

Universidad de las Ciencias Informáticas

Facultad 6



**Título: LIMS de Calidad del Centro de Ingeniería Genética y
Biotecnología: Desarrollo de la Base de Datos del Módulo
Liberación Analítica**

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

Autores: Anabel Vega Calcines

Alfredo Rodríguez Ruiz

Tutores: MsC. Maypher Román Durán

Ing. Niusvel Acosta Mendoza

Co-tutor: Manuel de Jesús Luis Díaz

Ciudad de La Habana, 24 de junio de 2008

“Año 50 de la Revolución”

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los 24 días del mes de junio del año 2008.

Anabel Vega Calcines (Autor)

Alfredo Rodríguez Ruiz(Autor)

MsC. Maypher Román Durán (Tutor)

Ing. Niusvel Acosta Mendoza(Tutor)



“Todo lo que enaltece y honra implica sacrificio”.

Jesús Suárez Gayol

DATOS DE CONTACTO

Msc.: Maypher Román Durán.

Universidad de Ciencias Informáticas, La Habana, Cuba.

E-mail: maypher@uci.cu

Ingeniero: Niusvel Acosta Mendoza.

Universidad de Ciencias Informáticas, La Habana, Cuba.

E-mail: nacosta@uci.cu

Licenciado: Manuel de Jesús Luis Díaz

Universidad de Ciencias Informáticas, La Habana, Cuba.

E-mail: mluis@uci.cu

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

Título: LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Desarrollo de la Base de Datos del Módulo Liberación Analítica.

Autores: Anabel Vega Calcines

Alfredo Rodríguez Ruiz

AGRADECIMIENTOS

A nuestro eterno Comandante en Jefe Fidel y a la Revolución Cubana por ser paradigma, guía y permitirnos estudiar en esta Universidad de Excelencia.

A nuestros profesores de la UCI por formarnos como profesionales y jóvenes dignos de estos tiempos.

A nuestros tutores por su asesoría y valioso aporte.

Al profesor Léster por su colaboración sistemática en la realización de este trabajo.

A los trabajadores del CIGB por dedicarnos atentamente parte de su tiempo.

A nuestro grupo de clases de estos cinco años por su apoyo.

A nuestros compañeros del proyecto LIMS.

DEDICATORIA

De Anabel:

A mi mamá María Elena y mi papá Manuel por su amor infinito, dedicación y apoyo constante, por enseñarme a no creer en imposibles e inspirarme con su ejemplo.

A mami Chichi, mi abuelo Diego, mi tía Mabel y mi hermano Boris por su cariño sin límites, apoyo incondicional y confianza absoluta, no los defraudaré.

A mi familia toda por ser tan especial.

A mis amigos de la UCI por regalarme su valiosa amistad en estos cinco inolvidables años, siempre los llevaré en mi corazón.

De Alfredo:

A mi mamá con mucho cariño y amor, por su confianza infinita, por ser quien me ha enseñado a rebasar todas las barreras que la vida nos presenta, a querer ser mejor cada día, a entender que no hay nada imposible y que solo hay que sacrificarse para lograr las metas que nos planteemos.

A mi papá por su apoyo incondicional y sus consejos de amigo fiel, su disponibilidad siempre que lo necesito y a su esposa Elena.

A mi abuelita Lola por ser la fuente de inspiración.

A mi hermano Claritin para que le sirva de ejemplo y motivación en su vida como estudiante.

A mi novia Yuneisi por su cariño, amor y dedicación en todos estos años a mi lado.

A mi tía Eloísa y su esposo Ibardito por brindarme su cariño y amor cada día, su eterna confianza, así como su apoyo incondicional.

A mis abuelos Rosa y Ovilio.

A mis compañeros de aulas que día tras día compartimos momentos inolvidables.

RESUMEN

La Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología (CIGB) tiene como objetivo lograr un Sistema de Gestión de la Calidad eficiente y para ello 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 manualmente y en ocasiones se dificulta su acceso. Por 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 Universidad de las Ciencias Informáticas (UCI). El presente trabajo describe el diseño e implementación de la Base de Datos del departamento de Liberación Analítica, uno de los grupos de Control de la Calidad, como parte del desarrollo del LIMS.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Proceso de gestión de información en el Grupo de Liberación Analítica de la Dirección de Calidad del CIGB.	6
1.2 Surgimiento y desarrollo de Bases de Datos.	7
1.3 Principales componentes de las Bases de Datos.	8
1.4 Modelos de Bases de Datos.	8
1.5 Arquitectura de los Sistemas de Bases de Datos.	11
1.6 Tendencias actuales y futuras de los Sistemas de Bases de Datos.	12
1.7 Sistemas Gestores de Bases de Datos.	12
1.7.1 Oracle.	13
1.7.2 MySQL.	14
1.7.3 PostgreSQL.	15
1.8 Mapeo de objetos a bases de datos con Symfony.	17
1.9 Rol de Diseñador de BD según el Proceso Unificado de Desarrollo.	19
1.10 Lenguaje Unificado de Modelado (UML).....	20
1.11 Herramientas CASE.....	21
1.11.1 Visual Paradigm for UML 6.1 Enterprise Edition.....	21
1.12 Herramientas de desarrollo.....	22
1.12.1 EMS SQL Manager 2005 for PostgreSQL.....	22
1.12.2 EMS Data Generator 2005 for PostgreSQL.....	22
Conclusiones	24
CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA	25
2.1 Requisitos funcionales y no funcionales del Módulo Liberación Analítica.	25
2.1.1 Requisitos funcionales	25
2.1.2 Requisitos no funcionales aplicables a la BD	30
2.2 Estrategia de integración de la solución con otros módulos.	31
2.3 Descripción de la Arquitectura de la Base de Datos.	33
2.4 Modelo de objetos del negocio.	35
2.5 Diagrama de clases del diseño.	35

2.6 Clases persistentes.....	35
2.6.1 Diagrama de Clases Persistentes.....	35
2.6.2 Descripción de las Clases Persistentes.....	38
2.7 Diseño de la BD.	51
2.7.1 Descripción de las tablas de la BD.	53
Conclusiones	69
CAPÍTULO 3: VALIDACIÓN DEL DISEÑO REALIZADO	70
3.1 Validación teórica del diseño.	70
3.1.1 Integridad.....	70
3.1.2 Normalización de los datos.....	71
3.1.3 Análisis de redundancia de datos.	73
3.1.4 Análisis de seguridad de la BD.	74
3.1.5 Trazabilidad de las acciones.....	75
3.2 Validación funcional del diseño.....	75
Conclusiones	81
CONCLUSIONES	82
RECOMENDACIONES.....	83
REFERENCIAS BIBLIOGRÁFICAS	84
BIBLIOGRAFÍA	86
GLOSARIO DE TÉRMINOS.....	88
ANEXOS.....	91

ÍNDICE DE FIGURAS

Fig. 1 Integración de la solución con otros módulos.....	32
Fig. 2 Estructura de la Capa de Mapeo Objeto-Relacional implementada por Symfony.....	34
Fig. 3 Diagrama de Despliegue.	34
Fig. 4 Diagrama de Clases Persistentes. (Parte 1).....	36
Fig. 5 Diagrama de Clases Persistentes. (Parte 2).....	37
Fig. 6 Diagrama de Clases Persistentes. (Parte 3).....	37
Fig. 7 Diagrama de Clases Persistentes. (Parte 4).....	38
Fig. 8 Modelo Físico de Datos. (Parte 1)	51
Fig. 9 Modelo Físico de Datos. (Parte 2)	52

INTRODUCCIÓN

El impetuoso desarrollo en el campo de la Biotecnología, la Ingeniería Genética y la industria farmacéutica a nivel mundial, ha provocado un crecimiento explosivo de la información científica en los archivos de los laboratorios tradicionales, dificultando en ocasiones su almacenamiento organizado, confiable y duradero.

Para facilitar el almacenamiento de estos importantes datos y ayudar al funcionamiento de un laboratorio surgen los Sistemas de Gestión de Información de los Laboratorios (LIMS).

Un LIMS proporciona un conjunto de herramientas basadas en sistemas informáticos, que permiten gestionar, evaluar y almacenar información de cualquier elemento que haya entrado al laboratorio, las pruebas aplicadas, los resultados obtenidos, así como el registro de los instrumentos usados, los nombres y roles de quienes ejecutaron el proceso. También puede automatizar funciones y procesos de rutina del laboratorio.

El objetivo de todo laboratorio debe ser la obtención de productos de calidad por medio del uso de mediciones precisas, fiables y adecuadas a tal fin. La implantación de sistemas de gestión de la calidad facilita la obtención de resultados óptimos y el reconocimiento externo de la competencia técnica del laboratorio a través de inspecciones o auditorías realizadas por órganos independientes con capacidad para emitir una acreditación.

El incremento en los estándares de calidad ha provocado un aumento en la cantidad de información requerida en un laboratorio de control de calidad. En consecuencia se enfrentan a elevadas cargas de trabajo, a una demanda creciente en la velocidad de generación de resultados elaborados y debidamente procesados, así como a un control riguroso de la validez y adecuación de los procedimientos utilizados en la adquisición y el tratamiento de los datos, con el fin de asegurar tanto la integridad de la información como la validez de las decisiones tomadas. [1]

Para dar solución a esta situación, en la actualidad se han creado algunos LIMS específicos para el área de control de calidad.

Representando a nuestro país en la rama de la biotecnología y la genética se destaca el Centro de Ingeniería Genética y Biotecnología de La Habana (CIGB), prestigiosa institución con más de 20 años en la investigación científica, en la producción y comercialización de productos biológicos reconocidos

a nivel mundial. Su impacto incide principalmente sobre la salud humana, las producciones agropecuarias, acuícolas, y al medio ambiente.

La calidad es la imagen del CIGB, los productos desarrollados y elaborados en el Centro se caracterizan por su eficacia y seguridad. Para garantizar que se cumpla con las pautas de calidad requeridas se creó la *Dirección de Calidad*, cuyas funciones se ponen de manifiesto a través de los *Departamentos de Control de la Calidad y Aseguramiento de la Calidad*. (Ver Anexo 1)

El *Departamento de Control de la Calidad* tiene entre sus funciones fundamentales las relacionadas con el muestreo, las especificaciones, los ensayos y la evaluación de la calidad de los productos que se generan en el Centro. Para el desempeño de las mismas, cuenta con la ayuda de dos grupos de trabajo y dos secciones: [2]

- Grupo de Desarrollo
- Grupo de Liberación Analítica
- Sección biológica compuesta por cinco laboratorios
- Sección físico-química compuesta por tres laboratorios

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

- Sección de Mejoramiento de la Calidad (SMC)
- Sección de Inspección, Auditoría y Liberación de lotes
- Grupo de Documentación
- Grupo de Metrología

Para facilitar el manejo y el control de la información generada en cada uno de los grupos y secciones anteriormente mencionadas, la Universidad de las Ciencias Informáticas (UCI) está desarrollando desde el año 2005 un LIMS para la Dirección de Calidad del CIGB.

El **Grupo Liberación Analítica** juega un papel muy importante en el control de la información que debe ser procesada para determinar la calidad de los productos analizados por los laboratorios. Es el

primer componente de la cadena lógica de procesamiento y gestión de los datos referentes a los procesos de Control de la Calidad.

El Grupo de Liberación Analítica manipula y genera grandes volúmenes de información referente a la recepción y distribución de las muestras en los laboratorios que realizarán los procesos de Control de la Calidad, así como la recogida de los resultados analíticos emitidos por los laboratorios. Para el almacenamiento de los datos utilizan libros, varias planillas y expedientes para lotes de productos en formato duro y hojas de cálculo de Microsoft Excel, documentos Microsoft Word, archivos .pdf y .txt para conservar la información en formato digital.

En el año 2007 se analizaron en los laboratorios de calidad 94399 muestras. Cada muestra genera datos primarios, reactivos y soluciones empleados en el ensayo, resultados de control para verificar que el ensayo fue satisfactorio, resultados de la muestra y análisis de los resultados, lo que significa un cúmulo de información considerable.

Los documentos generados en este departamento deben ser revisados y supervisados por la dirección del grupo. La información consultada no se obtiene con la inmediatez requerida, cuando se necesita de un tiempo de respuesta mínimo para la toma de decisiones.

La elaboración de reportes rutinarios u ocasionales es una tarea que se realiza diariamente y que se ve afectada por los lentos métodos de búsqueda entre los numerosos documentos a consultar. La cantidad de material de oficina empleado para el desempeño laboral de los trabajadores es considerable.

Luego de analizar cómo se gestiona la información en el Grupo de Liberación Analítica y las dificultades para su rápido acceso se identifica como **problema científico**: ¿Cómo mejorar el proceso de almacenamiento de la información en el Grupo de Liberación Analítica de la Dirección de Calidad del CIGB?

El problema planteado se enmarca en el **objeto de estudio**: El proceso de gestión de la información en el Grupo de Liberación Analítica de la Dirección de Calidad del CIGB.

El objeto de estudio delimita el **campo de acción**: El proceso de almacenamiento de la información en el Grupo de Liberación Analítica de la Dirección de Calidad del CIGB.

Para dar solución al problema se define como **objetivo general**: Desarrollar una Base de Datos para el Módulo Liberación Analítica del Sistema de Gestión de la Información de los Laboratorios de la Dirección de Calidad del CIGB.

Objetivos específicos:

- Diseñar el Modelo de Datos para la Base de Datos.
- Implementar las funciones de la Base de Datos.
- Validar teórica y funcionalmente el diseño de la Base de Datos.

Para lograr este objetivo, se realizarán las siguientes **tareas de investigación**:

1. Análisis de los procesos y la documentación generada en el Grupo Liberación Analítica de la Dirección de Calidad del CIGB.
2. Actualización sobre las últimas tendencias de la tecnología, métodos y herramientas en el diseño, implementación y validación de bases de datos.
3. Estudio de los artefactos y actividades asociadas al rol de Diseñador de Base de Datos propuestos por el Proceso Unificado de Desarrollo.
4. Estudio sobre la generación de la Capa de Mapeo Objeto-Relacional en Symfony.
5. Diseño de Diagrama de Clases Persistentes.
6. Diseño del Modelo Físico de Datos.
7. Análisis de la integridad de la BD.
8. Análisis de redundancia de información en la BD.
9. Normalización de la BD.
10. Implementación de disparadores y procedimientos almacenados de la BD.
11. Análisis de la seguridad de la base de datos.
12. Análisis de trazabilidad de la acciones.
13. Análisis de optimización de consultas.
14. Búsqueda de herramientas para pruebas de carga intensiva.

El trabajo consta de introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y anexos.

En el **Capítulo 1 Fundamentación Teórica** se realiza un estudio sobre el surgimiento y desarrollo de las bases de datos. Se selecciona el Sistema Gestor de Base de Datos (SGBD) atendiendo a las necesidades del sistema que se está desarrollando y la herramienta CASE más factible para el diseño de la BD. Se mencionan los artefactos y tareas para el Rol de Diseñador de Base de Datos según RUP. Se describe la herramienta de validación a utilizar.

En el **Capítulo 2 Descripción de la solución propuesta** se describen las clases persistentes, el modelo físico de datos y las tablas más importantes entre otros aspectos.

En el **Capítulo 3 Validación de la propuesta** se realizan validaciones teóricas y funcionales a la solución propuesta.

Capítulo 1

Fundamentación Teórica

En este capítulo se brinda información sobre el origen y desarrollo de las Bases de Datos (BD), las metodologías utilizadas en el mundo para su creación, las herramientas empleadas para el diseño de bases de datos y se definen las que se utilizarán en la confección de la Base Datos del Módulo Liberación Analítica del LIMS de Calidad del CIGB.

1.1 Proceso de gestión de información en el Grupo de Liberación Analítica de la Dirección de Calidad del CIGB.

El Grupo de Liberación Analítica tiene la función de recepción de muestras de lotes de productos que se elaboran en el CIGB y en otros centros del Polo Científico de la capital, como el Centro Nacional de Biopreparados (BioCen) y el Centro de Inmunología Molecular (CIM) y distribuirlas a los diferentes laboratorios dentro o fuera del CIGB donde se le realizan los ensayos correspondientes. Se encarga además, de recibir los resultados analíticos emitidos por los laboratorios que determinarán si un lote de producto está apto o no para ser liberado y puesto en el mercado. Con estos resultados se confecciona el expediente analítico de cada lote de producto. Los técnicos de este grupo almacenan también los resultados de todos los ensayos a los que fueron sometidas las muestras en los diferentes laboratorios.

Si el producto está No Conforme y se decide por el grupo de Mejoramiento de la Calidad que no es necesario realizar el resto de los ensayos, se invalidan los ensayos que estaban pendientes y se cierra el Expediente.

En el caso de que el producto esté Conforme, cuando se haya completado el Expediente del Producto se procede a elaborar un Informe de Análisis. El expediente es revisado por el Jefe del Departamento de Control de la Calidad y finalmente el Grupo de Liberación Analítica lo envía a Liberación de Lotes.

De manera general los trabajadores de este grupo son responsables de la recepción, conservación, control y destrucción de las muestras testigos de los diferentes productos terminados.

Todas estas operaciones se realizan bajo procedimientos patrones de operación, con métodos validados, equipos calificados, y materiales de referencia traceables para garantizar el cumplimiento de

las Buenas Prácticas de Laboratorio y las regulaciones internacionales que rigen el trabajo en los laboratorios de control. [4]

La información solo puede ser consultada por los trabajadores del grupo y sus superiores.

Los laboratorios de la Dirección de Calidad del CIGB y particularmente del Grupo de Liberación Analítica demandan altas velocidades en la generación de resultados elaborados y debidamente procesados, siempre garantizando la integridad de los datos.

Hasta el momento el Grupo de Liberación Analítica no posee una base de datos automatizada, pero guardan la mayoría de la información en formato digital en archivos .doc, .xls, .pdf y .txt y conservan salvadas de ellos en los discos de las computadoras del departamento.

La información almacenada en formato duro sobre los lotes de producción debe conservarse por un período no menor de 5 años, según lo estipulado. En el centro existe un archivo pasivo para guardar documentos por un período de tiempo más amplio, donde actualmente se tiene un cúmulo bastante completo de los datos generados a partir del año 1997. Sería más factible para este grupo conservar los archivos en formato electrónico permanentemente, es por ello que cada vez se hace más necesaria la puesta en marcha del LIMS.

1.2 Surgimiento y desarrollo de Bases de Datos.

Con el constante desarrollo de las tecnologías fue aumentando la necesidad de almacenamiento de información en las computadoras y se comenzaron a utilizar archivos secuenciales como almacenes de datos, que accedían rápido a la información pero de forma secuencial, y como alternativa a esta dificultad surgen los archivos indexados, que accedían directamente a la posición deseada del archivo.

Estos métodos de almacenamiento no cubrían las necesidades dada la creciente complejidad de los programas y datos a almacenar. Se necesitaba un sistema de almacenamiento que permitiera realizar operaciones complejas sin violar las restricciones, que garantizara la seguridad en el acceso de los usuarios e integridad de la información.

Para dar solución a estas necesidades surgen las bases de datos.

Una **Base de Datos** (BD) es un conjunto de datos organizados y relacionados entre sí, almacenados en la computadora con carácter más o menos permanente, puede ser utilizado por uno o varios programas de aplicación que permiten a los usuarios acceder y modificar los datos.

Las primeras bases de datos que surgieron fueron las jerárquicas que seguían una jerarquía para la organización de los datos. Esta topología tenía como inconveniente que el acceso a los datos era unidireccional, lo que hacía más complejo el camino inverso. Como variante a esta topología se crearon las bases de datos de red que incorporaban la característica de que cada nodo podía tener más de un nodo padre, pero no era aún una solución factible. Con el propósito de dar absoluta libertad a las relaciones entre tablas surgieron las bases de datos relacionales que incorporaron un lenguaje común de acceso a los datos: SQL (del inglés *Structured Query Language*, o *Lenguaje Estructurado de Consultas*) y las propiedades ACID (Atomicity, Consistency, Isolation, Durability).

1.3 Principales componentes de las Bases de Datos.

Los principales elementos que componen una BD son:

Datos: es lo que se conoce como BD propiamente dicha. Los datos serán tanto integrados, cuando se unifican varios archivos que de otra forma serían diferentes, como compartidos, cuando se comparten los datos individuales para que sean utilizados por los usuarios en diferentes fines.

Hardware: son los dispositivos de almacenamiento de la BD y los periféricos necesarios para su uso.

Software: lo constituye un conjunto de programas que se conocen como Sistemas Gestores de Bases de Datos (SGBD) que tienen como objetivo la creación de las estructuras para almacenar y manipular los datos y atender las solicitudes formuladas por los usuarios.

Usuarios: con las bases de datos se relacionan tres tipos de usuarios:

- Administrador de la BD (DBA, de sus siglas en inglés Data Base Administrator): se encarga del control total de la BD.
- Programador de aplicaciones: se encarga de crear programas de aplicación que utilizan la BD.
- Usuario final: es quien accede a la base de datos, ya sea por programas de aplicación o por un lenguaje de consultas.

1.4 Modelos de Bases de Datos.

Las BD pueden clasificarse de acuerdo a su modelo conceptual. Un modelo conceptual es una abstracción que permiten implementar un sistema eficiente de BD basándose generalmente en algoritmos y conceptos matemáticos. Los modelos más frecuentemente usados en las bases de datos son:

Bases de datos jerárquicas

Son bases de datos que almacenan su información en una estructura jerárquica, en una forma similar a un árbol invertido. Las bases de datos jerárquicas son útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. El modelo de datos jerárquico es un modelo rígido.

La poca flexibilidad de este modelo puede obligar a la introducción de redundancias cuando es preciso instrumentar, mediante el modelo jerárquico, situaciones del mundo real que no responden a una jerarquía. [5]

Base de datos de red

Es un modelo ligeramente distinto al jerárquico; su diferencia fundamental es la modificación del concepto de nodo, se permite que un mismo nodo tenga varios nodos padres.

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos, pero es difícil administrar la información utilizando este modelo.

Base de datos relacional

Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de "relaciones". Las BD relacionales son autodescriptivas por medio de sus tablas. Contiene elementos de datos (tuplas) y relaciones entre ellos, que se expresan a través de atributos comunes.

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Entre los principales Sistemas Gestores de Bases de Datos que implementan bases de datos relacionales se encuentran Oracle y MySQL.

Bases de datos orientadas a objetos (BDOO)

Este modelo es bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos, su estado y comportamiento. Incorpora todos los conceptos importantes del paradigma de objetos:

- Encapsulamiento: Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos.
- Herencia: Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- Polimorfismo: Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

Son una buena elección para aquellos sistemas que necesitan un buen rendimiento en la manipulación de tipos de datos complejos, como imágenes u objetos multimedia.

La optimización en este tipo de bases de datos en un proceso complejo, y sufren problemas de escalabilidad. Entre los Sistemas Gestores de Bases de Datos que implementan BDOO más conocidos se encuentran Gemstone y Vbase.

Bases de Datos Objeto – Relacionales

Este modelo combina las características del modelo Relacional y el Orientado a Objetos para dar una mejor solución al tratamiento de datos complejos. Le incorpora al modelo relacional, características típicas del paradigma orientado a objetos. Permite que se creen tipos de datos, funciones y operadores, dándole mayor funcionalidad y mejor desempeño. PostgreSQL es uno de los gestores de bases de datos que más se utiliza en la actualidad para este tipo de bases de datos.

Bases de Datos Multidimensionales

Se utilizan principalmente para crear aplicaciones OLAP (de sus siglas en inglés, *On Line Analytical Processing*), que se encargan de desarrollar procesos de análisis de información, orientados al acceso en modo de consulta.

Las Bases de Datos Multidimensionales poseen una estructura basada en dimensiones para realizar consultas complejas con alto rendimiento. Pueden utilizar un Sistema Gestor de Bases de Datos Relacional en estrella o Sistema Gestor de Bases de Datos Multidimensional.

El modelo seleccionado para el desarrollo de la BD fue el Relacional ya que se adapta al nivel de complejidad del problema a resolver, es un modelo maduro, ha tenido un buen desempeño y ha alcanzado un gran éxito reflejado en su amplio mercado.

1.5 Arquitectura de los Sistemas de Bases de Datos.

Para alcanzar la separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la base de datos, en 1975, el comité ANSI-SPARC (del inglés *American National Standard Institute - Standards Planning and Requirements Committee*) propuso una arquitectura de tres niveles.

En esta arquitectura, el esquema de una base de datos se define en tres niveles de abstracción:

- **Nivel interno:** Describe la estructura física de la BD mediante un esquema interno. Este esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso. Es el más cercano al almacenamiento.
- **Nivel conceptual:** Describe la estructura de toda la base de datos para una comunidad de usuarios mediante un esquema conceptual. Este esquema oculta los detalles de las estructuras de almacenamiento y describe entidades, atributos, relaciones, operaciones de los usuarios y restricciones. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar el esquema.
- **Nivel externo:** Describe varios esquemas externos o vistas de usuario. Cada uno describe la parte de la base de datos que interesa a un grupo determinado de usuarios y oculta a ese grupo el resto de la base de datos. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas. Es el más cercano al usuario.

La arquitectura de tres niveles es útil para explicar el concepto de **independencia de datos** que podemos definir como la capacidad para modificar el esquema en un nivel del sistema sin tener que modificar el esquema del nivel inmediato superior. [6]

Se pueden definir dos tipos de independencia de datos:

- **Independencia lógica:** Es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación ya sea para ampliar la base de datos o para reducirla.
- **Independencia física:** Es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual o los esquemas externos.

La arquitectura de tres niveles puede facilitar la obtención de la verdadera independencia de datos, tanto física como lógica. Sin embargo, los dos niveles de correspondencia implican un gasto extra durante la ejecución de una consulta o de un programa, lo cual reduce la eficiencia del SGBD. Es por esto que muy pocos SGBD han implementado esta arquitectura completamente. [6]

1.6 Tendencias actuales y futuras de los Sistemas de Bases de Datos.

En los últimos años, las tecnologías de bases de datos se han desarrollando vertiginosamente, han surgido BD multimedia, activas, deductivas, orientadas a objetos, seguras, temporales, móviles, paralelas, etc.

Esta nueva generación de bases de datos, se conoce como la “tercera generación” y proporciona capacidades de gestión de datos, objetos y conocimiento. Pretende responder a las necesidades de aplicaciones como: Sistemas de Información Geográfica (SIG), aplicaciones científicas, sistemas médicos, publicación digital, educación y formación, sistemas estadísticos, comercio electrónico, etc.

Dentro de los avances en el campo de las bases de datos podemos identificar tres dimensiones principales:

- **Rendimiento:** Debido a los avances en el hardware y el incremento exponencial de los datos almacenados que alcanzan petabytes, la tecnología ha evolucionado a bases de datos de alto rendimiento, destacándose las bases de datos paralelas, bases de datos en tiempo real y bases de datos en memoria principal.
- **Inteligencia:** Gran parte de la semántica de los datos que se encontraba en los programas de aplicación ha migrado hacia el servidor de datos, surgiendo así las bases de datos temporales, seguras, difusas, los almacenes de datos (*datawarehousing*) y la minería de datos (*datamining*).
- **Distribución:** Con el avance espectacular de las comunicaciones y la difusión cada día mayor del fenómeno Internet/Web también han evolucionado las bases de datos, surgieron las bases de datos distribuidas, multibases de datos; bases de datos móviles, y bases de datos web.

