en su totalidad. Vale aclarar que no es responsable de implementar dicha interfaz de usuario, solo de diseñar la forma visual de la misma. Los artefactos que genera son: prototipo de interfaz de usuario y mapa de navegación.

- Arquitecto de software.

Es el responsable de la arquitectura del software, que incluye las principales decisiones técnicas que limitan el diseño global y la ejecución del proyecto.[9]

Artefactos a Realizar.

Un producto o artefacto: es un trozo de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los productos son los resultados tangibles del proyecto, las cosas que va creando y usando hasta obtener el producto final.[10]

Para el desarrollo del proyecto que da lugar a la creación de este trabajo, se han tenido en cuenta varios de los artefactos fundamentales realizados por los roles de Especificador de Requerimientos, Analista del Sistema y Diseñador, que por interés del equipo de desarrollo serán abordados. (Ver Anexo 4).

Artefactos que obtiene el Analista del Sistema:

Modelo de casos de uso: describe bajo la forma de acciones y reacciones el comportamiento del sistema desde el punto de vista del usuario, permitiendo de esta forma el establecimiento de un acuerdo entre clientes y desarrolladores sobre las condiciones y requisitos que debe cumplir el sistema. Este modelo está formado por actores, casos de usos y las relaciones que se establecen entre estos, es decir, representa gráficamente a los procesos y su interacción con los actores y constituye una entrada de gran valor para las siguientes fases de construcción de un software.

Contiene:

- Paquetes de Casos de Uso.
- Caso de Uso.
- Actor: Representa terceros fuera del sistema que colabora con éste.

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

Glosario: es un documento que define los términos comunes que se utilizan para describir el proyecto.

Artefactos del Especificador de Requerimientos:

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

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

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

Artefactos que obtiene el Diseñador:

Modelo de Diseño: el modelo de diseño es un modelo de objetos que describe la realización de los casos de uso, y sirve como una abstracción del modelo de implementación y su código fuente. El modelo de diseño se utiliza como insumo esencial a las actividades de implementación y prueba.

Diseño de clases: es una parte fundamental de la aproximación de un diseño orientado a objetos. Una clase es una descripción de un conjunto de objetos que comparten las mismas responsabilidades, las relaciones, las operaciones, atributos, y la semántica. Las clases del diseño son una parte fundamental de un diseño orientado a objetos.

Una clase de diseño es una construcción similar en la implementación del sistema:

- El lenguaje utilizado para especificar una clase del diseño es lo mismo que el lenguaje de programación.

- Las relaciones de aquellas clases del diseño implicadas por otras clases, a menudo tienen un significado directo cuando la clase es implementada.
- Los métodos de una clase del diseño tienen correspondencia directa con el correspondiente método en la implementación de las clases.
- Una clase de diseño puede posponer el manejo de algunos requisitos para las siguientes actividades de implementación, indicándolos como requisitos de implementación de la clase.
- Una clase de diseño puede proporcionar interfaces si tiene sentido hacerlo en el lenguaje de programación.

Paquete de diseño: un paquete de diseño es una colección de clases, relaciones, realizaciones de caso de uso, diagramas, y otros paquetes. Se usa para estructurar el modelo de diseño y dividirlo en partes más pequeñas.

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

Diagramas de interacción: los diagramas de secuencia y los diagramas de colaboración (ambos llamados diagramas de interacción) son dos de los cinco tipos de diagramas de UML que se utilizan para modelar los aspectos dinámicos de los sistemas. Un diagrama de interacción muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes; un diagrama de colaboración es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes.

Artefactos que obtiene el Diseñador de Interfaz de Usuario:

Prototipo de interfaz de usuario: este artefacto representa un ejemplo visual de la interfaz de usuario que se construye para explorar y/o validar el diseño de interfaz de usuario, permitiéndole al cliente verificar que el sistema satisfaga sus necesidades.

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

Artefactos del Arquitecto de Software.

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

Diagrama de Despliegue: muestra los nodos y sus relaciones. Un nodo es un conjunto de componentes. Se utiliza para reducir la complejidad de los diagramas de clases y componentes de un gran sistema. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos.

1.2.3 Herramientas

En las últimas décadas se ha trabajado para encontrar técnicas que permitan incrementar la productividad y el control de la calidad en cualquier proceso de elaboración de software, y hoy día, las herramientas CASE (Computer Aided Software Engineering) sustituyen al papel y al lápiz para convertir la actividad de desarrollar software en un proceso automatizado.

Entre las herramientas CASE para el modelado de artefactos existentes a nivel mundial están las herramientas Umbrello, MagicDraw, Rational Rose y Visual Paradigm que ha sido la definida para utilizar en el desarrollo del proyecto.

1.2.3.1 Herramienta CASE.

Visual Paradigm Suite 3.1

Es una potente herramienta CASE (Computer-Aided Software Engineering) para visualizar y diseñar elementos de software, para ello utiliza UML como lenguaje de notación, proporciona a los desarrolladores una plataforma que les permita diseñar un producto con calidad de una forma rápida. Facilita la interoperabilidad con otras herramientas CASE y se integra con los diferentes Software, así como con varios lenguajes de programación, entre los que se pueden mencionar: PHP, Java y C# aunque más indirectamente. Está disponible en varias ediciones: Enterprise, Professional, Community, Standard, Modeler y Personal.

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Está diseñada para dar soporte a arquitectos de sistemas, diseñadores, desarrolladores, analistas de procesos de negocio y modeladores de datos en los procesos de desarrollo de software. Permite modelar diferentes diagramas como: diagramas de clases, diagramas de interacción entre otros.

Entre sus principales características están:

- Modelación de procesos del negocio
- Administración de requerimientos
- Generación de la capa Objeto-Relacional
- Generación de código e ingeniería inversa, incluye 10 lenguajes de programación, entre ellos: Java, C++, .NET, PHP y XML.
- Generación de reportes PDF, MS Word, HTML
- Permite importar y exportar archivos XML
- Permite importar archivos desarrollados con Rational Rose
- Posee una interfaz de usuario amigable
- Es multiplataforma, disponible para los Sistemas Operativos Linux, Windows, y Mac OS [11]

Por todo lo antes planteado se decide utilizar como herramienta de modelado para el desarrollo de la aplicación Web, Visual Paradigm; además de que este presenta una interfaz de usuario de fácil uso y muy amigable que permite realizar los diagramas y artefactos que se generan durante el desarrollo del software.

1.2.4 Lenguaje de Modelado.

UML (v.2.0)

El lenguaje de modelado a utilizar es el **Lenguaje Unificado de Modelado (UML)** que se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de software. Dicho lenguaje es una notación unificada con la que se permite lograr un entendimiento que propicie el intercambio entre los usuarios y los desarrolladores. Se ha convertido en un estándar de la industria del software. El UML estándar está compuesto por tres partes: elementos de construcción (tales como clases, objetos,

mensajes), relaciones entre los bloques (tales como asociación, generalización), y diagramas (por ejemplo, diagrama de actividad).[12] (Ver anexo 5).

Los principales **beneficios de UML** son:

- Mejores tiempos totales de desarrollo.
- Permite la modelación de sistemas utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.

1.2.5 Herramienta para la Gestión de Requerimientos

La Ingeniería de Requerimientos cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.

Según la IEEE, un requerimiento es la condición o capacidad que debe poseer un sistema o un componente de un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto. En este sentido, la gestión de requerimientos comprende al conjunto de actividades que intentan entender las necesidades de los usuarios y traducirlas en afirmaciones precisas (no ambiguas), que se usarán en el desarrollo del sistema.

El proceso de gestión de requerimientos implica tres tipos de tareas:

- **Elicitación:** Se trabaja estrechamente con los usuarios a fin de conocer la problemática en detalle. La esencia de esta etapa consiste en extraer el conocimiento relevante del problema.

- **Especificación:** Es el proceso de documentación del comportamiento deseado del sistema. Una especificación puede ser vista como un acuerdo entre usuarios y desarrolladores del software.
- **Validación:** Permite asegurar que las especificaciones reflejan correctamente las intenciones de clientes y usuarios. (Ver Figura 3).

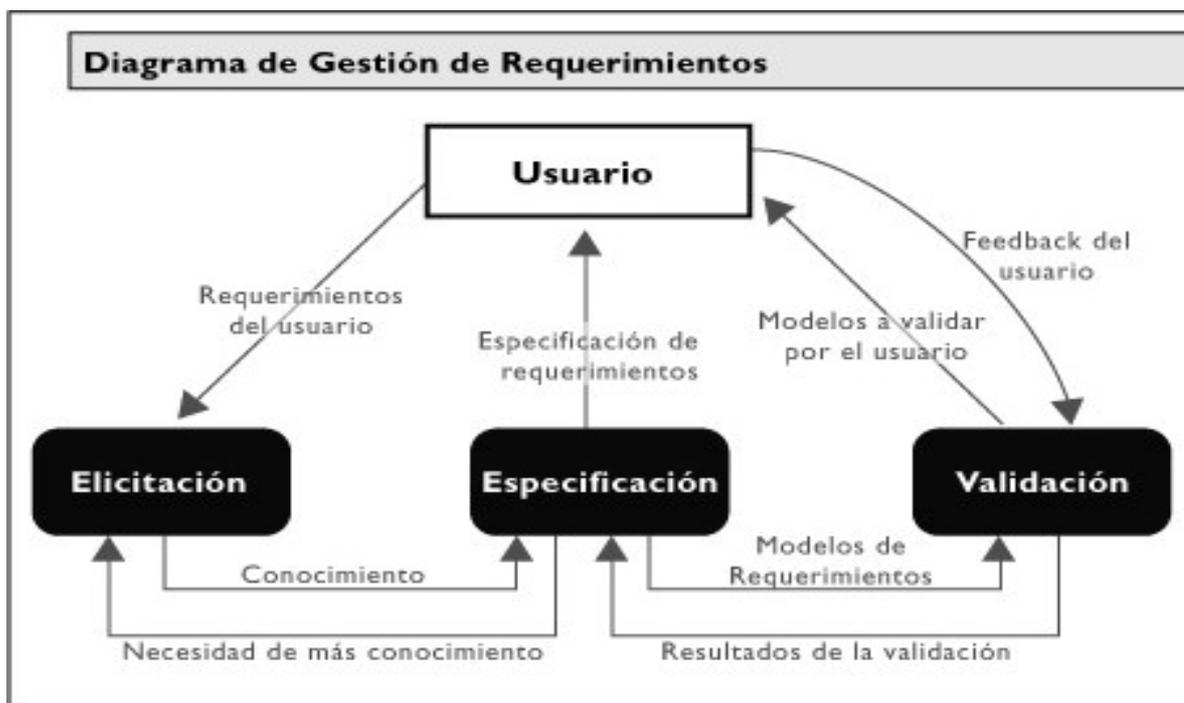


Fig 3. Diagrama de Gestión de Requerimientos.

Se espera que una especificación de requerimientos que fue aprobada por clientes y/o usuarios tenga al menos las siguientes características:

- Que contenga todos los requerimientos deseados.
- Que cada requerimiento solo tenga una interpretación posible (esto apunta a eliminar ambigüedades).
- Que el cumplimiento de cualquier requerimiento no provoque conflictos con el cumplimiento de otro requerimiento, es decir, que sea consistente.
- Que se definan prioridades.

Para la gestión de requisitos se utiliza la herramienta: Open Source Requirement Management Tool.

Open Source Requirement Management Tool (OSRMT v.1.5):

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

Ventajas y desventajas

La herramienta presenta las siguientes ventajas:

- La visualización de requisitos en forma jerárquica es intuitiva y fácil de manejar.
- Existen diversas distribuciones, tanto para un equipo en local como para un servidor de aplicaciones J2EE para permitir desarrollo colaborativo.
- Su licencia es GPL.
- Es un desarrollo basado en Java, por lo que es multiplataforma.
- Como herramienta open source de gestión de requisitos no tiene mucha competencia en cuanto a la funcionalidad ofrecida.
- Tiene una buena documentación pese a tratarse de una herramienta muy reciente.
- Existe un gran soporte para mantener la trazabilidad entre los documentos.
- Existen mecanismos que facilitan la importación y exportación de la información en XML

Las principales **desventajas** que se han observado son las siguientes:

- Es un desarrollo llevado a cabo por una persona individual, por lo que existe el riesgo de que no sea sostenible a lo largo del tiempo.
- No existe un soporte empresarial.
- No es posible generar automáticamente un documento de requisitos para entregar al cliente
- Algunas funcionalidades no han sido desarrolladas completamente y están a medias.
- La interfaz de usuario es en ocasiones lenta.

1.2.6 Lenguaje de Programación.

PHP 5

Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación, además es usada normalmente para la creación de páginas web dinámicas. PHP es un lenguaje interpretado.

PHP 5 mejorara los mecanismos de POO para solucionar las carencias de las anteriores versiones. Incluye todas las ventajas tales como:

- Soporte sólido y real para Programación Orientada a Objetos (u OOP) con PHP Data Objects.
- Mejoras de rendimiento.
- Mejor soporte a XML (XPath, DOM...).
- Soporte integrado para SOAP.
- Iteradores de datos.
- Excepciones de errores.

1.2.7 Framework de Desarrollo

Symfony 1.0.11

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

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones Web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.[13]

Symfony posee las siguientes características:

- Fácil de instalar y configurar en la mayoría de plataformas.

- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos
- Sigue las mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

1.3 Patrones.

Patrones de Diseño

Un patrón de diseño proporciona un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe estructuras comunes de la comunicación de los componentes que soluciona un problema de diseño dentro de un contexto particular, son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Soluciones basadas en la experiencia y que se ha demostrado que funcionan.

Ventajas

- Los patrones de diseño proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren de forma frecuente en el desarrollo.
- Por tanto, están basados en la recopilación del conocimiento de los expertos en desarrollo de software.
- Es una experiencia real, probada y que funciona y ayuda a no cometer los mismos errores.

1.3.1 Patrones de Casos de Usos

Patrón CRUD (Creating, Reading, Updating and Deleting / Crear, Buscar, Modificar y Eliminar): este patrón plantea la creación de un caso de uso, llamado "Información CRUD" o "Gestionar Información",

que modele todas las operaciones que se pueden realizar sobre una parte de información de un tipo determinado, ya sea crearla, leerla, actualizarla y eliminarla. Es importante destacar que este patrón se usa cuando todos los flujos contribuyen al mismo valor de negocio y son cortos y simples.

Aplicación

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

1.3.2 El patrón MVC

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Este estilo es aplicable a la realización del diseño según el framework Symfony que basa toda su estructura de diseño en este estilo.

Symfony está basado en este patrón clásico de arquitectura, que está formado por tres niveles:

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

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

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

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.[13]. (Ver Figura 4).

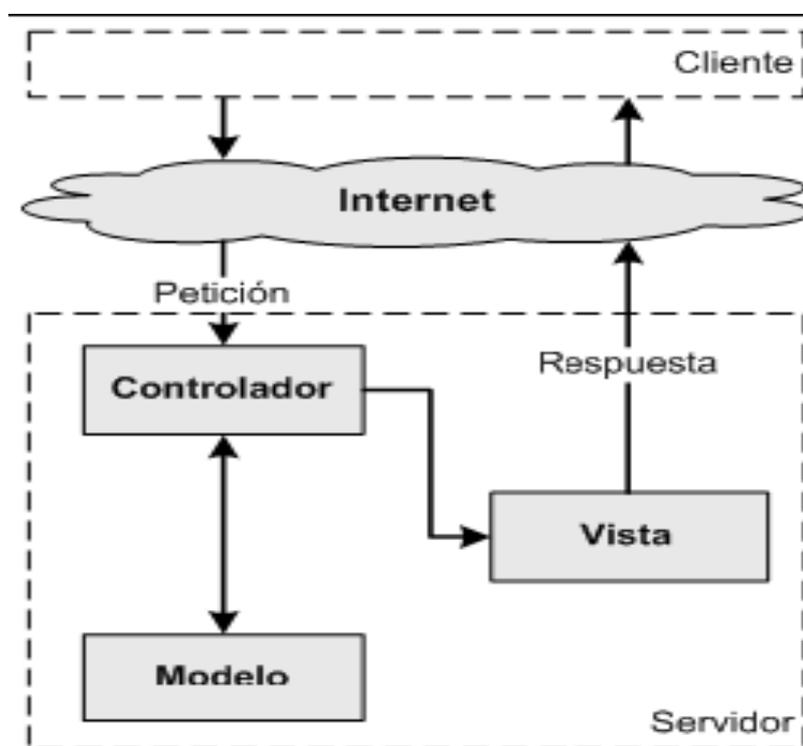


Fig 4. Funcionamiento del patrón MVC.

1.3.3 Patrones GRASP

Los patrones GRASP (por sus siglas en inglés, General Responsibility Assignment Software Patterns), patrones generales de software para asignar responsabilidades, describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones:

- **Bajo acoplamiento**

Solución: Asignar una responsabilidad para mantener bajo acoplamiento.

Surge ante la problemática de cómo dar soporte a una mínima dependencia y a un aumento en la reutilización. El acoplamiento mide qué tan fuerte está una clase conectada con otras (es decir, cuántas clases conoce y necesita). Una clase con bajo (o débil) acoplamiento no depende de "muchas otras" clases. Una clase con alto (o fuerte) acoplamiento recurre a muchas otras clases. Este tipo de clase no es conveniente, pues: cambios en las clases relacionadas ocasionan cambios en la clase local; son más difíciles de entender y de reutilizar.

- **Alta Cohesión**

Solución: Asignar una responsabilidad de modo que la cohesión siga siendo alta.

