

Universidad de las Ciencias Informáticas

Facultad 6



**Título: Sistema para la Gestión de la Información de los Laboratorios de la
Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología:
Desarrollo de la Base de Datos del
Módulo de Microbiología.**

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

Autores: José Luis Pita Lambertt.

Victor Manuel Gómez González.

Tutor: Ing. Alieski Sarmiento Almenares.

Co-tutor: Lic. Miulkenia Navarro Reyes.

Junio, 2008

“Año 50 de la Revolución”

“Acumular información es sólo el primer paso hacia la sabiduría. Pero compartir información es el primer paso hacia la comunidad. ”

Henry Lewis Gates.

DECLARACIÓN DE AUTORÍA

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

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

José Luis Pita Lambertt Víctor Manuel Gómez González Ing. Alieski Sarmiento Almenares.

Firma del Autor

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Ingeniero: Alieski Sarmiento Almenares.

Universidad de Ciencias Informáticas, Habana, Cuba.

E-mail: asarmiento@uci.cu

Licenciada: Miulkenia Navarro Reyes.

Universidad de Ciencias Informáticas, Habana, Cuba.

E-mail: mnavarror@uci.cu

DEDICATORIA

De Jose Luis:

A mis padres, por haber sido fuente de inspiración y amor, por depositar toda su confianza en mí, por brindarme su apoyo incondicional, por ser mi ejemplo y guía.

A mi hermana, para que le sirva de paradigma en la continuidad de sus estudios.

A Solainy, por su comprensión, dedicación y apoyo.

De Victor:

A mis padres y hermanos por su amor, comprensión, que siempre me apoyaron y fueron mi inspiración.

A toda mi familia por lo mucho que me quieren y toda la ayuda que me han brindado.

A mis amigos de toda la vida por estar conmigo en los momentos buenos y malos.

A mis amigos y compañeros de la universidad por estos años inolvidables que hemos compartido.

AGRADECIMIENTOS

Al claustro de profesores de este centro docente que, con su esfuerzo, nos ayudaron a desarrollar nuestros estudios.

A nuestro Tutor, Ingeniero Alieski Sarmiento Almendares, por su competencia y dedicación profesional con la que nos guió en el proceso del trabajo.

A todos, los que de una forma u otra contribuyeron al desarrollo de esta tesis, pues el trabajo científico, en general, es una obra colectiva.

RESUMEN

La Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología, tiene como objetivo principal lograr un Sistema de Gestión de la Calidad en los productos que elaboran para el desarrollo de la biotecnología en Cuba. En los laboratorios se manipulan diariamente grandes volúmenes de información que son archivadas manualmente, lo que dificulta la gestión de los datos y provoca pérdidas y duplicación de los mismos. Para contribuir a la gestión y control de la información que se genera, se desarrolla un Sistema para la Gestión de Información de los Laboratorios para el área de Control de la Calidad. El presente trabajo describe el desarrollo de la Base de Datos para el Módulo de Microbiología como parte del perfeccionamiento del Sistema para la Gestión de la Información de los Laboratorios.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.	5
1.1 Proceso de gestión de la información en el laboratorio de Microbiología.	5
1.2 Bases de Datos.	5
1.2.1 Bases de Datos y sus componentes principales.	7
1.2.2 Modelos de Bases de Datos.	7
1.2.3 Sistemas de Gestión de la Información de Laboratorios.	10
1.3 Sistemas Gestores de Bases de Datos.	12
1.3.1 PostgreSQL.	13
1.4 Mapeo de Objetos a Bases de Datos.	14
1.5 Metodología de desarrollo de software: Proceso Unificado de Desarrollo de Software (RUP).	16
1.6 Rol diseñador de Base de Datos.	17
1.7 Actividades del rol diseñador de Base de Datos.	17
1.8 Artefactos del rol diseñador de Base de Datos.	18
1.9 Herramientas CASE.	18
1.9.1 Visual Paradigm para UML.	19
1.10 Herramientas de desarrollo.	20
1.10.1 EMS SQL Manager for PostgreSQL.	20
1.10.2 EMS Data Generator 2005 for PostgreSQL.	21
1.11 Conclusiones.	22
CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.	23