Se espera que en los próximos años se extiendan los modelos de datos existentes o aparezcan otros nuevos.

1.7 Sistemas Gestores de Bases de Datos.

Un Sistema Gestor de Base de Datos (SGBD) es un software que sirven de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Estos sistemas se componen de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. Permiten a los

usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos garantizando además su seguridad.

Objetivos de los SGBD

- Lograr la independencia de los datos, los programas y procesos.
- Minimizar la redundancia de datos.
- Garantizar la integridad de los datos.
- Garantizar el acceso autorizado a la BD y la recuperación ante fallas.
- Permitir el acceso simultáneo a los datos.
- Acceder a los datos eficientemente, al mismo tiempo que se brinda a los usuarios facilidad de manipulación de datos.
- Mantener el control centralizado de la BD.

Dentro de los principales SGBD propietarios se encuentra Oracle, y dentro del software libre, se destacan MySQL y PostgreSQL.

1.7.1 Oracle.

El SGBD Oracle, fabricado por Oracle Corporation, utiliza la arquitectura cliente/servidor. Ha incorporado en su sistema el modelo objeto-relacional, pero al mismo tiempo garantiza la compatibilidad con el tradicional modelo relacional de datos. Así ofrece un servidor de bases de datos híbrido. Es uno de los más conocidos y ha alcanzado un buen nivel de madurez y de profesionalidad. Se destaca por su soporte de transacciones, estabilidad y escalabilidad.

Los tipos objeto de Oracle son tipos de datos definidos por el usuario que permiten modelar entidades complejas del mundo real en una estructura que trata cada entidad como una unidad atómica simple en la base de datos. [7]

A partir de la versión 10g del año 2004, se añade a los servidores la capacidad de funcionar según el paradigma de "Grid" (o rejilla) y se ofrecen mejoras en la administración e integración de algunos elementos que previamente no funcionaban correctamente juntos.

Ventajas

- Las entidades complejas del mundo real y la lógica se pueden modelar fácilmente, lo que permite reutilizar objetos para el desarrollo de base de datos de una forma más rápida y con mayor eficiencia.

- Los programadores de aplicaciones pueden acceder directamente a tipos de objetos Oracle, sin necesidad de ninguna capa adicional entre la base de datos y la capa cliente.
- Las aplicaciones que utilizan objetos de Oracle son fáciles de entender y mantener porque soportan las características del paradigma orientado a objetos.
- Tiene buen rendimiento y hace buen uso de los recursos.
- Posee un rico diccionario de datos.
- Brinda soporte a la mayoría de los lenguajes de programación.
- Es un sistema multiplataforma, disponible en Windows, Linux y Unix.
- Permite tener copias de la base de datos productiva en lugares lejanos a la ubicación principal. Las copias de la Base de Datos productiva pueden estar en modo de lectura solamente. [8]

Desventajas

- Es un producto de elevado precio por lo que por lo general se utiliza en empresas muy grandes y multinacionales.
- Los costos de soporte técnico y mantenimiento son elevados.
- Vulnerabilidades en la seguridad de la plataforma, se hace necesario aplicar parches de seguridad.

1.7.2 MySQL.

Es un SGBD relacional, multihilo y multiusuario, creado por la compañía MYSQL AB, una de las más grandes empresas en el desarrollo de Software libre en el mundo y es la responsable de su desarrollo en un esquema de licenciamiento dual. MySQL en los últimos años ha tenido un crecimiento vertiginoso. Es la base de datos de código abierto más popular del mundo.

Esta compuesto por un servidor SQL, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (APIs). Se puede obtener también como una biblioteca multihilo que se puede enlazar dentro de otras aplicaciones para obtener un producto más pequeño, más rápido, y más fácil de manejar.

Ventajas:

- Acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.

- Seguridad: Determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas en forma de permisos y privilegios, lo que permite compartir datos sin que peligre la integridad y protegiendo determinados contenidos.
- Escalabilidad: Es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla.
- Conectividad: Permite conexiones entre máquinas con distintos sistemas operativos. Servidores Linux o Unix, usando MySQL, pueden servir datos para ordenadores con Windows, Linux, Solaris.
- Recuperación automática ante fallas: Si se da de baja de forma anormal, no suele perder información ni corromper los datos y completa las transacciones que no se terminaron.
- Utiliza el potente lenguaje de consultas SQL.
- Registros de longitud fija y variable, sin límites en el tamaño, soporta hasta 32 índices por tabla.
- Integridad referencial. Se pueden definir llaves foráneas entre tablas relacionadas para asegurarse de que un registro no puede ser eliminado de una tabla si aún está siendo referenciado por otra.
- Permite implementar subconsultas a partir de la versión 4.1 y desencadenadores desde la versión 5.0.2.
- Brinda soporte para transacciones a través de sus tablas InnoDB.

Desventajas:

En enero del presente año se anunció que la empresa norteamericana Sun Microsystems compraba MySQL, a partir de ese momento, por las leyes del bloqueo norteamericano hacia Cuba, se prohibió la descarga de MySQL desde computadoras ubicadas en el país.

1.7.3 PostgreSQL.

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

Este proyecto lleva más de una década de desarrollo, siendo hoy día, el sistema libre más avanzado con diferencia, soportando la gran mayoría de las transacciones SQL y control concurrente. [9]

A diferencia de la mayoría de los sistemas de bases de datos que usan bloqueos para el control de concurrencia, PostgreSQL mantiene la consistencia de los datos en un modelo multiversión. Esto significa que mientras se consulta una base de datos, cada transacción ve una imagen de los datos (una versión de la base de datos) como si fuera tiempo atrás, sin tener en cuenta el estado actual de los datos que hay por debajo. Esto evita que la transacción vea datos inconsistentes que pueden ser causados por la actualización de otra transacción concurrente en la misma fila de datos, proporcionando aislamiento transaccional para cada sesión de la base de datos.

Ventajas

- Soporta distintos tipos de datos además de los tipos base, como son: fechas, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, entre otros. También permite la creación de tipos propios.
- Incorpora la estructura de datos array.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios web que atienden un gran número de solicitudes.
- Puede ser instalado un número ilimitado de veces sin temor de sobrepasar la licencia.
- Posee estabilidad y confiabilidad legendarias.
- Es extensible a través del código fuente disponible sin costos adicionales.
- Es multiplataforma, disponible en Linux, *nix, Mac Os X y Windows, entre otros sistemas operativos.
- Permite la duplicación de bases de datos maestras en múltiples sitios de réplica.
- Posee interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby.
- Permite implementar reglas, vistas, disparadores, subconsultas y procedimientos almacenados.
- Posee una API abierta.

- Incluye la autenticación Kerberos y brinda soporte nativo para Secure Sockets Layer (SSL) garantizando que la comunicación con el servidor y la autenticación sean seguras.
- Posee herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL.
- Implementa el control de concurrencia multiversión, que permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente mientras la base de datos permanece disponible para consultas.

Desventajas

- La sintaxis de algunos de sus comandos no es intuitiva.
- La velocidad de respuesta que ofrece con bases de datos relativamente pequeñas puede parecer un poco deficiente, pero la mantiene al gestionar bases de datos realmente grandes.

Teniendo en cuenta la necesidad de utilizar una herramienta libre, para el desarrollo de la BD del Módulo Liberación Analítica y luego de analizar las características de diferentes SGBD, la dirección del proyecto seleccionó PostgreSQL 8.2, un gestor multiplataforma, confiable, estable, con gran escalabilidad, control de concurrencia y funcionalidades que lo destacan como uno de los SGBD más potentes en la actualidad.

1.8 Mapeo de objetos a bases de datos con Symfony.

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. También proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener.

Symfony es un completo framework, diseñado para optimizar, el desarrollo de las aplicaciones web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente con PHP 5. Es compatible con la mayoría de los SGBD como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix como en plataformas Windows.

Está basado en el patrón de diseño web Modelo Vista Controlador (MVC), que está formado por tres niveles:

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

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

Symfony accede a las bases de datos desde un contexto orientado a objetos, para que este acceso sea efectivo, se necesita una interfaz que traduzca la lógica de los objetos a la lógica relacional, la ORM (de sus siglas en inglés *Object-Relational Mapping*) o “mapeo de objetos a bases de datos”, que está formada por objetos que permiten acceder a los datos y que contienen en sí mismos el código necesario para hacerlo. Symfony utiliza Propel como ORM que permite encapsular la lógica de los datos y añadir métodos accesorios en los objetos que no tienen relación directa con una tabla. Propel utiliza la capa de abstracción de bases de datos Creole que obliga a utilizar una sintaxis específica para las consultas y a cambio optimiza y adapta el lenguaje SQL a la base de datos concreta que se está utilizando, lo que evita reescribir parte de las consultas en caso de que sea necesario cambiar de gestor. Ambos paquetes se incluyen en Symfony.

Para traducir el modelo relacional de las base de datos a un modelo de objetos de datos se necesita una descripción del modelo relacional, que se llama “esquema” (schema). La sintaxis para definir los esquemas usa el formato YAML, aunque también puede trabajar con el formato nativo de los esquemas en Propel, basado en XML. Con este esquema se construyen las clases del modelo.

La capa del modelo que es la que se encarga de la manipulación de los datos, puede dividirse en una capa de acceso a los datos y otra capa de abstracción de la BD, de manera que las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de un gestor de BD en específico, sino que utilizan otras funciones, implementadas en la capa de abstracción para realizar las consultas, incrementando la portabilidad del sistema. [10]

El LIMS de Calidad se está implementando sobre el framework de desarrollo de aplicaciones web Symfony y por tanto se generará la capa del modelo, donde los programadores desarrollarán las consultas de acceso a datos.

1.9 Rol de Diseñador de BD según el Proceso Unificado de Desarrollo.

La metodología de desarrollo de software seleccionada por la dirección del proyecto fue el Proceso Unificado de Desarrollo (*RUP*, del inglés *Rational Unified Process*).

El rol de Diseñador de BD según esta metodología debe desarrollar una serie de artefactos y actividades para garantizar el cumplimiento satisfactorio de su labor.

El Diseñador de BD es el responsable de definir el diseño detallado de la BD, incluyendo tablas, vistas, restricciones, disparadores, procedimientos almacenados y otras construcciones que se necesitan almacenar, recuperar y borrar. Debe asegurarse de que los datos persistentes se almacenan en forma coherente y eficiente.

El alcance de las actividades realizadas por el Diseñador de BD varía según el tamaño y complejidad de la aplicación y el mecanismo de almacenamiento de datos persistentes que será utilizado en el proceso.

El Diseñador de BD debe tener sólido conocimiento sobre:

- Modelado de datos, diseño de BD.
- Análisis y diseño de técnicas orientadas a objetos.
- Arquitectura del sistema, incluyendo la mejora del rendimiento de la BD y del sistema.
- Administración de BD.
- Comprensión del entorno y lenguaje de implementación.

Actividades que debe realizar:

- Desarrollar el modelo lógico de la BD.

Es una actividad opcional, su propósito es dar una visión idealizada de la estructura lógica de los datos, en términos de entidades y relaciones, es independiente de la implementación de la BD.

- Desarrollar el diseño físico de la BD.

Su refiere a modelar los elementos que representan la estructura física detallada de la BD:

- Definir Dominios.
- Crear los elementos iniciales del diseño físico de la BD.
- Definir las tablas de referencia.
- Crear clave primaria y restricciones de integridad.
- Definir las reglas de integridad referencial y de la información.
- Normalizar el diseño de la BD para su optimización.
- Optimizar el acceso a los datos.

- Definir las características de almacenamiento.
- Diseño de procedimientos almacenados.
- Revisar los resultados.

Se debe revisar regularmente que la estructura de la BD sea consistente.

El artefacto que se obtiene como resultado de las actividades desarrolladas por el Diseñador de BD es el Modelo de Datos. (Ver Anexo 2)

El Modelo de Datos describe las representaciones lógicas y físicas de los datos persistentes utilizados por la aplicación. Puede ser creado a partir de un conjunto de clases del diseño, a partir de un modelo entidad-relación creado por el diseñador de BD o mediante ingeniería inversa a partir de una BD existente.

Es necesario el Modelo de Datos aún cuando el mecanismo de almacenamiento de los datos persistentes no se base en la tecnología orientada a objetos. Se utiliza para definir la persistencia de las estructuras de datos y la correspondencia entre las clases de diseño y las estructuras de datos persistentes.

El Diseñador de BD es responsable de la integridad del Modelo de Datos y de asegurar que los datos modelados sean correctos, coherentes y comprensibles.

1.10 Lenguaje Unificado de Modelado (UML).

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. [11]

No es una guía para realizar el análisis y diseño orientado a objetos. Es un lenguaje que permite modelar sistemas con tecnología orientada a objetos.

Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementarlo.

UML es el lenguaje de modelado aplicado por la Herramientas CASE utilizada en el proyecto.

1.11 Herramientas CASE.

Las siglas CASE corresponden a las iniciales de: Computer Aided Software Engineering; y en su traducción al Español significa Ingeniería de Software Asistida por Computación. Las herramientas CASE permiten modelar los procesos de negocios de las empresas y desarrollar los Sistemas de Información. [12]

Una herramienta CASE suele incluir:

- Un diccionario de datos para almacenar información sobre los datos de la aplicación de bases de datos.
- Herramientas de diseño para dar apoyo al análisis de datos.
- Herramientas que permitan desarrollar el modelo de datos, así como los esquemas conceptual y lógico.
- Herramientas para desarrollar los prototipos de las aplicaciones.

El uso de herramientas CASE en el modelado de bases de datos, permite realizar las tareas de diseño de un modo más eficiente, si nos referimos a los costes de producción y efectivo teniendo en cuenta el grado en que el sistema satisface las necesidades de los usuarios.

1.11.1 Visual Paradigm for UML 6.1 Enterprise Edition.

Visual Paradigm for UML 6.1 Enterprise Edition fue la herramienta CASE seleccionada en el proyecto para modelar los procesos de negocio y el desarrollo del sistema. Se utilizará para realizar el diseño del modelo de datos. Utiliza UML como lenguaje de modelado.

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.

Visual Paradigm for UML permite:

- Modelar procesos de negocio
- Administrar requerimientos.
- Importar archivos desarrollados con Rational Rose.
- Importar y exportar archivos XML.
- Generar código e ingeniería inversa.
- Generar una capa Objeto- Relacional fiable, escalable, y de alto rendimiento.
- Modelar visualmente el diseño lógico y físico de datos.
- Automatizar el mapeo entre el modelo de objetos y el modelo de datos.

Soporta una amplia gama de bases de datos donde se incluyen: Oracle, DB2, Cloudscape / Derby, Sybase Adaptive Server Enterprise, Sybase SQL Anywhere, Microsoft SQL Server, PostgreSQL, MySQL y otros. Permite una fácil migración entre bases de datos ya que el mismo conjunto de objetos ORM Java puede trabajar con diferentes bases de datos.

No sólo es una aplicación independiente, se puede integrar a los principales IDEs: Eclipse/WebSphere, Borland JBuilder, NetBeans / Sun ONE, IntelliJ IDEA, Oracle JDeveloper, BEA WebLogic Workshop.

Es multiplataforma, disponible para los Sistemas Operativos Linux, Windows, y Mac OS.

1.12 Herramientas de desarrollo.

1.12.1 EMS SQL Manager 2005 for PostgreSQL.

Para la administración del servidor de BD PostgreSQL se seleccionó la herramienta *EMS SQL Manager 2005 for PostgreSQL versión 3.1*.

Esta aplicación trabaja con versiones de PostgreSQL superior a la 8.2 y soporta sus últimas características como espacios de tablas y nombres de argumentos en funciones. Su interfaz gráfica es amigable e incluye un modo guiado de trabajo muy claro. [13]

Características principales:

- Administración fácil de todos los objetos PostgreSQL.
- Administración efectiva de seguridad.
- Capacidades de exportación e importación de datos.
- Modo guiado para labores de mantenimiento.
- Acceso al Servidor PostgreSQL a través del protocolo HTTP.

1.12.2 EMS Data Generator 2005 for PostgreSQL.

La herramienta *EMS Data Generator 2005 for PostgreSQL versión 2.3* se seleccionó para la generación de datos en pruebas de carga intensiva a la BD.

Permite generar datos para varias tablas a la vez, seleccionar las tablas y campos que se quieren llenar, definir rango de valores para los mismos, generar campos de tipo char utilizando máscara, además de obtener los valores que son resultado de las consultas SQL. Brinda la posibilidad de definir plantillas de valores para su utilización futura.

Características:

- Posee una interfaz de usuario amigable.
- Genera datos para BD diferentes, ubicadas en una misma computadora.
- Soporta todos los tipos de datos que posee PostgreSQL: arreglos, direcciones de red, tipos geométricos, etc.
- Genera datos de diferentes formas para cada campo, ya sea desde una lista de valores predefinida, valores aleatorios haciendo uso de caracteres previamente definidos o valores incrementales.
- Control automático sobre la integridad referencial.
- Permite generar valores nulos para un por ciento definido de los valores generados para cada campo.
- Permite almacenar todos los parámetros de generación en la sesión del asistente en uso.

Conclusiones

Luego de realizar un estudio profundo sobre las bases de datos y otros aspectos necesarios para obtener una BD robusta para el almacenamiento de los datos del Módulo Liberación Analítica se seleccionó el modelo de base de datos relacional, RUP como metodología para el desarrollo de la actividad diseñar BD y UML como lenguaje de modelado. Se escogió Visual Paradigm for UML 6.1 Enterprise Edition para la realización del diagrama de clases persistentes y para el modelo de datos. Se decidió utilizar el SGBD PostgreSQL 8.2, utilizando para la administración de la BD EMS SQL Manager for PostgreSQL y para la generación de datos de prueba EMS Data Generator for PostgreSQL. Se determinó utilizar el framework Symfony para la generación de las clases de la capa de acceso a datos.

Capítulo **2**

Descripción y análisis de la solución propuesta

En este capítulo se presentan los requisitos funcionales y no funcionales que debe satisfacer la BD, las clases persistentes del módulo Liberación Analítica y se describe el diseño físico de la BD.

2.1 Requisitos funcionales y no funcionales del Módulo Liberación Analítica.

2.1.1 Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. A continuación se relacionan los requisitos funcionales obtenidos del trabajo de los analistas que necesitan intercambiar información con la BD.

1. Gestionar muestras.

- 1.1. Registrar recepción de nueva muestra.
- 1.2. Modificar datos de muestra.
 - 1.2.1. Registrar traza.
- 1.3. Buscar y visualizar muestra.
- 1.4. Buscar y visualizar expediente analítico.

2. Administrar libro de registros de cada producto:

- 2.1. Crear nuevo libro por etapa.
- 2.2. Insertar datos en el libro.
- 2.3. Modificar datos del libro.
 - 2.3.1. Registrar traza.
- 2.4. Buscar y visualizar libro.
- 2.5. Generar reportes.

3. Administrar libro de registros por centros:

- 3.1. Crear nuevo libro.
- 3.2. Insertar datos en el libro.
- 3.3. Modificar estructura general del libro.
 - 3.3.1. Registrar traza.
- 3.4. Modificar datos de muestras del libro.

3.4.1. Registrar traza.

3.5. Buscar y visualizar libro.

3.6. Generar reportes.

4. Gestionar datos de muestras testigos:

4.1. Registrar nueva muestra.

4.2. Modificar datos de muestra.

4.2.1. Registrar traza.

4.3. Registrar salida de muestra.

4.4. Registrar destrucción de muestra.

4.5. Registrar revisión de estado de las muestras.

5. Administrar libro de registros de muestras testigos por producto:

5.1. Crear nuevo libro por etapa.

5.2. Buscar y visualizar.

5.3. Modificar los datos del libro.

5.3.1. Registrar traza.

5.4. Generar reportes.

6. Gestionar Solicitud de muestras testigos:

6.1. Registrar Solicitud en planilla SIC-0935.

6.2. Buscar y visualizar planilla.

6.3. Modificar los datos de la planilla.

6.3.1. Registrar traza.

6.4. Generar reportes.

6.4.1. Imprimir reporte.

7. Gestionar Control de Salida de muestras testigos:

7.1. Registrar Control de Salida de muestras testigos en planilla SIC-0825.

7.2. Buscar y visualizar planilla.

7.3. Modificar los datos de la planilla.

7.3.1. Registrar traza.

7.4. Generar reportes.

8. Gestionar Revisión del Estado de muestras testigos:

8.1. Registrar Revisión de Estado de muestras testigos en modelo SIC-0824.

8.2. Buscar y visualizar planilla.

8.3. Modificar los datos de la planilla.

8.3.1. Registrar traza.

8.4. Generar reportes.

8.5. Revisar modelo SIC-0824.

9. Gestionar Solicitud de destrucción de productos:

9.1. Registrar Solicitud de destrucción de productos en modelo SIC-0830.

9.2. Generar automáticamente.

9.3. Buscar y visualizar modelo.

9.4. Modificar los datos del modelo.

9.4.1. Registrar traza.

9.5. Generar reportes.

10. Gestionar Análisis de Características Organolépticas:

10.1. Registrar análisis en planilla SIC-0123.

10.2. Buscar y visualizar planilla.

10.3. Modificar datos de planilla.

10.3.1. Registrar traza.

11. Gestionar resultados de las pruebas:

11.1. Buscar y visualizar resultados de pruebas.

11.2. Gestionar resultados fuera de especificación

11.3. Registrar resultado de ensayos realizados fuera del CIGB.

12. Gestionar Análisis de resultados fuera de especificación:

12.1. Registrar Análisis en planilla SIC-0837.

12.2. Buscar y visualizar planilla.

12.3. Modificar los datos de la planilla.

12.3.1. Registrar traza.

12.4. Generar reportes.

12.5. Revisar planillas SIC-0837

13. Gestionar Cálculo de la actividad específica:

13.1. Crear nueva planilla SIC-0709.

13.2. Generar automáticamente el cálculo.

13.3. Buscar y visualizar planilla.

13.4. Modificar datos de planilla.

13.4.1. Registrar traza.

14. Gestionar Cálculo relación PRP:TT:

14.1. Crear nueva planilla SIC-0795.

14.2. Generar automáticamente el cálculo.

14.3. Buscar y visualizar planilla.

14.4. Modificar datos de planilla.

14.4.1. Registrar traza.

15. Gestionar Cálculo de grupos funcionales residuales:

15.1. Crear nueva planilla SIC-0794.

15.2. Generar automáticamente el cálculo.

15.3. Buscar y visualizar planilla.

15.4. Modificar datos de planilla.

15.4.1. Registrar traza.

16. Gestionar Cálculo de la actividad biológica por bulbo:

16.1. Crear nueva planilla SIC-0938.

16.2. Generar automáticamente el cálculo.

16.3. Buscar y visualizar planilla.

16.4. Modificar datos de planilla.

16.4.1. Registrar traza.

17. Gestionar Notificación de resultados no satisfactorios:

17.1. Registrar Notificación de Resultados no satisfactorios en planilla SIC-0931.

17.2. Generar planilla automáticamente.

17.3. Buscar y visualizar planilla.

17.4. Modificar los datos de la planilla.

17.4.1. Registrar traza.

17.5. Generar reportes.

18. Gestionar Informe de Análisis:

18.1. Registrar Informe de Análisis en planilla SIC-0092.

18.2. Buscar y visualizar planilla.

18.3. Modificar los datos de la planilla.

18.3.1. Registrar traza.

18.4. Generar reportes.

19. Gestionar Certificado de Análisis del Ingrediente Farmacéutico:

19.1. Registrar Certificado en planilla SIC-0300.

19.2. Buscar y visualizar planilla.

19.3. Modificar los datos de la planilla.

19.3.1. Registrar traza.

19.4. Generar reportes.

20. Gestionar Certificado de Análisis del Producto Final:

- 20.1. Registrar Certificado en planilla SIC-0301.
- 20.2. Buscar y visualizar planilla.
- 20.3. Modificar los datos de la planilla.
 - 20.3.1. Registrar traza.
- 20.4. Generar reportes.

21. Gestionar Documento que avala la liberación para la comercialización:

- 21.1. Registrar Documento en planilla SIC-0001.
- 21.2. Buscar y visualizar planilla.
- 21.3. Modificar los datos de la planilla.
 - 23.2.1. Registrar traza.
- 21.4. Generar reportes.

22. Gestionar Informe analítico semanal de resultados:

- 22.1. Generar informe.
- 22.2. Buscar y visualizar informe.
- 22.3. Modificar datos del informe.

23. Generar reportes generales:

- 23.1. Visualizar nuevos resultados registrados por analistas del CIGB.
- 23.2. Visualizar resultados de ensayos pendientes y atrasados.
- 23.3. Visualizar Informe de análisis que ya pueden ser generados.
- 23.4. Visualizar cuándo una concentración está lista para repartir las muestras que dependían de ella.
- 23.5. Visualizar Resultados no conformes.
- 23.6. Avisar la finalización del almacenamiento de muestras testigos.
- 23.7. Avisar que está en tiempo de generar reporte de costos (mensual).

24. Gestionar informe de costos mensuales:

- 24.1. Generar informe.
- 24.2. Buscar y visualizar informe.
- 24.3. Modificar datos del informe.
 - 24.3.1. Registrar traza.

25. Gestionar resumen semestral de productos:

- 25.1. Generar resumen semestral por producto.
- 25.2. Almacenar resumen.
- 25.3. Buscar y visualizar resumen.

25.4. Modificar resumen.

25.4.1. Registrar traza.

2.1.2 Requisitos no funcionales aplicables a la BD

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Son las características que hacen al producto atractivo, usable, rápido o confiable.

- **Usabilidad.**

Se debe garantizar un acceso fácil y rápido a los usuarios.

- **Rendimiento.**

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

- **Soporte.**

Se requiere de la instalación de un servidor Web Apache 1.3 o superior con funcionalidades relacionadas con el manejo de la Base de Datos con un gestor PostgreSQL y el servicio de interpretación de códigos con PHP 5 o superior.

- **Portabilidad.**

El sistema debe ser multiplataforma.

- **Seguridad.**

Se debe garantizar que la información sensible solo pueda ser vista por los usuarios con el nivel de acceso adecuado. El sistema debe contar con protección contra acciones no autorizadas o que puedan afectar la integridad de los datos. Contará con un registro de trazas de documentos y planillas modificadas por los usuarios, para garantizar el control de las operaciones de este tipo en el sistema. No se podrá acceder al sistema desde ninguna computadora personal que esté fuera del CIGB.

- **Confiabilidad.**

El sistema será usado y administrado solamente por trabajadores del área de Calidad del CIGB, por lo tanto la información que fluirá en el mismo, será la real emitida por cada uno de los grupos o departamentos. Podrán acceder a visualizar ciertas informaciones, directivos de otras áreas, previa consulta con la dirección del proyecto y los desarrolladores de la aplicación.

- **Restricciones en el diseño.**

El SGBD debe ser PostgreSQL. Se usará Visual Paradigm for UML como herramienta CASE para el modelado de los diferentes artefactos.