En la perspectiva del diseño orientado a objetos, la cohesión (o, más exactamente, la cohesión funcional) es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

Patrones GOF:

El grupo de GoF clasificaron los patrones en 3 grandes categorías basadas en su PROPÓSITO: creacionales, estructurales y de comportamiento.

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

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

En este trabajo sólo se utilizó el patrón Decorator que forma parte de los patrones estructurales.

Decorator: Patrón de diseño utilizado por Symfony, que es implementado por el layout, archivo denominado también plantilla global que almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde otro punto de vista, el layout decora la plantilla.

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

Aplicación:

El Decorator se debe usar para:

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

Ventajas del patrón Decorator

- Más flexible que la herencia: responsabilidades pueden añadirse y eliminarse en tiempo de ejecución.
- Diferentes decoradores pueden ser conectados a un mismo objeto.
- Reduce el número de propiedades en las clases de la parte alta de la jerarquía.

- Es simple añadir nuevos decoradores de forma independiente a las clases que extienden.
- Un objeto decorador tiene diferente OID (identificador único que proporciona el sistema) que el objeto que decora.

1.4 Vistas Arquitectónicas.

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

La arquitectura de un sistema consta de múltiples vistas, asociadas a diferentes dimensiones o perspectivas del sistema. En RUP, ésta se compone de 4+1 vistas: Vista Lógica, Vista de Procesos, Vista de Implementación, Vista de Despliegue y estas cuatro regidas por la Vista de Casos de Uso.

Vista de casos de uso.

Esta vista representa un subconjunto del artefacto Modelo de casos de uso y lista los casos de usos o escenarios del modelo de casos de uso más significativos, con las funcionalidades centrales del sistema. Si el sistema se hace extenso entonces se debería organizarse en paquetes, lo cual facilitaría la comprensión de la vista de casos de uso.

Vista lógica.

Esta vista representa un subconjunto del artefacto Modelo de diseño, la cual representa los elementos de diseño más importantes para la arquitectura del sistema. Ésta describe las clases más importantes, su organización en paquetes y subsistemas, y estos en capas. También describe las realizaciones de casos de uso más importantes como por ejemplo las que describen aspectos dinámicos del sistema.

Vista de procesos.

Esta vista suministra una base para la comprensión de la organización de los procesos de un sistema, ilustrados en el mapeo de las clases y subsistemas en procesos e hilos. Solo suele usarse cuando el sistema presenta procesos concurrentes o hilos.

Vista de despliegue.

Esta vista suministra una base para la comprensión de la distribución física de un sistema a través de nodos. Suele utilizarse cuando el sistema está distribuido. Esto incluye la asignación de tareas provenientes de la vista de procesos en los nodos.

Vista de implementación.

Esta vista describe la descomposición del software en capas y subsistemas de implementación. También provee una vista de la trazabilidad de los elementos de diseño de la vista lógica ahora para la implementación.

Esta contiene:

- Una enumeración de los subsistemas.
- Diagramas de componentes que ilustran la organización en capas y jerarquías de los subsistemas.
- Dependencia entre subsistemas.

1.5 Conclusiones

En este capítulo se profundizó en los conceptos y definiciones necesarias para lograr comprender los procesos que se realizan en el módulo del LBM. Para el desarrollo de la investigación se empleará RUP como metodología de desarrollo, UML como lenguaje de modelado, Visual Paradigm como herramienta case, PHP 5 como lenguaje de programación, OSRMT para la gestión de requerimientos y Symfony como framework de desarrollo.

CAPÍTULO 2: REQUERIMIENTOS

El flujo de trabajo **Requerimientos** del Proceso Unificado de Desarrollo (RUP), es muy importante para la elaboración de un software porque en él, los desarrolladores identifican y clasifican los requerimientos, encuentran actores y casos de uso, priorizan y detallan los casos de uso, se encargan de prototipar la interfaz de usuario, y estructurar el modelo de casos de uso del sistema.

En este capítulo se definen los requisitos funcionales, los no funcionales, los actores del sistema y el diagrama de casos de uso del sistema, la descripción textual de los casos de uso del sistema aprobados y los prototipos no funcionales de los mismos.

2.1 Actores del sistema a automatizar: Laboratorio de Biología Molecular (LBM)

Para el módulo de BM se define como actor del sistema al analista, a continuación se muestra su descripción:

NOMBRE DEL ACTOR	DESCRIPCIÓN
Analista	Realiza los ensayos en el laboratorio, registra todos los datos y resultados importantes, pide aprobación de ensayos al Jefe de Grupo y escribe los Procedimientos Patrones de Operación de los diferentes ensayos.

Tabla 1. Descripción de Actores del Sistema

2.2 Requerimientos Funcionales y No Funcionales del Sistema.

Los requerimientos funcionales son condiciones o capacidades que el sistema debe cumplir, los cuales en la fase de construcción deben ser posibles de probar o verificar.

En este módulo se identificaron un total de 31 requerimientos como parte del proyecto de los cuales se presentan en este documento sólo 10 de ellos. Para ver los demás remitirse al Expediente de Proyecto.

2.2.1 Requerimientos funcionales: módulo de BM.

1. Gestionar Solicitud de destrucción de bancos de células (SIC-0840):
 - 1.1 Crear nueva Solicitud de destrucción de bancos de células
 - 1.2 Modificar datos del registro de Solicitud de destrucción de bancos de células
 - 1.2.1 Registrar traza
 - 1.3 Buscar y visualizar el registro de Solicitud de destrucción de bancos de células
 - 1.4 Imprimir Solicitud de destrucción de bancos de células

2. Gestionar registro de Determinación del marcador de selección (SIC-0152):
 - 2.1 Crear nuevo registro de Determinación del marcador de selección
 - 2.2 Modificar datos del registro de Determinación del marcador de selección
 - 2.2.1 Registrar traza
 - 2.3 Buscar y visualizar el registro de Determinación del marcador de selección
 - 2.4 Imprimir el registro de Determinación del marcador de selección

3. Gestionar registro de Descargo para Banco de Células (SIC-0190):
 - 3.1 Crear nuevo registro de Descargo para Banco de Células
 - 3.2 Modificar datos del registro de Descargo para Banco de Células
 - 3.2.1 Registrar traza
 - 3.3 Buscar y visualizar el registro de Descargo para Banco de Células
 - 3.4 Imprimir el registro de Descargo para Banco de Células.

4. Gestionar registro de Temperatura (SIC-0125R):
 - 4.1 Crear nuevo registro de Temperatura
 - 4.2 Modificar datos del registro de Temperatura
 - 4.2.1 Registrar traza
 - 4.3 Buscar y visualizar el registro de Temperatura
 - 4.4 Imprimir el registro de Temperatura

5. Gestionar registro de Muestreo de los bancos de Células (SIC-0951):
 - 5.1 Crear nuevo registro de Muestreo de los bancos de células
 - 5.2 Modificar datos del registro de Muestreo de los bancos de células
 - 5.2.1 Registrar traza
 - 5.3 Buscar y visualizar el registro de Muestreo de los bancos de células
 - 5.4 Imprimir el registro de Muestreo de los bancos de células.

6. Gestionar Cronograma de chequeo de bancos de células (SIC-0130):
 - 6.1 Crear nuevo Cronograma de chequeo de bancos de células.
 - 6.2 Modificar datos del Cronograma de chequeo de bancos de células
 - 6.2.1 Registrar traza
 - 6.3 Buscar y visualizar el Cronograma de chequeo de bancos de células
 - 6.4 Imprimir el Cronograma de chequeo de bancos de células

7. Gestionar Certificado de Liberación de los bancos de Células transformados (SIC-0229):
 - 7.1 Crear nuevo Certificado de Liberación de los bancos de Células transformados
 - 7.2 Modificar datos del Certificado de Liberación de los bancos de Células transformados
 - 7.2.1 Registrar traza
 - 7.3 Buscar y visualizar el Certificado de Liberación de los bancos de Células transformados
 - 7.4 Imprimir el Certificado de Liberación de los bancos de Células transformados.

8. Gestionar registro de Determinación de la Estabilidad Plasmídica (SIC-0149):
 - 8.1 Crear nuevo registro de Determinación de la Estabilidad Plasmídica
 - 8.2 Modificar datos del registro de Determinación de la Estabilidad Plasmídica
 - 8.2.1 Registrar traza
 - 8.3 Buscar y visualizar el registro de Determinación de la Estabilidad Plasmídica
 - 8.4 Imprimir el registro de Determinación de la Estabilidad Plasmídica

9. Gestionar registro de Destrucción de Banco de Células (SIC-0841):
 - 9.1 Crear nuevo registro de Destrucción de Banco de Células
 - 9.2 Modificar datos del registro de Destrucción de Banco de Células
 - 9.2.1 Registrar traza
 - 9.3 Buscar y visualizar el registro de Destrucción de Banco de Células
 - 9.4 Imprimir el registro de Destrucción de Banco de Células

10. Gestionar Informe de análisis de los bancos de células (SIC-0950):
 - 10.1 Crear nuevo Informe de análisis de los bancos de células
 - 10.2 Modificar datos del Informe de análisis de los bancos de células
 - 10.2.1 Registrar traza
 - 10.3 Buscar y visualizar el Informe de análisis de los bancos de células

10.4 Imprimir el Informe de análisis de los bancos de células

2.2.2 Requerimientos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Entre los requerimientos no funcionales que se definieron en el proyecto se pueden mencionar los siguientes:

- ***Apariencia o interfaz externa.***

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

- ***Usabilidad.***

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

- ***Rendimiento.***

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

- ***SopORTE.***

Se requiere de la instalación de un servidor Web que interprete código PHP 5 o superior. Las computadoras clientes requerirán de un navegador capaz de interpretar código Java Script.

- ***Portabilidad.***

El sistema podrá ser usado sobre los sistemas operativos Windows 98, sus versiones superiores y Linux.

- ***Seguridad.***

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

El sistema debe contar con protección contra acciones no autorizadas o que puedan afectar la

integridad de los datos; y con un registro de trazas de los documentos modificados por los usuarios, para garantizar el control de las operaciones de este tipo. Se podrá acceder a algunas funcionalidades del sistema desde cualquier computadora personal que esté fuera del CIGB.

- **Políticos-culturales.**

El idioma que se empleará en la aplicación será el español. El sistema tendrá logotipos e imágenes en correspondencia con el carácter científico y profesional del CIGB. Algún cambio que se quiera realizar en la aplicación, será tramitado por la dirección del proyecto por parte del CIGB y canalizado por los directivos de Producción de la UCI.

- **Legales.**

El sistema se dará a conocer a todos los trabajadores de la Dirección de Calidad como la herramienta para gestionar la información de cada uno de los grupos y laboratorios. Se estará usando para el desarrollo de la aplicación herramientas de software libre con licencia GNU/GPL.

- **Confabilidad.**

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

Podrán acceder a visualizar ciertas informaciones, directivos de otras áreas, con previa consulta a la dirección del proyecto y a los desarrolladores de la aplicación. El sistema validará la entrada de datos para evitar entradas inadecuadas.

- **Software.**

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

Para el servidor de la aplicación los Sistemas Operativos recomendados son: Windows Server 2003 o superior o Linux. Se debe instalar un servidor Web Apache 1.3 o superior.

- **Hardware.**

Se deberá contar con impresora y escáner en las computadoras clientes que interactúen con la aplicación. Se deberán incluir más computadoras personales en los grupos y laboratorios de la Dirección de Calidad, las que deben cumplir con las siguientes características: se necesita como mínimo 1 GB de espacio libre en el disco duro, microprocesador Pentium IV a 2.0 GHz o superior y 1.0 GB de RAM o superior.

El servidor debe tener las siguientes características: capacidad de disco duro de 120.0 GB o superior, microprocesador Pentium IV superior a 2.0 GHz y como mínimo 1.0 GB de RAM.

- **Restricciones en el diseño**

La lógica de presentación constituirá una capa independiente de la lógica de negocio, centrandó su función en la interfaz de usuario y validaciones de los datos de entrada. Se utilizarán herramientas de desarrollo que garanticen la calidad de todo el ciclo de desarrollo del producto.

2.3 Modelo de Casos de Uso del Sistema

A continuación se muestra el diagrama de Casos de Uso del Sistema para la representación de 10 requerimientos funcionales definidos en el trabajo.

Diagrama de casos de uso del sistema a automatizar: LBM.

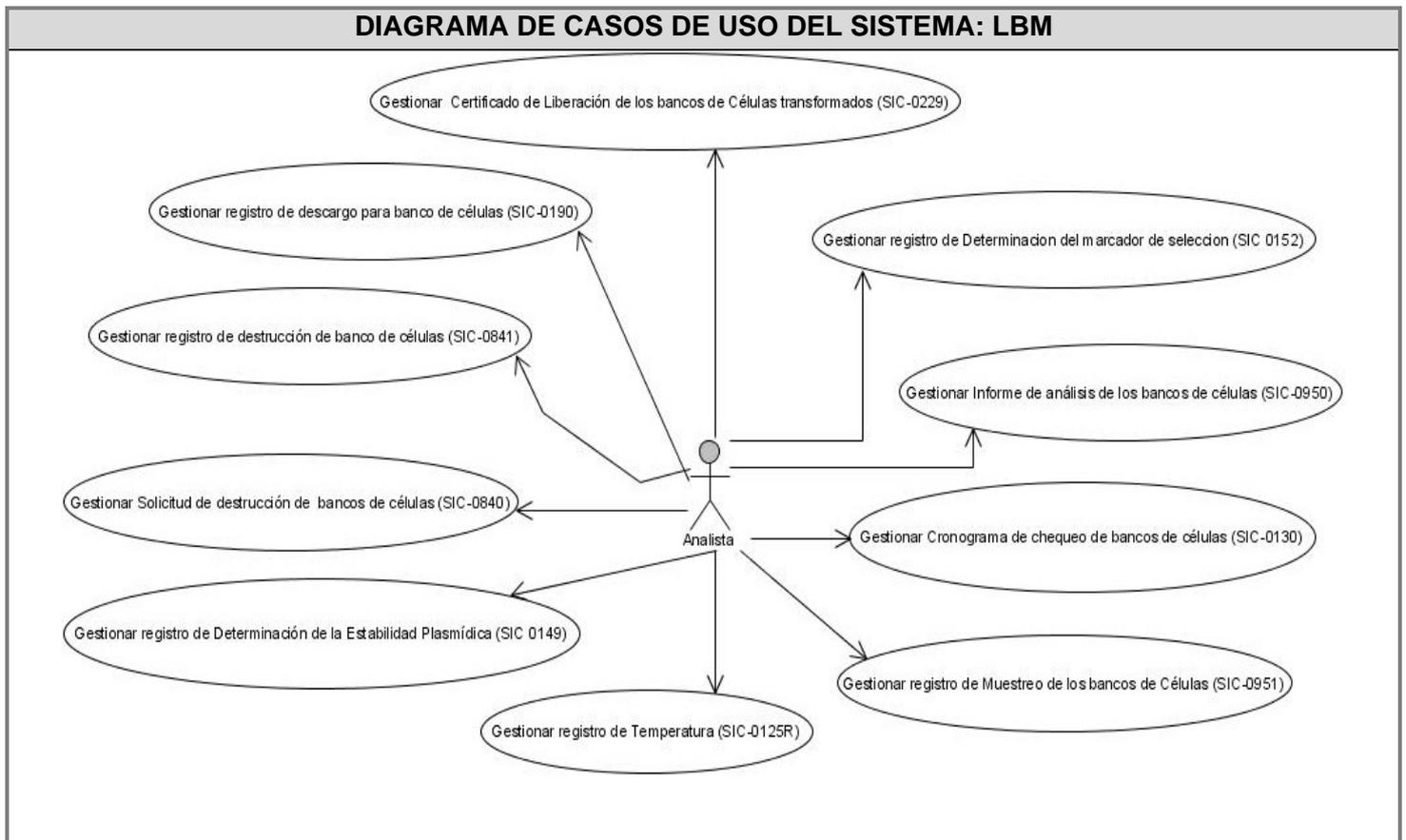


Fig 5. Diagrama de Casos de Uso del Sistema: LBM.

2.4 Descripción de los casos de uso del módulo: BM.

En este epígrafe se presentan las descripciones textuales resumidas de 10 casos de uso y sólo la descripción textual extendida del caso de uso “*Gestionar Solicitud de destrucción de bancos de células (SIC-0840)*”, para ver las demás descripciones remitirse al Expediente del Proyecto.