2.1	Estrategia de integración de la solución con otros módulos o sistemas.....	23
2.2	Descripción de la arquitectura y fundamentación.	24
2.3	Modelo de objetos del negocio.....	26
2.4	Requisitos Funcionales y No Funcionales.	27
2.5	Diagrama de clases del diseño.....	32
2.6	Identificación de las clases persistentes.....	32
2.7	Descripción de las clases persistentes.....	36
2.8	Diseño de la BD.....	55
2.9	Descripción de las tablas.....	58
2.10	Conclusiones.....	78
CAPÍTULO 3: VALIDACIÓN DEL DISEÑO REALIZADO.....		79
3.1	Validación teórica del diseño.....	79
3.1.1	Integridad.....	79
3.1.2	Normalización de la Base de Datos.....	81
3.1.3	Análisis de la seguridad de la Base de Datos.....	82
3.1.4	Trazabilidad de las acciones.....	84
3.1.5	Disparadores.....	84
3.2	Validación funcional.....	87
3.2.1	Generación de datos para el llenado voluminoso de la BD.....	87
3.2.2	Análisis de optimización de consultas.....	90
3.3	Conclusiones.....	92
CONCLUSIONES.....		93
RECOMENDACIONES.....		94

REFERENCIAS BIBLIOGRÁFICAS	95
BIBLIOGRAFÍA	96
ANEXOS	98
Anexo 1. Organigrama del Departamento de Calidad del CIGB.	98
Anexo 2. Artefacto del rol diseñador de BD.	99
Anexo 3. Diagrama de clases del diseño del CUS Realizar Chequeo de Viabilidad. ...	100
Anexo 4. Diagrama de Clases Persistentes.	101
Anexo 5. Modelo Físico de Datos.	107
GLOSARIO DE TÉRMINOS	112

ÍNDICE DE FIGURAS

Figura 2.1 Integración de la solución con otros módulos.....	23
Figura 2.2 Diagrama de despliegue.	25
Figura 2.3 Diagrama de ORM implementada por Symfony.....	26
Figura 2.4 Realizar Ensayo Viabilidad	27
Figura 2.5 Diagrama de Clases Persistentes. (Parte 1).....	33
Figura 2.6 Diagrama de Clases Persistentes. (Parte 2).....	34
Figura 2.7 Diagrama de Clases Persistentes. (Parte 3).....	35
Figura 2.8 Diagrama de Clases Persistentes. (Parte 4).....	36
Figura 2.9 Modelo Físico de Datos. (Parte 1)	55
Figura 2.10 Modelo Físico de Datos. (Parte 2).....	56
Figura 2.11 Modelo Físico de Datos. (Parte 3).....	56
Figura 2.12 Modelo Físico de Datos. (Parte 4).....	57
Figura 2.13 Modelo Físico de Datos. (Parte 5).....	58

INTRODUCCIÓN

“A partir de la década de los años 80 se produjo en Cuba una explosión en la actividad científica en el campo de la salud, llevándose a cabo un intenso proceso inversionista y de formación de personal.” (1)

“En 1992 surgió el Polo Científico en el oeste de La Habana, que permite disponer en la actualidad de un complejo e integrado sistema de investigación-producción en varias esferas, entre las que se destaca la Biotecnología aplicada a diferentes ramas de la sociedad.” (1)

“El Polo Científico cuenta, entre otros centros con el **Centro de Ingeniería Genética y Biotecnología (CIGB)**, insertado dentro de un proceso de desarrollo de instituciones de biotecnología en las 14 provincias cubanas y el municipio especial de Isla de la Juventud.” (1)

“El Centro de Ingeniería Genética y Biotecnología de Cuba, es una institución de desarrollo dinámico que le ha permitido alcanzar un alto nivel en la investigación, desarrollo, producción y comercialización de productos biológicos obtenidos a través de los métodos de la biotecnología moderna.” (2)