2.2 Estrategia de integración de la solución con otros módulos.

Para la integración de todos los módulos de la BD del sistema, se realizarán las siguientes actividades (Ver Fig. 1):

Identificar las CP (Clases Persistentes) para cada módulo: Se identifican las clases persistentes de cada módulo a partir del diagrama de clases del diseño que se tiene del trabajo realizado por los diseñadores.

Integrar las Clases Persistentes de todos los Módulos: Se integran las clases persistentes de todos los módulos en único diagrama de clases que incluye las relaciones entre clases de varios módulos en caso de que existan. Al finalizar, la integración debe validarse y si no constituye una correcta representación de las clases persistentes del sistema en su totalidad, deben revisarse los diagramas de clases persistentes de cada módulo nuevamente.

Diseñar el Modelo Físico de Datos de cada Módulo: Se realiza un modelo para cada módulo para representar la estructura Física de la BD, a través de tablas y relaciones.

Describir los campos a utilizar en cada Módulo: Cada módulo debe realizar una descripción de los campos que poseen cada una de sus tablas.

Integrar los Modelos de Datos de todos los Módulos: Se integran los Modelos de Datos realizados en cada módulo en único modelo. La integración debe validarse y si no constituye una correcta representación de la estructura de la BD del sistema deben revisarse los modelos de cada módulo nuevamente.

Integrar las descripciones de los campos de todos los Módulos: Con las descripciones de todos los campos de las tablas de los diferentes módulos debe conformarse un diccionario de datos, para que todos los diseñadores de BD y desarrolladores de la aplicación utilicen la misma nomenclatura.

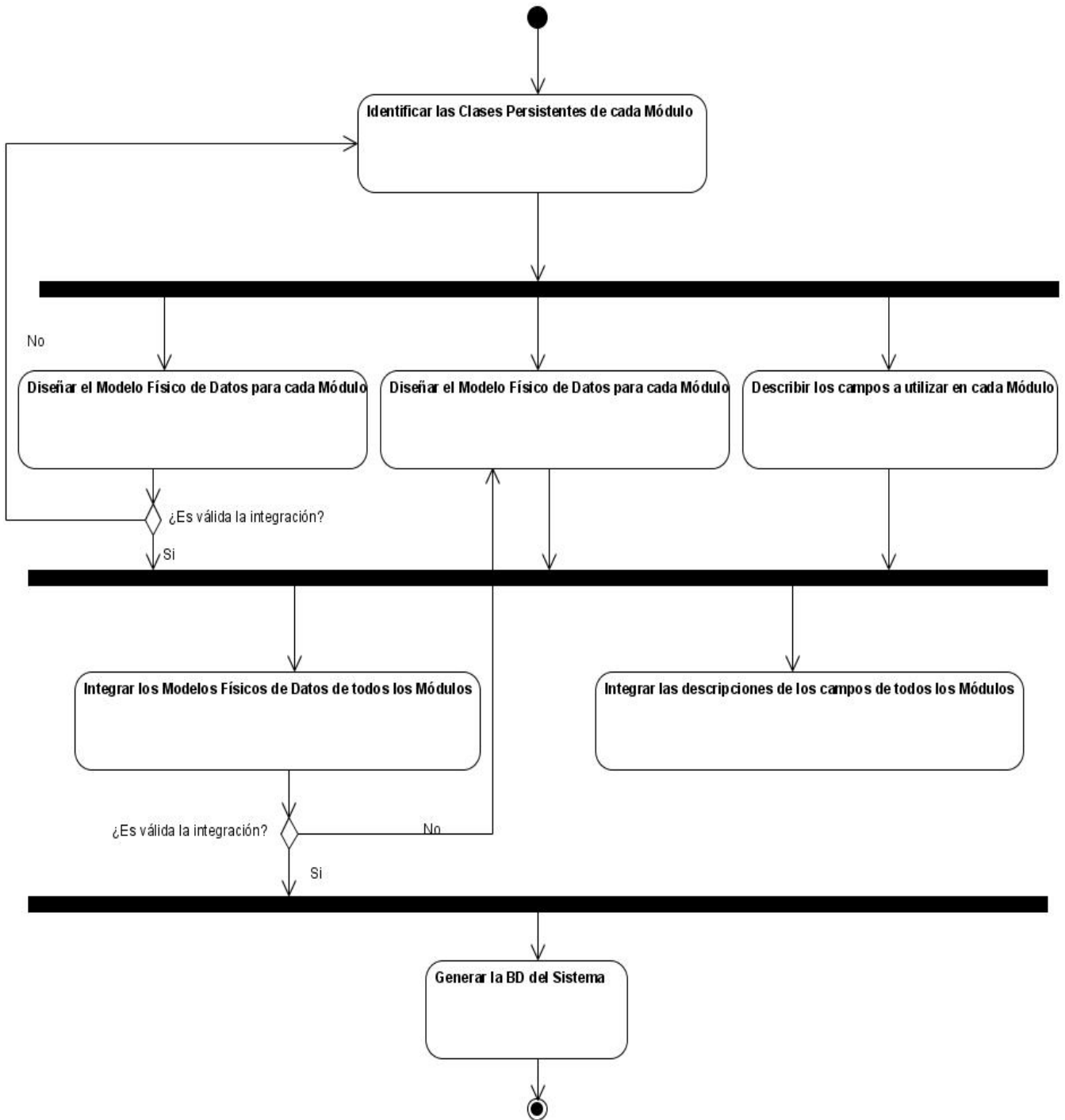


Fig. 1 Integración de la solución con otros módulos.

2.3 Descripción de la Arquitectura de la Base de Datos.

La Base de Datos contiene 49 tablas en el modelo físico que serán utilizadas en el Módulo Liberación Analítica y algunas de ellas serán utilizadas en otros módulos cuando se realice la integración, utilizando principalmente las tablas de *lote*, *muestra* y *resultado*, en las que se almacena el mayor volumen de información de este grupo.

En el SGBD PostgreSQL se implementaron funciones disparadoras para validar la integridad referencial de las tablas, registrar automáticamente las trazas cuando se realizan modificaciones en los datos, así como funciones para comprobar la validez de los datos introducidos.

La implementación de la capa de acceso a datos, es responsabilidad de los programadores de acuerdo a la distribución del proyecto y se realizará en la capa del Modelo, según la arquitectura Modelo-Vista-Controlador que implementa Symfony. Para obtener la capa de Mapeo Objeto-Relacional se necesita el esquema que describe el modelo relacional donde se definen las tablas, relaciones y las características de sus columnas. El archivo `schema.yml` se generó en este trabajo mediante la tarea de línea de comandos `< symfony propel-build-schema` y se guardan en el directorio `config/` del proyecto que se definió. Las opciones de conexión con la base de datos se especifican en el archivo `databases.yml`.

Las clases del modelo se generaron a partir del esquema mediante la tarea `< symfony propel-build-model`.

Al ejecutar ese comando, se analiza el esquema y se generan las clases base del modelo, que se almacenan en el directorio `lib/model/om/` del proyecto. Estas clases no se modifican porque se sobrescriben cada vez que se genera el modelo.

Para la tabla *lote* se generaron las clases base:

- BaseLote
- BaseLotePeer

También se crean las clases objeto y las clases peer en el directorio `lib/model/`, que heredan de las clases base y son a las que se les añaden los métodos propios, ya que no se modifican cuando se vuelve a construir el modelo.

Las clases objeto representan un registro de la base de datos. Permiten acceder a las columnas de un registro y a los registros relacionados.

Las clases peer permiten obtener los registros de las tablas de la base de datos. Sus métodos devuelven normalmente un objeto o una colección de objetos de la clase objeto relacionada.

Para la tabla *lote* se generó la clase objeto:

- Lote

Y la clase peer:

- LotePeer

De manera general, para todas las tablas de la BD se generaron las clases del modelo de acuerdo a la Fig. 2.

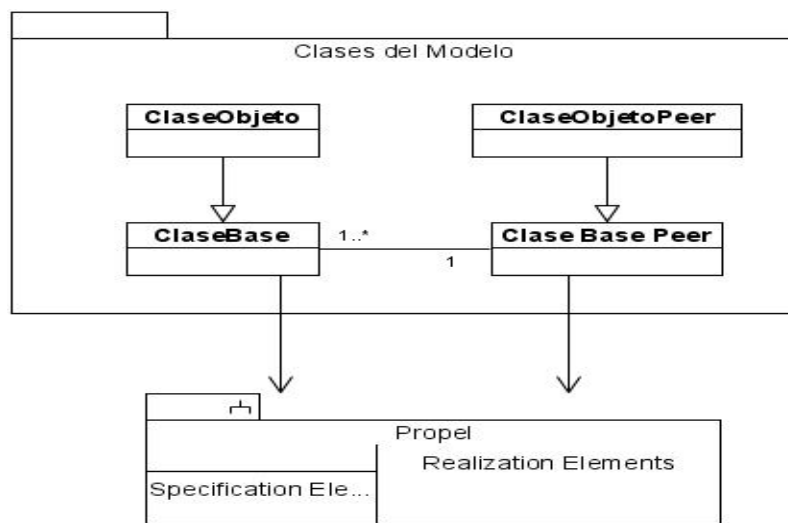


Fig. 2 Estructura de la Capa de Mapeo Objeto-Relacional implementada por Symfony.

El despliegue de la aplicación se realizará como describe el siguiente diagrama (Ver Fig.3):

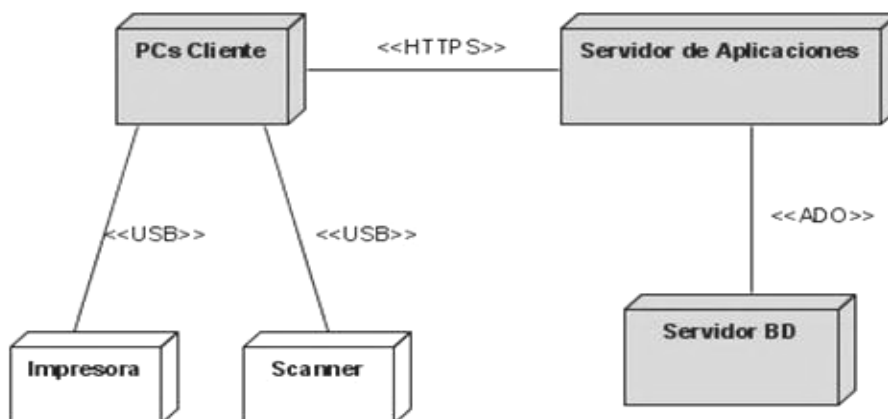


Fig. 3 Diagrama de Despliegue.

2.4 Modelo de objetos del negocio.

El modelo de objetos es uno de los artefactos que se genera en el flujo de trabajo Modelamiento del Negocio, en el que se representan las relaciones entre los trabajadores y las entidades del negocio. Para la identificación de las clases persistentes que representan entidades a almacenar en la BD se consultó el modelo de objetos diseñado por los analistas.

2.5 Diagrama de clases del diseño.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar.

Una clase de diseño es una construcción similar en la implementación del sistema. Esta describe un conjunto de objetos que comparten las mismas responsabilidades, relaciones, operaciones, atributos y semántica. Se realizó un análisis de los diagramas de clases del diseño obtenidos del trabajo de los diseñadores del proyecto que facilitó la tarea de identificar las clases persistentes.

2.6 Clases persistentes.

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Todas las clases identificadas en el análisis no son persistentes. Por lo general las clases persistentes tienen como origen las clases identificadas como entidad porque ellas modelan la información y el comportamiento asociado de algún fenómeno o concepto. Las clases persistentes referencian directamente las entidades lógicas y sus atributos.

2.6.1 Diagrama de Clases Persistentes.

Luego de interactuar directamente con el cliente y realizar un estudio de la documentación generada por los analistas del módulo, se refinaron las clases persistentes. (Ver Fig. 4-7).

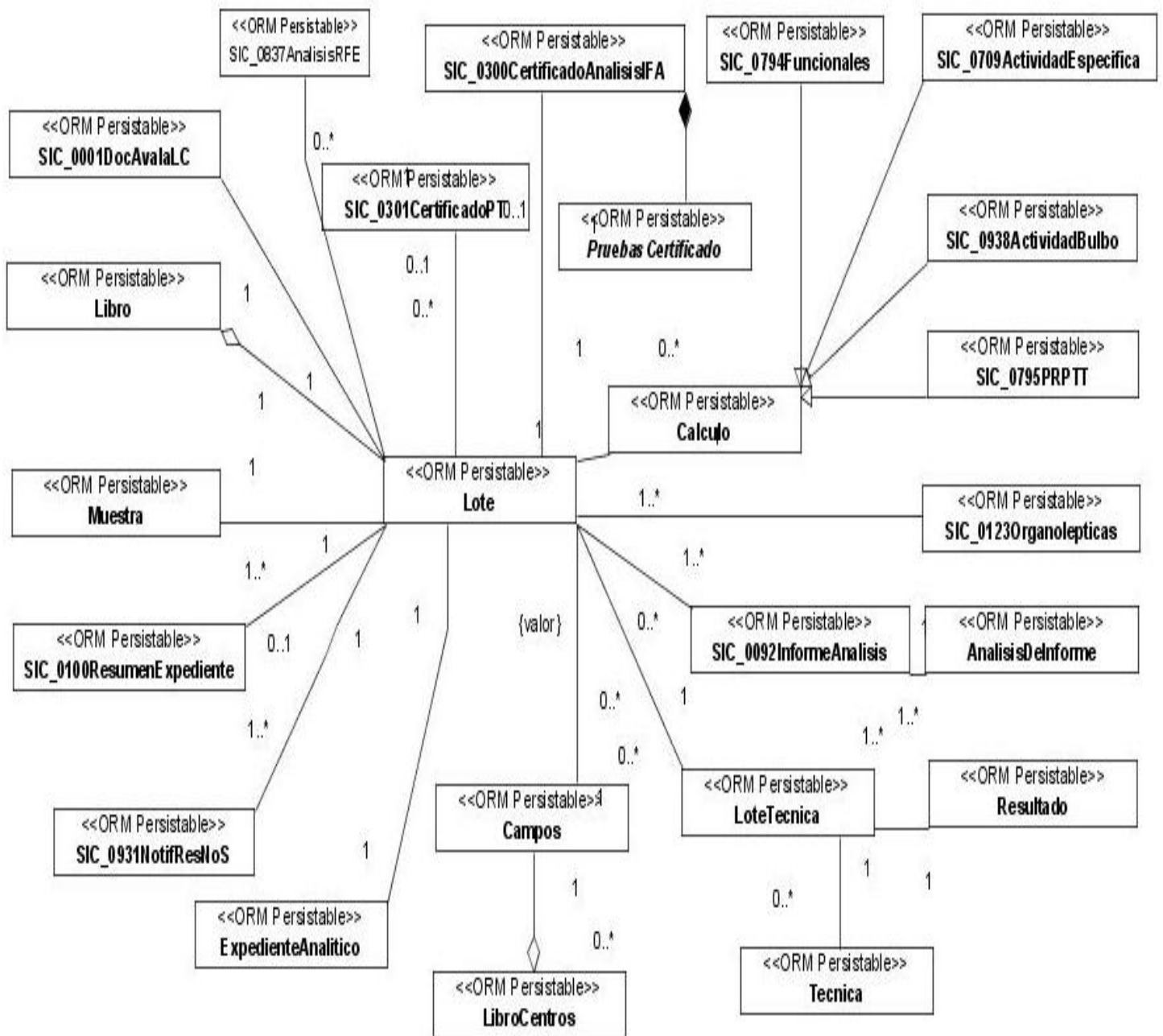


Fig. 4 Diagrama de Clases Persistentes. (Parte 1)

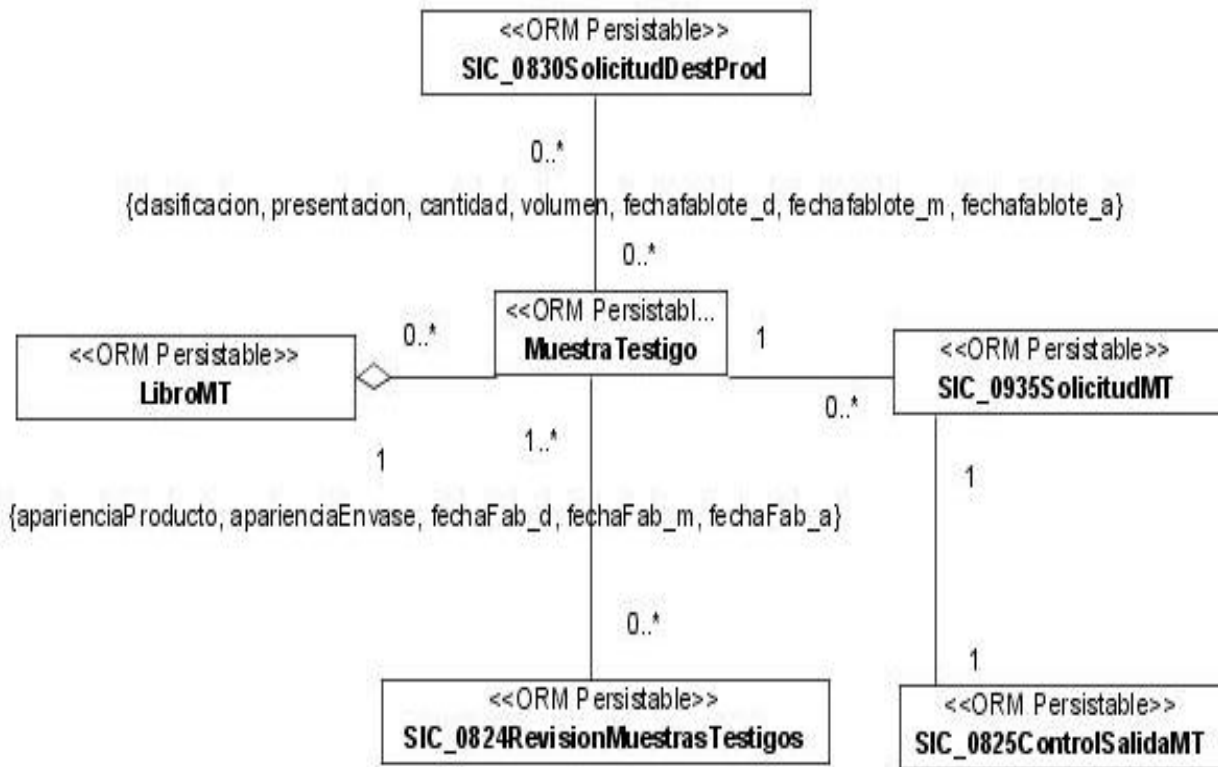


Fig. 5 Diagrama de Clases Persistentes. (Parte 2)

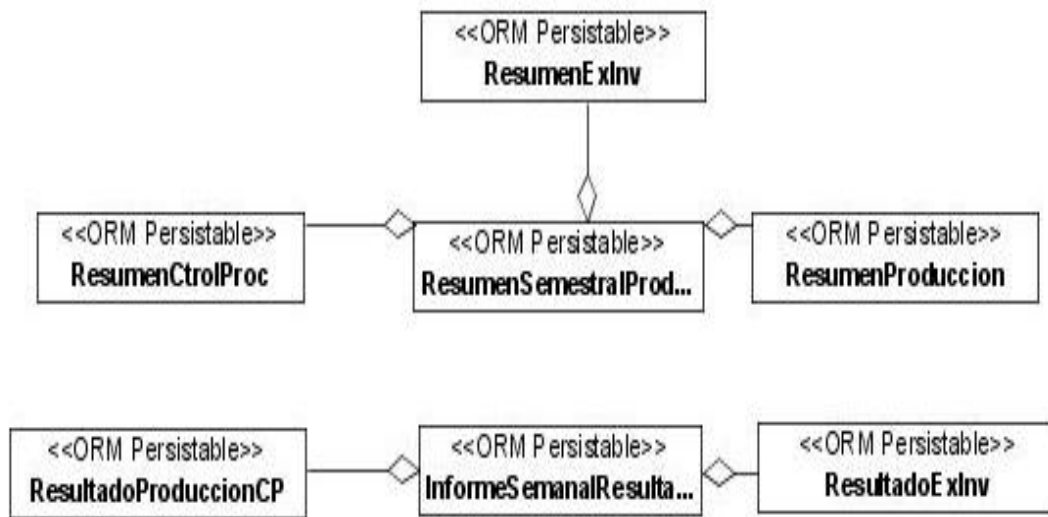


Fig. 6 Diagrama de Clases Persistentes. (Parte 3)

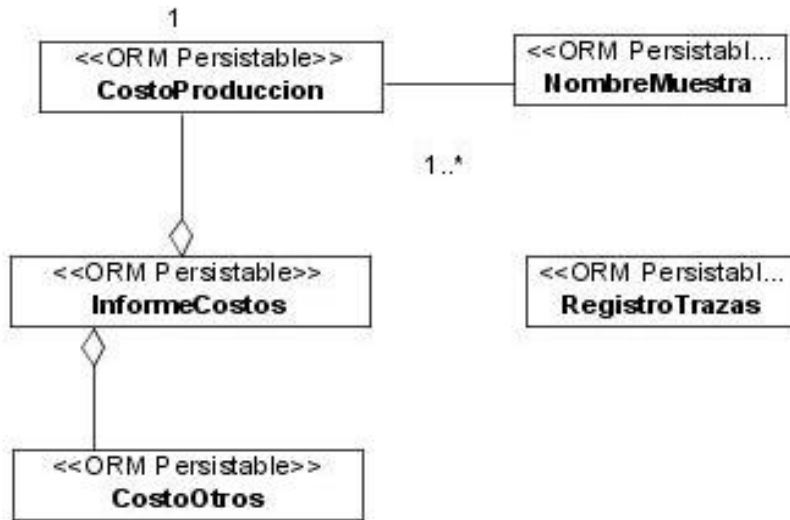


Fig. 7 Diagrama de Clases Persistentes. (Parte 4)

2.6.2 Descripción de las Clases Persistentes.

Todas las clases persistentes que se describen incluyen entre sus funcionalidades los métodos Get() y Set() para cada uno de sus atributos.

Nombre: Libro	
Tipo de clase: Entidad	
Atributo	Tipo
nombreProducto	String
nombreEtapa	String
origenProducto	String
Responsabilidades:	
Nombre:	Libro()
Descripción:	Constructor

Nombre: Muestra	
Tipo de clase: Entidad	
Atributo	Tipo
noControl	Integer
nombreRecibe	String
nombreEntrega	String
fechaRecLote_D	Short
fechaRecLote_M	Short
fechaRecLote_A	Integer
almacenamiento	Integer
cantidad	Integer
observaciones	String
autorizadoNo	Boolean

Responsabilidades:	
Nombre:	Muestra()
Descripción:	Constructor

Nombre: Lote	
Tipo de clase: Entidad	
Atributo	Tipo
lote	String
origenProducto	String
destinoProducto	String
loteOrigen	String
fechaFabLote_D	Short
fechaFabLote_M	Short
fechaFabLote_A	Integer
vistoNo	Boolean
Responsabilidades:	
Nombre:	Lote()
Descripción:	Constructor

Nombre: Tecnica	
Tipo de clase: Entidad	
Atributo	Tipo
tecnica	String
ppo	String
Responsabilidades:	
Nombre:	Tecnica()
Descripción:	Constructor

Nombre: LoteTecnica	
Tipo de clase: Entidad	
Atributo	Tipo
Responsabilidades:	
Nombre:	LoteTecnica ()
Descripción:	Constructor

Nombre: Resultado	
Tipo de clase: Entidad	
Atributo	Tipo
idResultado	Integer
conformeoNo	Boolean
fechaSalResultado_D	Short
fechaSalResultado_M	Short
fechaSalResultado_A	Integer
ubicacionFueraCIGB	String
fechaRecepcion_D	Short
fechaRecepcion_M	Integer

fechaRecepcion_A	Integer
realizadoPor	String
conformeono	Boolean
Responsabilidades:	
Nombre:	Resultado ()
Descripción:	Constructor

Nombre: SIC_0123Organolepticas	
Tipo de clase: Entidad	
Atributo	Tipo
idSIC	Integer
Folio	Integer
fechaRec_D	Short
fechaRec_M	Short
fechaRec_A	Integer
cantBulbos	Integer
recibidosPor	String
disolventeUsado	Integer
cantDisolvente	Integer
realizaResuspension	String
filtracion	Boolean
tamannoPoros	Integer
filtradoPor	String
tipoAguja	String
especificaciones	String
cumpleoNo	Boolean
observaciones	String
confPor	String
fechaConf_D	Short
fechaConf_M	Short
fechaConf_A	Integer
fechaVen_D	Short
fechaVen_M	Short
fechaVen_A	Integer
revisadoPor	String
fechaRevision_D	Short
fechaRevision_M	Short
fechaRevision_A	Integer
Responsabilidades:	
Nombre:	SIC_0123Organolepticas()
Descripción:	Constructor

Nombre: Calculos	
Tipo de clase: Entidad	
Atributo	Tipo
idSIC	Integer
folio	Integer

conformeoNo	Boolean
limiteAceptacion	Float
realizadoPor	String
fechaRealizado_D	Short
fechaRealizado_M	Short
fechaRealizado_A	Integer
recibidoPor	String
fechaRecibido_D	Short
fechaRecibido_M	Short
fechaRecibido_A	Integer
revisadoPor	String
fechaRevisado_D	Short
fechaRevisado_M	Short
fechaRevisado_A	Integer
observaciones	String
valorObtenido	Float
estado	String
Responsabilidades:	
Nombre:	Calculos()
Descripción:	Constructor

La clase *Calculos* es una generalización de las clases *SIC_0795PRP_TT*, *SIC_0709ActividadEspecificas*, *SIC_0794Funcionales*, *SIC_0938ActividadBulbo*.

Nombre: SIC_0795PRPTT	
Tipo de clase: Entidad	
Atributo	Tipo
concentracionTT	Float
resConSIC107	String
concPRPTT	Float
resSIC762	String
concCarbLib	Float
resCarbSIC762	String
Responsabilidades:	
Nombre:	SIC_0795PRPTT()
Descripción:	Constructor
Responsabilidades:	
Nombre:	CalcularRelacionPRPTT()
Descripción:	Calcula la relación PRP/TT.

Nombre: SIC_0709ActividadEspecificas	
Tipo de clase: Entidad	
Atributo	Tipo
concA	Float
metodoA	String

pPOA	String
sICA	String
folioA	Integer
concB	Float
metodoB	String
pPOB	String
sICB	String
folioB	Integer
Responsabilidades:	
Nombre:	SIC_0709ActividadEspecifica()
Descripción:	Constructor
Responsabilidades:	
Nombre:	CalcularActividadEspecifica()
Descripción:	Calcula la Actividad Específica.

Nombre: SIC_0794Funcionales	
Tipo de clase: Entidad	
Atributo	Tipo
valorGS	Float
folioSIC0759C	Integer
Responsabilidades:	
Nombre:	SIC_0794Funcionales()
Descripción:	Constructor
Responsabilidades:	
Nombre:	CalcularGruposFuncionales()
Descripción:	Calcula el valor de los grupos funcionales residuales.