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

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> <ul style="list-style-type: none"> ✓ Crear nueva Solicitud de destrucción de bancos de células. ✓ Modificar la Solicitud de destrucción de bancos de células. ✓ Buscar y visualizar la Solicitud de destrucción 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	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
<p>1. El Analista, quiere realizar una de las siguientes operaciones:</p> <ul style="list-style-type: none"> a) Crear una nueva Solicitud de destrucción de bancos de células (SIC-0840). b) Buscar y visualizar la Solicitud de destrucción de bancos de células (SIC-0840). c) Modificar datos del SIC-0840. 	<p>2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> 1. Si el Analista realiza (a), crea una nueva Solicitud de destrucción de bancos de células (SIC-0840) ir a la Sección “Crear nuevo registro” 2. Si el Analista realiza (b), busca y visualiza la Solicitud de destrucción de bancos de células (SIC-0840) ir a la Sección “Buscar y

	<p>visualizar registro”.</p> <p>3. 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>
Sección “Crear nuevo registro”	
Acción del actor	Respuesta del Sistema
	1. El sistema muestra la interfaz correspondiente a la creación de la nueva registro de Solicitud de destrucción de bancos de células (SIC-0840).
	2. El sistema genera el campo <u>Folio</u> consecutivamente.
	3. El sistema genera automáticamente los campos: <ul style="list-style-type: none"> ✓ <u>Laboratorio</u> (con las siglas: BM). ✓ <u>Área</u> (con la palabra: Calidad)
<p>4. El Analista edita los siguientes campos:</p> <ul style="list-style-type: none"> ✓ <u>Código del banco a destruir</u> ✓ <u>Motivo de destrucción</u> ✓ <u>Número de viales destruir</u> ✓ <u>Fecha destrucción</u> ✓ <u>Hora de destrucción</u> ✓ <u>Cargo autoriza</u> ✓ <u>Cargo responsable</u> ✓ <u>Fecha responsable destrucción</u> ✓ <u>Responsable destrucción</u> 	
	<p>5. El sistema genera automáticamente el campo:</p> <ul style="list-style-type: none"> ✓ <u>Autoriza destrucción</u> <p>Con el nombre y los apellidos de la persona autenticada.</p>

CAPÍTULO 2: REQUERIMIENTOS

	6. El sistema actualiza el campo y <i>fecha autoriza destrucción</i> 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: <ul style="list-style-type: none"> ✓ Imprimir registro. ✓ Modificar registro. 	11. El sistema en correspondencia a lo que haya escogido el analista realiza lo siguientes: <ul style="list-style-type: none"> ✓ 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	(Ver Anexo 6)
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: <ul style="list-style-type: none"> ✓ Imprimir registro. ✓ Modificar registro. 	8. El sistema en correspondencia con la operación que escogió el analista realiza lo siguiente: <ul style="list-style-type: none"> ✓ 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	(Ver Anexo 6.1)
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: <ul style="list-style-type: none"> ✓ <u>Código del banco a destruir</u> ✓ <u>No viales destruir</u> 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. <ul style="list-style-type: none"> ✓ <u>Código del banco a destruir</u> ✓ <u>No viales destruir</u> 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: <ul style="list-style-type: none"> ✓ Imprimir registro. ✓ Modificar registro. 	9. El sistema en correspondencia con la operación escogida por el analista realiza lo siguiente: <ul style="list-style-type: none"> ✓ 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"
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	(Ver Anexo 6.2)

Tabla 2. Gestionar Solicitud de destrucción de bancos de células (SIC-0840).

2.4.2 Gestionar registro de Determinación del marcador de selección (SIC-0152).

Nombre del Caso de Uso	Gestionar registro de Determinación del marcador de selección (SIC-0152)
Actores	Analista (Inicia)
Propósito	Registrar los datos en el registro de Determinación del marcador de selección (SIC-0152), 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 Determinación del marcador de selección:</p> <ul style="list-style-type: none"> ✓ Crear nuevo registro de Determinación del marcador de selección. ✓ Modificar el registro de Determinación del marcador de selección. ✓ Buscar y visualizar el registro de Determinación del marcador de selección <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 de Determinación del marcador de selección (SIC-0152), modifica, imprime, busca y visualiza datos según los criterios indicados por el analista.
Requerimientos especiales	-

Tabla 3. Gestionar registro de Determinación del marcador de selección (SIC-0152).

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

Nombre del Caso de Uso	Gestionar registro de Descargo para Banco de Células (SIC-0190)
Actores	Analista (Inicia)
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	El caso de uso se inicia cuando el analista va a realizar alguna de

	<p>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	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 Descargo para Banco de Células (SIC-0190), modifica, imprime, busca y visualiza datos según los criterios indicados por el analista.
Requerimientos especiales	-

Tabla 4. Gestionar registro de Descargo para Banco de Células (SIC-0190).

2.4.4 Gestionar registro de Temperatura (SIC-0125R).

Nombre del Caso de Uso	Gestionar registro de Temperatura (SIC-0125R)
Actores	Analista (Inicia)
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 Descargo para Banco de Células.

	<ul style="list-style-type: none"> ✓ 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	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 registro de Temperatura (SIC-0125R), modifica, imprime, busca y visualiza datos según los criterios indicados por el analista.
Requerimientos especiales	-

Tabla 5. Gestionar registro de Temperatura (SIC-0125R).

2.4.5 Gestionar registro de Muestreo de los bancos de Células (SIC-0951).

Nombre del Caso de Uso	Gestionar registro de Muestreo de los bancos de Células (SIC-0951)
Actores	Analista (Inicia)
Propósito	Registrar los datos del registro de Muestreo de los bancos de células (SIC-0951), 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 Muestreo de los bancos de Células:</p> <ul style="list-style-type: none"> ✓ Crear nuevo registro de Muestreo de los bancos de Células. ✓ Modificar el registro de Muestreo de los bancos de Células. ✓ Buscar y visualizar el registro de Muestreo 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	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 registro de Muestreo de los bancos de células (SIC-0951), modifica, imprime, busca, visualiza datos según los criterios indicados por el analista.
Requerimientos especiales	-

Tabla 6. Gestionar registro de Muestreo de los bancos de Células (SIC-0951).

2.4.6 Gestionar Cronograma de chequeo de bancos de células (SIC-0130).

Nombre del Caso de Uso	Gestionar Cronograma de chequeo de bancos de células (SIC-0130)
Actores	Analista (Inicia)
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 el Cronograma de chequeo de bancos de células. ✓ Buscar y visualizar el 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</p>

	<p>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 Cronograma de chequeo de bancos de células (SIC-0130), modifica, imprime, busca y visualiza datos según los criterios indicados por el analista.
Requerimientos especiales	-

Tabla 7. Gestionar Cronograma de chequeo de bancos de células (SIC-0130).

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

Nombre del Caso de Uso	Gestionar Certificado de Liberación de los bancos de Células transformados (SIC-0229)
Actores	Analista (Inicia)
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 el Certificado de Liberación de los bancos de Células transformados. ✓ Buscar y visualizar el 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</p>

	las siguientes operaciones concernientes a la gestión.
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 Certificado de Liberación de los bancos de Células transformados (SIC-0229), modifica, imprime, busca y visualiza datos según los criterios indicados por el analista.
Requerimientos especiales	-

Tabla 8. Gestionar Certificado de Liberación de los bancos de Células transformados (SIC-0229).

2.4.8 Gestionar registro de Determinación de la Estabilidad Plasmídica (SIC-0149).

Nombre del Caso de Uso	Gestionar registro de Determinación de la Estabilidad Plasmídica (SIC-0149)
Actores	Analista (Inicia)
Propósito	Registrar los datos del Gestionar registro de Determinación de la Estabilidad Plasmídica (SIC-0149), 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 Determinación de la Estabilidad Plasmídica:</p> <ul style="list-style-type: none"> ✓ Crear nuevo registro de Determinación de la Estabilidad Plasmídica. ✓ Modificar registro de Determinación de la Estabilidad Plasmídica. ✓ Buscar y visualizar registro de Determinación de la Estabilidad Plasmídica. <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	R8.1, R8.2, R8.2.1, R8.3, R8.4
Precondiciones	Que el Analista se haya autenticado en el sistema.
Poscondiciones	El sistema crea un nuevo registro de Determinación de la Estabilidad Plasmídica (SIC-0149), modifica, imprime, busca, visualiza datos según el interés que tenga el analista.
Requerimientos especiales	-

Tabla 9. Gestionar registro de Determinación de la Estabilidad Plasmídica (SIC-0149).

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

Nombre del Caso de Uso	Gestionar registro de Destrucción de Banco de Células (SIC-0841)
Actores	Analista (Inicia)
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> <ul style="list-style-type: none"> ✓ Crear nuevo registro de Destrucción de Banco de Células. ✓ Modificar el registro de Destrucción de Banco de Células. ✓ Buscar y visualizar el registro de Destrucción de 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	R9.1, R9.2, R9.2.1, R9.3, R9.4
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	-

Tabla 10. Gestionar registro de Destrucción de Banco de Células (SIC-0841).

2.4.10 Gestionar Informe de Análisis de los bancos de células (SIC-0950).

Nombre del Caso de Uso	Gestionar Informe de Análisis de los bancos de células (SIC-0950)
Actores	Analista (Inicia)
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 el 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	R10.1, R10.2, R10.2.1, R10.3, R10.4
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 los criterios indicados por el analista.
Requerimientos especiales	-

Tabla 11. Gestionar Informe de Análisis de los bancos de células (SIC-0950).

2.5 Conclusiones.

En este capítulo se identificaron los requerimientos funcionales y no funcionales del sistema, se identificó y describió al actor del sistema y los 10 casos de uso del sistema, quedando elaborados y aprobados los prototipos de los casos de uso descritos. Se realizó el diagrama de casos de uso del sistema, donde se muestra la relación entre los casos de uso y los actores del sistema.

CAPÍTULO 3: DISEÑO DEL SISTEMA

En este capítulo se estudiarán los casos de uso del sistema para diseñar las clases que se implementarán, se representan los diagramas de las clases diseñadas con sus relaciones, agrupados en paquetes de diseño, con la elaboración además de los diagramas de interacción y la validación de los mismos. Se realiza además el mapa de navegación, se hace referencia a la arquitectura del sistema a través de las Vista Lógica y de Despliegue.

3.1 Propósitos del diseño.

La fase de diseño mantiene la estructura definida en el análisis para el sistema y brindando un soporte para la implementación de la arquitectura del sistema como un todo. O sea, hasta ahora se tienen las bases para formar el esqueleto del sistema y a partir de éste todo lo demás de forma iterativa e incremental hasta la culminación del software.

Se pueden definir como **propósitos del diseño**:

- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.
- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
- Contribución a una arquitectura estable y sólida, y crear un plano del modelo de implementación.
- Capturar las interfaces entre los subsistemas antes en el ciclo de vida del software, lo cual es muy útil cuando utilizamos interfaces como elementos de sincronización entre diferentes equipos de desarrollo.
- Ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

Para modelar el diseño se utilizan las clases del diseño que son una construcción similar en la implementación del sistema:

- Adaptar el diseño para que se corresponda con el entorno de implementación, diseñando sus funcionalidades.

- Los métodos de una clase del diseño tienen correspondencia directa con el correspondiente método en la implementación de las clases.

3.2 Principios de diseño.

UML posee una extensión para el modelado de aplicaciones Web, propuesta por Conallen [14], dicha extensión es usada para el diseño de las clases. Los estereotipos que usa esta extensión son:



<<Server Page>> Representa la página Web que tiene código que se ejecuta en el servidor. Este código interactúa con recursos en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página.

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



<<Client Page>> Una instancia de Página Cliente es una página Web, con formato HTML; mezcla de datos, presentación y lógica. Son interpretadas por el browser. Cada página cliente solo puede ser construida por una página servidor.



<<Form>> Grupo de elementos de entrada que son parte de una página cliente. Se relaciona directamente con la etiqueta de igual nombre del HTML. Sus atributos son los elementos de entrada del formulario (Text Field, Text Area, Button, Label, Radio Button, Radio Group, Select, Check Box y Hidden Fields).

3.3 Referencia a la Arquitectura del Sistema.

3.3.1 Vista lógica.

En la vista lógica se muestran los componentes principales del diseño y sus relaciones, de forma independiente de los detalles técnicos y de cómo la funcionalidad será implementada en la plataforma de ejecución. Los modelos de diseño de la aplicación son vistas lógicas del modelo funcional y

describen la solución. Se describe la solución en términos de paquetes y clases del diseño; dentro de esta vista se describe la realización de los casos de uso, subsistemas, paquetes y clases de los casos de uso más significativos arquitectónicamente

Para estructurar la vista del sistema, no se identificaron en este módulo (BM) subsistemas funcionales, se organizó por paquetes a partir de los casos de uso arquitectónicamente significativos, que constituyen la totalidad de los casos de uso identificados para este módulo. Existe un paquete formado por su diagrama de clases. En este trabajo se muestra la vista lógica de uno de los 3 paquetes identificados, para ver la vista completa remitirse al Expediente de Proyecto.

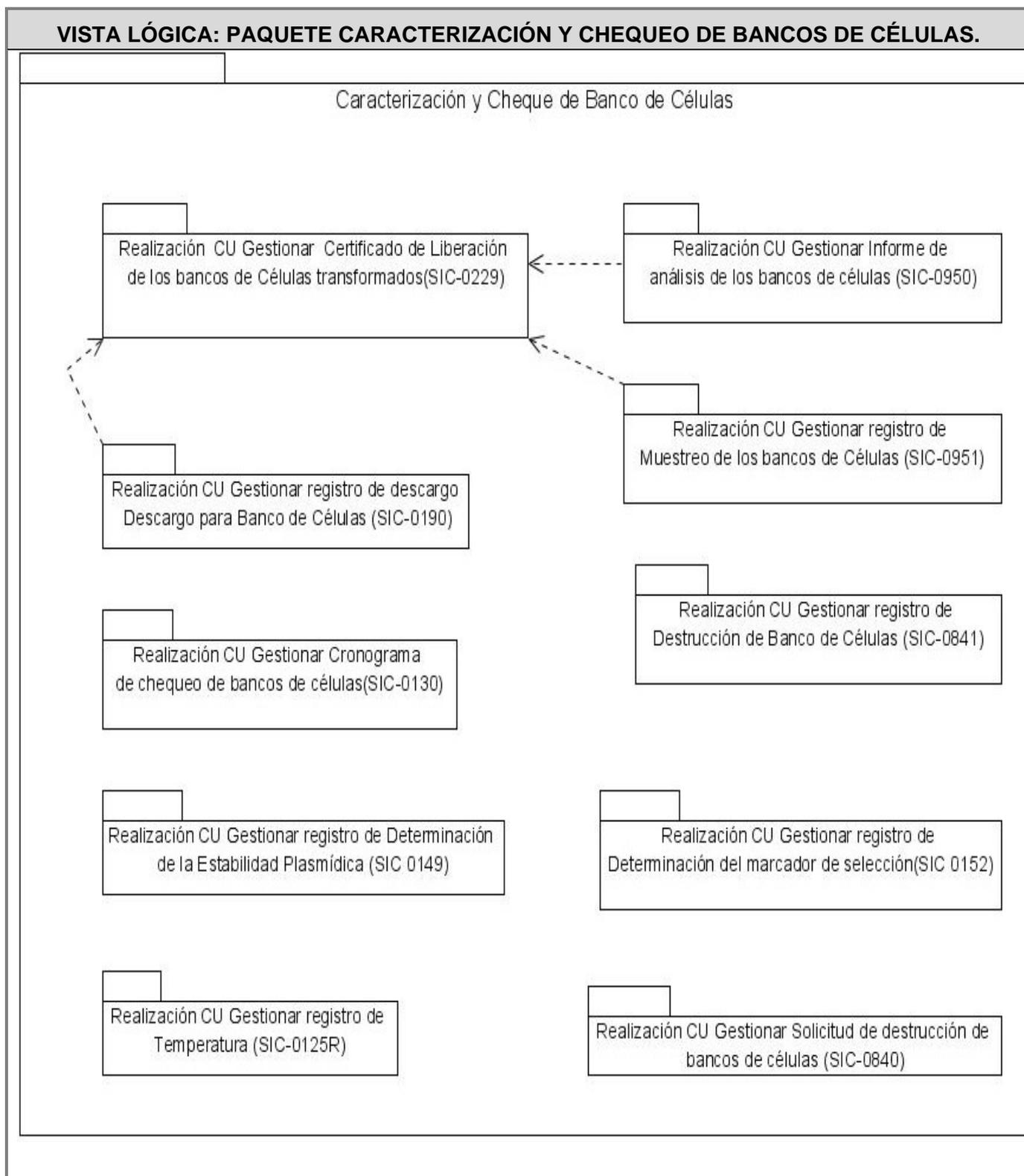


Fig 6. Vista Lógica: Paquete Caracterización y Chequeo de Bancos de Células.

Cada paquete contiene la realización de un caso de uso. Por ejemplo el paquete relacionado con el caso de uso Gestionar Solicitud de destrucción de bancos de células (SIC-0840), contiene la realización del mismo.

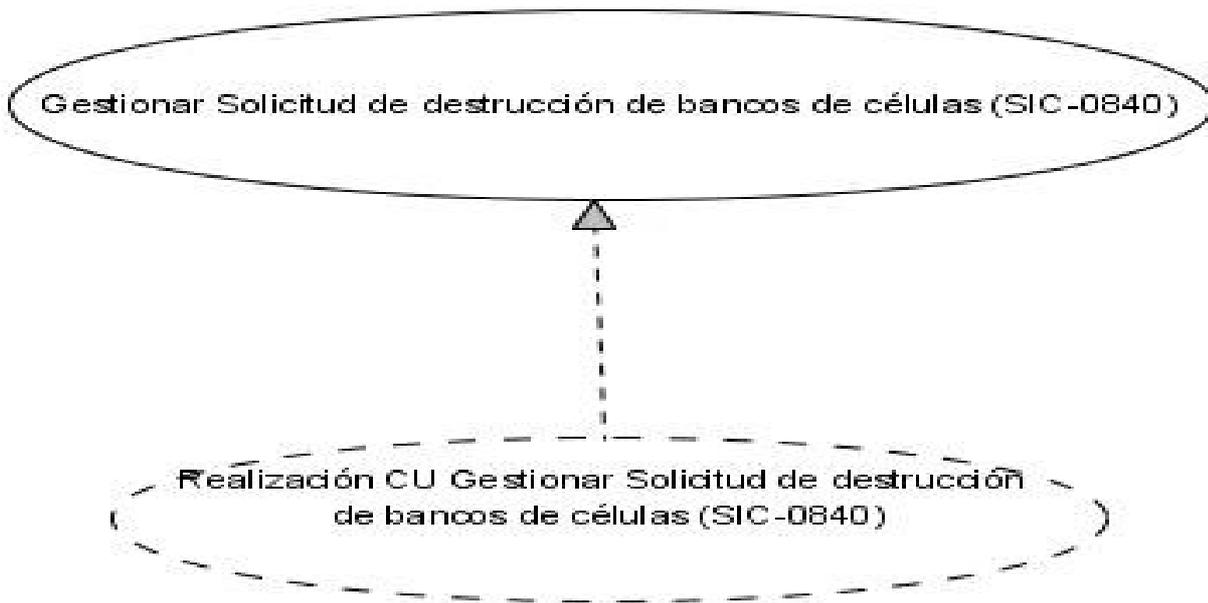


Fig 7. Realización del CU Gestionar Solicitud de destrucción de bancos de células (SIC_0840).