Nombre: SIC_0938ActividadBulbo	
Tipo de clase: Entidad	
Atributo	Tipo
nP	String
concentración	Float
actBiolA	Float
pPO	String
sic	String
folioSIC	Integer
volumenBulboB	Float
pNO	String
aC	String
folioAC	Integer
Responsabilidades:	
Nombre:	SIC_0938ActividadBulbo()
Descripción:	Constructor
Nombre:	ActBiol()
Descripción:	Calcula el valor de la actividad biológica.
Responsabilidades:	

Nombre:	CalcularActividadBiologica()
Descripción:	Calcula el valor de la actividad biológica por bulbo

Nombre: LibroCentros	
Tipo de clase: Entidad	
Atributo	Tipo
idCentro	Integer
nombreCentro	String
anno	Integer
Responsabilidades:	
Nombre:	LibroCentros()
Descripción:	Constructor

Nombre: Campos	
Tipo de clase: Entidad	
Atributo	Tipo
idNombre	Integer
nombreCampo	String
visibilidad	Boolean
Responsabilidades:	
Nombre:	Campos()
Descripción:	Constructor

Nombre: SIC_0092InformeAnalisis	
Tipo de clase: Entidad	
Atributo	Tipo
idSIC	Integer
folio	Integer
presentacion	String
observaciones	String
realizadoPor	String
fechaRealizado_D	Short
rechaRealizado_M	Short
fechaRealizado_A	Integer
recibidoPor	String
fechaRecibido_D	Short
fechaRecibido_M	Short
fechaRecibido_A	Integer
revisadoPor	String
fechaRevisado_D	Short
fechaRevisado_M	Short
fechaRevisado_A	Integer
Responsabilidades:	
Nombre:	SIC_0092InformeAnalisis()
Descripción:	Constructor

Nombre: AnalisisDelInforme	
Tipo de clase: Entidad	
Atributo	Tipo
espRef	String
resultadoDef	String
nombreEnsayo	String
Responsabilidades:	
Nombre:	AnalisisDelInforme ()
Descripción:	Constructor

Nombre: ExpedienteAnalitico	
Tipo de clase: Entidad	
Atributo	Tipo
idExpediente	Integer
estado	Boolean
Responsabilidades:	
Nombre:	ExpedienteAnalitico()
Descripción:	Constructor

Nombre: SIC_0837AnalisisRFE	
Tipo de clase: Entidad	
Atributo	Tipo
idSIC	Integer
Folio	Integer
Lote	String
tecnica	String
ppo	String
motivoInvestigacion	String
conclusiones	String
realizadoPor	String
cargoRealiza	String
revisadoPor	String
cargoRevisa	String
fechaRealizado_D	Short
fechaRealizado_M	Short
fechaRealizado_A	Integer
fechaRevisado_D	Short
fechaRevisado_M	Short
fechaRevisado_A	Integer
rfe	String
env	String
lotesInv	String
procedimientVigente	String
tecnicaSegunPpo	String
estaEntrenado	String
manipulacionMuestra	String
tempAlmacMuestra	String

matRefTraceables	String
cepasRefTraceables	String
controlPositivo	String
controlNegativo	String
graficoControl	String
calibracionEquipos	String
calibracionInst	String
datosPrimarios	String
aproxRedondeo	String
criteriosValidez	String
muestraLoteSatisf	String
controlesTemp	String
soluciones	String
reactivos	String
condAlmacReactivos	String
solucRefPpo	String
solucRefVigentes	String
mediosCultivo	String
coeficienteVariacion	String
controlEsterilidad	String
microorgAislado	String
chequeoDesempenno	String
ciclosEsterilizacion	String
controlesTemper	String
sistemasComerciales	String
estado	String
Responsabilidades:	
Nombre:	SIC_0837 AnalisisRFE()
Descripción:	Constructor

Nombre: SIC_0931NotifResNoS	
Tipo de clase: Entidad	
Atributo	Tipo
IdSIC	Integer
otrosResNC	String
notificadoPor	String
fechaNotificacion_D	Short
fechaNotificacion_M	Short
fechaNotificacion_A	Integer
recibidoPor	String
fechaRecibido_D	Short
fechaRecibido_M	Short
fechaRecibido_A	Integer
Responsabilidades:	
Nombre:	SIC_0931NotifResNoS()
Descripción:	Constructor

Nombre: SIC_0300CertificadoAnalisisIFA	
Tipo de clase: Entidad	
Atributo	Tipo
idSIC	Integer
folio	Integer
conclusiones	String
condicionesAlmacenamiento	String
observaciones	String
liberadoPor	String
cargoLibera	String
fechaLiberacion_D	Short
fechaLiberacion_M	Short
fechaLiberacion_A	Integer
emitidoPor	String
cargoEmite	String
fechaEmision_D	Short
fechaEmision_M	Short
fechaEmision_A	Integer
fechaVencimiento_D	Short
fechaVencimiento_M	Short
fechaVencimiento_A	Integer
Responsabilidades:	
Nombre:	SIC_0300CertificadoAnalisisIFA()
Descripción:	Constructor

Nombre: PruebasCertificado	
Tipo de clase: Entidad	
Atributo	Tipo
idPruebasCertificado	Integer
prueba	String
limiteAceptacion	String
metodo	String
resultado	String
Responsabilidades:	
Nombre:	PruebasCertificado ()
Descripción:	Constructor

Nombre: SIC_0301CertificadoPT	
Tipo de clase: Entidad	
Atributo	Tipo
idSIC	Integer
folio	Integer
lote	String
presentacion	String
nombreComercial	String
dosisBulbo	String
volumenDosis	String
fechaVencimiento_D	Short

fechaVencimiento_M	Short
fechaVencimiento_A	Integer
lamInmuno	String
resultadoInmuno	String
lamRibavirina	String
resultadoRibavirina	String
lamInspeccionVisual	String
resultadoInspeccionVisual	String
conclusiones	String
observaciones	String
condAlmacenamiento	String
fechaConfeccion_D	Short
fechaConfeccion_M	Short
fechaConfeccion_A	Integer
liberadoPor	String
cargoLibera	String
emitidoPor	String
cargoEmite	String
fechaEmision_D	Short
fechaEmision_M	Short
fechaEmision_a	Integer
Responsabilidades:	
Nombre:	SIC_0301CertificadoAnalisisPT()
Descripción:	Constructor

Nombre: SIC_0001DocAvalaLC	
Tipo de clase: Entidad	
Atributo	Tipo
idSIC	Integer
folio	Integer
dosis	Float
cantidadKit	Integer
descripcion	String
conclusiones	String
observaciones	String
fechaLiberacion_D	Short
fechaLiberacion_M	Short
fechaLiberacion_A	Integer
liberadoPor	String
cargoLibera	String
emitidoPor	String
cargoEmite	String
fechaEmision_D	Short
fechaEmision_M	Short
fechaEmision_A	Integer
Responsabilidades:	
Nombre:	SIC_0001DocAvalaLC()
Descripción:	Constructor

Nombre: SIC_0100ResumenExpediente	
Tipo de clase: Entidad	
Atributo	Tipo
idResumen	Integer
problemas	String
desviacInvest	String
conclusiones	String
realizadoPor	String
cargo	String
fechaRealizado_D	Short
fechaRealizado_M	Short
fechaRealizado_A	Integer
Responsabilidades:	
Nombre:	SIC_0100ResumenExpediente()
Descripción:	Constructor

Nombre: LibroMT	
Tipo de clase: Entidad	
Atributo	Tipo
nombreProducto	String
nombreEtapa	String
Responsabilidades:	
Nombre:	LibroMT()
Descripción:	Constructor

Nombre: MuestrasTestigos	
Tipo de clase: Entidad	
Atributo	Tipo
lote	String
fechaVencimiento_D	Short
fechaVencimiento_M	Short
fechaVencimiento_A	Integer
unidades	Integer
presentacion	String
entrega	String
recibe	String
folio	Integer
salidaFolio	Integer
saldo	Integer
observaciones	String
noCaja	String
fechaEntrada_D	Short
fechaEntrada_M	Short
fechaEntrada_A	Integer
volumenMinimo	String
Responsabilidades:	
Nombre:	MuestraTestigo()
Descripción:	Constructor

Nombre: SolicitudDestProd	
Tipo de clase: Entidad	
Atributo	Tipo
idSIC	Integer
folio	Integer
fechaSolicitud_D	Short
fechaSolicitud_M	Short
fechaSolicitud_A	Integer
area	String
clasificacion	String
presentacion	String
cantidad	Integer
observaciones	String
volumen	String
solicitadoPor	String
cargoSolicita	String
revisadoPor	String
cargoRevisa	String
aprobadoPor	String
cargoAprueba	String
Responsabilidades:	
Nombre:	SolicitudDestProd()
Descripción:	Constructor

Nombre: ProductoDestruir	
Tipo de clase: Entidad	
Atributo	Tipo
idProductoDestruir	Integer
area	String
clasificacion	String
presentacion	String
cantidad	Integer
volumen	String
Responsabilidades:	
Nombre:	ProductoDestruir()
Descripción:	Constructor

Nombre: SolicitudMT	
Tipo de clase: Entidad	
Atributo	Tipo
idSIC	Integer
folio	Integer
fechaSolicitud_D	Short
fechaSolicitud_M	Short
fechaSolicitud_A	Integer
solicitadoPor	String

propositoEntrega	String
cantidadUnidades	Integer
fechaEntrega_D	Short
fechaEntrega_M	Short
fechaEntrega_A	Integer
recibidoPor	String
autorizadoPor	String
Responsabilidades:	
Nombre:	SolicitudMT()
Descripción:	Constructor

Nombre: ControlSalidaMT	
Tipo de clase: Entidad	
Atributo	Tipo
ildSIC	Integer
folio	Integer
cantBulbosEntregan	Integer
cantidadBulbosQuedan	Integer
propositoEntrega	String
entrega	String
recibe	String
fechaEntrega_D	Short
fechaEntrega_M	Short
fechaEntrega_A	Integer
Responsabilidades:	
Nombre:	ControlSalidaMT()
Descripción:	Constructor

Nombre: RevisionMuestrasTestigos	
Tipo de clase: Entidad	
Atributo	Tipo
idSIC	Integer
folio	Integer
lugar	String
fechaRevision_D	Short
fechaRevision_M	Short
fechaRevision_A	Integer
observaciones	String
realizadoPor	String
revisadoPor	String
caracOrg	Boolean
envase	Boolean
etiqueta	Boolean
Responsabilidades:	
Nombre:	RevisionMuestrasTestigos()
Descripción:	Constructor

Para ver las descripciones de las restantes Clases Persistentes remitirse al Anexo 3.

2.7 Diseño de la BD.

La BD para el Módulo Liberación Analítica es el medio para el almacenamiento de todos los datos relacionados con las muestras evaluadas por calidad y para acceder de forma controlada a los mismos.

El Modelo Físico de Datos se diseñó a partir de un análisis de las clases persistentes, realizaciones de los casos de uso del sistema, estructura de las planillas, libros, expedientes y otros documentos generados en el departamento así como la forma en que se gestiona la información en el Grupo de Liberación Analítica. Este modelo fue realizado con la herramienta Visual Paradigm for UML 6.1 Enterprise Edition que genera de forma automática la BD PostgreSQL. En las Fig.8-9 se muestra el Modelo Físico de Datos en forma simplificada ocultando los atributos de las tablas, para consultar el modelo en su forma ampliada remitirse al Anexo 4.

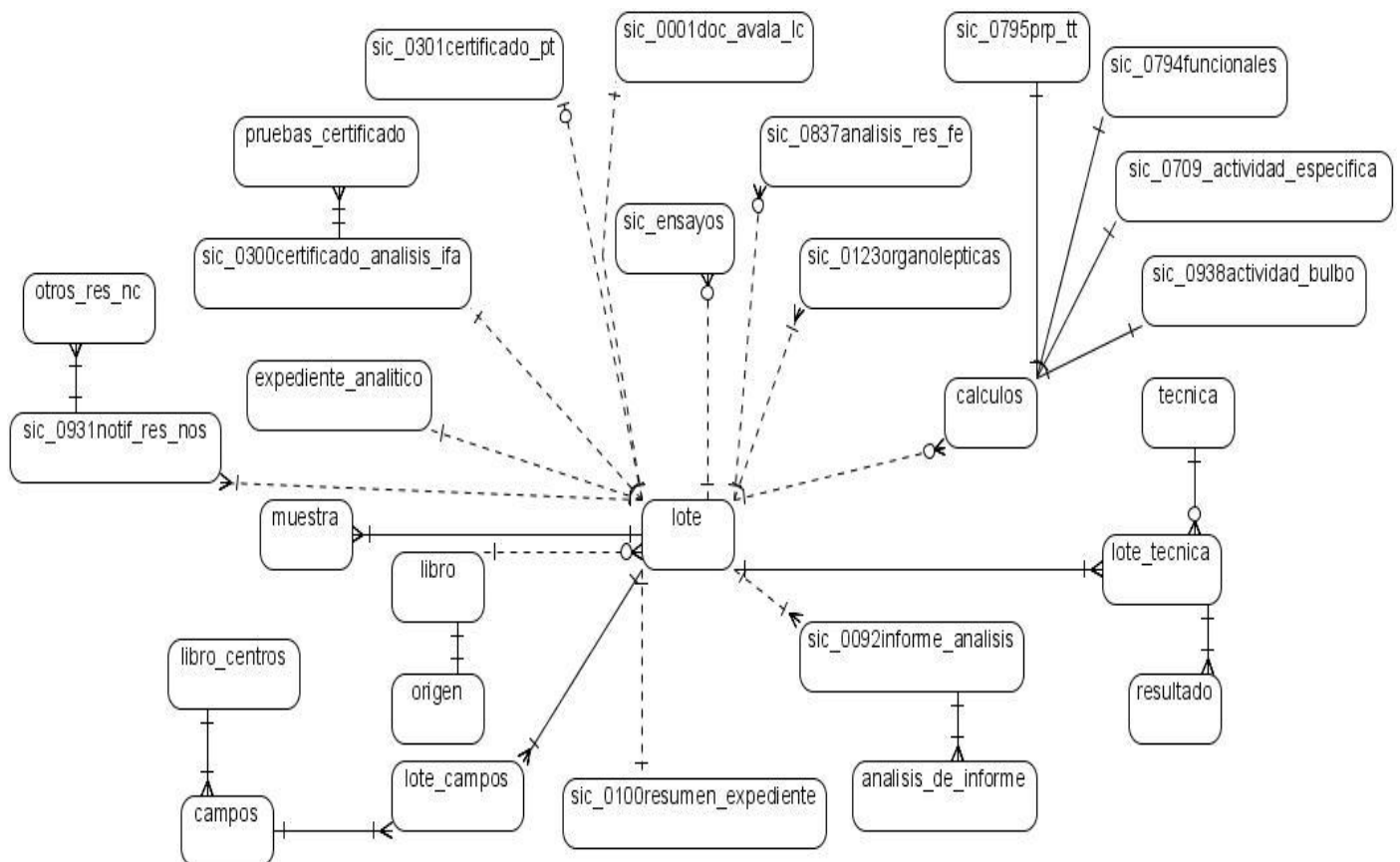


Fig. 8 Modelo Físico de Datos. (Parte 1)

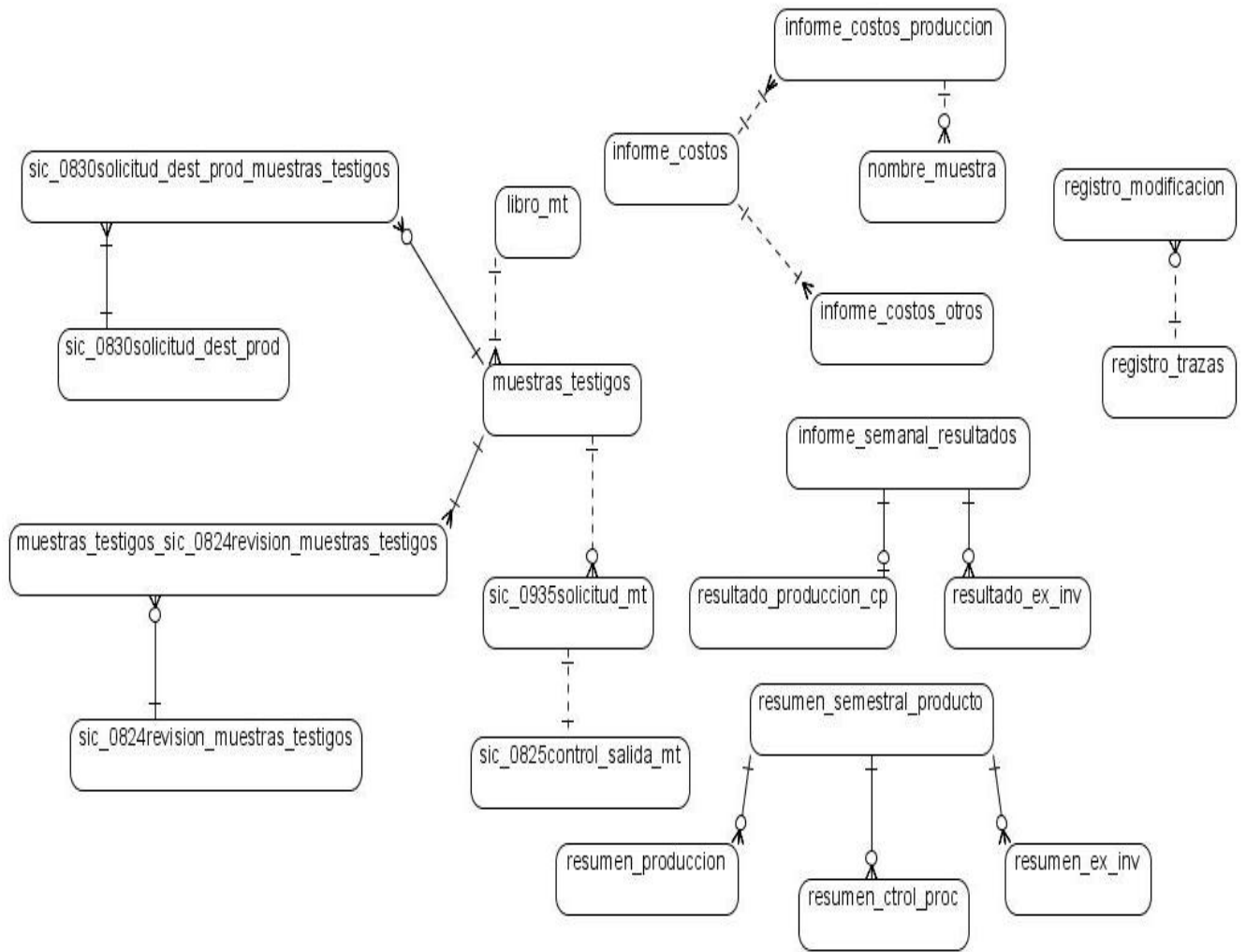


Fig. 9 Modelo Físico de Datos. (Parte 2)

2.7.1 Descripción de las tablas de la BD.

Nombre: libro		
Descripción: Representa un libro de recepción de muestras para un producto en una etapa determinada.		
Atributo	Tipo	Descripción
nombre_producto	Varchar(50)	Llave primaria compuesta de la tabla
nombre_etapa	Varchar(50)	Llave primaria compuesta de la tabla
idtipo	Int4	Llave foránea que toma de la entidad Origen

Nombre: origen		
Descripción: Nomenclador utilizado para guardar las posibles fuentes de origen de los productos.		
Atributo	Tipo	Descripción
idtipo	Int4	Identificador, llave primaria de la tabla
nombre_tipo	Varchar(50)	Nombre de la fuente de origen.

Nombre: muestra		
Descripción: Representa los datos de la muestra a registrar en el libro.		
Atributo	Tipo	Descripción
no_control	Int4	Llave primaria compuesta que identifican las filas
lote	Varchar(15)	Llave foránea que toma de la entidad Lote.
nombre_entrega	Varchar(50)	Nombre de la persona que entrega la muestra.
nombre_recibe	Varchar(50)	Nombre de la persona que recibe la muestra.
almacenamiento	Varchar(50)	Temperatura de almacenamiento requerida.
cantidad_bulbos	Int4	Cantidad de bulbos que se reciben.
observaciones	Varchar(255)	Observaciones generales.
autorizado_no	Bool	Para las muestras provenientes de Investigación y Extraplan de producción se especifica si fue autorizada o no.
fecha_recepcion_d	Int2	Día de recepción de la muestra.
fecha_recepcion_m	Int2	Mes de recepción de la muestra.
fecha_recepcion_a	Int4	Año de recepción de la muestra.

Nombre: lote		
Descripción: Es la entidad que recoge la información referente al lote de las muestras recepcionadas.		
Atributo	Tipo	Descripción
lote	Varchar(15)	Llave primaria de la tabla, lote al que pertenece la muestra.
lote_origen	Varchar(15)	Lote que le dio origen al lote en cuestión.
nombre_producto	Varchar(50)	Llave foránea que toma de la tabla Libro.
nombre_etapa	Varchar(50)	Llave foránea que toma de la tabla Libro.
idtipo	Int4	Llave foránea que toma de la tabla Libro.
fecha_fab_lote_d	Int2	Día de fabricación del lote.
fecha_fab_lote_m	Int2	Mes de fabricación del lote.
fecha_fab_lote_a	Int4	Año de fabricación del lote.
visto_no	Bool	Se utiliza para mantener el estado de las alertas

		sobre un lote, si se desea o no, que sea visto la próxima vez que se inicia la aplicación.
--	--	--

Nombre: lote_tecnica		
Descripción: Entidad que surge de la relación de muchos a muchos entre las entidades lote y tecnica, asocia las técnicas que se le van a realizar a cada lote.		
Atributo	Tipo	Descripción
lote	Varchar(15)	Llave primaria que toma de la entidad lote.
id_tecnica	Int4	Llave primaria que toma de la entidad tecnica.

Nombre: tecnica		
Descripción: Recoge las diferentes técnicas que se le pueden aplicar a las muestras en los procesos de control de la calidad.		
Atributo	Tipo	Descripción
tecnica	Varchar(25)	Nombre de la técnica, llave primaria
ppo	Varchar(15)	Procedimiento Patrón de Operaciones por el que se rige la técnica.

Nombre: resultado		
Descripción: Recoge la información de los resultados de los ensayos, esta entidad de llena automáticamente cuando se entran resultados de ensayos realizados en laboratorios del CIGB y manualmente cuando se entran resultados de ensayos realizados en laboratorios fuera del CIGB.		
Atributo	Tipo	Descripción
idresultado	Int4	Valor entero de autoincremento llave primaria.
lote	Varchar(15)	Llave foránea de la entidad lote_tecnica.
id_tecnica	Int4	Llave foránea de la entidad lote_tecnica.
conformeono	Bool	Si es Conforme o No Conforme
resultado	Varchar(50)	El valor del resultado obtenido.
nombre_analista	Varchar(50)	Nombre de la persona que realizó el análisis.
fecha_salida_resultado_d	Int2	Día de salida de resultados.
fecha_salida_resultado_m	Int2	Mes de salida de resultados.
fecha_salida_resultado_a	Int4	Año de salida de resultados.
ubicacion_fuera_cigb	Int4	Ubicación de la copia escaneada de la planilla que entregó un laboratorio fuera del CIGB con el/los resultado(s) de los ensayos.

Nombre: sic_0123organolepticas		
Descripción: Representa el SIC-0123 que describe el proceso de análisis de las características organolépticas de las muestras.		
Atributo	Tipo	Descripción
idsic	Int4	Valor entero de autoincremento llave primaria.
folio	Int4	Número de folio de la planilla, valor autoincremental que comienza de cero al iniciar cada año.
lote	Varchar(15)	Llave foránea que toma de la entidad lote.
fecha_rec_d	Int2	Día de Recepción en el laboratorio.
fecha_rec_m	Int2	Mes de Recepción en el laboratorio.

fecha_rec_a	Int4	Año de Recepción en el laboratorio.
cant_bulbos	Int4	Cantidad de bulbos a analizar.
recibidos_por	Varchar(50)	Nombre del analista que recibió los bulbos.
tipo_producto	Varchar(15)	Si es MPA, Liofilizado o Formulado.
disolvente_usado	Varchar(50)	Tipo de disolvente usado en el proceso (Agua filtrada, o Alcohol Bencílico).
cant_disolvente	Int4	Cantidad de disolvente usado en el proceso.
filtracion	Bool	Si necesita filtrarse o no.
tamanno_poro	Int4	Tamaño del poro del filtro.
filtrado_por	Varchar(50)	Nombre del analista que realizó la filtración.
tipo_aguja	Varchar(50)	Tipo de aguja.
fecha_vencimiento_d	Int2	Día de vencimiento.
fecha_vencimiento_m	Int2	Mes de vencimiento.
fecha_vencimiento_a	Int4	Año de vencimiento.
especificaciones	Varchar(255)	Especificaciones.
observaciones	Varchar(255)	Observaciones.
realiza_resuspension	Varchar(50)	Nombre del analista que realizó la resuspensión.
cumpleono	Boolean	Si cumple o no con las especificaciones.
conf_por	Varchar(50)	Nombre del analista que elaboró la planilla.
fecha_conf_d	Int2	Día de confección.
fecha_conf_m	Int2	Mes de confección.
fecha_conf_a	Int4	Año de confección.
revisado_por	Varchar(50)	Nombre de quien revisa el SIC.
fecha_revision_d	Int2	Día de revisión.
fecha_revision_m	Int2	Mes de revisión.
fecha_revision_a	Int4	Año de revisión.
estado	bool	Si ha sido revisado o no el SIC del cálculo.

Nombre: calculos		
Descripción: Guarda la información común de los cálculos Relación PRP: TT, Actividad Biológica por Bulbo, Actividad Específica y Grupos Funcionales Residuales.		
Atributo	Tipo	Descripción
idsic	Int4	Valor entero autoincremental, llave primaria.
folio	Int4	Valor entero autoincremental, folio del SIC comienza de cero al iniciar cada año.
lote	Varchar(15)	Llave foránea que toma de la entidad lote.
realizado_por	Int4	Nombre del analista que realizó el cálculo.
revisado_por	Int4	Nombre de la persona que revisó.
conformeono	Bool	Si es conforme o No Conforme.
limite_aceptacion	Float4	Valor límite para que el resultado sea conforme.
fecha_realizado_d	Int2	Día de realizado.
fecha_realizado_m	Int2	Mes de realizado.
fecha_realizado_a	Int4	Año de realizado.
fecha_revisado_d	Int2	Día de realizado.
fecha_revisado_m	Int2	Mes de realizado.
fecha_revisado_a	Int4	Año de realizado.
recibido_por	Varchar(50)	Nombre del analista del laboratorio que recibe la muestra.

fecha_recibido_d	Int2	Día de recibida la muestra en el laboratorio.
fecha_recibido_m	Int2	Mes de recibida la muestra en el laboratorio.
fecha_recibido_a	Int4	Año de recibida la muestra en el laboratorio.
observaciones	Varchar(255)	Observaciones.
valor_obtenido	Float4	Límite de aceptación según el NP vigente.
estado	bool	Si ha sido revisado o no el SIC-0123

Nombre: sic_0795prp_tt		
Descripción: Representa el SIC-0795 que describe la relación PRPTT.		
Atributo	Tipo	Descripción
idsic	Int4	Llave primaria que toma de la entidad calculos.
concentracion_tt	Float4	Concentración de proteínas totales
res_con_sic107	Int4	Folio del SIC-107 donde está descrito el resultado.
conc_prp_tt	Float	Concentración de carbohidratos totales.
res_sic762	Int4	Folio del SIC-762 donde está descrito el resultado.
conc_carb_lib	Float4	Concentración de carbohidratos libres.
res_carb_sic762	Int4	Folio del SIC-762 donde está descrito el resultado.
relacion_prp_tt	Float4	Valor de la relación (ConcPRPTT – ConcCarbLib)/ConcentracionTT.