La realización del caso de uso contiene el diagrama de clases del diseño del mismo, así como los diagramas de secuencia por escenario. En los diagramas de clases del diseño se aprecia claramente el estilo arquitectónico Modelo Vista Controlador, en la siguiente figura se muestra el diagrama de clases del diseño del caso de uso Gestionar Solicitud de destrucción de bancos de células (SIC-0840) donde se demuestra lo anterior.

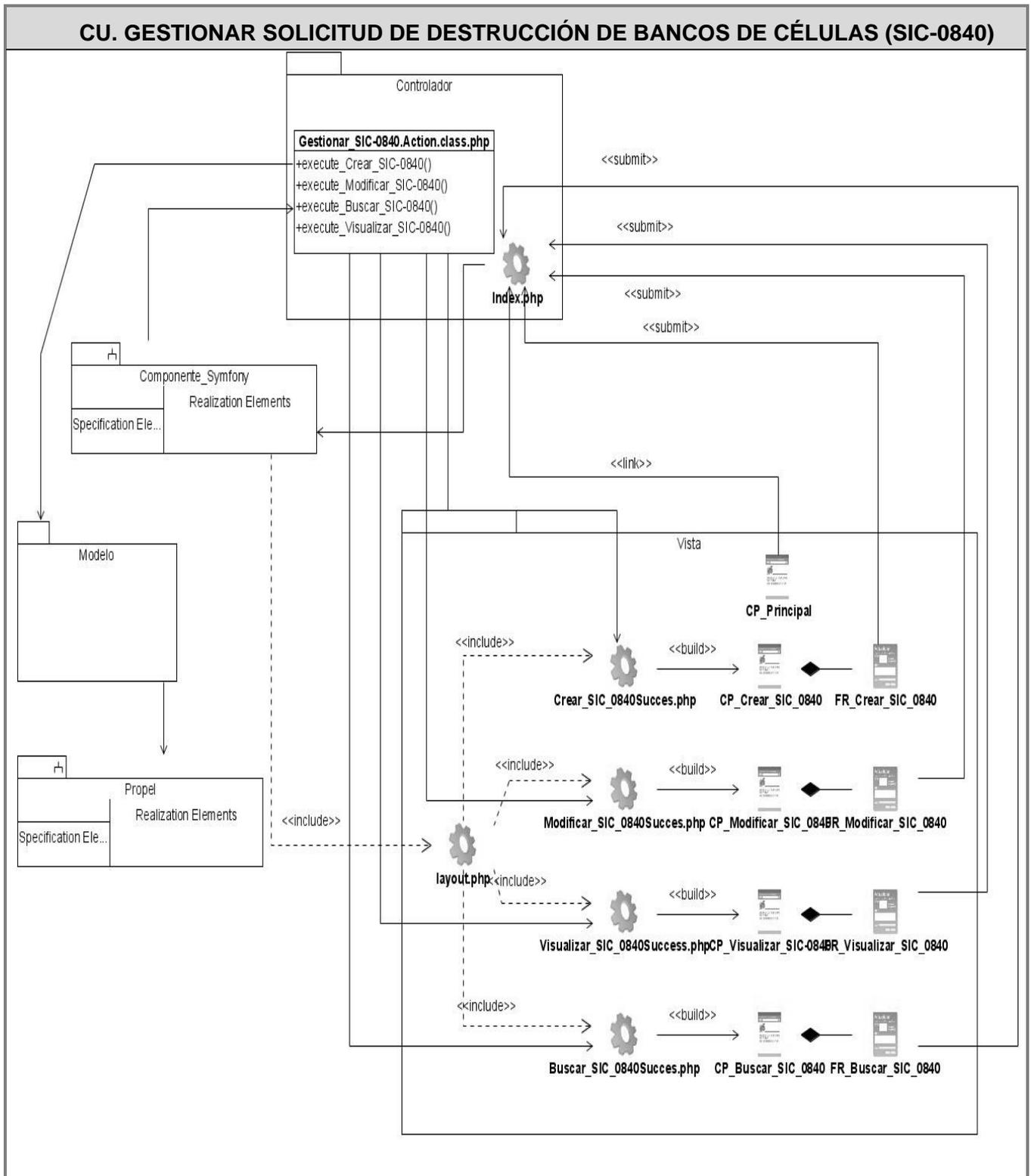


Fig 8. Gestionar Solicitud de Destrucción de Bancos de Células (SIC-0840).

3.3.2 Vista de despliegue.

Esta vista muestra la distribución física de cada uno de los recursos necesarios para la instalación del software desarrollado, representa un grafo de nodos unidos por conexiones de comunicación. Para el almacenamiento de los datos de la aplicación se utiliza un servidor de Base de Datos representado como un nodo y para lograr la conexión del sistema con la base de datos se utiliza TCP/IP + SSL como protocolo de comunicación. Además debe existir una conexión mediante protocolos de comunicación seguros HTTPS que permitan la integridad de la información que se manipula, así como dispositivos de impresión y escaneo.

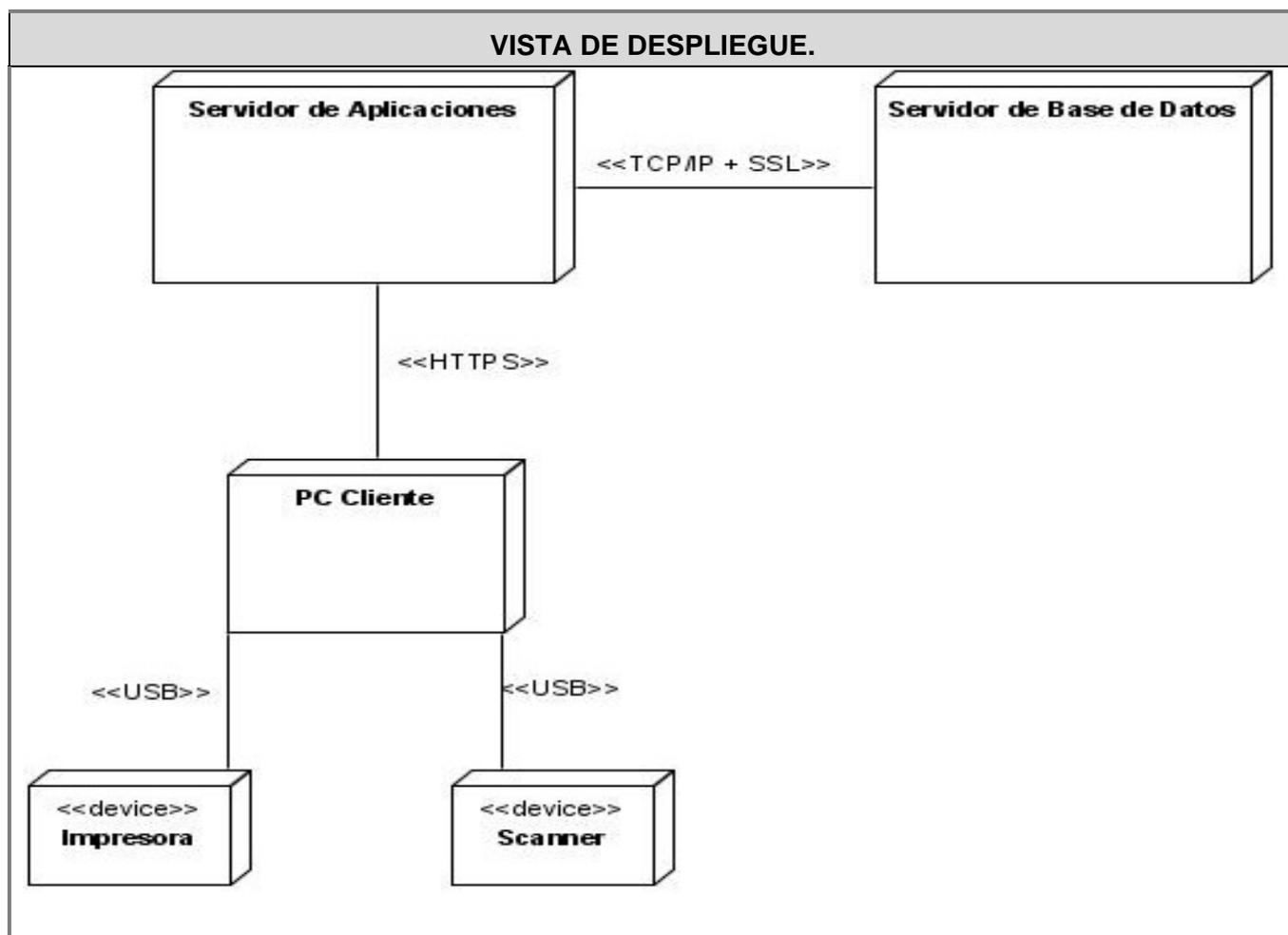


Fig 9. Vista de Despliegue.

3.4 Mapa de Navegación.

Un mapa de navegación es la representación gráfica de la organización de la información de una estructura web. Expresa todas las relaciones de jerarquía, secuencia y permite elaborar escenarios de comportamiento de los usuarios. También grafica, de modo que todos los profesionales participantes en un proyecto lo tengan claro, diferencias entre páginas dinámicas, administrables o estáticas. El mapa de navegación hace ver más fácil como muchos clics llevarán al usuario hacia una pantalla específica.

Para la creación de mapas de navegación para aplicaciones Web aún se carece de software mediante los cuales se logre dar una estructura completa al funcionamiento de los mismos. Microsoft Frontpage ofrece la posibilidad de ver y almacenar una representación gráfica de nodos y enlaces entre ellos, pero se basa en las conexiones físicas, no en la representación de la estructura de la información. Aparte, pueden encontrarse algunos webbots (mecanismos que invocan diversas funciones interactivas integradas en FrontPage), que son capaces de representar una estructura de carpetas y directorios que semeja la distribución física de un sitio, pero no son verdaderas herramientas de creación de mapas del web propio. En el proyecto después de un análisis profundo se decidió realizar el mapa en la herramienta CASE Visual Paradigm (Suite 3.1) debido a que brinda una buena representación dada por sus estereotipos y es una excelente forma de complacer a los usuarios.

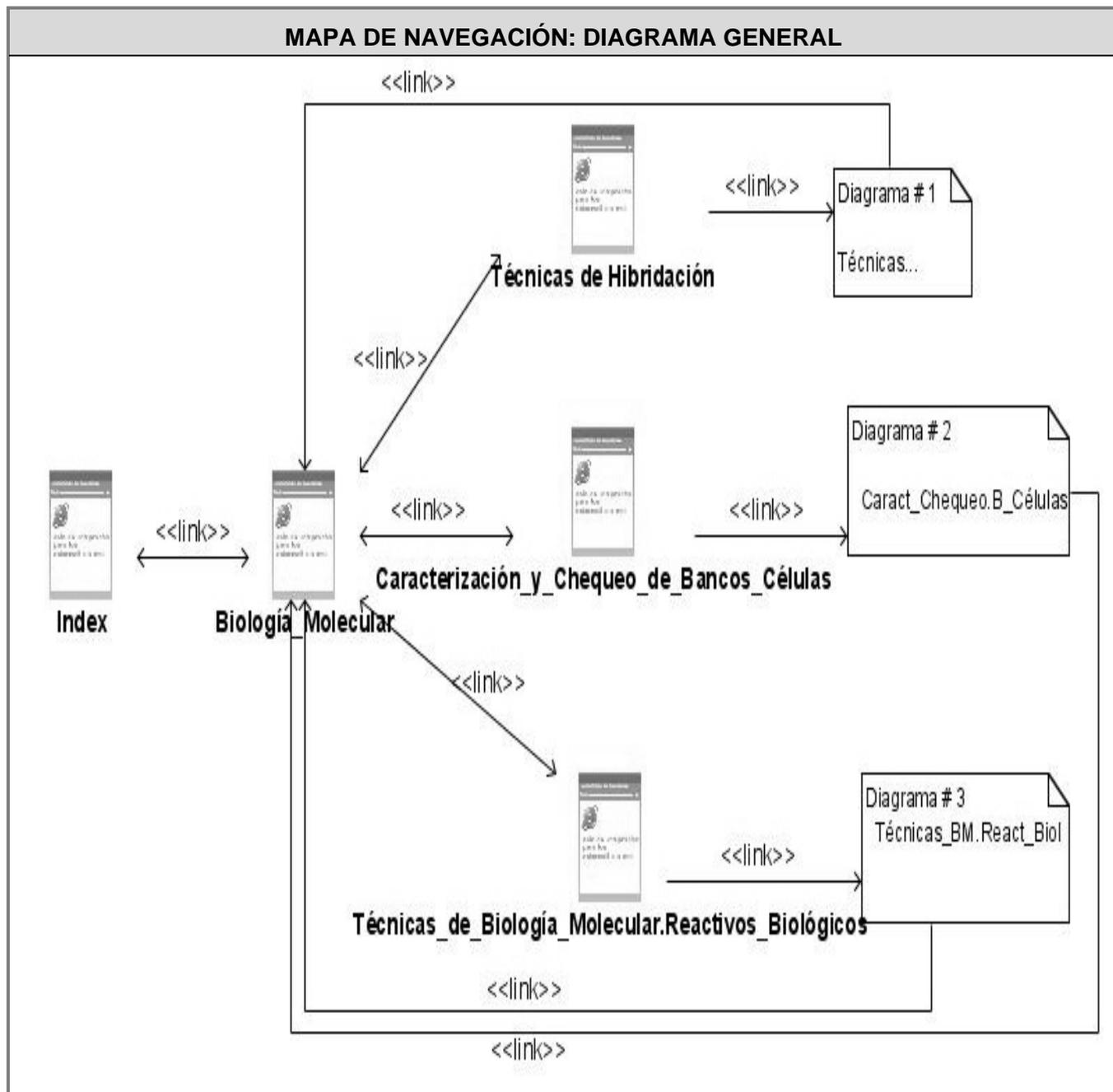


Fig 10. Mapa de navegación: Diagrama General.

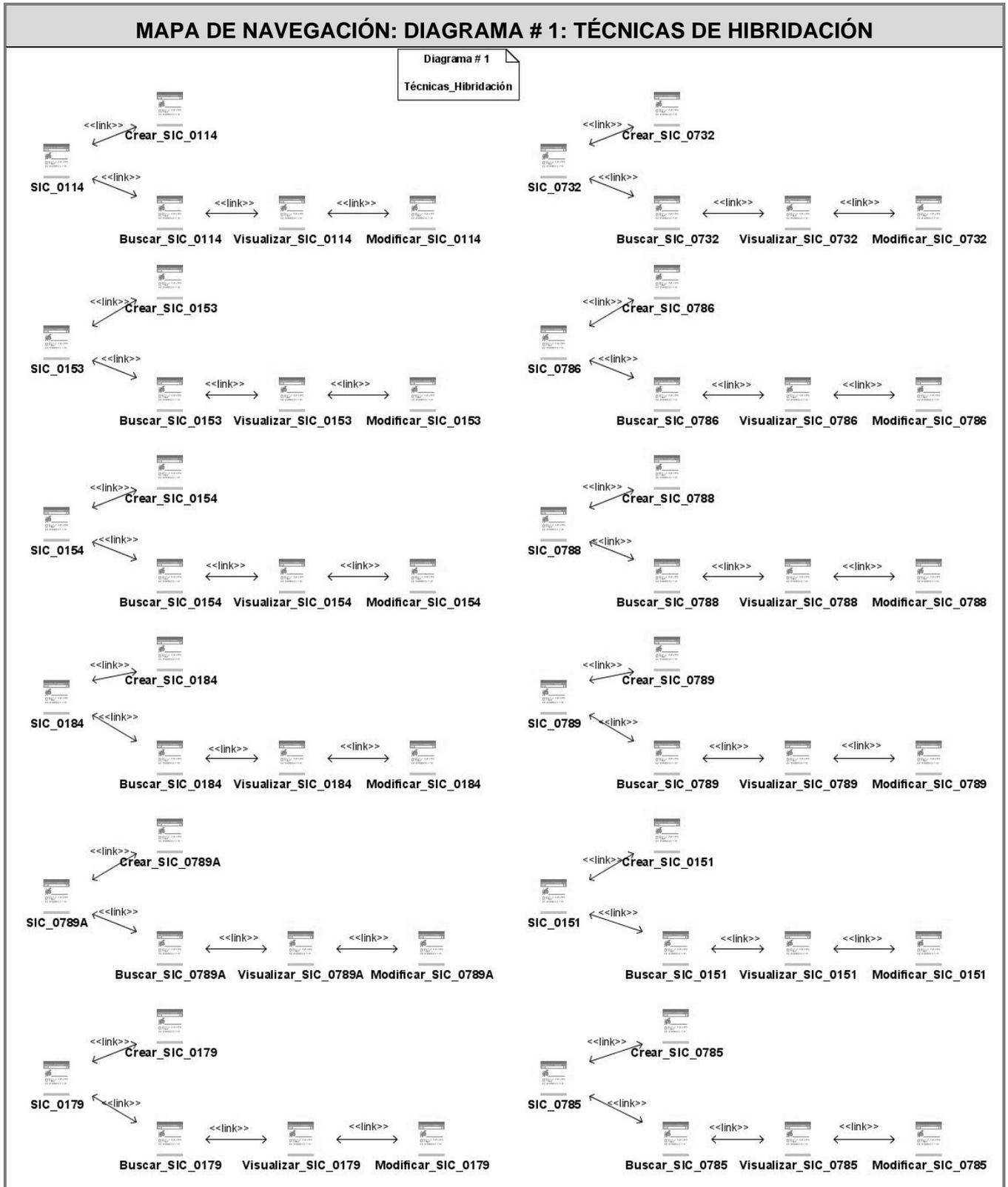


Fig 11. Mapa de Navegación: Diagrama # 1: Técnicas de Hibridación.