Nombre: sic_0709actividad_especifica		
Descripción: Representa el SIC-0709 que describe el cálculo de Actividad Específica.		
Atributo	Tipo	Descripción
idsic	Int4	Llave primaria que toma de la entidad calculos.
conca	Float4	Valor de concentración de la proteína de interés A.
metodoa	Varchar(50)	Método utilizado para obtener A.
ppoa	Varchar(50)	Procedimiento Patrón de Operación (PPO) según se calculó el valor de la concentración A.
sica	Varchar(50)	Nombre del SIC donde se describe la obtención de la concentración A.
folioa	Int4	Folio del SIC donde se describe la obtención de la concentración A...
concb	Float4	Valor de concentración de proteínas totales.
metodob	Varchar(50)	Método utilizado para obtener B.
ppob	Varchar(50)	PPO según se calculó el valor de la concentración B.
sicb	Varchar(50)	Nombre del SIC donde se describe la obtención de la concentración B.
foliob	Int4	Folio del SIC donde se describe la obtención de la concentración B.

Nombre: sic_0938actividad_biologica		
Descripción: Representa el SIC-0938 que describe el cálculo de Actividad Biológica por Bulbo.		
Atributo	Tipo	Descripción
idsic	Int4	Llave primaria que toma de la entidad calculos.
np	Varchar(50)	Número de Parte.
concentracion	Float4	Concentración en millones.
act_biol_a	Float4	Valor de la activad biológica A.

ppo	Varchar(50)	PPO según se calculó el valor de la actividad biológica A.
sic	Varchar(50)	Nombre del SIC donde se describe el cálculo de la actividad biológica A.
foliosic	Int4	Folio del SIC donde se describe el cálculo de la actividad biológica A.
volumen_bulbo_b	Float4	Volumen por bulbo reportado.
pno	Varchar(50)	PNO según se calculó el volumen por bulbo reportado.
ac	Varchar(50)	Nombre del AC donde se describe el cálculo de la actividad biológica A.
folio_ac	Int4	Folio del AC donde se describe el cálculo de la actividad biológica A.

Nombre: sic_0794funcionales		
Descripción: Representa el SIC-0938 que describe el cálculo de Actividad Biológica por Bulbo		
Atributo	Tipo	Descripción
idsic	Int4	Llave primaria que toma de la entidad calculos.
valorgs	Float4	Valor de los grupos sulfhídricos,
folio_sic0759c	Int4	Folio del SIC0759C.

Nombre: sic_0092informe_analisis		
Descripción: Guarda la información del Informe de Análisis, que puede crearse para entregar a los inspectores de calidad, o para cerrar el Expediente Analítico de un producto.		
Atributo	Tipo	Descripción
idsic	Int4	Valor entero de autoincremento llave primaria.
lote	Varchar(15)	Llave foránea que toma de la entidad lote.
folio	Int4	Número de folio de la planilla, valor autoincremental que comienza de cero al iniciar cada año.
presentacion	Varchar(50)	Forma en la de presentación de las muestras.
observaciones	Varchar(255)	Observaciones.
realizado_por	Varchar(50)	Nombre de la persona que confeccionó el Informe de Análisis.
fecha_realizado_d	Int2	Día de realización del informe.
fecha_realizado_m	Int2	Mes de realización del informe.
fecha_realizado_a	Int4	Año de realización del informe.
revisado_por	Varchar(50)	Nombre de la persona que revisó el Informe de Análisis.
fecha_revisado_d	Int2	Día de revisión del informe.
fecha_revisado_m	Int2	Mes de revisión del informe.
fecha_revisado_a	Int4	Año de revisión del informe.
recibido_por	Varchar(50)	Nombre de la persona que recibió el Informe de Análisis.
fecha_recibido_d	Int2	Día de recepción del informe
fecha_recibido_m	Int2	Mes de recepción del informe
decha_recibido_a	Int4	Año de recepción del informe
cierra_exp_no	Bool	Si cierra o no el expediente analítico del producto.

Nombre: analisis_de_informe		
Descripción: Entidad que guarda los análisis contenidos en un Informe de Análisis.		
Atributo	Tipo	Descripción
Nombre_ensayo	Varchar(50)	Nombre del ensayo, llave primaria.
idsic	Int4	Llave foránea que toma de la entidad sic_0092informe_analisis.
esp_ref	Varchar(50)	Especificaciones y referencias que debe cumplir el ensayo para que sea conforme.
resultado_def	Varchar(50)	Resultado definitivo del ensayo.

Nombre: expediente_analitico		
Descripción: Expediente analítico que se crea para cada lote de producto que se analiza.		
Atributo	Tipo	Descripción
idexpediente	Int4	Valor entero de autoincremento llave primaria
lote	Varchar(15)	Llave foránea que toma de la entidad lote.
estado	Bool	Estado en que se encuentra la confección del expediente (“Pendiente a cerrar”, “Pendiente a revisar” o “Completo“)

Nombre: SIC_0100resumen_expediente		
Descripción: Representa la planilla Resumen del Expediente Analítico		
Atributo	Tipo	Descripción
idresumen	Int4	Valor entero de autoincremento llave primaria
lote	Varchar(15)	Llave foránea que toma de la entidad lote.
problemas	Varchar(255)	Problemas detectados.
desviac_invest	Varchar(255)	Desviaciones e investigaciones.
conclusiones	Varchar(255)	Conclusiones.
realizado_por	Varchar(50)	Nombre de la persona que confeccionó el resumen del expediente.
cargo	Varchar(50)	Cargo de la persona que realizó el resumen.
fecha_realizado_d	Int2	Día de realización del informe
fecha_realizado_m	Int2	Mes de realización del informe
fecha_realizado_a	Int4	Año de realización del informe

Nombre: sic_0001doc_avala_lc		
Descripción: Representa el documento de liberación para la comercialización.		
Atributo	Tipo	Descripción
idsic	Int4	Valor entero de autoincremento llave primaria
folio	Int4	Valor entero de autoincremento número de folio.
lote	Int4	Llave foránea que toma de la entidad lote.
cantidad_kit	Int4	Cantidad de Kits a Liberar.
descripcion	Varchar(255)	Descripción.
observaciones	Varchar(255)	Desviaciones e Investigaciones.
fecha_liberacion_d	Int2	Día de liberación.
fecha_liberacion_m	Int2	Mes de liberación.
fecha_liberacion_a	Int4	Año de liberación.

liberado_por	Varchar(50)	Nombre de la persona que liberó para la comercialización.
cargo_libera	Varchar(50)	Cargo de la persona que liberó para la comercialización.
emitido_por	Varchar(50)	Nombre de la persona que emite el documento.
cargo_emite	Varchar(50)	Cargo de la persona que emite el documento.
fecha_emision_d	Int2	Día de emisión del documento.
fecha_emision_m	Int2	Mes de emisión del documento.
fecha_emision_a	Int4	Año de emisión del documento.

Nombre: sic_0300certificado_analisis_ifa		
Descripción: Almacena los datos del certificado de análisis para Ingredientes Farmacéuticos Activos (IFA).		
Atributo	Tipo	Descripción
idsic	Int4	Entero autoincremental, llave primaria.
folio	Int4	Número de folio de la planilla, valor autoincremental que comienza de cero al iniciar cada año.
lote	Int4	Llave foránea que toma de la entidad lote.
conclusiones	Varchar(255)	Conclusiones.
condiciones_almacenamiento	Varchar(255)	Condiciones de Almacenamiento
observaciones	Varchar(255)	Observaciones generales.
Liberado_por	Varchar(50)	Nombre de la persona que liberó el producto.
cargo_libera	Varchar(50)	Cargo de la persona que liberó el producto.
fecha_liberacion_d	Int2	Día en que se confeccionó el certificado de análisis para el IFA.
fecha_liberacion_m	Int2	Mes en que se confeccionó el certificado de análisis para el IFA.
Fecha_liberacion_a	Int4	Año en que se confeccionó el certificado de análisis para el IFA.
emitido_por	Varchar(50)	Nombre de la persona que emitió el certificado de análisis para el IFA.
fecha_emision_d	Int2	Día en que se emitió el documento.
fecha_emision_m	Int2	Mes en que se emitió el documento.
fecha_emision_a	Int4	Mes en que se emitió el documento.
cargo_emite	Varchar(50)	Cargo de la persona que emitió el certificado de análisis para el IFA.
lote	Varchar(15)	Llave foránea que toma de la tabla Lote
fecha_vencimiento_d	Int2	Día en que vence el IFA
fecha_vencimiento_m	Int2	Mes en que vence el IFA
Fecha_vencimiento_a	Int4	Año en que vence el IFA

Nombre: pruebas_certificado		
Descripción: Almacena los datos de las pruebas contenidas en la planilla de liberación de productos IFA.		
Atributo	Tipo	Descripción
idpruebascertificado	Int4	Entero autoincremental, llave primaria.
idsic	Int4	Llave foránea que toma de la tabla

		SIC_0300CertificadoAnalisisIFA.
prueba	Varchar(50)	Nombre de la prueba.
limite_aceptacion	Varchar(50)	Limite de aceptación para que el resultado de la prueba sea conforme.
metodo	Varchar(50)	Método por el que se realizó la prueba.
resultado	Varchar(50)	Resultado obtenido para esa prueba.

Nombre: sic_0301certificado_pt		
Descripción: Almacena los datos de la planilla de liberación de productos en fase Producto Terminado.		
Atributo	Tipo	Descripción
idsic	Int4	Entero autoincremental, llave primaria.
folio	Int4	Número de folio de la planilla, valor autoincremental que comienza de cero al iniciar cada año.
lote	Varchar(15)	Llave foránea que toma de la tabla Lote.
presentacion	varchar(25)	Presentación de la muestra.
nombre_comercial	Varchar(25)	Nombre comercial.
dosis_bulbo	Varchar(15)	Dosis por bulbo.
volumen_dosis	Varchar(15)	Volumen de la dosis.
fecha_vencimiento_d	Int2	Día de vencimiento.
fecha_vencimiento_m	Int2	Mes de vencimiento.
fecha_vencimiento_a	Int4	Año de vencimiento.
lam_inmuno	Varchar(50)	Límite de aceptación y método de la inmuno identificación de IFN.
resultado_inmuno	Varchar(25)	Resultado de la inmuno identificación de IFN.
lam_ribavirina	Varchar(50)	Límite de aceptación y método de la identificación de ribavirina.
resultado_ribavirina	Varchar(25)	Resultado de la identificación de ribavirina.
lam_inspeccion_visual	Varchar(50)	Límite de aceptación y método de la inspección visual.
resultado_inspeccion_visual	Varchar(25)	Resultado de la inspección visual.
conclusiones	Varchar(255)	Conclusiones.
observaciones	Varchar(255)	Observaciones.
cond_almacenamiento	Varchar(50)	Condiciones de almacenamiento.
fecha_confeccion_d	Int2	Día de confección.
fecha_confeccion_m	Int2	Mes de confección.
fecha_confeccion_a	Int4	Año de confección.
liberado_por	Varchar(50)	Nombre de la persona que libera.
cargo_libera	Varchar(50)	Cargo de la persona que libera.
emitido_por	Varchar(50)	Nombre de la persona que emite el certificado.
cargo_emite	Varchar(50)	Cargo de la persona que emite el certificado.
fecha_emision_d	Int2	Día de emisión.
fecha_emision_m	Int2	Mes de emisión.
fecha_emision_a	Int4	Año de emisión.

Nombre: sic_0837 analisis_res_fe		
Descripción: Esta entidad representa la planilla SIC-0837 Análisis de Resultados Fuera de Especificación que abre la investigación cuando un ensayo da No Conforme.		
Atributo	Tipo	Descripción
idsic	Int4	Valor entero de autoincremento llave primaria.
folio	Int4	Número de folio de la planilla, valor autoincremental que comienza de cero al iniciar cada año.
lote	Varchar(15)	Llave foránea que toma de la entidad Lote.
tecnica	Varchar(50)	Técnica aplicada en el ensayo.
ppo	Varchar(50)	Procedimiento patrón de operación.
motivo_investigacion	Varchar(255)	Motivo por el que se inició la investigación.
conclusiones	Varchar(255)	Conclusiones.
realizado_por	Varchar(50)	Nombre de la persona que confeccionó la planilla.
cargo_realiza	Varchar(50)	Cargo de la persona que confeccionó la planilla.
revisado_por	Varchar(50)	Nombre de la persona que revisó la planilla.
cargo_revisa	Varchar(50)	Cargo de la persona que revisó la planilla.
fecha_realizado_d	Int2	Día de realizado.
fecha_realizado_m	Int2	Mes de realizado.
fecha_realizado_a	Int4	Año de realizado.
fecha_revisado_d	Int2	Día de revisado.
fecha_revisado_m	Int2	Mes de revisado.
fecha_revisado_a	Int4	Año de revisado.
rfe	Varchar(255)	Resultado fuera de especificación.
env	Varchar(255)	
lotes_inv	Varchar(255)	Lotes involucrados.
procedimiento_vigente	Varchar(15)	Si se utilizó el procedimiento vigente. (Si, No ó No procede)
tecnica_segun_ppo	Varchar(15)	Si se ejecutó adecuadamente la técnica según el PPO aplicable. (Si, No ó No procede).
esta_entrenado	Varchar(15)	Si el analista que realizo la técnica está entrenado para hacerla. (Si, No ó No procede).
manipulacion_muestra	Varchar(15)	Si se manipuló adecuadamente la muestra. (Si, No ó No procede).
temp_almac_muestra	Varchar(15)	Si la muestra se encontraba almacenada a la temperatura adecuada. (Si, No ó No procede).
mat_ref_traceables	Varchar(15)	Si se usó un material de referencia traceable que estaba adecuadamente almacenado y cuyo control sistemático dio resultados satisfactorios. (Si, No ó No procede).
cepas_ref_traceables	Varchar(15)	Si se usaron cepas de referencia traceables con adecuados controles de viabilidad y pureza. (Si, No ó No procede).
control_positivo	Varchar(15)	Si cumple el control positivo con el rango establecido. (Si, No ó No procede).
control_negativo	Varchar(15)	Si los controles negativos dieron resultados satisfactorios. (Si, No ó No procede).
grafico_control	Varchar(15)	Si está el grafico de control bajo control estadístico. (Si, No ó No procede).

calibracion_equipos	Varchar(15)	Si está vigente la calibración del equipamiento. (Si, No ó No procede).
calibracion_inst	Varchar(15)	Si está vigente la calibración de los instrumentos. (Si, No ó No procede).
datos_primarios	Varchar(15)	Si se transcribieron adecuadamente los datos primarios. (Si, No ó No procede).
aprox_redondeo	Varchar(15)	Si se realizaron las aproximaciones y redondeo acorde al procedimiento establecido. (Si, No ó No procede).
criterios_validez	Varchar(15)	Si se cumplen los criterios de validez del ensayo. (Si, No ó No procede).
muestra_lotes_satisf	Varchar(15)	Si la muestra se incluyó junto a otras provenientes de otros lotes y/o productos que dieron resultados satisfactorios. (Si, No ó No procede).
controles_temp	Varchar(15)	Si existen registros de los controles de temperatura en los cuartos de incubación y/o incubadoras, de los freezer y refrigeradores. (Si, No ó No procede).
soluciones	Varchar(15)	Si las soluciones fueron preparadas, almacenadas y se encuentran vigentes. (Si, No ó No procede).
reactivos	Varchar(15)	Si los reactivos químicos y biológicos fueron evaluados, almacenados adecuadamente y se encuentran vigentes. (Si, No ó No procede).
cond_almac_reactivos	Varchar(15)	Si hay garantías sobre las condiciones de almacenamiento de los materiales y reactivos biológicos. (Si, No ó No procede).
soluc_ref_ppo	Varchar(15)	Si las soluciones de referencia se prepararon según el PPO establecido. (Si, No ó No procede).
soluc_ref_vigentes	Varchar(15)	Si las soluciones de referencia están vigentes. (Si, No ó No procede).
medios_cultivo	Varchar(15)	Si los medios de cultivo pasaron satisfactoriamente los ensayos de esterilidad y promoción del crecimiento. (Si, No ó No procede).
coeficiente_variacion	Varchar(15)	Si el coeficiente de variación entre los conteos estaba dentro del límite de variabilidad de la técnica. (Si, No ó No procede).
control_esterilidad	Varchar(15)	Si existe control de esterilidad de los materiales críticos
microorg_aislado	Varchar(15)	Si el microorganismo aislado en el monitoreo de ambiente, superficie y personal es el mismo que el aislado en el producto. (Si, No ó No procede).
chequeo_desempeno	Varchar(15)	Si se hizo el chequeo del desempeño si es aplicable. (Si, No ó No procede).
ciclos_esterilizacion	Varchar(15)	Si existen registros de los ciclos de esterilización. (Si, No ó No procede).
controles_temper	Varchar(15)	Si existen registros de los controles de temperatura en los cuartos de incubación y/o incubadoras, de los freezer y refrigeradores en microbiología. (Si, No ó No procede).
sistemas_comerciales	Varchar(15)	Si se usan sistemas comerciales en su periodo de

		vigencia y de acuerdo con las recomendaciones de los fabricantes. (Si, No ó No procede).
estado	Varchar(50)	Estado de elaboración del SIC-0937. (Si, No ó No procede).

Nombre: sic_0931_notif_res_nos		
Descripción: Esta entidad representa la planilla SIC-0931 Notificación de Resultados No Satisfactorios que se genera cada vez que llega al grupo un resultado de ensayo no conforme definitivamente.		
Atributo	Tipo	Descripción
idnotif	Int4	Valor entero de autoincremento llave primaria.
lote	Varchar(15)	Llave foránea que toma de la entidad lote.
notificado_por	Varchar(50)	Nombre del primer ensayo por el que se notificó el resultado no satisfactorio.
recibido_por	Varchar(50)	Nombre de la persona que recibió la planilla.
idotros	Int4	Identificador de los otros resultados no conformes que fueron notificados, llave foránea que toma de la tabla otros_res_nc.
fecha_notificacion_d	Int2	Día de notificación de la no conformidad.
fecha_notificacion_m	Int2	Mes de notificación de la no conformidad.
fecha_notificacion_a	Int4	Año Día de notificación de la no conformidad.
fecha_recibido_d	Int2	Día de recibido el resultado no satisfactorio.
fecha_recibido_m	Int2	Mes de recibido el resultado no satisfactorio.
fecha_recibido_a	Int4	Año de recibido el resultado no satisfactorio.

Nombre: otros_res_nc		
Descripción: Esta entidad representa otros ensayos no conformes además de por el que se notificó.		
Atributo	Tipo	Descripción
idotros	Int4	Valor entero de autoincremento llave primaria.
otros_res_nc	Varchar(50)	Nombre de otros ensayos que dieron no conforme para una muestra.

Nombre: libro_centros		
Descripción: Guarda los datos de centros que realizan ensayos para el CIGB.		
Atributo	Tipo	Descripción
idlibrocentro	Int4	Valor entero de autoincremento llave primaria.
nombre_centro	Varchar(50)	Nombre del centro.
anno	Int4	Año en que se crea el libro.

Nombre: campos		
Descripción: Almacena los campos que contiene el libro de registro de resultados de los ensayos realizados en otros centros.		
Atributo	Tipo	Descripción
idcampo	Int4	Valor entero de autoincremento llave primaria.
idlibrocentro	Int4	Llave foránea que toma de la tabla libro_centros.
nombreCampo	Varchar(15)	Nombre del campo.

Nombre: lote_campos		
Descripción: Se utiliza para almacenar el valor que toma el campo para determinado lote.		
Atributo	Tipo	Descripción
lote	Varchar(15)	Llave foránea que toma de la entidad lote.
idcampo	Int4	Valor entero de autoincremento llave primaria.
idlibrocentro	Int4	Llave primaria que toma de la tabla libro_centros.
valor	Varchar(255)	Valor que toma el campo para determinado lote.

Nombre: libro_mt		
Descripción: Representa un libro de Muestras Testigos.		
Atributo	Tipo	Descripción
nombre_producto	Varchar(50)	Llave primaria compuesta de la tabla.
nombre_etapa	Varchar(50)	Llave primaria compuesta de la tabla.

Nombre: muestras_testigos		
Descripción: Recoge los datos de las muestras testigos.		
Atributo	Tipo	Descripción
lote	Varchar(15)	Lote de la muestra testigo, llave primaria de la tabla.
nombre_producto	Varchar(50)	Llave foránea que toma de la tabla libro_mt.
nombre_etapa	Varchar(50)	Llave foránea que toma de la tabla libro_mt.
tiempo_almacenamiento	Varchar(50)	Tiempo que debe almacenarse la muestra testigo.
presentacion	Varchar(50)	Volumen que contiene el frasco o la concentración del producto.
entrega	Varchar(50)	Nombre de la persona que entrega la muestra testigo.
fecha_entrada_d	Int2	Día de entrada de la muestra testigo.
fecha_entrada_m	Int2	Mes de entrada de la muestra testigo.
fecha_entrada_a	Int4	Año de entrada de la muestra testigo.
fecha_vencimiento_d	Int2	Día de vencimiento de la muestra testigo.
fecha_vencimiento_m	Int2	Mes de vencimiento de la muestra testigo.
fecha_vencimiento_a	Int4	Año de vencimiento de la muestra testigo.
unidades	Int4	Cantidad de muestras testigos de ese lote que se van a almacenar.
recibe	Varchar(50)	Nombre de la persona que recibe la muestra testigo.
folio	Int4	Número de folio, entero autoincremental.
saldo	Int4	Cantidad de muestras testigos en existencia, se va modificando a medida que salen muestras testigos.
observaciones	Varchar(255)	Observaciones.
no_caja	Int4	Número de caja.
volumen_minimo	Varchar(15)	Volumen mínimo de muestra que se necesita almacenar.

Nombre: sic_0935solicitud_mt		
Descripción: Representa una solicitud de muestra testigo.		
Atributo	Tipo	Descripción
idsic	Int4	Entero autoincremental, llave primaria.
lote	Varchar(15)	Llave foránea que toma de la tabla LibroMuestrasTestigos.
folio	Int4	Número de folio de la planilla, valor autoincremental que comienza de cero al iniciar cada año.
solicitado_por	Varchar(50)	Nombre de la persona que solicita la muestra testigo.
proposito_entrega	Text	Propósito de la entrega.
cantidad_unidades	Integer	Cantidad de unidades solicitadas.
recibido_por	Varchar(50)	Nombre de la persona que recibe la solicitud.
autorizado_por	Varchar	Nombre de la persona que autoriza.
fecha_solicitud_d	Int2	Día en que se realiza la solicitud.
fecha_solicitud_m	Int2	Mes en que se realiza la solicitud.
fecha_solicitud_a	Int4	Año en que se realiza la solicitud.
fecha_entrega_d	Int2	Día en que se realiza la entrega
fecha_entrega_m	Int2	Mes en que se realiza la entrega.
fecha_entrega_a	Int4	Año en que se realiza la entrega.

Nombre: sic_0824revisión_muestras_testigos		
Descripción: Representa los datos de una planilla de revisión del estado de las muestras testigos.		
Atributo	Tipo	Descripción
idsic	Int4	Entero autoincremental, llave primaria.
folio	Int4	Número de folio de la planilla, valor autoincremental que comienza de cero al iniciar cada año.
observaciones	Varchar	Observaciones generales.
realizado_por	Varchar(50)	Nombre de la persona que realizó la revisión del estado de las muestras testigos.
revisado_por	Varchar(50)	Nombre de la persona que revisó la planilla de Revisión del Estado de las Muestras Testigos.
fecha_revisión_d	Int2	Día en que se realiza la revisión.
fecha_revisión_m	Int2	Mes en que se realiza la revisión.
fecha_revisión_a	Int4	Año en que se realiza la revisión.
carac_org	Bool	Si se van a revisar o no las características organolépticas de las muestras testigos.
envase	Bool	Si se van a revisar o no el estado del envase de las muestras testigos.
sello	Bool	Si se van a revisar o no el estado de los sellos de las muestras testigos.
etiqueta	Bool	Si se van a revisar o no el estado de las etiquetas de las muestras testigos.
lugar	Varchar(50)	Lugar donde están almacenadas las muestras testigos.

Nombre: muestras_testigos_sic_0824revisión_muestras_testigos		
Descripción: Representa los datos de las muestras que fueron analizadas en la revisión del estado de las muestras testigos.		
Atributo	Tipo	Descripción
lote	Varchar(15)	Llave foránea que toma de la tabla muestras_testigos.
idsic	Int4	Llave foránea que toma de la tabla SIC_824RevisiónMuestrasTestigos
apariciencia_producto	Varchar(255)	Apariciencia que tiene el producto.
apariciencia_envase	Varchar(255)	Apariciencia que tiene el envase.
unidades_revisadas	Int4	Cantidad de unidades revisadas.
fechafablote_d	Int2	Día de fabricación del lote al que pertenece la muestra testigo
fechafablote_m	Int2	Mes de fabricación del lote al que pertenece la muestra testigo
fechafablote_a	Int4	Año de fabricación del lote al que pertenece la muestra testigo

Nombre: sic_0825ControlSalidaMT		
Descripción: Representa la planilla que mantiene el control de la salida de las muestras testigos.		
Atributo	Tipo	Descripción
idsic	Int4	Entero autoincremental, llave primaria.
Folio	Int4	Número de folio de la planilla, valor autoincremental que comienza de cero al iniciar cada año.
idsic0935	Int4	Llave foránea que toma de la tabla sic_0935solicitud_mt.
cantidad_bulbos_entregan	Int4	Cantidad de bulbos que se entregan.
proposito_entrega	Varchar(255)	Propósito de la entrega
recibe	Varchar(50)	Nombre de la persona que recibe las muestras testigos.
entrega	Varchar(50)	Nombre de la persona que entrega las muestras testigos.
fecha_entrega_d	Int2	Día de entrega las muestras testigos.
fecha_entrega_m	Int2	Mes de entrega las muestras testigos.
fecha_entrega_a	Int4	Año de entrega las muestras testigos.

Nombre: sic_0830solicitud_dest_prod		
Descripción: Representa la planilla de solicitud de destrucción de las muestras testigos.		
Atributo	Tipo	Descripción
idsic	Int4	Entero autoincremental, llave primaria.
lote	Varchar(15)	Llave foránea que toma de la tabla muestras_testigos.
fecha_solicitud_d	Int2	Día que se solicita la destrucción de muestras testigos.
fecha_solicitud_m	Int2	Mes que se solicita la destrucción de muestras testigos.
fecha_solicitud_a	Int4	Año que se solicita la destrucción de muestras

		testigos.
observaciones	Varchar(255)	Observaciones generales.
solicitado_por	Varchar(50)	Nombre de la persona que solicita la destrucción de productos.
cargo_solicita	Varchar(50)	Cargo de la persona que solicita la destrucción de productos.
revisado_por	Varchar(50)	Nombre de la persona que revisa la solicitud de destrucción de productos.
cargo_revisa	Varchar(50)	Cargo de la persona que revisa la solicitud de destrucción de productos.
aprobado_por	Varchar(50)	Nombre de la persona que aprueba la destrucción de productos.
cargo_aprueba	Varchar(50)	Cargo de la persona que aprueba la destrucción de productos.

Nombre: producto_destruir		
Descripción: Entidad que surge de la relación de muchos a muchos entre muestras_testigos y sic_0830solicitud_dest_prod.		
Atributo	Tipo	Descripción
lote	Int4	Llave primaria que toma de la tabla muestras_testigos.
idsic	Int4	Llave primaria que toma de la tabla sic_0830solicitud_dest_prod.
clasificacion	Varchar(50)	Clasificación de la muestra testigo.
presentacion	Varchar(50)	Volumen que contiene el frasco o la concentración del producto.
cantidad	Int4	Cantidad de muestra que se solicita destruir.
volumen	Varchar(50)	Volumen de muestra a destruir.

Nombre: registro_trazas		
Descripción: Tabla utilizada para mantener el registro de las trazas de modificación de los datos de las tablas.		
Atributo	Tipo	Descripción
idtraza	Int4	Entero autoincremental, llave primaria.
documento_modificado	Varchar(50)	Nombre del documento modificado.
id_doc_modif	Int4	Identificador del documento modificado.
campo_modificado	Int4	Nombre del campo modificado.
fecha_modificacion_d	Int2	Día de modificación del campo.
fecha_modificacion_m	Int2	Mes de modificación del campo.
fecha_modificacion_a	Int4	Año de modificación del campo.
usuario	Varchar(50)	Nombre del usuario que realizó el cambio.

Nombre: registro_modificacion		
Descripción: Tabla utilizada para almacenar los valores de los datos modificados		
Atributo	Tipo	Descripción
idmod	Int4	Entero autoincremental, llave primaria.

idtraza	Int4	Identificador del documento modificado.
campo_modificado	Int4	Nombre del campo modificado.
valor_viejo	Varchar(255)	Valor que tenía ese campo antes de la modificación.
valor_nuevo	Varchar(255)	Nuevo valor del campo después de ser modificado.
usuario	Varchar(50)	Nombre del usuario que realizó el cambio.

Para ver las descripciones de las restantes tablas de la BD remitirse al Anexo 5.

Conclusiones

En este capítulo se mostró el Diagrama de Clases Persistentes del Módulo Liberación Analítica, los atributos y métodos de las 43 clases persistentes identificadas y las relaciones entre ellas.

Además se ha descrito el Modelo Físico de Datos y los atributos de todas las tablas. Se definieron 49 tablas que representan los datos relevantes a almacenar por el Módulo Liberación Analítica del LIMS.

Capítulo **3**

Validación del diseño realizado

En este capítulo se brinda información sobre la validación realizada al diseño de la BD propuesta, se muestran las reglas a tener en cuenta para obtener un diseño de óptima calidad, así como la selección de consultas para un llenado voluminoso e inteligente de la BD para validarla funcionalmente.

3.1 Validación teórica del diseño.

Para realizar un buen diseño de BD es necesario velar por la integridad de los datos, la normalización del diseño, la redundancia de la información, la trazabilidad de las acciones que se realizan en la BD y la seguridad.

3.1.1 Integridad.

La integridad de una BD se refiere a la exactitud, completitud, validez y consistencia de los datos almacenados de acuerdo a un conjunto de restricciones semánticas.

Integridad de dominio: Está determinada por la validez de las entradas para una columna. Se puede restringir el tipo mediante tipos de datos, el formato mediante reglas y restricciones CHECK o el intervalo de valores posibles mediante restricciones FOREIGN KEY y CHECK, definiciones DEFAULT y NOT NULL. En la BD del Módulo Liberación Analítica se definió la fecha por los campos: día, mes y año de cada suceso, el campo día puede tomar valores enteros de tamaño 2 bytes restringiendo su dominio entre los valores 1 y 31, para el mes el rango de valores admisibles está entre 1 y 12 y el dominio del campo año incluye los valores enteros de 4 bytes en el intervalo de 1000 a 9999. Cuando la longitud de una cadena no puede exceder los n caracteres se usa un varchar(n), por ejemplo el lote esta en el dominio varchar(15). Cuando el tipo de dato es varchar, acepta números, letras o caracteres, pero existen campos que es necesario validar los datos que pueden almacenar. Para realizar validaciones PostgreSQL posee los “obligadores”, que chequean el cumplimiento de determinadas condiciones que deben cumplir los campos de una tabla y disparan errores en caso de ser violadas. Los “disparadores” (triggers) pueden validar campos de más de una tabla a la vez y disparan un error, en caso de no cumplirse alguna(s) de las condiciones que valida. A través de funciones especiales, verifican las condiciones que debe cumplir una tupla para ser aceptada como correcta e insertarla en la

BD o retornan valor nulo en caso de violarse alguna condición. Pueden ser programadas para dispararse antes o después de insertar, actualizar o eliminar una tupla en la tabla.

También es necesario definir el conjunto de valores lo más específico que se pueda para que no se cometan errores que afecten la integridad del sistema, por ejemplo el campo *estado* de un *expediente_analitico*, que puede ser: completo o pendiente.

Integridad de Entidad: Establece que cada entidad de la BD debe ser identificable de modo único a través de la clave primaria, garantizándose siempre su unicidad y que nunca tome valores nulos.

Datos Requeridos: Establece que una columna no puede tomar un valores nulos. Se define declarando la columna como NOT NULL cuando la tabla que la contiene se crea por primera vez. Un ejemplo de dato requerido es la columna *lote_origen* de la tabla *lote*.

Chequeo de Validez: Cuando se crea una tabla a cada columna se le asigna un tipo de datos y el SGBD se encarga de asegurar que solamente los datos del tipo especificado sean ingresados en la tabla.

Integridad Referencial: Asegura la correspondencia entre las claves foráneas y primarias de tablas relacionadas para garantizar que los registros de dichas tablas son válidos y que no se borren o cambien datos relacionados de forma accidental. Por ejemplo, la tabla *sic_0092informe_analisis* posee un campo *lote* como llave foránea que toma de la tabla *lote* por lo que se necesita validar la existencia del lote, antes de insertar o actualizar alguna tupla en la tabla *sic_0092informe_analisis*. A pesar de que PostgreSQL valida que se cumpla la integridad referencial, en la BD se hace uso de “disparadores” (triggers) para el tratamiento de errores, que permiten validar la existencia de las llaves foráneas antes de insertar información en la BD.

3.1.2 Normalización de los datos.

La teoría de la normalización se ha desarrollado para obtener estructuras de datos eficientes. El concepto de normalización fue introducido por E.D. Codd y fue pensado para aplicarse a sistemas relacionales. Sin embargo, tiene aplicaciones más amplias.

La normalización es la expresión formal del modo de realizar un buen diseño. Provee los medios necesarios para describir la estructura lógica de los datos en un sistema de información.

Ventajas:

- Evita anomalías de actualización.

- Mejora la independencia de los datos, permitiendo realizar extensiones de la BD, afectando muy poco, o nada, a los programas de aplicación existentes que accedan a la base de datos.[14]

Durante la normalización los esquemas de relación que no cumplen las condiciones se descomponen repartiendo sus atributos entre esquemas de relación más pequeños que cumplen las condiciones establecidas. En este proceso, se somete un esquema de relación a una serie de pruebas para determinar si pertenece o no a cierta forma normal.

En un principio, Codd propuso tres formas normales, primera, segunda y tercera formas normales (1FN, 2FN, 3FN). Posteriormente Boyce y Codd propusieron una definición más estricta de la 3FN a la que se conoce como forma normal Boyce-Codd (FNBC). Todas estas formas normales se basan en las dependencias funcionales entre atributos de una relación. Más adelante se propusieron la cuarta forma normal (4FN) y una quinta (5FN), con fundamento en los conceptos de dependencias multivaluadas y dependencias de reunión, respectivamente. Existe además la Sexta Forma Normal, conocida como Forma Normal de Dominio-Clave (DKFN).

Primera Forma Normal (1FN)

Una relación está en 1FN si cumple la propiedad de que sus dominios no tienen elementos que a su vez sean conjuntos, todos los atributos poseen valores atómicos y no incluye ningún grupo repetitivo.

Teniendo la BD en Primera Forma Normal se resuelve el problema de los encabezados de columna múltiples y se garantiza la inserción de los registros sin un exceso de datos en la mayoría de las tablas.

Segunda Forma Normal (2FN)

Una relación se dice que está en 2FN si está en 1FN y los atributos no llaves (ni primarias, ni candidatas), si los tiene, son funcional y completamente dependientes de la llave primaria. Este segundo paso se aplica sólo con relación a llaves compuestas.

Una vez alcanzada la 2FN, se controlan la mayoría de los problemas de lógica.

Las bases de datos pueden llevarse hasta niveles de normalización no muy elevados por razones de rendimiento, el hecho de descomponer muchas relaciones podría incidir desfavorablemente sobre la complejidad en el acceso a los datos.

La BD para el Módulo Liberación Analítica está normalizada hasta la Segunda Forma Normal, ya que en la medida que se normaliza el diseño se obtiene una BD más relacional, pero con un esquema complejo para trabajar; resultaría engorroso el acceso a los datos desde el contexto orientado a objetos que implementa Symfony, generándose clases en la capa ORM que por sí solas no representan entidades significativas del negocio y complican el acceso a los datos. Normalizando

hasta la 2FN se controlan la mayoría de los problemas de lógica y se puede insertar un registro sin exceso de datos en la mayoría de las tablas.

Inicialmente se parte del cumplimiento de la 1NF, ya que todos los valores de los atributos de cada tabla son atómicos y no presenta valores multivaluados o atributos compuestos.

En la planilla SIC-0092 se registran entre otros datos, los múltiples resultados de los ensayos realizados a determinado lote de producto, por lo que se crearon dos tablas: *sic_0092informe_analisis* que almacena los datos generales de la planilla y *analisis_de_informe* que guarda los datos referentes a cada ensayo realizado al lote en cuestión, garantizando así la atomicidad de los atributos.

Además cumple con la Segunda Forma Normal (2NF), ya que todas sus tablas están en 1FN y sus atributos no llave, dependen única y completamente de la llave primaria en las tablas con llave primaria compuesta, permitiendo que la ocurrencia de cada fila en la tabla sea única y quede representada por la llave.

3.1.3 Análisis de redundancia de datos.

La redundancia de datos es aquella información repetida que puede generar inconsistencia en la BD y requiere más espacio en memoria física. Uno de los objetivos de los Sistemas de Bases de Datos es minimizar la redundancia. En la realidad es difícil evitarla completamente; en ocasiones es deseable cierta redundancia para disminuir el tiempo de acceso a los datos o para simplificar el método de direccionado.

Debe ser controlada de forma que no existan duplicidades perjudiciales ni innecesarias y que las redundancias físicas sean tratadas por el mismo sistema, de modo que no puedan producirse incoherencias. Por tanto, un dato se actualizará lógicamente por el usuario de forma única y el sistema se preocupará de cambiar físicamente todos aquellos campos en los que el dato estuviese repetido, en caso de existir redundancia física. [15]

El sistema de planillas y libros que maneja el Grupo de Liberación Analítica posee redundancia de datos que se minimizan en la BD diseñada. La BD evita las entradas dobles de datos, por ejemplo en el SIC-0300 se registra la fecha de fabricación del lote, dato que se almacena cuando se registra el lote de producto y está en la tabla *lote* por lo que no es necesario redundar en ese dato y no se incluye como campo de la tabla *sic_0300certificado_analisis_ifa* sino que se accede a él a través del número de lote (*lote*, llave foránea que toma de la tabla *lote*).

La tabla *resultado* almacena los datos de los ensayos realizados en los laboratorios: resultado obtenido (*resultado*), nombre del analista que lo realizó (*realizado_por*) y si dio conforme o no conforme

(conformeono). Estos datos se encuentran además en las tablas de registros de los SIC de los laboratorios pero se hace necesario duplicar la información para disminuir el tiempo de acceso a ellos, pues son datos utilizados con frecuencia por el Grupo de Liberación Analítica y la forma de acceder a ellos desde las tablas de los laboratorios resulta compleja. Para controlar esta redundancia, la inserción y actualización de los datos de esta tabla se realiza de forma automática desde que se insertan los datos en los laboratorios, lo que garantiza la integridad y evita inconsistencias.

3.1.4 Análisis de seguridad de la BD.

Las BD están bajo constante amenaza de sufrir ataques de personas o sistemas no autorizados que ponen en riesgo su integridad y confidencialidad; por eso deben tomarse una serie de medidas que impidan estos sucesos y permitan conservar salvadas de la BD para restaurar los datos si se pierden o corrompen por alguna razón.

Los SGBD permiten definir autorizaciones o derechos de acceso teniendo en cuenta los usuarios, ubicaciones desde donde se puede acceder, así como asignar privilegios que tendrán los usuarios una vez autenticados.

Lo primero que debe garantizarse es que solo puedan acceder a los datos los usuarios autorizados. PostgreSQL, permite configurar los permisos de los usuarios utilizando tres niveles de acceso. En el primer nivel se configuran los permisos de conexión para los host y los usuarios a la o las BD en el archivo `pg_hba.conf`, se define que dirección o direcciones IP tendrán acceso a cuál o cuales BD y en que modo podrán conectarse: conexión sin contraseña, validando el usuario y la contraseña o que rechace cualquier conexión desde el IP o rangos IP y usuarios seleccionados. En el segundo nivel, se configuran a que BD pueden acceder determinados usuarios, utilizando las opciones del archivo `pg_ident.conf`. En el tercer nivel, permite configurar los accesos de los usuarios a las tablas de la BD, utilizando las sentencias GRANT y REVOKE se pueden asignar o denegar respectivamente los permisos para insertar, eliminar y actualizar datos, entre otros.

En la BD propuesta se establecen controles de seguridad para los datos almacenados, garantizando que solo los usuarios autorizados puedan efectuar operaciones correctas sobre algunas tablas o sobre la BD. Por la importancia de conservar los datos por un período ilimitado de tiempo, se denegó el permiso para eliminar sobre las tablas que contienen datos del Grupo de Liberación Analítica, permitiendo ejecutar sentencias INSERT, SELECT y UPDATE.

Para evitar la pérdida de los datos en caso de ocurrir alguna falla, PostgreSQL permite realizar salvadas de la BD mediante el volcado (dump), pueden realizarse de forma automática o manualmente.

Para el volcado de la BD se utiliza el comando `pg_dump` que cuenta con una serie de parámetros adicionales para indicar el nombre de la BD, el usuario y contraseña para conectarse a ella, el nombre

que tomará el archivo de salva y donde se desea guardarla. El volcado de la BD puede realizarse con el servicio PostgreSQL corriendo, se puede realizar a una o varias BD y mantiene compatibilidad entre versiones; este proceso puede ser un poco lento cuando se hacen salvas a BD de gran volumen.

Para restaurar las salvas, se utiliza el comando `pg_restore` indicando la salva que deseamos restaurar entre otros parámetros.

3.1.5 Trazabilidad de las acciones.

La trazabilidad es la cualidad que permite que todas las acciones realizadas sobre un sistema informático sean asociadas inequívocamente a un individuo o entidad. Es la capacidad que tiene una organización o sistema para rastrear, reconstruir o establecer relaciones entre objetos monitoreados, para identificar y analizar situaciones específicas o generales en los mismos. [16]

El framework de desarrollo web Symfony, sobre el que se está desarrollando la aplicación para el LIMS, guarda información sobre los eventos realizados en archivos de log que se encuentran en el directorio log del proyecto. Por cada evento realizado se genera una nueva línea en el archivo que incluye la fecha y hora a la que se ha producido, el tipo de evento, el objeto que ha sido procesado y otros detalles que dependen de cada tipo de evento y/o objeto procesado.

Cuando se realizan actualizaciones a la BD los archivos de log de Symfony no registran los valores que tenían los datos antes de ocurrir la actualización y en este proyecto se necesita conservarlos para dar cumplimiento a los requisitos funcionales relacionados con registros de trazas. Es por ello que en la BD se crearon las tablas `registro_trazas` y `registro_modificacion` para almacenar el historial completo de las operaciones realizadas, es decir, momento en que se llevo a cabo, acción realizada, usuario que inició la acción y el resultado de la misma (valores de los datos antes de la actualización y después de la actualización).

3.2 Validación funcional del diseño.

Para la realización de pruebas de un llenado voluminoso e inteligente de la BD, sin violar la integridad de la misma, se empleó la herramienta EMS PostgreSQL Data Generator 2005 que permite seleccionar tablas para generar datos, definir rangos de valores admisibles, así como la cantidad de tuplas que se desea generar. El empleo de esta herramienta posibilitó chequear la integridad de los datos y se obtuvieron muy buenos resultados.

Se generó un volumen de datos similar a lo acumulado en cinco años en el Grupo de Liberación Analítica y se muestran los tiempos de inserción en las tablas más usadas: 200 datos para la tabla *libro* (1s), 13000 para *lote* (1,42min), 14000 para *muestra* (1,56min), 200 para *tecnica* (1s), 52000 para

lote_tecnica (5,72min), 78000 para *resultados* (8,54min). Para las restantes tablas se generaron diferentes cantidades de datos y se obtuvieron tiempos de respuesta adecuados, teniendo en cuenta el gran volumen de datos insertados y la búsqueda que realiza el software en el catálogo de llaves para validar las llaves foráneas.

Se realizaron algunas consultas SQL con el objetivo de validar funcionalmente el diseño de la BD y obtener los tiempos de ejecución de las mismas, a pesar de que no serán utilizadas por la aplicación para el acceso a los datos.

Las consultas generadas para la realización de las pruebas, se escogieron de acuerdo a las operaciones que se esperan sean las más utilizadas una vez implantada la BD, por ejemplo:

Generar el Informe Analítico Semanal de resultados de productos de producción.

SELECT

```
public.muestra.fecha_recepcion_d,  
public.muestra.fecha_recepcion_m,  
public.muestra.fecha_recepcion_a,  
public.tecnica.nombre_tecnica,  
public.resultado.resultado,  
public.resultado.fecha_recepcion_d,  
public.resultado.fecha_recepcion_m,  
public.resultado.fecha_recepcion_a,  
public.lote.nombre_producto,  
public.lote.nombre_etapa
```

FROM

```
public.expediente_analitico
```

```
INNER JOIN public.lote ON (public.expediente_analitico.lote=public.lote.lote)
```

```
INNER JOIN public.lote_tecnica ON (public.lote.lote=public.lote_tecnica.lote)
```

```
INNER JOIN public.resultado ON (public.lote_tecnica.lote=public.resultado.lote)
```

```
INNER JOIN public.muestra ON (public.lote.lote=public.muestra.lote)
```

```
INNER JOIN public.tecnica ON (public.resultado.id_tecnica=public.tecnica.id_tecnica)
```

WHERE

```
(public.expediente_analitico.estado = 'Pendiente a completar') AND
```

```
(public.lote.idtipo = 1) AND
```

```
(public.muestra.fecha_recepcion_d < 8) AND
```

```
(public.muestra.fecha_recepcion_d > 1) AND  
(public.muestra.fecha_recepcion_m = 9) AND  
(public.resultado.fecha_salida_resultado_a = 2008)
```

ORDER BY

```
public.lote.nombre_producto;
```

El resultado de esta consulta fue de 122 filas y se realizó en un tiempo de 5,8 s en una búsqueda secuencial en la BD. Posteriormente se indexaron los campos fecha_salida_resultado_m y fecha_salida_resultado_a de la tabla resultado, idtipo de la tabla lote y fecha_recepcion_m y fecha_recepcion_a de la tabla muestra por ser columnas usadas frecuentemente como parámetros de búsqueda. El tipo de índice utilizado fue B-Tree pues involucra múltiples columnas y los atributos indexados utilizan los operadores <, <=, =, >=, >. También se ejecutó el comando VACUUM que realiza un proceso de limpieza en la BD para recuperar el espacio de disco ocupado por filas modificadas o borradas, actualizar las estadísticas usadas por el optimizador y evitar la pérdida de datos antiguos debido a la reutilización del identificador de transacción; esta tarea debe ejecutarla el administrador de la BD periódicamente sobre todas las tablas. Con esto se logró disminuir el tiempo de respuesta a 1,4s.

Generar el Resumen Semestral de Productos de Producción.

SELECT

```
public.muestra.observaciones,  
public.muestra.fecha_recepcion_d,  
public.muestra.fecha_recepcion_m,  
public.muestra.fecha_recepcion_a,  
public.lote.fechafablote_d,  
public.lote.fechafablote_m,  
public.lote.fechafablote_a,  
public.sic_0092informe_analisis.fecha_realizado_d,  
public.sic_0092informe_analisis.fecha_realizado_m,  
public.sic_0092informe_analisis.fecha_realizado_a
```

FROM

```
public.muestra
```

```
INNER JOIN public.lote ON (public.muestra.lote = public.lote.lote)
```

```
INNER JOIN public.sic_0092informe_analisis ON (public.lote.lote =  
public.sic_0092informe_analisis.lote)
```


WHERE

```
public.sic_0092informe_analisis.cierra_exp_no = 'true' AND  
public.muestra.fecha_recepcion_m > 6 AND  
public.muestra.fecha_recepcion_m < 13 AND  
public.muestra.fecha_recepcion_a = 2008 AND  
public.lote.nombre_producto = 'Heberpenta' AND  
public.lote.nombre_etapa = 'IFA' AND  
public.lote.idtipo = 1
```

El resultado de esta consulta fue de 210 filas y se realizó en un tiempo de 5,4s en una búsqueda secuencial. Luego de indexar los campos anteriormente mencionados y realizar la tarea VACUMM disminuyó el tiempo de respuesta a 706ms.

Las pruebas realizadas a una BD son una aproximación a los resultados reales que podrían obtenerse de la interacción directa con los usuarios y su grado de acercamiento depende de varios factores, entre ellos: cantidad de usuarios conectados y el grado de concurrencia de las solicitudes realizadas. Los resultados obtenidos en las pruebas no pueden ser motivo de confianza, la mejor prueba a realizar es la interacción directa del usuario en situaciones reales, de aquí la importancia del mantenimiento del software y la realización de versiones buscando mejorar o resolver posibles problemas que se detecten durante su utilización.

Luego de las pruebas realizadas se verificó el cumplimiento de los requisitos funcionales propuestos por los analistas.

3.3 Análisis de optimización de consultas.

Las consultas son peticiones que se realizan a la BD, para almacenar, visualizar y actualizar los datos. Una consulta puede expresarse de varias maneras, donde cada una sugiere una forma diferente de hallar el resultado y por tanto algunas pueden ser más óptimas que otras.

El acceso a los datos de la BD del LIMS se implementará mediante objetos, utilizando la capa ORM de Propel que transforma las llamadas a los objetos en consultas SQL optimizadas para el SGBD que se está utilizando y la capa de abstracción de la BD Creole que proporciona una interfaz entre el código PHP y el código SQL de la base de datos.

Las clases objeto base que genera Symfony para cada tabla, contienen una serie de constructores y accesores por defecto en función de la definición de cada columna: los métodos new(), getColumn() y

setColumna() permiten crear y obtener las propiedades de los objetos. Estos métodos utilizan una variante del estándar de codificación camelCase aplicada al nombre de las columnas.

Cuando se quiere obtener más de un registro de la BD, se puede utilizar el método doSelect() de la clase peer correspondiente a los objetos que se quieren obtener. El primer parámetro de este método es un objeto de la clase Criteria, que se utiliza para definir consultas simples sin utilizar SQL, para conseguir la abstracción de la base de datos.

Invocar el método doSelect() es más potente que una consulta SQL ya que optimiza el código SQL, devuelve un arreglo de objetos, todos los valores pasados a Criteria se escapan antes de insertarlos en el código SQL para evitar la inyección SQL, además de que la capa ORM crea y rellena de automáticamente los objetos en función del result set de la base de datos, lo que se conoce como *hidratación (hidratyng, en inglés)*.

Este proceso requiere una cantidad de tiempo directamente proporcional al número de resultados que devuelve la consulta. Debido a esto, las funciones del modelo deberían optimizarse para devolver solamente los resultados necesarios. Si no se necesitan todos los resultados devueltos, debe refinarse el objeto Criteria mediante los métodos setLimit() y setOffset() o automatizarse esta acción utilizando un paginador. El objeto sfPropelPager gestiona de forma automática los valores offset y limit para una consulta Propel, de forma que solamente se crean los objetos mostrados en cada página.

También es importante controlar el número de consultas que se realizan a la base de datos por cada petición para optimizar el código. Si se utiliza el método doSelect() para ejecutar una consulta que involucra campos de dos tablas relacionadas, posterior a la ejecución del método se debe hacer una consulta a la BD por cada objeto del que se necesite buscar datos de la tabla relacionada. Para obtener estos datos utilizando una única consulta se deben utilizar Joins. El método doSelectJoinXXX() realiza una consulta más compleja que el doSelect() donde las columnas adicionales que están presentes en el resultado obtenido permiten a Propel "*hidratar*" los objetos que hacen referencia a las tablas relacionadas. Este método se genera automáticamente cuando se ejecuta la tarea propel-build-model, para cada clave externa en las clases peer.

El resultado devuelto al emplear los métodos doSelect() y doSelectJoinXXX() es el mismo array de objetos. La diferencia se hace evidente cuando se utiliza un método *getter* asociado con una clave externa. En el caso del método doSelect(), se realiza una consulta a la base de datos y se crea un nuevo objeto con el resultado; en el caso del método doSelectJoinXXX(), el objeto asociado ya existe y no se realiza la consulta con la base de datos, por lo que el proceso es mucho más rápido. [10]

Evidentemente, la llamada al método `doSelectJoinXXX()` es un poco más lenta que la llamada a un método simple `doSelect()`, por lo que solamente mejora el rendimiento global de la página si se utilizan los objetos relacionados. [10]

En ocasiones es necesario realizar una consulta que involucre campos de varias tablas, donde la cantidad de registros a recuperar es muy grande, para estos casos es preferible obtener directamente el resultado y no obtener los objetos porque los tiempos de respuestas serían demasiado altos debido al proceso de hidratación. Bajo estas circunstancias resultaría desventajoso utilizar la ORM (Propel), pero es recomendable usar la capa de abstracción de la BD (Creole) pues se encarga de mantener la seguridad de la base de datos y el buen rendimiento y portabilidad de la aplicación, mientras que si se utilizan instrucciones PHP que accedan directamente a la base de datos se corre el riesgo de sufrir ataques de inyección SQL.