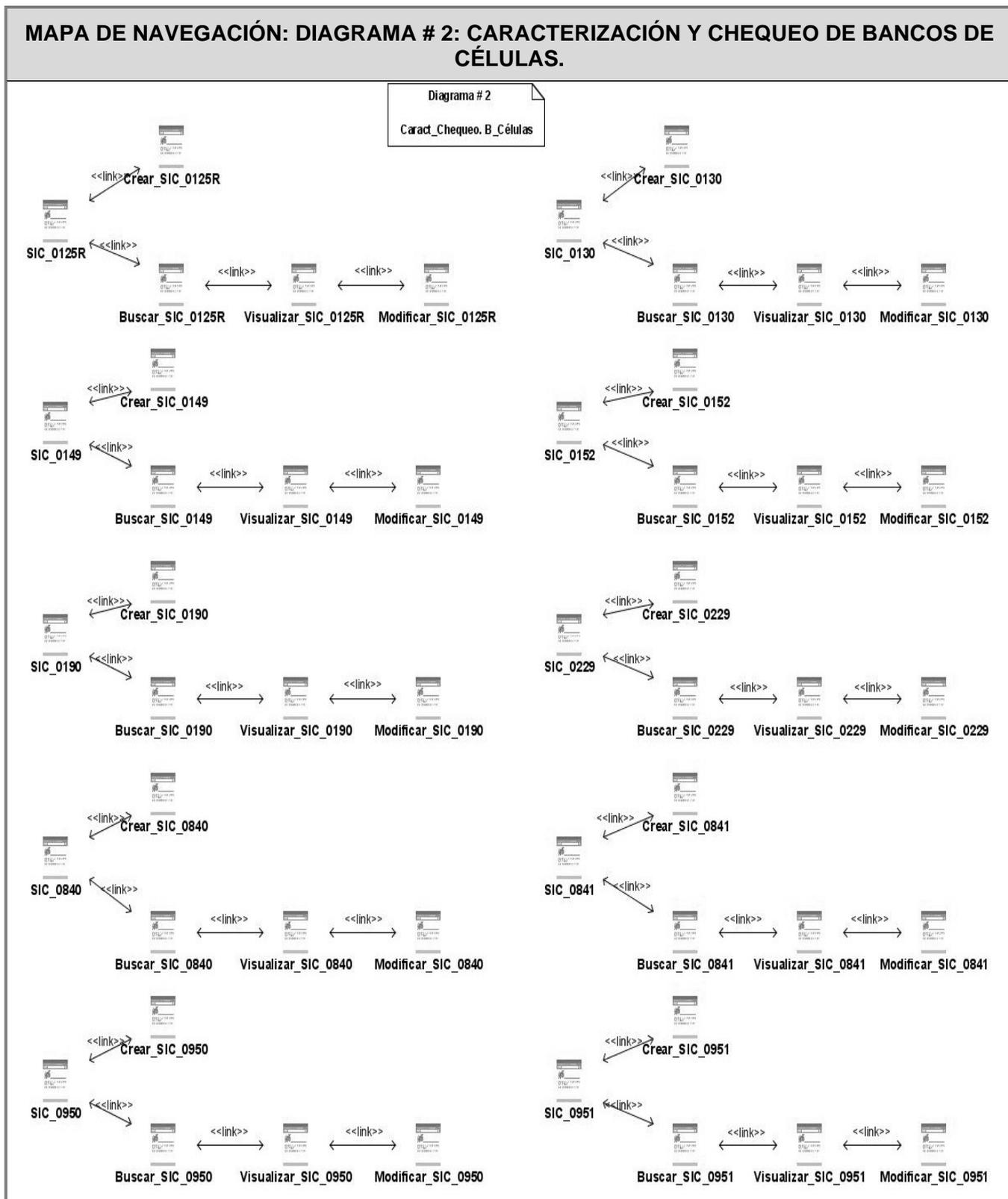


Fig 12. Mapa de Navegación: Diagrama # 2: Caracterización y Chequeo de Bancos de Células.

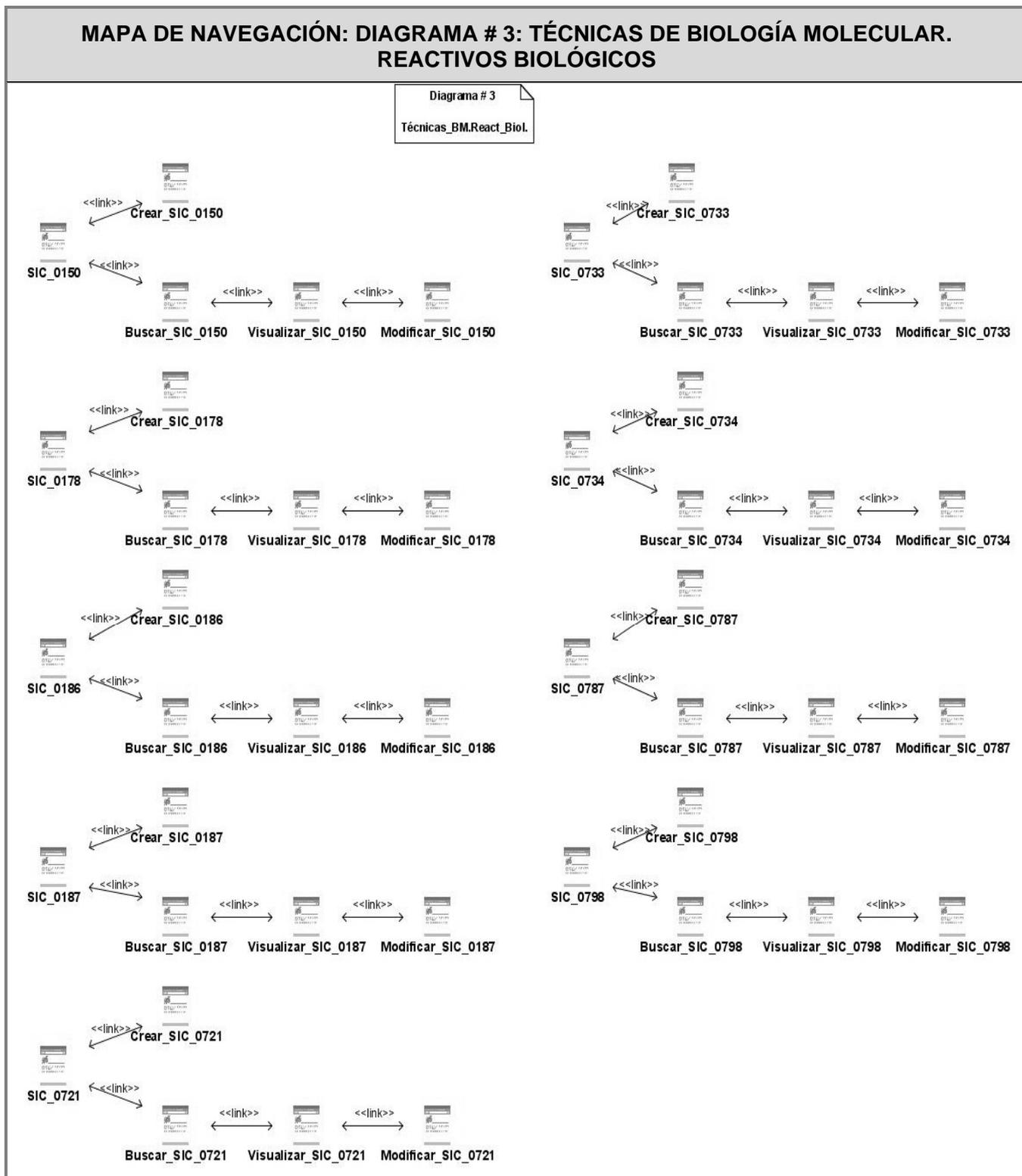


Fig 13. Mapa de Navegación: Diagrama # 3: Técnicas de Biología Molecular. Reactivos Biológicos.

3.5 Diagramas de Clases del Diseño.

Para todos los diagramas de clases se realizó un único paquete de Modelo, para lograr un mejor entendimiento de los mismos. En la Vista se encuentran: las páginas clientes, los formularios, páginas servidoras y el layout, aquí se representa la información con la que trabaja la aplicación. El Controlador se encuentra conformado por el controlador frontal que es el único punto de entrada a la aplicación, carga la configuración y determina la acción a ejecutarse, y las acciones: verifican la integridad de las peticiones, preparan los datos requeridos por la capa de presentación y contienen la lógica de la aplicación. La lógica de negocio de las aplicaciones Web depende del modelo de datos en él se encuentran todas las clases con sus atributos y las relaciones que existen entre ellas con las clases que genera Symphony: Objeto, ObjetoPeer, BaseObjeto y BaseObjetoPeer.

- Clases Base: Son las que se generan directamente a partir del esquema, no se deben modificar esas clases, porque cada vez que se genera el modelo, estas se actualizan.
- Clases Objeto: Heredan de las clases con nombre Base, estas no se modifican cuando se genera el modelo, por lo que en las mismas se añaden los métodos propios.
- Clases Peer: tienen métodos estáticos para trabajar con las tablas de la base de datos, proporcionan los medios necesarios para obtener los registros de las tablas y sus métodos. Se devuelven normalmente un objeto o una colección de objetos de la clase objeto relacionada.

A continuación se muestran los diagramas de clases del diseño de los CUS presentados en el capítulo anterior.

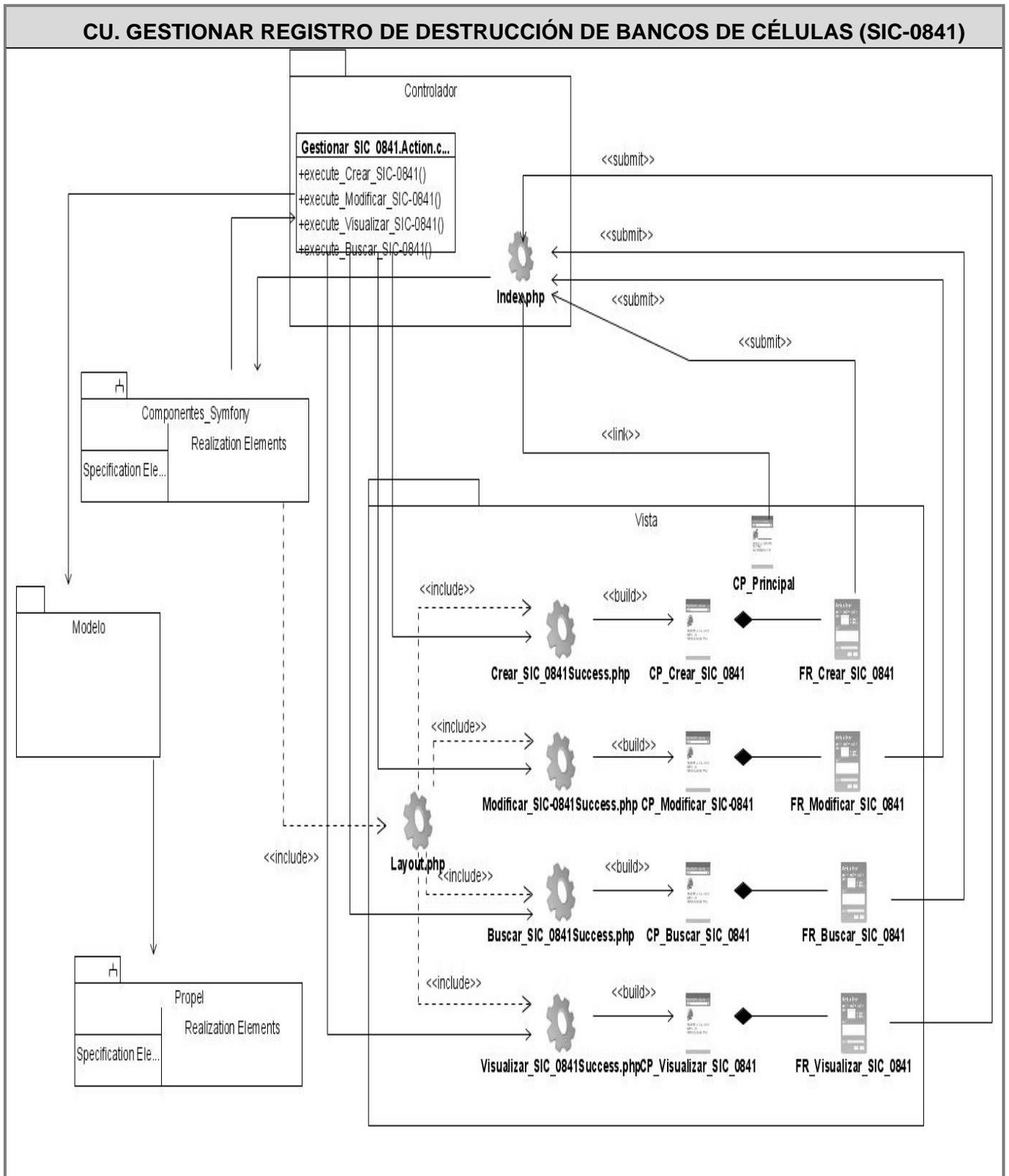


Fig 14. Gestionar registro de Destrucción de Banco de Células (SIC-0841).

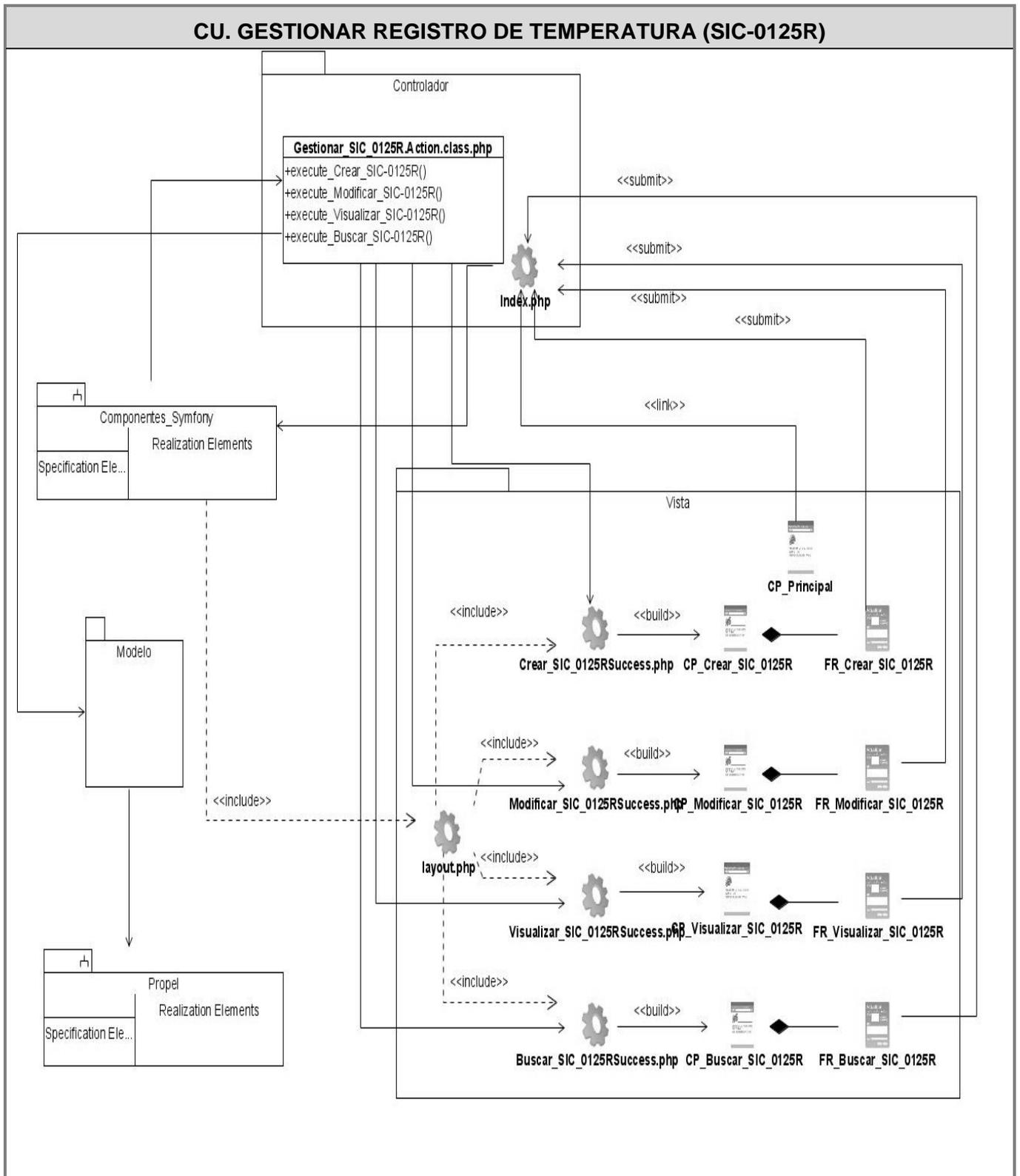


Fig 15. Gestionar Registro de Temperatura (SIC-0125R).

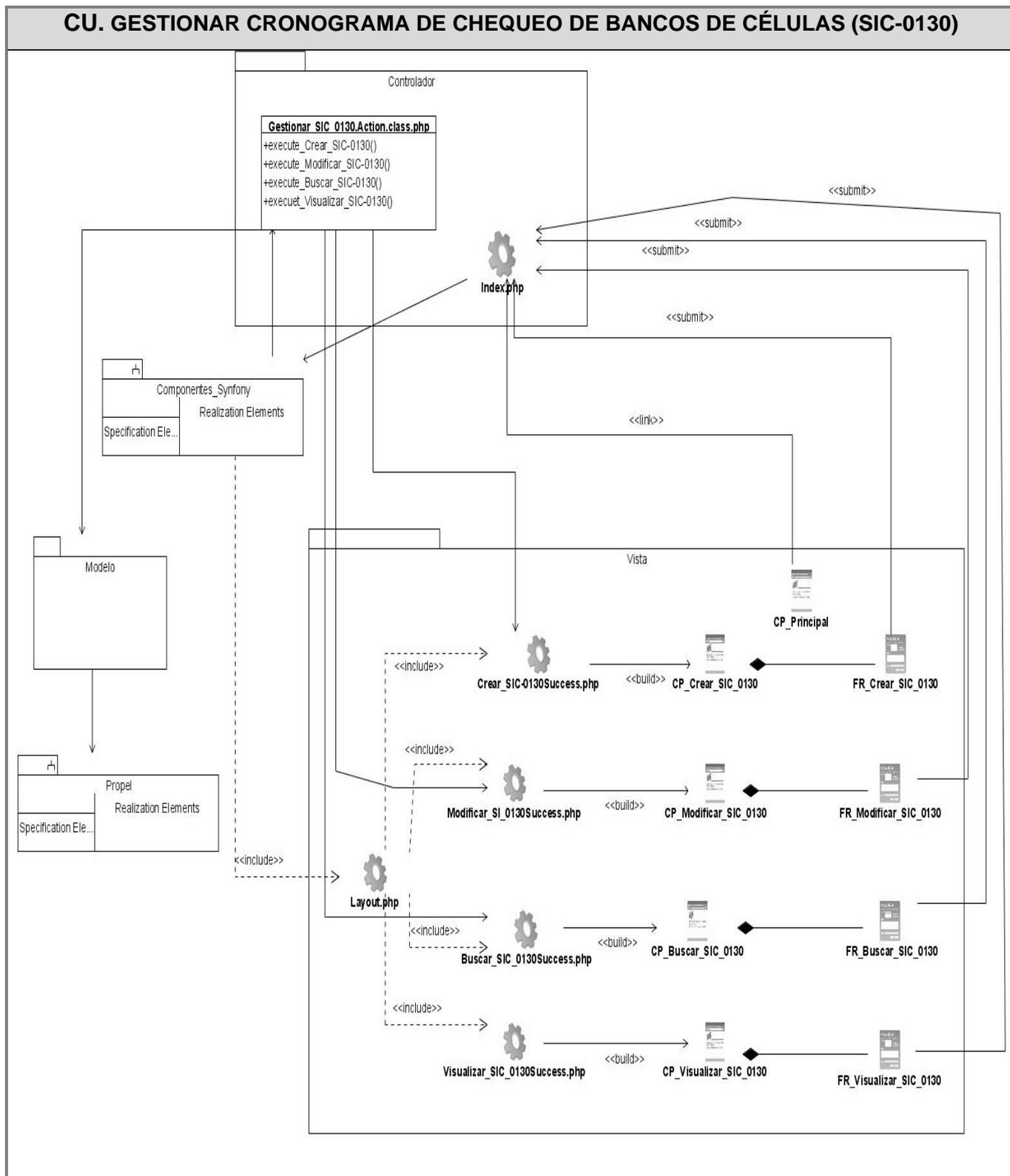


Fig 16. Gestionar Cronograma de Chequeo de Bancos de Células (SIC-0130).

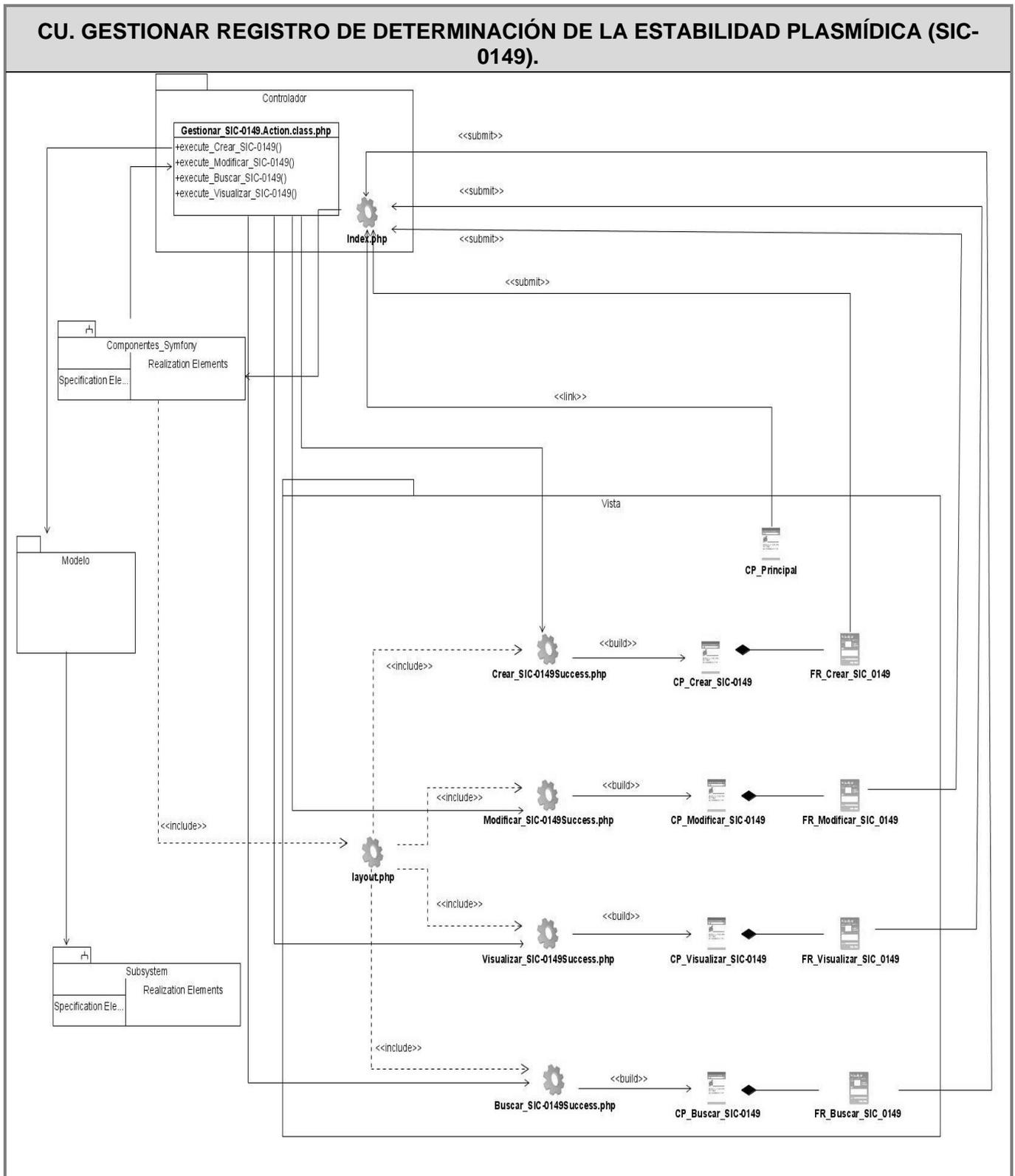


Fig 17. Gestionar registro de Determinación de la Estabilidad Plasmática (SIC-0149).

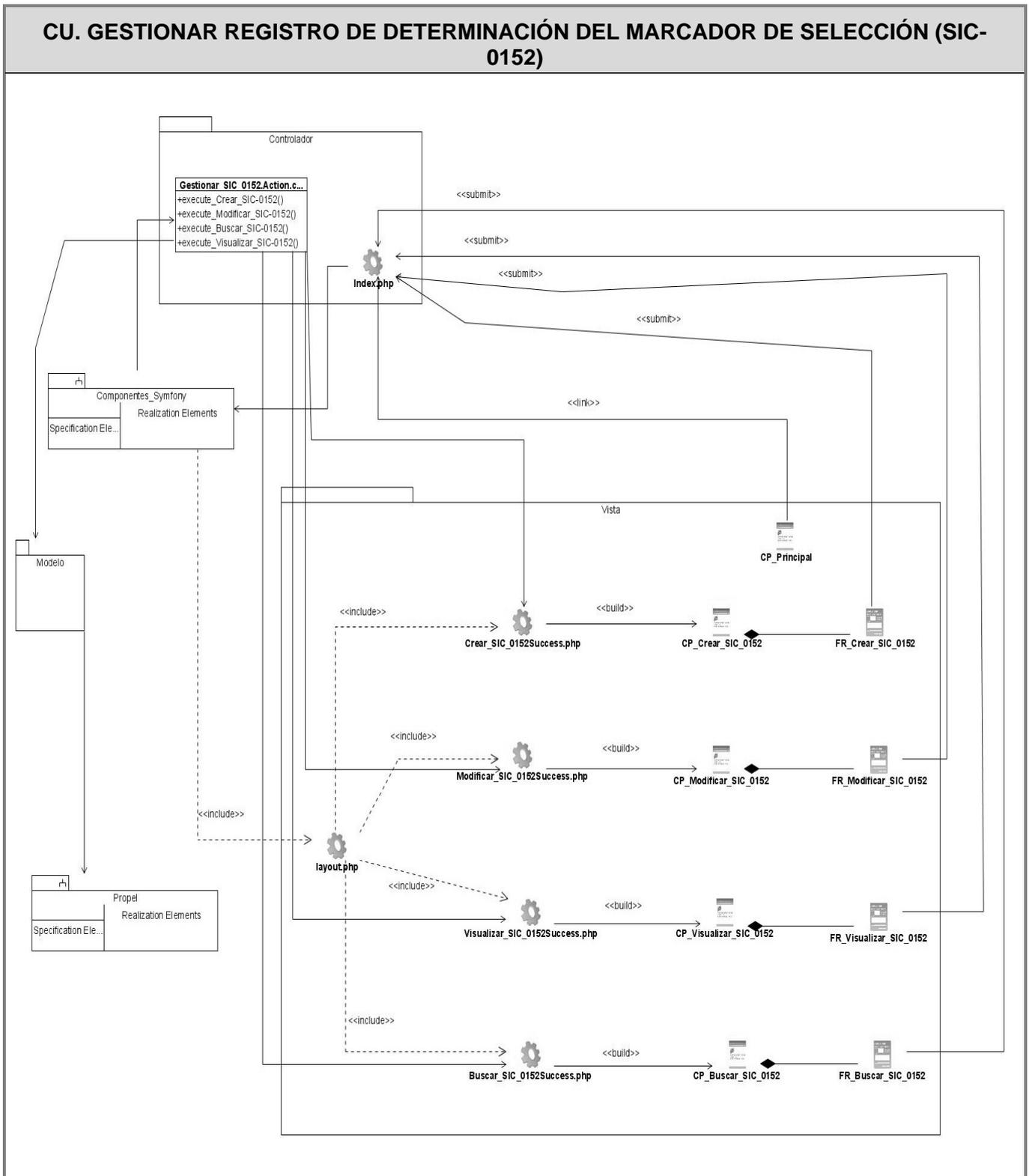


Fig 18. Gestionar registro de Determinación del Marcador de Selección (SIC-0152).

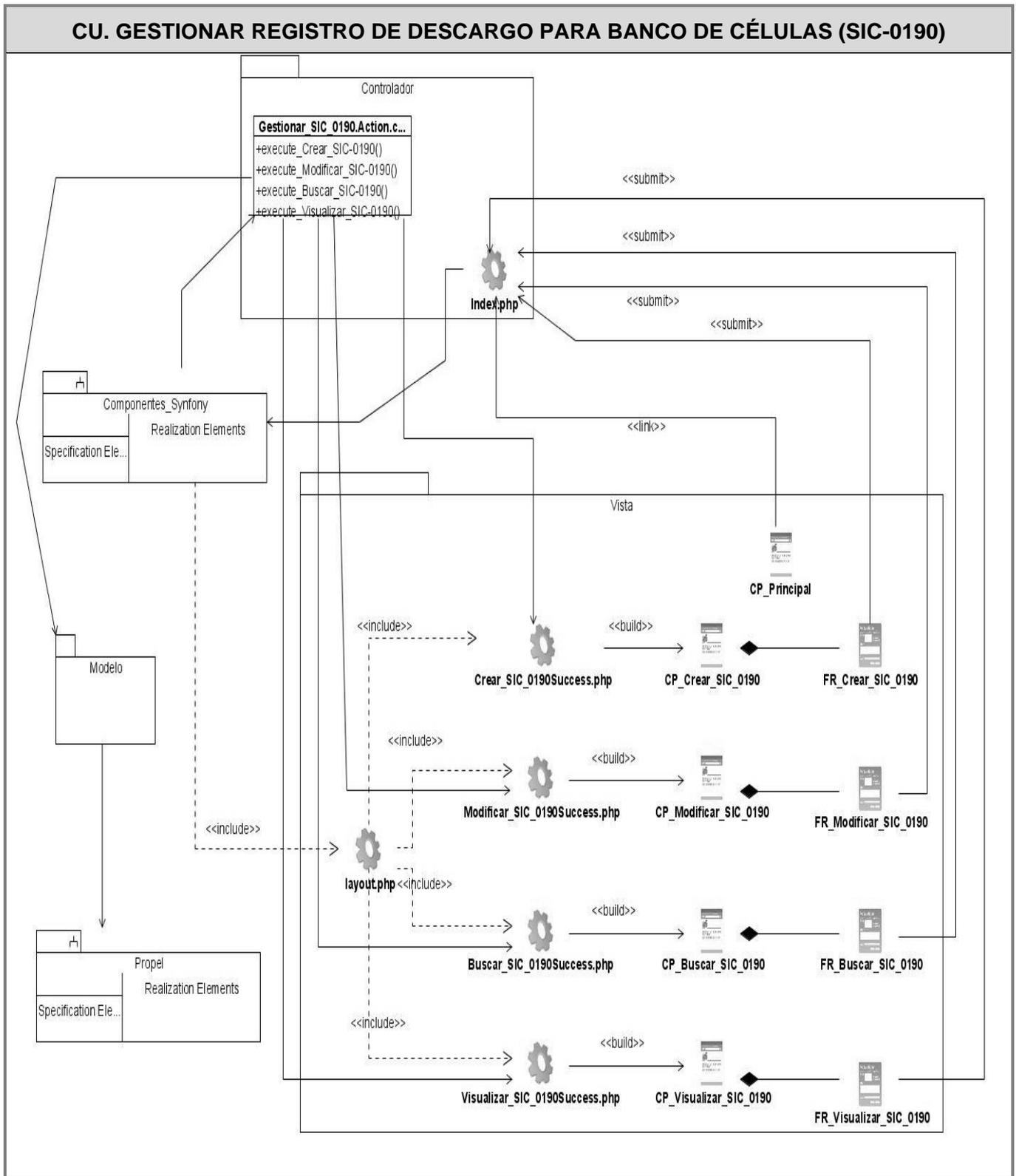


Fig 19. Gestionar registro de Descargo para Banco de Células (SIC-0190).