La implementación de las consultas a la BD en Symfony la realizarán los programadores del proyecto, a partir de la capa ORM generada en este trabajo. Se propone realizar un análisis de las vías más óptimas de acceso a los datos para obtener mejores tiempos de respuestas a las solicitudes.

Conclusiones

En este capítulo se han analizado aspectos a tener en cuenta para realizar un buen diseño de BD, como son: la integridad de los datos, la redundancia y la normalización, la BD propuesta se llevó hasta la segunda forma normal (2NF) por razones de rendimiento de la aplicación. También se analizó la seguridad de la BD, realizando un control de acceso y limitando los permisos de los usuarios. Para tener constancia de las acciones realizadas por los usuarios en la BD en cada momento, se hizo un análisis de la trazabilidad de las acciones, almacenándose los datos necesarios para realizar auditorías de control.

Se realizó un llenado voluminoso de la BD con la herramienta EMS Data Generator For PostgreSQL 2005 para la realización de pruebas funcionales y se probaron algunas consultas SQL obteniéndose tiempos de respuesta adecuados.

Conclusiones

Como resultado de este trabajo se diseñó una BD relacional, utilizando el SGBD PostgreSQL por la alta escalabilidad que nos brinda, permitiendo almacenar y manejar el gran volumen de información referente al Grupo de Liberación Analítica. El diseño obtenido posee 49 tablas, que almacenan la información necesaria para el trabajo correcto de este grupo.

Para garantizar el correcto funcionamiento de la BD se implementaron funciones y disparadores.

El diseño de la BD se validó teóricamente analizando su integridad, normalización, seguridad, redundancia de datos y trazabilidad de las acciones.

Se realizó una validación funcional utilizando herramientas para el llenado voluminoso de la BD y ejecutando consultas complejas. Se obtuvo como resultado una BD que permite satisfacer los requisitos funcionales planteados por los analistas.

Se concluye que todos los objetivos propuestos en la investigación se cumplieron satisfactoriamente y se incluyen una serie de recomendaciones para el trabajo futuro.

Recomendaciones

Lograr la integración con el resto de los módulos del LIMS de Calidad del CIGB.

Desarrollar un Data Warehouse para el LIMS donde se almacenen los datos que hayan permanecido por más de cinco años en el sistema y funcione como una plataforma sólida para el procesamiento informático y el análisis estadístico sobre los datos históricos.

Referencias bibliográficas

1. **Franco, Yaritza y López, Yuleimis.** *LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis de los laboratorios Sistemas Críticos y Cromatografía y Electroforesis.* 2006.
2. **CIGB.** Departamento de Control de la Calidad. [En línea] 2003. [Citado el: 10 de noviembre de 2007.] <http://www.cigb.edu.cu/pages/ccalidad.htm>.
3. —. Departamento de Aseguramiento de la Calidad. [En línea] 2003. [Citado el: 10 de noviembre de 2007.] <http://www.cigb.edu.cu/pages/acalidad.htm>.
4. —. Grupo de Recepción de Muestras. [En línea] [Citado el: 10 de noviembre de 2007.] <http://www.cigb.edu.cu/pages/grmme.htm>.
5. **Ruiz, Francisco.** El modelo de datos jerárquico. [En línea] 2001. [Citado el: 20 de noviembre de 2007.] http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T0001_MAMoraga.pdf.
6. **Márquez, María.** Apuntes de Ficheros y Bases de Datos. [En línea] 2001. [Citado el: 9 de febrero de 2008.] <http://www3.uji.es/~mmarques/f47/apun/>.
7. **Costilla, C.** Características Objeto-Relacionales del Sistema de Gestión de Bases de Datos Oracle en 2002. [En línea] 2006. [Citado el: 10 de diciembre de 2007.] <http://sinbad.dit.upm.es/docencia/grado/curso0607/BDOR%20Primera%20Parte%20documento%20Oracle%20en%202002%E2%80%9393Nov%202006.pdf>.
8. **Domínguez, Arodys y Miranda, Duniel.** SIMDEC. Sistema de Manejo de Datos de Ensayos Clínicos: Diseño e implementación de la Base de Datos. Ciudad de La Habana., 2007.
9. **Pecos, Daniel.** PostGreSQL vs. MySQL. [En línea] [Citado el: 4 de diciembre de 2007.] http://www.netpecos.org/docs/mysql_postgres/index.html.
10. **Potencier, Fabien y Zaninotto, François.** Symfony, la guía definitiva. [En línea] 2008. [Citado el: 15 de mayo de 2008.] <http://www.librosweb.es/symfony/index.html>.
11. **Booch, G., Rumbaugh, J. y Jacobson, I.** El Lenguaje Unificado de Modelado. 2000.
12. **Périsse, Marcelo Claudio.** Proyecto informático. Una metodología simplificada. [En línea] febrero de 2001. [Citado el: 15 de diciembre de 2007.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/>.
13. **SQLMANAGER.** EMS Manager for PostgreSQL Overview. [En línea] [Citado el: 4 de diciembre de 2007.] <http://www.sqlmanager.net/products/postgresql/manager>.

14. **Mato García, Rosa María.** Sistemas de Bases de Datos.
15. **Norick.** Sistemas de Bases de datos. [En línea] 2003. [Citado el: 25 de mayo de 2008.]
<http://usuarios.lycos.es/cursosgbd/UD2.htm>.
16. **Cano, Jeimy J.** Trazabilidad de las Operaciones Electrónicas. [En línea] [Citado el: 26 de mayo de 2008.]
<http://www.isaca.org/Template.cfm?Section=Home&CONTENTID=34423&TEMPLATE=/ContentManagement/ContentDisplay.cfm>.
17. **Sarmiento, Alieksy y Cutiño, Elián.** LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis del Grupo de Recepción de Muestras. 2006.
18. Guía Internacional Armonizada para IFA. *Buenas Prácticas de Manufactura.* 1999.
19. **Lellelid, Hans.** Propel- Guía de usuario. [En línea] 2004. [Citado el: 31 de mayo de 2008.]
http://propel.phpdb.org/docs/es/user_guide/.

Bibliografía

- **Alarcón, Pedro Pablo y Garbajosa, Juan.** Administración de Bases de Datos. Tema 6. Restricciones y Control de Integridad. [En línea] [Citado el: 16 de mayo de 2008.]
- **Alarcón Medina, José Manuel.** Administración SGBD PostgreSQL. [En línea] 2006. [Citado el: 1 de junio de 2008.] Administración SGBD PostgreSQL.
- Ayuda Extendida del Rational Enterprise Edition 2007.
- **Benítez, Rafael, Roa, Iván y Romero, Diego.** Bases de Datos de Orientadas a Objetos. [En línea] 2002. [Citado el: 5 de diciembre de 2007.] <http://www.dei.uc.edu.py/tai2002/BDOO>.
- **Castillo, Leonel de Jesús y Álvarez, Yaniel.** Base de Datos para la Residencia de la UCI. Ciudad de La Habana., 2007.
- **Creole Project.** Creole Project. [En línea] 2006. [Citado el: 31 de mayo de 2008.] <http://creole.phpdb.org/trac/>.
- **De La Herrán, Manuel.** Administración y Optimización de Bases de Datos Oracle. Conceptos de Bases de Datos Multidimensionales. [En línea] 2004. [Citado el: 11 de febrero de 2008.] <http://www.redcientifica.com/oracle/c0001p0005.html>
- **Devarakonda, Ramakanth S.** Sistemas de Base de datos Relacionado a Objeto - El Camino hacia Adelante. [En línea] [Citado el: 20 de diciembre de 2007.] <http://www.acm.org/crossroads/espanol/xrds7-3/ordbms.html>.
- **Díaz, Elennis y Nieto, Liusmila.** LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis de la Sección de Mejoramiento de la Calidad y del Grupo de Desarrollo. Ciudad de La Habana, 2007.
- **Domínguez, Arodys y Miranda, Duniel.** SIMDEC. Sistema de Manejo de Datos de Ensayos Clínicos: Diseño e implementación de la Base de Datos. Ciudad de La Habana., 2007. http://www.oei.eui.upm.es/Asignaturas/BD/ABD/doc/temas/abdTema6_05-06.pdf.
- **Durantes, Victor Hugo, Magariño, Fernando y Neif, José.** Curso de BD y PostgreSQL. [En línea] 21 de 4 de 2001. [Citado el: 20 de febrero de 2008.] http://alfa.facyt.uc.edu.ve/computacion/pensum/cs0347/Laborat/Taller_PostgreSQL.pdf.

- **Equipo de desarrollo de PostgreSQL.** PostgreSQL. [En línea] 2002. [Citado el: 15 de marzo de 2008.] <http://www.ibiblio.org/pub/linux/docs/LuCaS/Postgresqls/web/navegable/todopostgresql/postgres.htm>.
- **Gutiérrez, Javier J.** ¿Qué es un framework? [En línea] [Citado el: 10 de mayo de 2008.] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
- **MYSQL AB.** MySQL Documentation. [En línea] 2008. [Citado el: 5 de junio de 2008.] <http://dev.mysql.com/doc/>.
- **Nipotti, Claudio.** ¿Por qué servidores de datos? [En línea] 2006. [Citado el: 20 de noviembre de 2007.] <http://www.esbiza.com.ar/dl/cliente-servidor.pdf>
- **Potencier, Fabien y Zaninotto, François.** Symfony, la guía definitiva. [En línea] 2008. [Citado el: 15 de mayo de 2008.] <http://www.librosweb.es/symfony/index.html>.
- **Pozo, Salvador.** MySQL con clase. Gestión de Bases de Datos. [En línea] 2004. [Citado el: 15 de noviembre de 2007.] <http://mysql.conclase.net/curso/index.php>.
- **Sánchez, Jorge.** Unidad 1 Gestión y Diseño de Bases de Datos. Sistemas Gestores de Bases de Datos. [En línea] 2007. [Citado el: 10 de enero de 2008.] <http://www.jorgesanchez.net/bd/apuntes07/sghbd0107.pdf>
- **Sarmiento, Aliksy y Cutiño, Elián.** LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis del Grupo de Recepción de Muestras. 2006.
- **SQLMANAGER.** EMS Manager for PostgreSQL Overview. [En línea] [Citado el: 4 de diciembre de 2007.] <http://www.sqlmanager.net/products/postgresql/manager>.
- —. EMS SQL Query for PostgreSQL Overview. [En línea] [Citado el: 3 de febrero de 2008.] <http://www.sqlmanager.net/en/products/postgresql/query>.

Glosario de términos

ACID: (*Atomicity, Consistency, Isolation, Durability*). Propiedades importantes de las BD relacionales.

ANSI-SPARC: (*American National Standard Institute - Standards Planning and Requirements Committee*). Comité de estandarización de Bases de Datos.

API: (*Application Programming Interface*). Especificación de una librería o utilidad que documenta su interfaz y permite su uso sin conocimiento de su interior.

Array: Tipo de dato que representa una colección de datos del mismo tipo a los que se accede mediante índices.

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. [17]

Atomicity: (*Atomicidad*). Propiedad de las BD que asegura que las transacciones se realicen en su totalidad o no se realicen.

BD: (*Base de Datos*). Colección de datos interrelacionados, variables en el tiempo, que pueden ser utilizados por uno o varios programas de aplicación.

Conforme: Condición de un lote de producto que indica que cumple con las especificaciones requeridas.

Consistency: (*Consistencia*). Propiedad de las BD que asegura que después de ejecutarse las transacciones los datos deben quedar en un estado consistente.

Creole: Capa de abstracción de bases de datos para PHP5.

Durabilidad: (*Durability*). Propiedad de las BD que asegura que una vez ejecutada una transacción ésta persistirá aunque falle el sistema.

Expediente del Producto: Conjunto de todos los resultados analíticos y las investigaciones analíticas realizadas a un lote según se han descrito previamente en una especificación de acuerdo con las normas nacionales o internacionales establecidas para ese producto en cuestión. [17]

Framework: Es una estructura de soporte, utilizada en el desarrollo de software, sobre la cual pueden organizarse y desarrollarse proyectos. Representa una arquitectura de software y provee una estructura y una metodología de trabajo.

IDE: (*Integrated Development Environment*). Entorno de desarrollo, es programa compuesto por un conjunto de herramientas para un programador de software.

Ingredientes Farmacéuticos Activos (IFA) o Producto Intermedio: Sustancia farmacéutica activa o producto intermedio de calidad y pureza establecidas en comparación con estándar primario, que se utiliza como sustancia de referencia para los ensayos rutinarios de laboratorio. [18]

Informe de Análisis: Informe que se realiza como resumen de los ensayos a que ha sido sometido un lote.

Isolation: (*Aislamiento*). Propiedad de las BD que asegura que una transacción no puede utilizar datos de otra que no ha concluido su ejecución.

Kerberos: Protocolo de autenticación de redes de computadoras que permite a dos ordenadores demostrar su identidad mutuamente de manera segura en una red insegura

Log: Registro de eventos ocurridos en un periodo de tiempo en particular.

Lote: Cantidad definida de materia prima, material de envase o producto terminado, elaborado en un solo proceso o una serie de procesos de tal manera que puede esperarse que sea homogéneo. En el caso de un proceso continuo de fabricación, el lote debe corresponder a una fracción definida de la producción, caracterizada por la homogeneidad que se busca en el producto. [17]

MVC: (*Modelo Vista Controlador*). 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.

No Conforme: Condición un lote de producto que indica que no cumple con las especificaciones requeridas.

ORM: (*Object-Relational Mapping*) Mapeo de objetos a bases de datos, es una técnica de programación utilizada para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos. Crea una base de datos orientada a objetos virtual, por encima de la base de datos relacional.

Procedimiento Patrón de Operación: Documento en el que se refleja la secuencia detallada de pasos a seguir para llevar a cabo una actividad.

Producto: Es el resultado de una investigación al que finalmente se le establecen dos nombres: uno genérico y otro comercial. Está destinado al uso que su investigador haya propuesto y definido. [17]

Producto terminado: Producto que ha sido sometido a todas las etapas de producción, incluyendo el envase final y el etiquetado. [17]

Propel: Es un servicio de objeto persistente y de consulta que provee un sistema para almacenar objetos en una base de datos y un sistema para búsqueda y restauración de objetos desde una base de datos. [19]

RUP: (*Rational Unified Process*) Proceso unificado de desarrollo de software. Metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

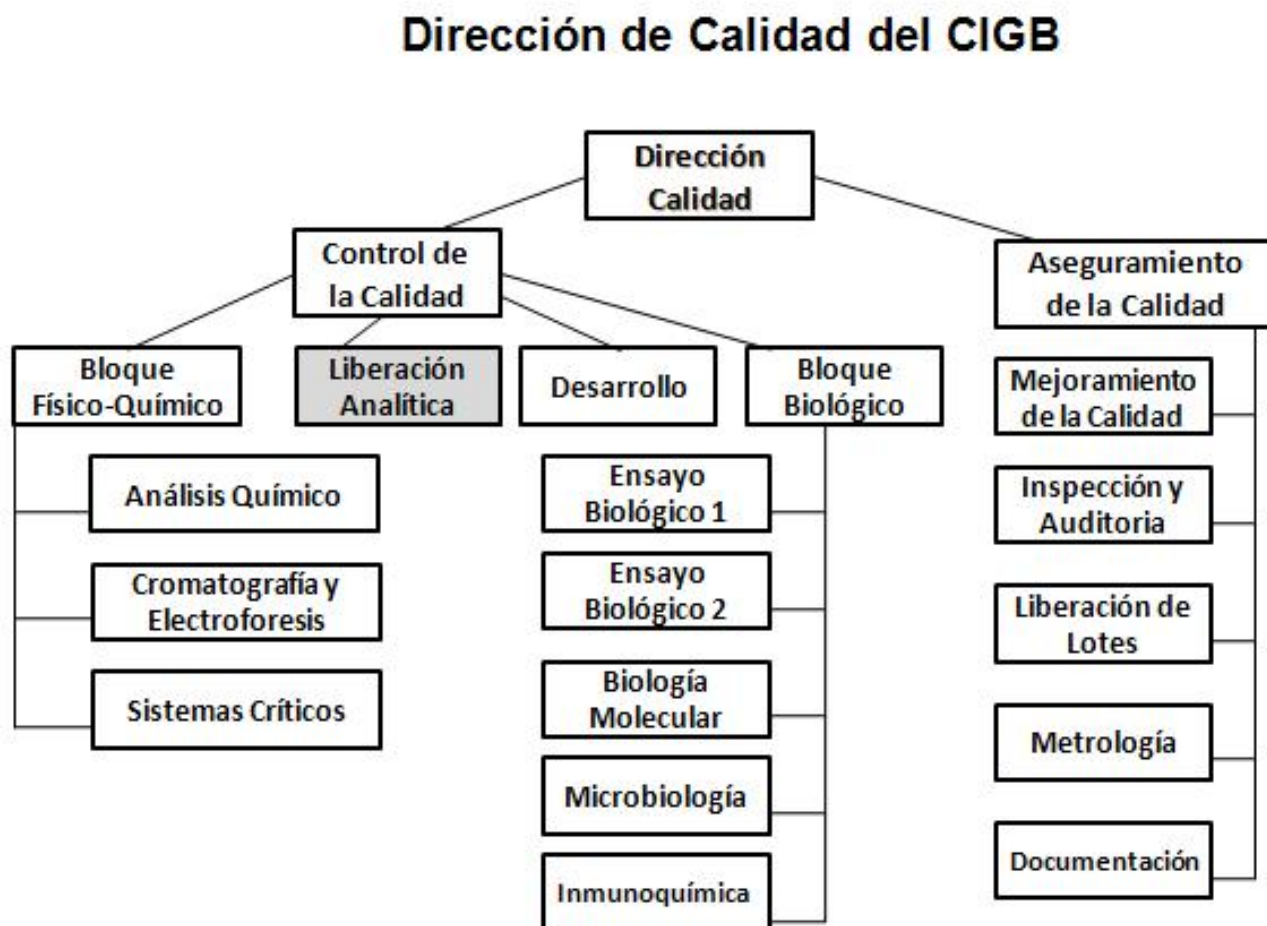
SGBD: (*Sistema Gestión de Bases de Datos*). Aplicación que permite a los usuarios definir, crear y mantener bases de datos, y proporciona acceso controlado a la misma. [6]

SIC: Sistema de Información y Control.

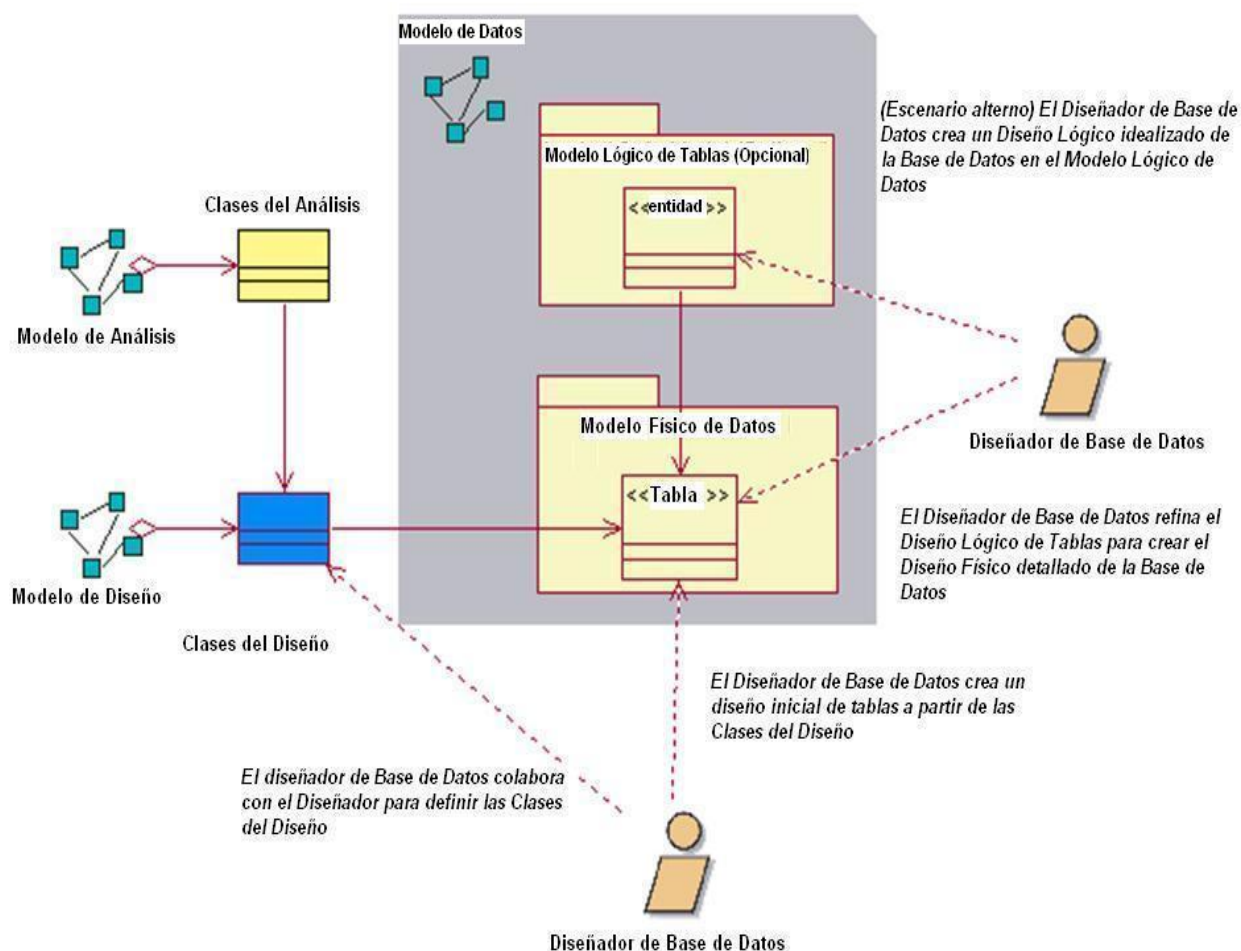
SSL: (*Secure Sockets Layer*) Protocolo que proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía.

Anexos

Anexo 1. Organigrama de la Dirección de Calidad del CIGB



Anexo 2. Actividades del Rol de Diseñador de Base de Datos según RUP



Anexo 3. Descripción de Clases Persistentes.

Nombre: InformeCostos	
Tipo de clase: Entidad	
Atributo	Tipo
idInforme	Integer
fechaConfeccion_D	Short
fechaConfeccion_M	Short
fechaConfeccion_A	Integer
Responsabilidades:	
Nombre:	InformeCostos()
Descripción:	Constructor

Nombre: CostoProduccion	
Tipo de clase: Entidad	
Atributo	Tipo
idProducto	Integer
producto	String
etapa	String
cant_Lotes	Integer
cant_Muestras	Integer
Responsabilidades:	
Nombre:	CostoProduccion()
Descripción:	Constructor

Nombre: NombreMuestra	
Tipo de clase: Entidad	
Atributo	Tipo
muestra	Integer
Responsabilidades:	
Nombre:	NombreMuestra()
Descripción:	Constructor

Nombre: CostoOtros	
Tipo de clase: Entidad	
Atributo	Tipo
idProducto	Integer
producto	String
ensayo	String
cantidad	String
Responsabilidades:	
Nombre:	CostoOtros()
Descripción:	Constructor

Nombre: InformeSemanalResultados	
Tipo de clase: Entidad	
Atributo	Tipo
idInforme	Integer
producto	String
etapa	String
fechaEntrada_D	Short
fechaEntrada_M	Short
fechaEntrada_A	Integer
fechaConfeccion_D	Short
fechaConfeccion_M	Short
fechaConfeccion_A	Integer
Responsabilidades:	
Nombre:	InformeSemanalResultados()

Descripción:	Constructor
---------------------	-------------

Nombre: ResultadoProduccionCP	
Tipo de clase: Entidad	
Atributo	Tipo
idResultado	Integer
ensayo	String
estadoEnsayo	String
fechaAnalista_D	Short
fechaAnalista_M	Short
fechaAnalista_A	Integer
Responsabilidades:	
Nombre:	ResultadosProduccionCP()
Descripción:	Constructor

Nombre: ResultadoExInv	
Tipo de clase: Entidad	
Atributo	Tipo
idResultado	Integer
fechaAnalista_D	Short
fechaAnalista_M	Short
fechaAnalista_A	Integer
fechaSalidaResultado_D	Short
fechaSalidaResultado_M	Short
fechaSalidaResultado_A	Integer
valorResultado	String
Responsabilidades:	
Nombre:	ResultadoExInv()
Descripción:	Constructor

Nombre: ResumenSemestralProducto	
Tipo de clase: Entidad	
Atributo	Tipo
idResumen	Integer
fechaConfeccion_D	Short
fechaConfeccion_M	Short
fechaConfeccion_A	Integer
cantLotes	Integer
Responsabilidades:	
Nombre:	ResumenSemestralProducto()
Descripción:	Constructor

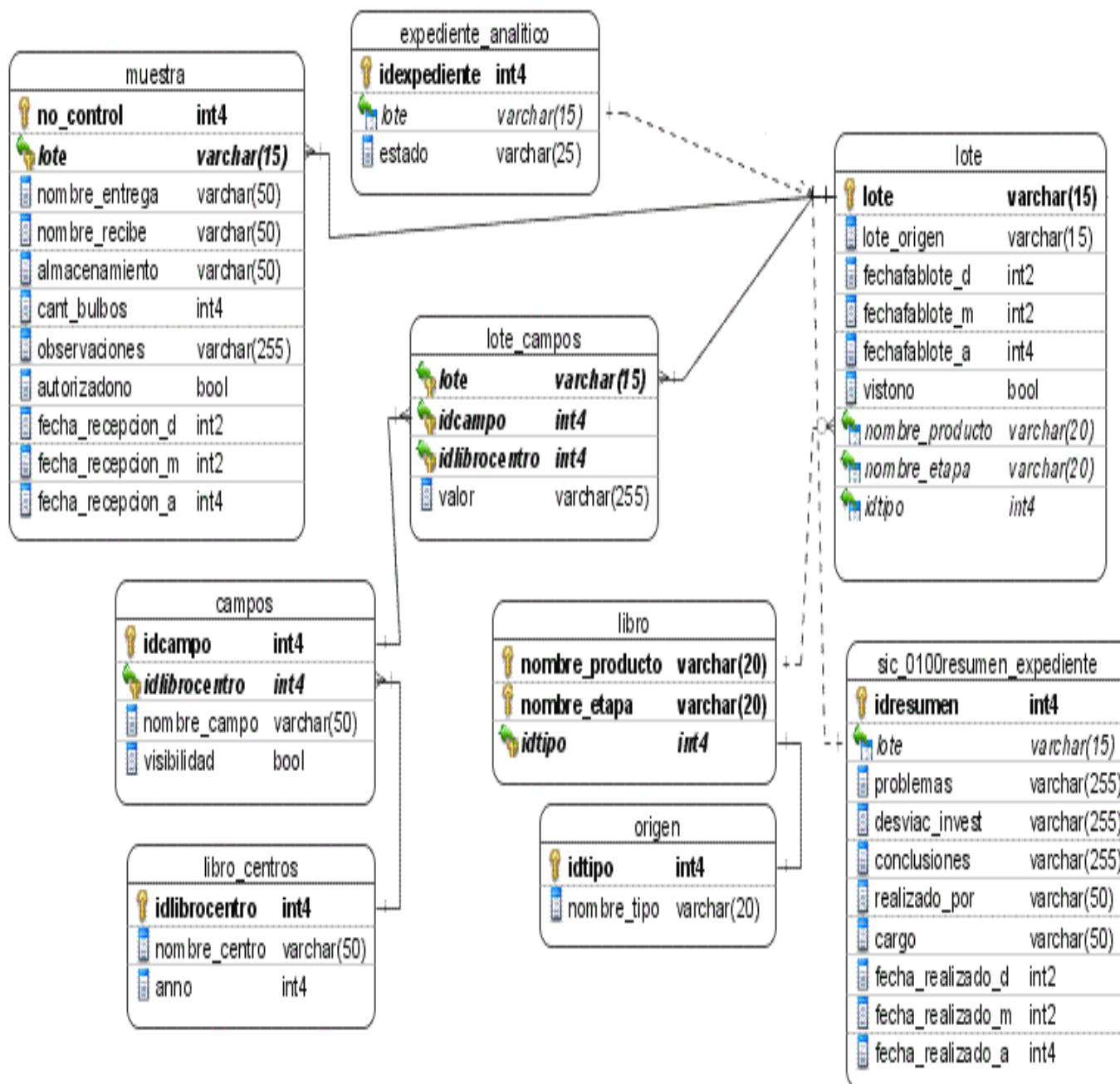
Nombre: ResumenProduccion	
Tipo de clase: Entidad	
Atributo	Tipo