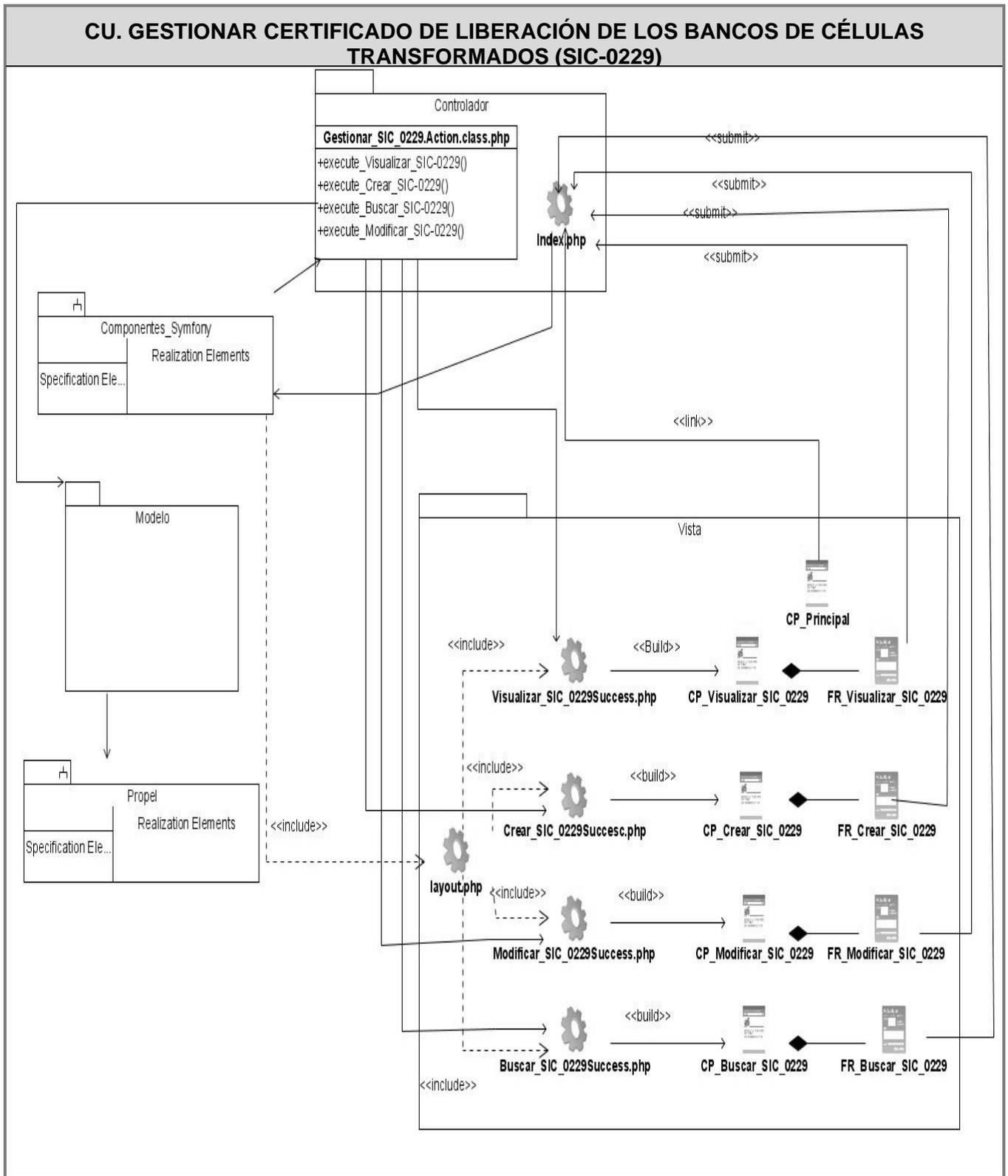


Fig 20. Gestionar Certificado de Liberación de los Bancos de Células transformados (SIC-0229).

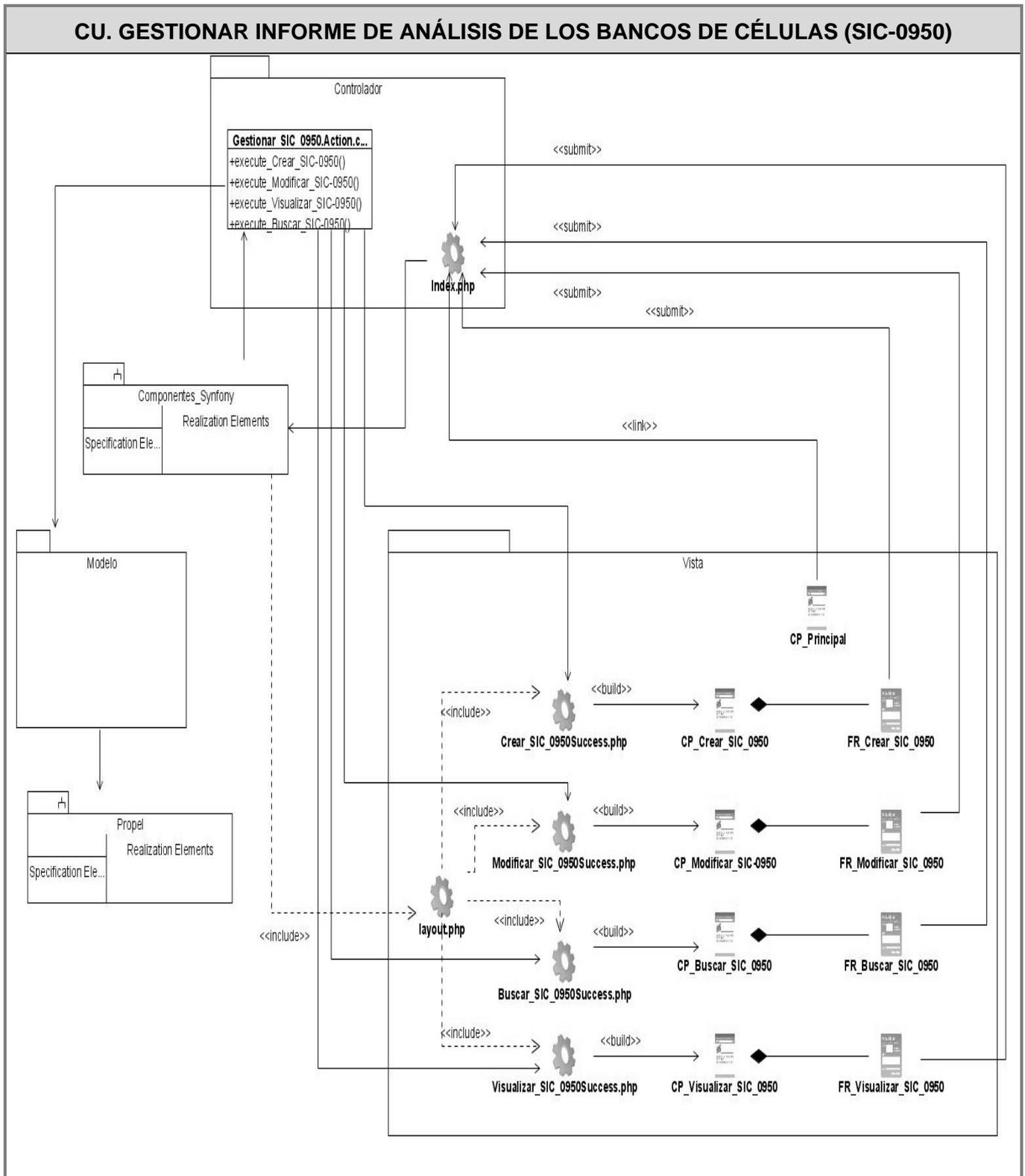


Fig 21. Gestionar Informe de análisis de los bancos de células (SIC-0950).

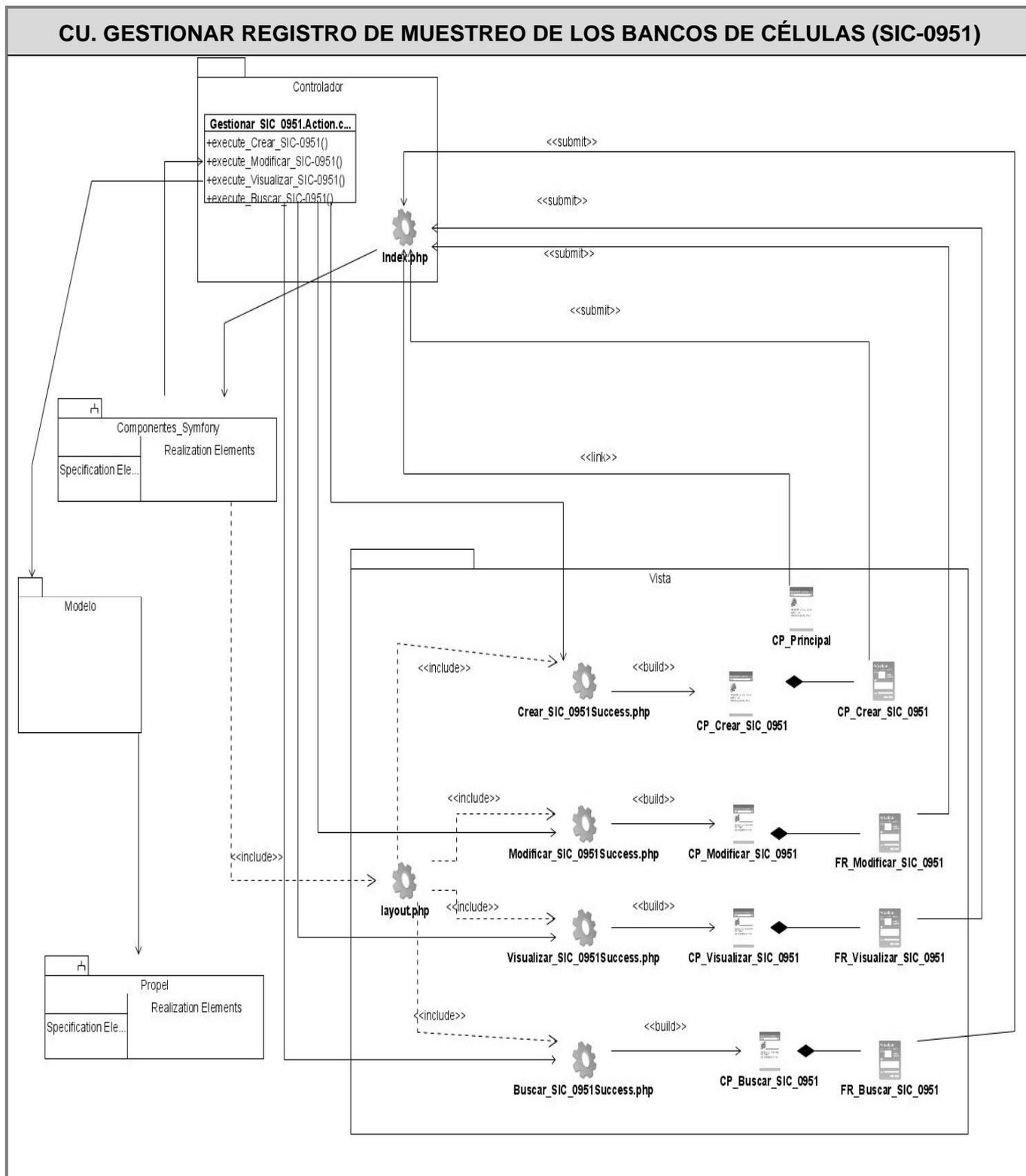


Fig 22. Gestionar registro de Muestreo de los bancos de Células (SIC-0951).

3.6 Diagrama de Secuencia.

Los diagramas de secuencia tienen como objetivo principal lograr que los programadores tengan una idea más segura de la traducción de las funcionalidades que debe tener el sistema.

Características de los diagramas de secuencia:

- Línea de vida de los objetos: representa la existencia de un objeto sobre un período de tiempo.
- Foco de control: muestra el período de tiempo durante el cual un objeto está representando una acción.
- Pueden servir para visualizar, especificar, construir y documentar los aspectos dinámicos de una sociedad particular de objetos, o pueden ser usados para modelar un flujo particular de control de un caso de uso.
- Son importantes para modelar los aspectos dinámicos de un sistema y para construir sistemas ejecutables a través de ingeniería hacia adelante e ingeniería inversa.

3.7 Evaluación de la calidad del diseño

Después de culminado el flujo de trabajo de Análisis y Diseño de un software es importante la evaluación de la calidad del mismo. La realización de los Casos de Uso que incluye; diagrama de clases del diseño y los diagramas de interacción deben quedar validados con el objetivo de que el sistema sea implementado sin ambigüedades y con la calidad requerida.

Para validar el diseño se han definido diversas maneras entre las que se pueden mencionar, la generación del código a través de la herramienta CASE de modelado utilizada, utilizando los diagramas de interacción para comprobar si el realizado el diseño satisface las necesidades del programador. En el caso de que el sistema ya se esté implementando, es posible comparar el diseño con el código fuente y analizar si existe una correcta correspondencia entre ellos.

El diseño realizado a las clases diseñadas en el módulo de BM, que da lugar a la realización de este trabajo se puede comprobar mediante la realización de encuestas a los programadores, responsables de lograr la operacionalidad del sistema. A continuación se muestran algunos aspectos en los cuales fue basada la encuesta y que arrojaron los siguientes resultados:

- *Interpretación del diseño:* Los programadores han estudiado los diagramas de clases del diseño y secuencia realizados, considerando que el envío de mensaje y las definiciones de las clases han sido correcta y fácil de comprender para realizar la implementación del sistema.
- *Correspondencia de los requerimientos no funcionales:* En el diseño se evidencia el uso de los requisitos no funcionales; como el de Seguridad, ya que cada usuario que se autentifique en el sistema tendrá acceso solamente a la página que le corresponde o sea, sólo realizará las tareas que ya le han sido asignadas por su jefe de grupo. También se evidencia el RNF de restricciones del diseño, pues el diseño cumple con la característica de que la capa de presentación esté separada de la capa de lógica.

A continuación se muestran los diagramas de secuencia para cada uno de los escenarios que responden al diseño realizado para el CUS: *Gestionar Solicitud de destrucción de bancos de células (SIC-0840)*. Para ver los demás diagramas remitirse al Expediente de Proyecto.



Fig 23. Escenario Crear SIC_0840: Gestionar Solicitud de Destrucción de Bancos de Células.

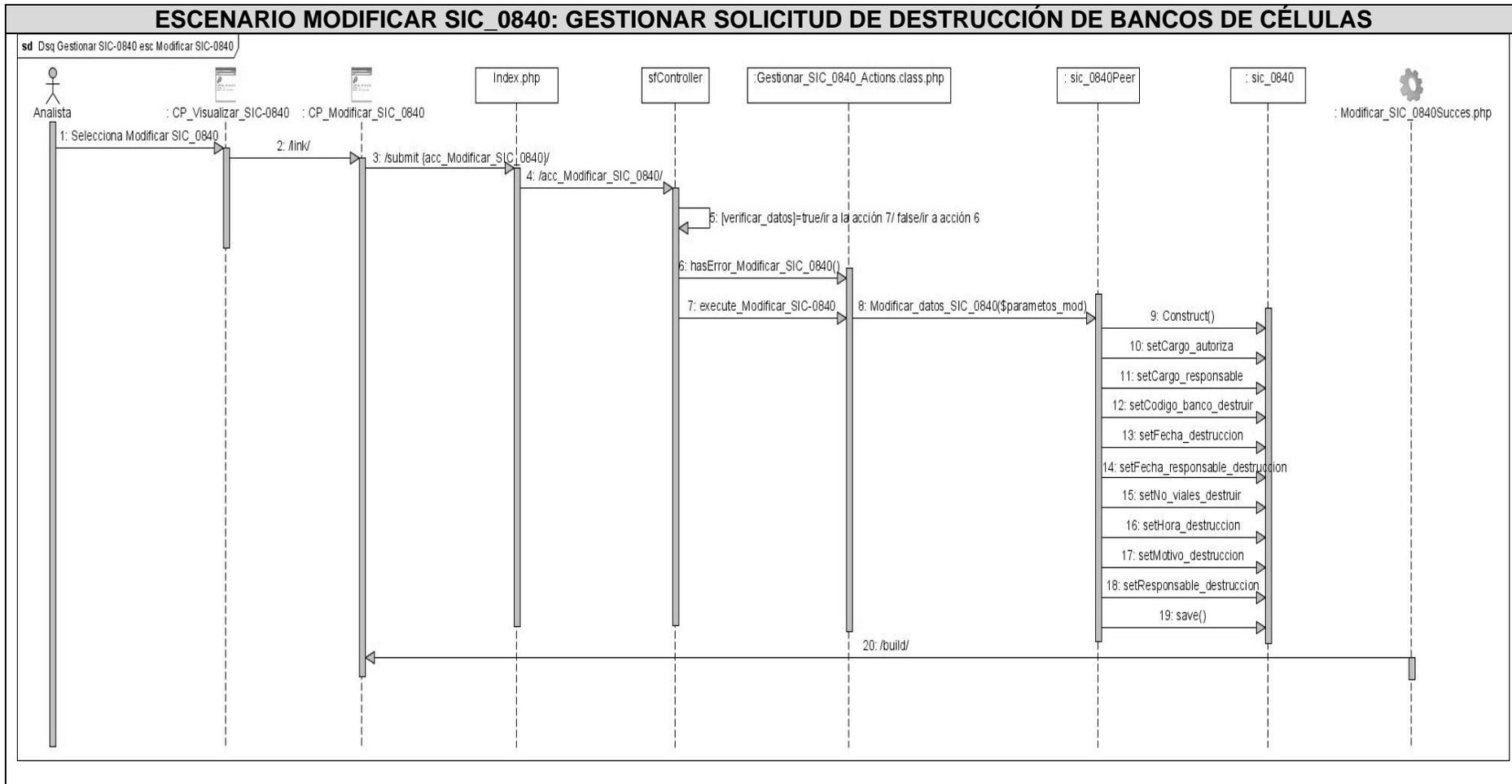


Fig 25. Escenario Modificar SIC_0840: Gestionar Solicitud de Destrucción de Bancos de Células.

3.8 Conclusiones

En este capítulo se definieron las clases que se corresponden con el diseño del sistema, se muestran los diagramas de clases del diseño y los diagramas de interacción aplicando el patrón MVC, para dar validez al diseño y brindar la entrada principal para la implementación del módulo. Se realizó el mapa de navegación, así como el diagrama de Despliegue y el diagrama de la Vista Lógica.

CONCLUSIONES

- Se realizó el análisis de los procesos que se desarrollan en el LBM, logrando una completa familiarización con los mismos.
- Se identificaron los requerimientos correspondientes al módulo Biología Molecular y se realizó la descripción del sistema.
- Se terminó la realización del diseño de los 10 CU desarrollados en este trabajo.

RECOMENDACIONES

- Se continúe la realización de los casos de uso del sistema definidos en el módulo Biología Molecular.
- Se refine el diseño teniéndose en cuenta la validación para una futura implementación conforme con el diseño realizado.
- Se use para el refinamiento del flujo de trabajo Análisis y Diseño la herramienta CASE Visual Paradigm pues resulta de interés para la facultad migrar al Sistema Operativo Linux.

REFERENCIA BIBLIOGRÁFICA

- [1]. Díaz Laurencio Elennis, Nieto Cervantes Liusmila; [Citado: 3/11/2007] *LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis de la Sección de Mejoramiento de la Calidad y del Grupo de Desarrollo*, La Habana 2007, Cuba.
- [2]. Acosta Mendoza Niusvel, Rodríguez León Alexis René; [Citado: 3/11/2007] *LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis de los Laboratorios de Biología Molecular y Ensayos Biológicos I*, La Habana 2007, Cuba.
- [3]. Cuevas Melis Olga, "*Formación y cualificación de personal*" con estructura y contenidos de acuerdo con la norma UNE-EN ISO/IEC 17025:2005. [En Línea, Citado: 1/12/2007], Disponible en: <http://www.onlabweb.com/Lims.htm>.
- [4]. Rodríguez Perojo¹ Lic. Keilyn y Ronda León², Lic. Rodrigo, *El web como sistema de información* Recibido: 12 de enero del 2006. Aprobado: 16 de enero del 2006 [Citado: 13/1/2008].
- [5]. Quass, *¿POR QUÉ NECESITO UN LIMS?*, [En Línea, Citado: 1/12/2007], Disponible en: http://www.quaass.com/web/unlims_1.htm.
- [6]. Entorno Virtual de Aprendizaje, *Características de RUP*, [En Línea, Citado: 8/11/2007], Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=6655>.
- [7]. Ricardo Williams: Coordinador general del SIU. Licenciado en Sistemas. Especialista en Ingeniería de Software. *Sistemas de Información y Mapa de Navegación\SIU.htm*. [Citado: 5/3/2008].
- [8]. Hanz Cocchi Guerrero, *RoI* [En Línea, Citado: 27/1/2008] disponible en: <http://hancocchi.net/el-rol-del-analista-en-rup/>.
- [9]. *Ayuda extendida del Rational Rose Enterprise Edition 2003*, [Citado: 25/1/2008].
- [10]. *Un producto o artefacto*, [Citado: 27/1/2008] disponible en: *Ayuda extendida del Rational Rose Enterprise Edition 2003*.
- [11]. *Manual del Visual Paradigm*, [Citado: 17/2/2008].

- [12]. Modelado de Sistemas con UML: ¿Qué es UML?, [En Línea, Citado: 24/1/2008], disponible en: <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c12.html>
- [13]. Fabien Potencier, François Zaninotto, *Symfony, la guía definitiva*, [En Línea, Citado: 20/2/2008].
- [14]. Conallen, Jim. *UML Extension for Web Applications v 0.91*. Marzo, 1999, [Citado: 26/4/2008].

BIBLIOGRAFÍA

- Acosta Mendoza Niusvel, Rodríguez León Alexis René; [Citado: 3/11/2007] LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis de los Laboratorios de Biología Molecular y Ensayos Biológicos I, La Habana 2007, Cuba.
- *Arquitectura de Software*, [Citado: 2/12/2007], disponible en: http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.mspx.
- CIGB, Dirección de Calidad, 2003, [En Línea, Citado: 21/10/2007], disponible en: <http://www.cigb.edu.cu/pages/calidad.htm>.
- Conallen, Jim. *UML Extension for Web Applications v 0.91*. Marzo, 1999, [Citado: 26/4/2008].
- Cuevas Melis Olga, "Formación y cualificación de personal" con estructura y contenidos de acuerdo con la norma UNE-EN ISO/IEC. [En Línea, Citado: 1/12/2007], Disponible en: <http://www.onlabweb.com/Lims.htm>.
- Desarrollo Orientado a Objetos con UML, [Citado: 20/3/2008], disponible en: <http://www.clikear.com/manuales/uml/index.asp>.
- Díaz Laurencio Elennis, Nieto Cervantes Liusmila; [Citado: 3/11/2007] LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis de la Sección de Mejoramiento de la Calidad y del Grupo de Desarrollo, La Habana 2007, Cuba.
- Entorno Virtual de Aprendizaje, *Características de RUP*, [En Línea, Citado: 8/11/2007], Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=6655>.
- Fabien Potencier, François Zaninotto, *Symfony, la guía definitiva*, [En Línea, Citado: 20/2/2008].
- *FLUJO DE TRABAJO DE REQUERIMIENTOS*. Departamento de Ingeniería y Gestión de Software, Universidad de las Ciencias Informáticas. 2007-2008. Conferencia, [Citado: 23/1/2008].
- Hanz Cocchi Guerrero, *Rol* [En Línea, Citado: 27/1/2008] disponible en: <http://hancocchi.net/el-rol-del-analista-en-rup/>.
- *Herramientas de Gestión de Requisitos*, [En Línea, Citado: 16/4/2008], disponible en: <http://www.ines.org.es/vulcano/wp-content/uploads/2007/09/d6-estudio-de-herramientas-de-certificacion-tid.pdf>.
- Jacobson, Ivar; Booch, Grady y Rumbaugh, James: *El Proceso Unificado de Desarrollo Volumen I*, The Addison Wesley Longman Inc., 1999, [Citado: 26/2/2008].
- Larman, C. *UML y Patrones*, Tomo I, Capítulo 18, [Citado: 18/3/2008].
- *Manual del Visual Paradigm*, [Citado: 17/2/2008].

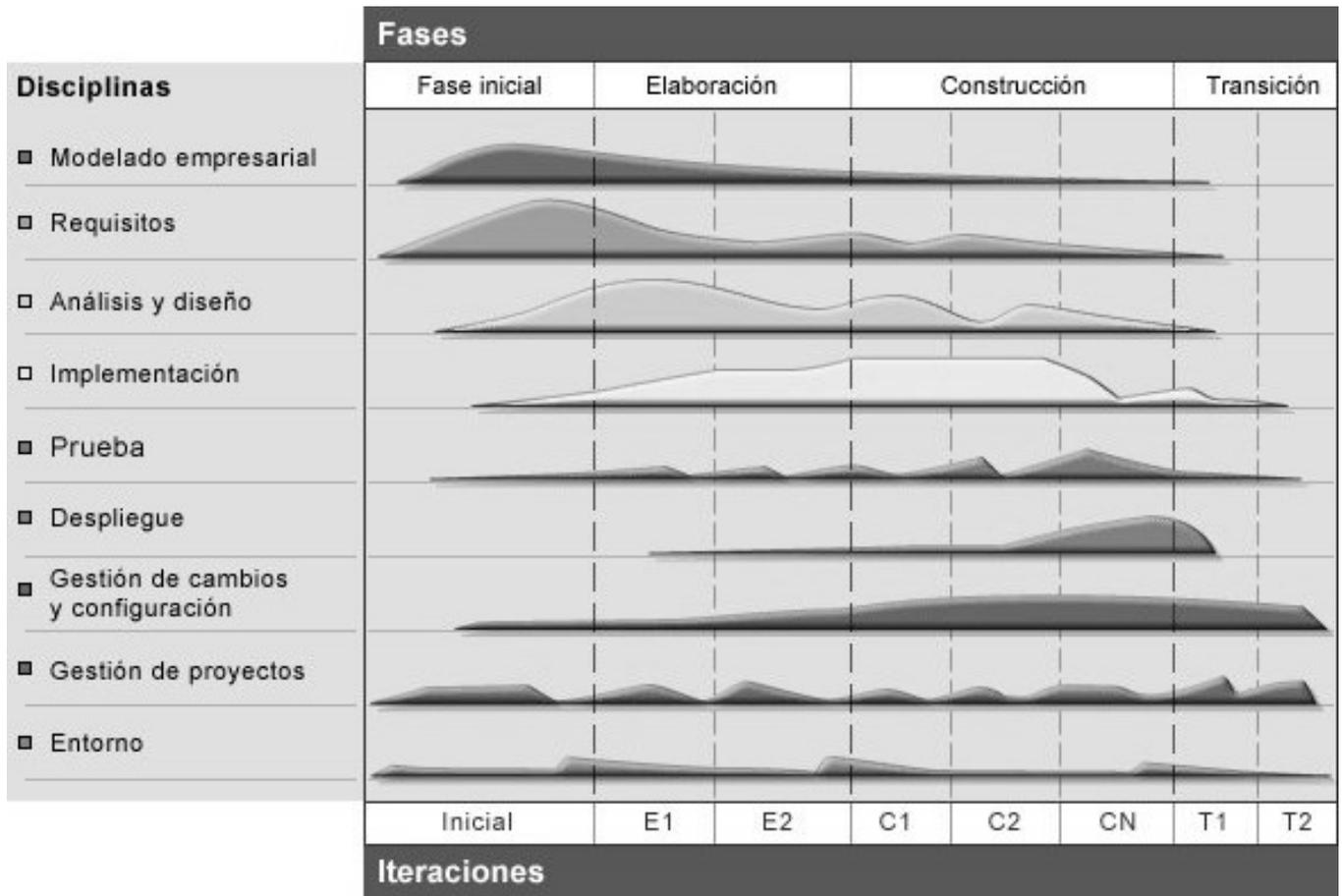
- Modelado de Sistemas con UML: ¿Qué es UML?, [En Línea, Citado: 24/1/2008], disponible en: <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c12.html>.
- Patrones de diseño. [En línea, Citado: 14/3/2008] <http://mit.ocw.universia.net/6.170/6.170/f01/pdf/lecture-12.pdf>.
- Patrones para asignación de responsabilidades, [En Línea, Citado: 25/3/2008], disponible en: <https://s3.amazonaws.com/ppt-download/gonzalorojas-12-uml-patrones-de-diseno1574.ppt>.
- Procesos de Desarrollo: RUP, XP y FDD. 2002, [En línea, Citado: 20/1/2008], disponible en: http://www.javahispano.org/contenidos/es/procesos_de_desarrollo.
- Quass, ¿POR QUÉ NECESITO UN LIMS?, [En Línea, Citado: 1/12/2007], Disponible en: http://www.quaass.com/web/unlims_1.htm.
- Ricardo Williams: Coordinador general del SIU. Licenciado en Sistemas. Especialista en Ingeniería de Software. Sistemas de Información y Mapa de Navegación\SIU.htm. [Citado: 5/3/2008].
- Rodríguez Perojo1 Lic. Keilyn y Ronda León, Lic. Rodrigo, *El web como sistema de información* Recibido: 12 de enero del 2006. Aprobado: 16 de enero del 2006, [Citado: 15/5/2008].
- *Sitio Teleformación de La Universidad de Las Ciencias Informáticas*, [En Línea, Citado: 19/11/2007], disponible en : <http://teleformacion.uci.cu/mod/resource/view.php?id=8865>.
- *Un producto o artefacto*, [Citado: 27/1/2008] disponible en: Ayuda extendida del Rational Rose Enterprise Edition 2003.
- Ventajas e inconvenientes de PHP, [Citado: 27/11/2007], disponible en: <http://ascii.eii.us.es/docs/2002-03/php/php4.html>.

ANEXOS

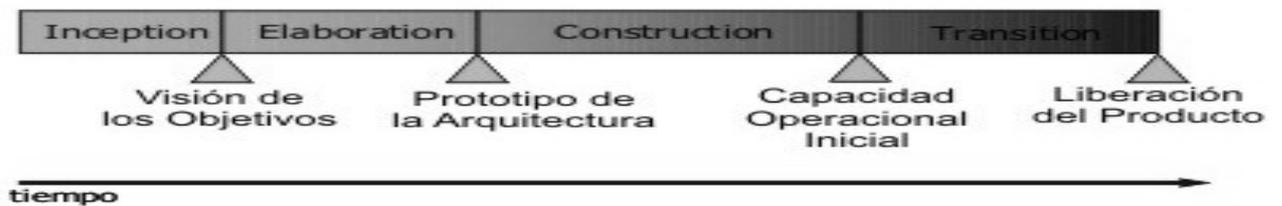
Anexo 1. Organigrama del Departamento de Calidad del CIGB.



Anexo 2. Fases e Iteraciones de RUP.



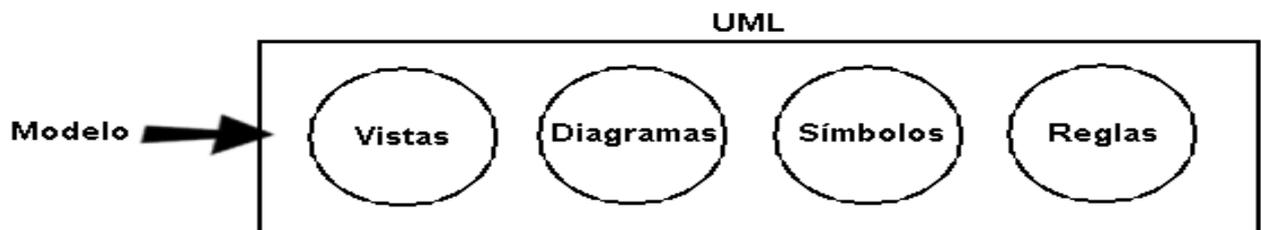
Anexo 3. Fases e hitos en RUP.



Anexo 4. Artefactos a realizar según los roles a desempeñar.



Anexo 5. Representación gráfica del Lenguaje de Modelado.



Anexo 6. Prototipos no funcionales del CUS

Anexo 6 Escenario Crear: *Gestionar Solicitud de Destrucción de Bancos de Células (SIC_0840)*.

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:	
Laboratorio BCP	Area:		
Código del Banco a Destruir		Número de viales a destruir :	
Motivos de la Destrucción: <div style="border: 1px solid black; height: 150px; width: 100%;"></div>			
Fecha de destrucción:		Hora de la destrucción	
Persona responsable de la destrucción			
Nombre y apellido		Firma	
Cargo que ocupa		Fecha	
Autorización de la destrucción			
Nombre y apellido		Firma	
Cargo que ocupa		Fecha	

Anexo 6.1 Escenario Modificar: *Gestionar Solicitud de Destrucción de Bancos de Células (SIC_0840)*.

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:	
Laboratorio BCP	Area:	xxxxxxxx	
Código del Banco a Destruir	xxxxxxxx	Número de viales a destruir :	xxxxxxxx
Motivos de la Destrucción: <div style="border: 1px solid black; height: 150px; width: 100%;"></div>			
Fecha de destrucción:	xxxxxxxx	Hora de la destrucción	xxxxxxxx
Persona responsable de la destrucción			
Nombre y apellido	xxxxxxxx	Firma	xxxxxxxxxxxxxxxxxxxx
Cargo que ocupa	xxxxxxxx	Fecha	xxxxxxxx
Autorización de la destrucción			
Nombre y apellido	xxxxxxxx	Firma	xxxxxxxxxxxxxxxxxxxx
Cargo que ocupa	xxxxxxxx	Fecha	xxxxxxxx

Anexo 6.2 Escenario Buscar-Visualizar: *Gestionar Solicitud de Destrucción de Bancos de Células (SIC_0840)*.

CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA		
SUBDIRECCIÓN DE CALIDAD		
SOLICITUD DE DESTRUCCIÓN DE BANCOS DE CÉLULAS		
Area :	<input type="text"/>	Código: <input type="text"/>
		<input type="button" value="Cancelar"/> <input type="button" value="Buscar"/>
Area	Código	
<input type="text"/>	<input type="text"/>	Visualizar

GLOSARIO DE TÉRMINOS

Artefacto: Pieza de información utilizada o producida por un proceso de desarrollo de software, como un documento externo o el producto de un trabajo. Un artefacto puede ser un modelo, una descripción o un software.

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

BD: Base de Datos.

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

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

Desviación: Alteración no prevista, resultado de variaciones accidentales negligentes o aleatorias que afecta o puede afectar potencialmente la calidad de un producto o proceso.

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

Especificación: Lista detallada de requisitos con los cuales los productos o materiales usados u obtenidos deben estar conformes y sirven como base para la evaluación de la calidad de los mismos.

Estabilidad: Propiedad de cualquier forma farmacéutica contenida en un determinado material de envase de mantener, dentro de ciertos límites y durante el tiempo de almacenamiento y uso, las características físicas, químicas, microbiológicas, toxicológicas y terapéuticas que tenía en el momento de su fabricación.

Estudio de Estabilidad: Serie de ensayos que permiten obtener información para establecer el período de validez de un medicamento en su envase original y en las condiciones de almacenamiento especificadas.

Lote: Cantidad definida de materia prima, material de envase o producto elaborado en un solo proceso o en una serie de procesos, de tal manera que pueda esperarse que sea homogéneo.

Material: Es aquel material que se utiliza para el envase productos, ya sean intermedios, terminados o ingredientes farmacéuticos activos.

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

No. de Lote: Combinación bien definida de números, letras y/o símbolos que identifican inequívocamente un lote en las etiquetas, registros de lotes, certificados de análisis, etc. Que a su vez permite determinar la historia completa de su producción.

PPO: Procedimiento patrón de operación.

Producción: Todas las operaciones involucradas en la preparación de un producto farmacéutico, desde la recepción de los materiales, el procesado y el envasado, etiquetado y reetiquetado, hasta llegar al producto terminado.

Producto: Resultado de una etapa del proceso de producción.

SIC: Sistema Informativo de Calidad.

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

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