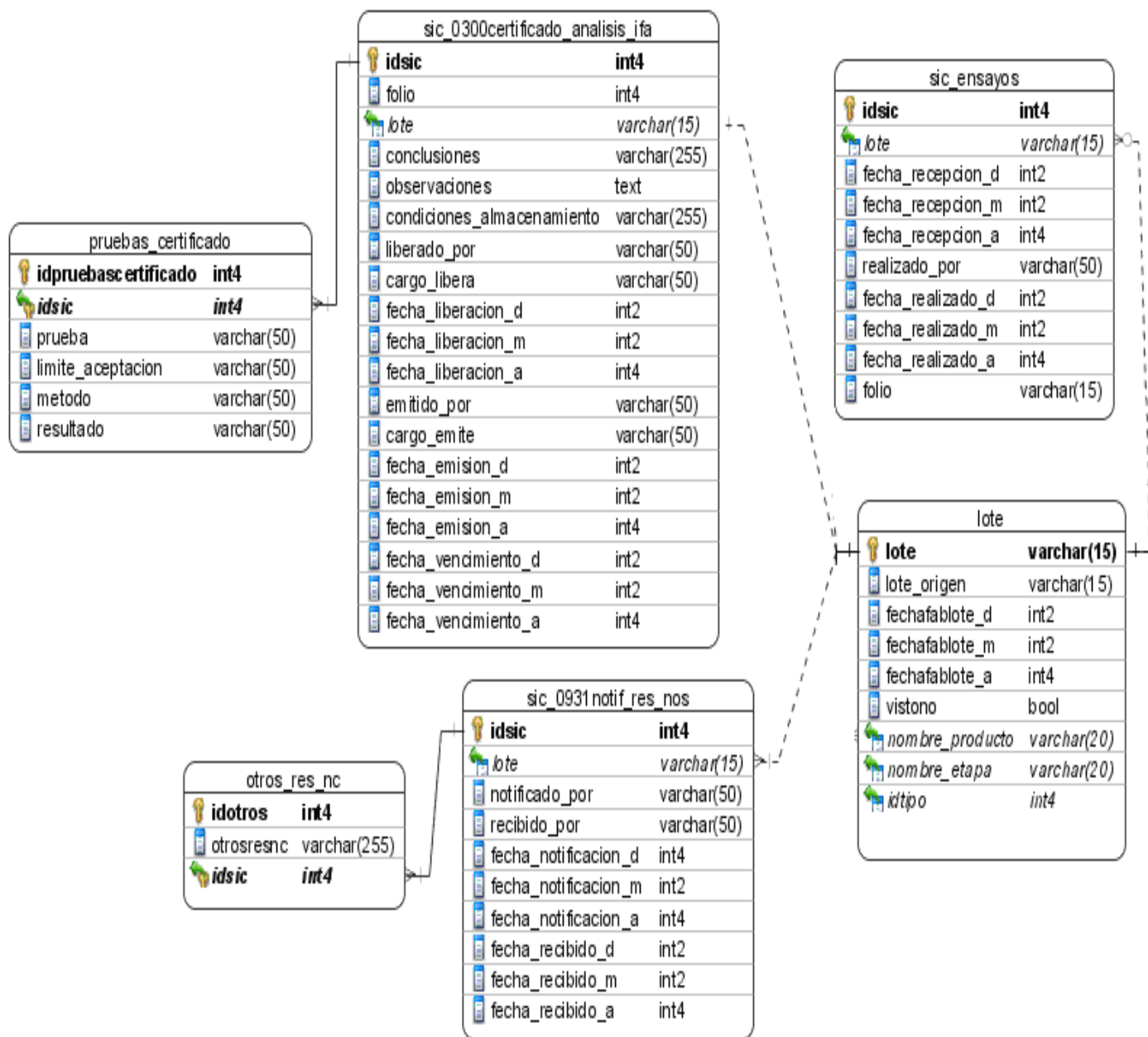
idResumenProduccion	Integer
lote	String
producto	String
etapa	String
fechaFab_D	Short
fechaFab_M	Short
fechaFab_A	Integer
fechaEntrada_D	Short
fechaEntrada_M	Short
fechaEntrada_A	Integer
fechaSIC0092_D	Short
fechaSIC0092_M	Short
fechaSIC0092_A	Integer
difTiempo	Integer
observaciones	String
Responsabilidades:	
Nombre:	ResumenProduccion()
Descripción:	Constructor

Nombre: ResumenCtrolProc	
Tipo de clase: Entidad	
Atributo	Tipo
idResumenCtrolProc	Integer
lote	String
producto	String
etapa	String
fechaFab_D	Short
fechaFab_M	Short
fechaFab_A	Integer
fechaEntrada_D	Short
fechaEntrada_M	Short
fechaEntrada_A	Integer
fechaUltimoEnsayo_D	Short
fechaUltimoEnsayo_M	Short
fechaUltimoEnsayo_A	Integer
difTiempo	Integer
observaciones	String
Responsabilidades:	
Nombre:	ResumenCtrolProc()
Descripción:	Constructor

Nombre: ResumenExInv	
Tipo de clase: Entidad	
Atributo	Tipo
idResumenExInv	Integer
cantLotes	Integer
cantMuestras	Integer
Responsabilidades:	
Nombre:	ResumenExInv()
Descripción:	Constructor

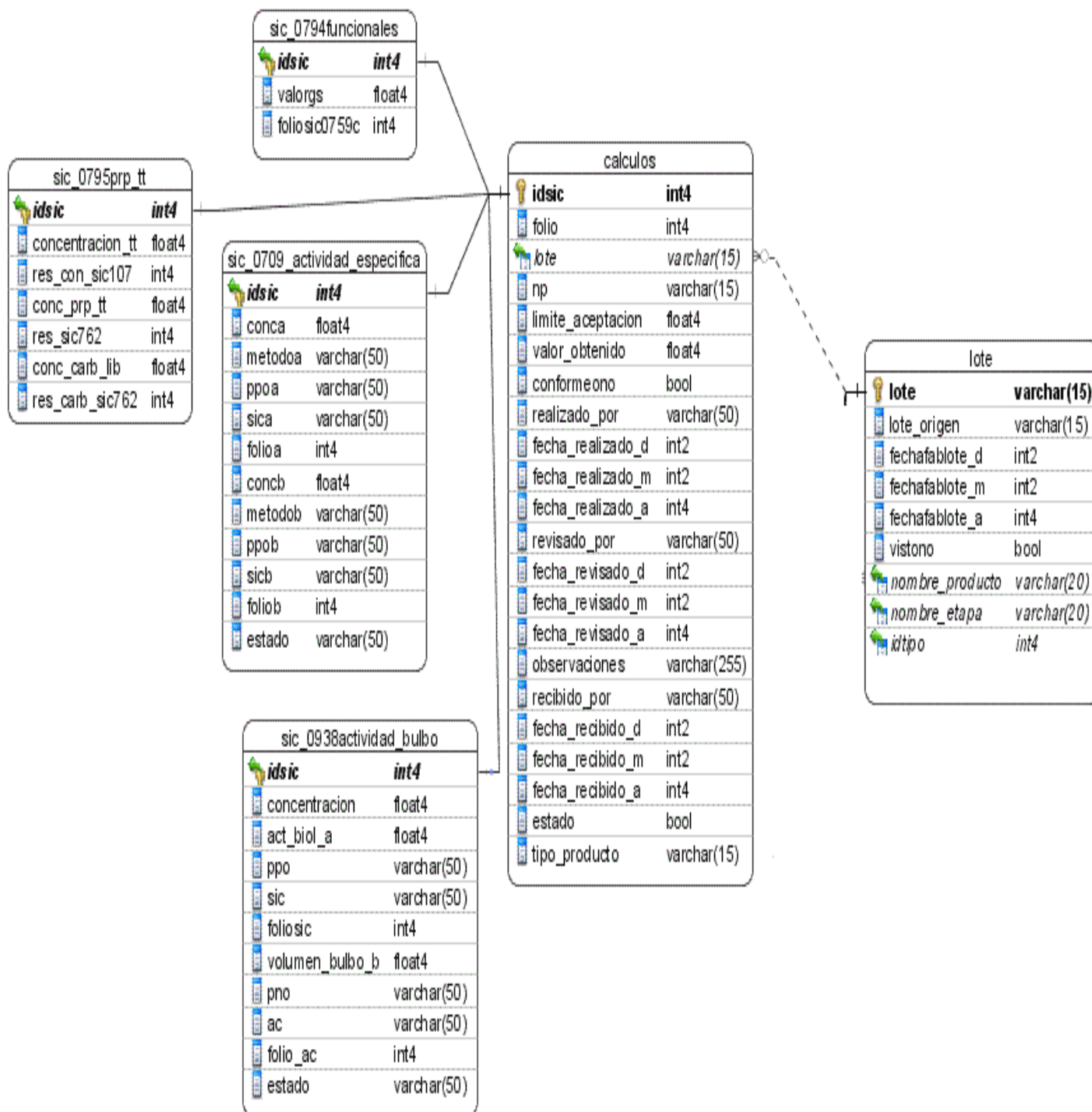
Anexo 4. Modelo Físico de Datos.

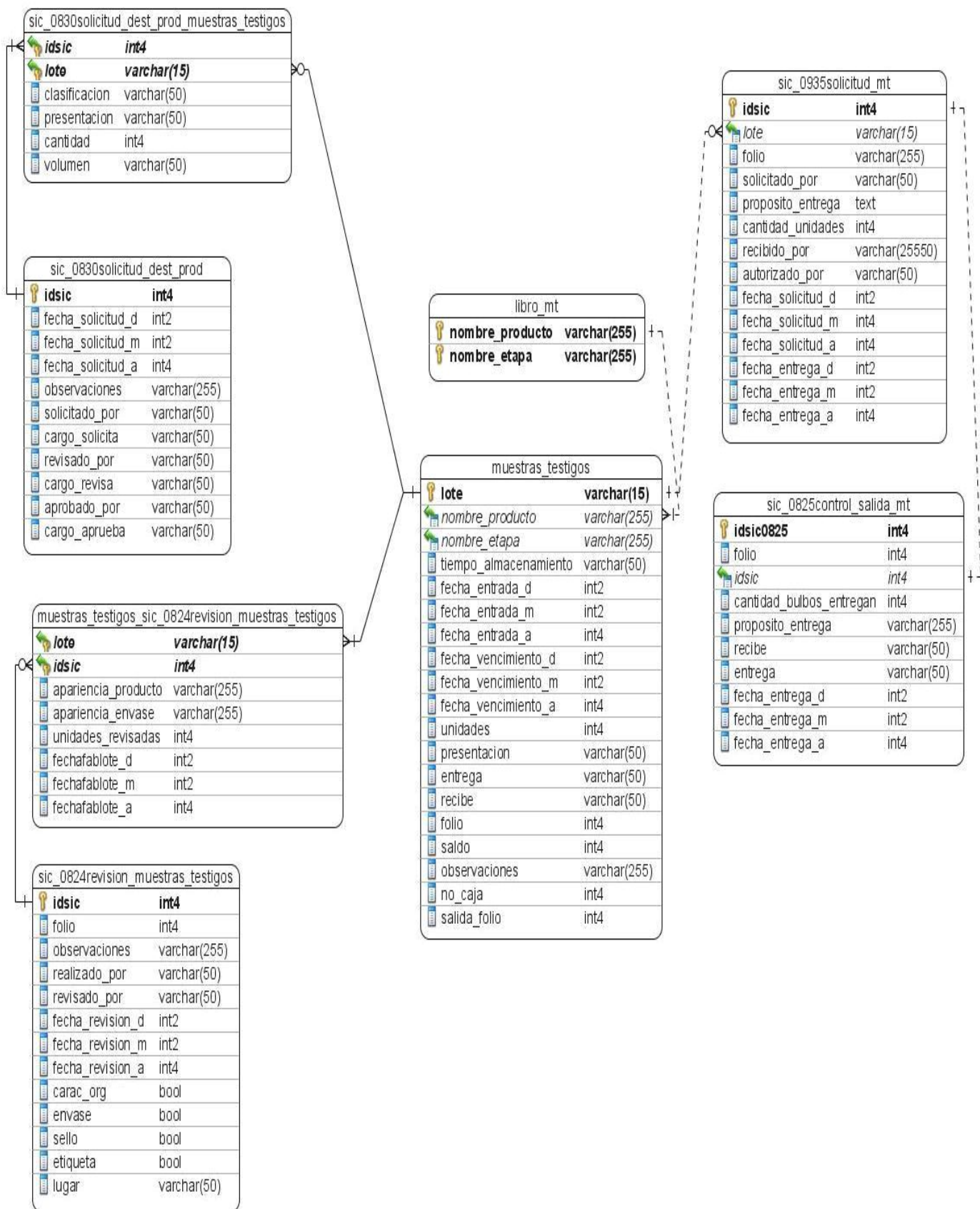


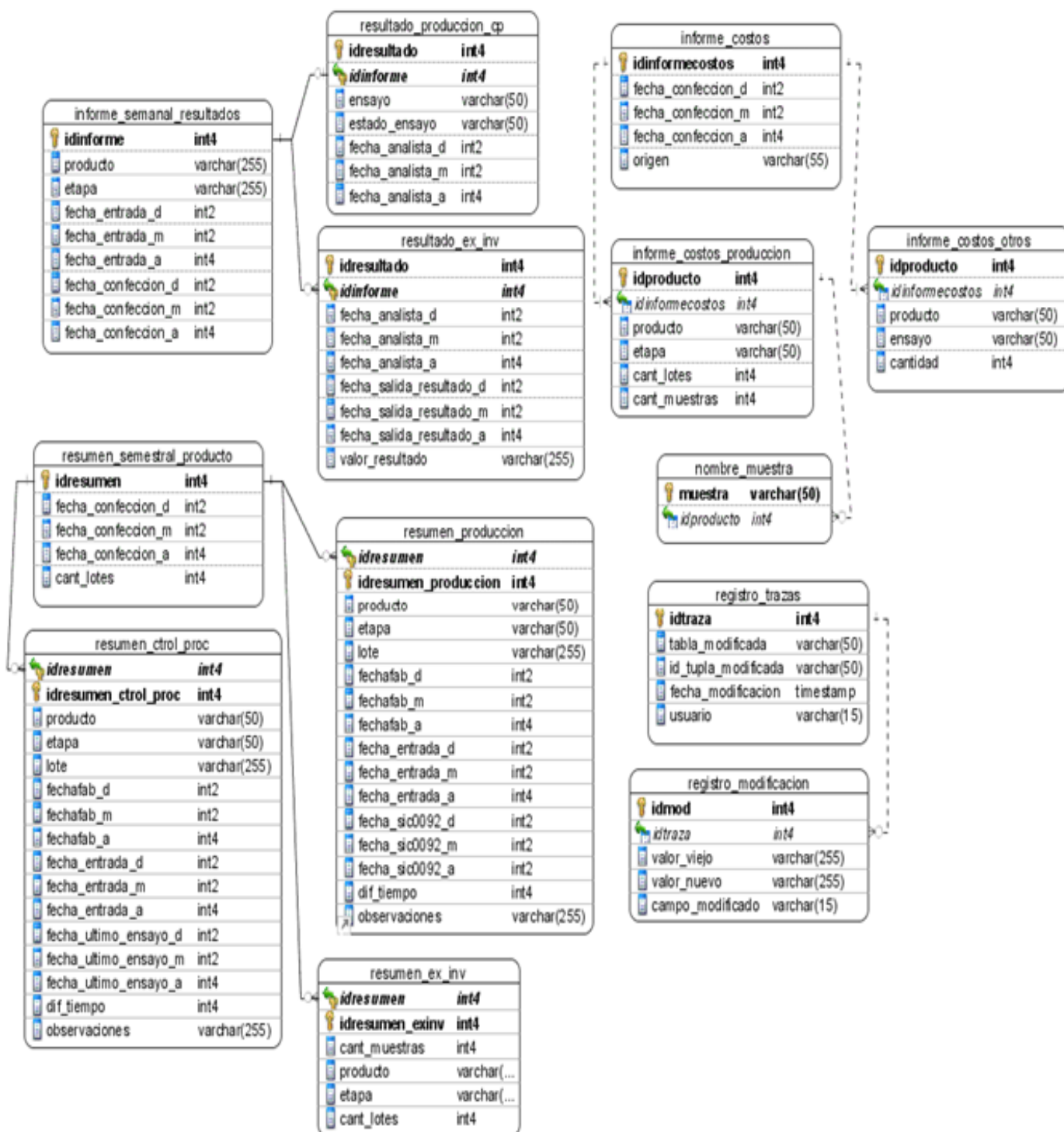












Anexo 5. Descripción de tablas de la BD.

Nombre: informe_costos		
Descripción: Son datos del informe de costos mensuales para determinada fuente de origen de productos en un mes específico.		
Atributo	Tipo	Descripción
idinformecostos	Int4	Valor entero de autoincremento llave primaria.
fecha_confeccion_d	Int2	Día de confección del informe.
fecha_confeccion_m	Int2	Mes de confección del informe.
fecha_confeccion_a	Int4	Año de confección del informe.

Nombre: costos_produccion		
Descripción: Representa la información de un producto que se almacena en el informe de costos mensuales para la fuente de origen de productos: Producción.		
Atributo	Tipo	Descripción
idproducto	Int4	Valor entero de autoincremento llave primaria.
idinformecostos	Int4	Llave foránea que toma de la entidad informe_costos.
producto	Varchar(50)	Nombre del producto.
etapa	Varchar(50)	Nombre de la etapa en que se encuentra la muestra.
cant_lotes	Int4	Cantidad de lotes del producto recepcionados en el mes.
cant_muestras	Int4	Cantidad de ensayos realizados al total de lotes.

Nombre: nombre_muestra		
Descripción: Representa los nombres de los ensayos realizados a los lotes del producto para el que se genera el informe mensual de costos de determinado mes.		
Atributo	Tipo	Descripción
muestra	Varchar(50)	Nombre del ensayo, llave primaria
idproducto	Int4	Llave foránea que toma de la entidad costos_produccion.

Nombre: costos_otros		
Descripción: Representa la información de un producto que se almacena en el informe de costos mensuales para la fuente de origen de productos que no sean de Producción.		
Atributo	Tipo	Descripción
idproducto	Int4	Valor entero de autoincremento llave primaria.
idinformecostos	Int4	Llave foránea que toma de la entidad informe_costos.
producto	Varchar(50)	
ensayo	Varchar(50)	
cantidad	Int4	

Nombre: informe_semanal_resultados		
Descripción: Almacena datos de un informe semanal de resultados para los productos que tienen el Expediente Analítico abierto, o sea, que tienen ensayos en desarrollo.		
Atributo	Tipo	Descripción
idinforme	Int4	Valor entero de autoincremento llave primaria.
producto	Varchar(50)	Nombre del producto.
etapa	Varchar(50)	Nombre de la etapa en que se encuentra el producto.
fecha_entrada_d	Int2	Día de entrada del producto a liberación analítica.
fecha_entrada_m	Int2	Mes de entrada del producto a liberación analítica.
fecha_entrada_a	Int4	Año de entrada del producto a liberación analítica.
fecha_confeccion_d	Int2	Día de confección del informe.
fecha_confeccion_m	Int2	Mes de confección del informe.
fecha_confeccion_a	Int4	Año de confección del informe.

Nombre: resultado_produccion_cp		
Descripción: Almacena los datos de un resultado contenido en el informe semanal de resultados para los productos de Control de Procesos de Producción.		
Atributo	Tipo	Descripción
idresultado	Int4	Valor entero de autoincremento llave primaria.
idinforme	Int4	Llave foránea que toma de la tabla informe_semanal_resultados.
ensayo	Varchar(50)	Nombre del producto
estado_ensayo	Varchar(50)	Nombre de la etapa en que se encuentra el producto
fecha_analista_d	Int2	Día de recepción de la muestra por el analista del laboratorio.
fecha_analista_m	Int2	Mes de recepción de la muestra por el analista del laboratorio.
fecha_analista_a	Int4	Año de recepción de la muestra por el analista del laboratorio.

Nombre: resultado_ex_inv		
Descripción: Almacena los datos de un resultado contenido en el informe semanal de resultados para los productos de Extraplan de Producción e Investigaciones.		
Atributo	Tipo	Descripción
idresultado	Int4	Valor entero de autoincremento llave primaria.
idinforme	Int4	Llave foránea que toma de la tabla informe_semanal_resultados.
fecha_analista_d	Int2	Día de recepción de la muestra por el analista del laboratorio.
fecha_analista_m	Int2	Mes de recepción de la muestra por el analista del laboratorio.
fecha_analista_a	Int4	Año de recepción de la muestra por el analista del laboratorio.
fecha_salida_resultado_d	Int2	Día de salida del resultado.

fecha_salida_resultado_m	Int2	Mes de salida del resultado.
fecha_salida_resultado_a	Int4	Año de salida del resultado.

Nombre: resumen_semestral_producto		
Descripción: Representa un resumen semestral de los lotes de un producto.		
Atributo	Tipo	Descripción
idresumen	Int4	Entero autoincremental, llave primaria.
producto	Varchar(50)	Nombre del producto.
etapa	Varchar(50)	Nombre de la etapa de la muestra.
origen	Varchar(50)	Fuente de donde proviene el producto.
fecha_confeccion_d	Int2	Día de confección del resumen semestral.
fecha_confeccion_m	Int2	Mes de confección del resumen semestral.
fecha_confeccion_a	Int4	Año de confección del resumen semestral.
cant_lotes	Int4	Cantidad de lotes del producto recepcionados en el semestre.

Nombre: resumen_produccion		
Descripción: Son los datos que se recogen en el resumen semestral del producto para los productos de Producción		
Atributo	Tipo	Descripción
idresumen_produccion	Int4	Entero de autoincremento, Llave primaria.
idresumen	Int4	Llave primaria que toma de la tabla resumen_semestral_producto.
lote	Varchar(50)	Lote del producto.
producto	Varchar(50)	Etapa en que está el producto.
etapa	Varchar(15)	Lote del producto.
fecha_fab_d	Int2	Día de fabricación del producto.
fecha_fab_m	Int2	Mes de fabricación del producto.
fecha_fab_a	Int4	Año de fabricación del producto.
fecha_entrada_d	Int2	Día de entrada a liberación analítica.
fecha_entrada_m	Int2	Mes de entrada a liberación analítica.
fecha_entrada_a	Int4	Año de entrada a liberación analítica.
dif:_tiempo	Int4	Diferencia de tiempo desde que entró la muestra hasta que se cerró su expediente.
fecha_sic0092_d	In2	Día de confección del SIC-0092 que cerró el expediente analítico del lote en cuestión.
fecha_sic0092_m	In2	Mes de confección del SIC-0092 que cerró el expediente analítico del lote en cuestión.
fecha_sic0092_a	In2	Año de confección del SIC-0092 que cerró el expediente analítico del lote en cuestión.
observaciones	Varchar(255)	Observaciones generales.

Nombre: resumen_ctrol_proc		
Descripción: Son los datos que se recogen en el resumen semestral del producto para los productos de Controles de Procesos		
Atributo	Tipo	Descripción
idresumen_ctrol_proc	Int4	Entero autoincremental, llave primaria.

idresumen	Int4	Llave foránea que toma de la tabla resumen_semestral_producto.
lote	Varchar(15)	Lote del producto.
producto	Varchar(50)	Etapa en que está el producto.
etapa	Varchar(15)	Lote del producto.
fecha_fab_d	Int2	Día de fabricación del producto.
fecha_fab_m	Int2	Mes de fabricación del producto.
fecha_fab_a	Int4	Año de fabricación del producto.
fecha_entrada_d	Int2	Día de fabricación del producto.
fecha_entrada_m	Int2	Mes de fabricación del producto.
fecha_entrada_a	Int4	Año de fabricación del producto.
fecha_ultimo_ensayo_d	Int2	Día en que se realizó el último ensayo.
fecha_ultimo_ensayo_m	Int2	Mes en que se realizó el último ensayo.
fecha_ultimo_ensayo_a	Int4	Año en que se realizó el último ensayo.
dif_tiempo	Int4	Diferencia de tiempo desde que entró la muestra hasta que se cerro su expediente.
observaciones	Varchar(255)	Observaciones generales.

Nombre: resumen_ex_inv		
Descripción: Son los datos que se recogen en el resumen semestral del producto para los productos de Extraplan e Investigaciones.		
Atributo	Tipo	Descripción
Idresumen_exinv	Int4	Entero autoincremental, llave primaria.
idresumen	Int4	Llave foránea que toma de la tabla resumen_semestral_producto.
cant_muestras	Int4	Cantidad de ensayos realizados al lote del producto en el semestre.

Nombre: sic_ensayos		
Descripción: Tabla que guarda los datos generales de todos los SIC de ensayos de los laboratorio del CIGB.		
Atributo	Tipo	Descripción
idsic	Int4	Entero autoincremental, llave primaria.
lote	Varchar(15)	Llave foránea que toma de la tabla lote.
fecha_recepcion_d	Int2	Día de recepción de la muestra en el laboratorio.
fecha_recepcion_m	Int2	Mes de recepción de la muestra en el laboratorio.
fecha_recepcion_a	Int2	Año de recepción de la muestra en el laboratorio.
realizado_por	Varchar(50)	Nombre del analista que realizó el ensayo.
fecha_realizado_d	Int2	Día de realizado el ensayo.
fecha_realizado_m	Int2	Mes de realizado el ensayo.
fecha_realizado_a	Int4	Año de realizado el ensayo.
folio	Int4	Folio del SIC.