La Dirección de Calidad del CIGB, creada a fines del año 1990 tiene como su principal recurso, un colectivo de trabajadores dedicado a un conjunto de objetivos y tareas en función de lograr un Sistema de Gestión de la Calidad eficiente en el Centro. Las funciones específicas que desarrolla se distribuyen en los departamentos de Control de la Calidad y Aseguramiento de la Calidad. (Ver Anexo 1. Organigrama del Departamento de Calidad del CIGB)

“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. Además debe comprobar y poner en práctica todos los procedimientos de control, evaluar, mantener y almacenar los materiales de referencia, asegurar que se controle la estabilidad de los ingredientes farmacéuticos activos y los productos terminados. El Departamento de Control de la Calidad es el responsable de autorizar o rechazar las materias primas y los productos intermedios.” (3) Para acometer estas funciones, está compuesto por dos Secciones y dos Grupos de trabajo:

- Grupo de Recepción de Muestras y Manipulación de Expedientes.
- Grupo de Desarrollo.
- Sección Biológica.
- Sección Físico-Química.

Sección Biológica: “En esta Sección es donde se ejecutan las técnicas analíticas relacionadas con la determinación de la actividad biológica in vivo e in vitro, cuantificación e identificación de proteínas contaminantes y proteínas de interés por métodos inmunológicos tipo ELISA, Western blot, Inmunodot, detección de ADN contaminante del hospedero por métodos de hibridización, reacciones en cadena de la polimerasa (PCR), almacenamiento y control de los Bancos de Células Primarios.” (3)

Para ejecutar sus funciones esta Sección posee cinco grupos de trabajo:

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

Laboratorio de Microbiología: Este grupo es el encargado de realizar el control microbiológico de todas las producciones del Centro, además brinda servicio a las áreas de investigación y desarrollo de la producción. Dentro de las principales tareas realizadas se cita el monitoreo ambiental, aspecto que incluye: control microbiológico del aire, superficies, personal, agua, aire y vapor.

Es el responsable del Programa de Limpieza e Higienización de las áreas de producción, y asesora al personal de producción en la selección, utilización y evaluación de los diferentes higienizantes. El control microbiológico de los procesos de fermentación y purificación de los principios activos así como de los bancos de células utilizados para los procesos fermentativos, son aspectos esenciales ya que los mismos permiten asegurar la calidad del producto resultante, tarea que es realizada por los técnicos de este laboratorio.

En cuanto a las técnicas de control de productos terminados, se realiza el ensayo de esterilidad y límite microbiano según la especificación de cada producto. Todos los medios de cultivos utilizados para los controles son analizados por esterilidad y promoción del crecimiento lo que garantiza la confiabilidad de los resultados emitidos además de contar con ensayos de límite microbiano y esterilidad validados.

Los trabajadores del Laboratorio de Microbiología (LM) son responsables del control microbiológico de todas las producciones del Centro, brindan servicios a las áreas de investigación y desarrollo de la producción, por lo que manejan una gran cantidad de información además del conjunto de datos de oficina y elaboración de reportes ocasionales o de rutinas que se realizan diariamente, los cuales son almacenados y que, aparejado a una ineficiente forma de búsqueda en el gran cumulo de libros a consultar, genera dificultad en la búsqueda y constante perdida de tiempo.

Toda la información es almacenada en documentos que son archivados de forma manual, en muchos casos se dificulta la gestión de la información que se recopila y se producen duplicaciones. Al considerar esta situación se realiza el análisis correspondiente para evaluar la aplicación de una base de datos general que recoja toda la información diaria que se genera en el laboratorio de Microbiología.

Se identifica como **problema científico** ¿Cómo mejorar el proceso de almacenamiento de la información en el Laboratorio de Microbiología de la Dirección de Calidad del CIGB?

Este **problema** se enmarca en el **objeto de estudio**: El proceso de gestión de la información en el Laboratorio de Microbiología de la Dirección de Calidad del CIGB.

El objeto delimita el **campo de acción**: El proceso de almacenamiento de la información en el Laboratorio de Microbiología de la Dirección de Calidad del CIGB.

Para la solución del **problema** se plantea como **objetivo**: Desarrollar una base de datos para el Módulo de Microbiología del Sistema de Gestión de la Información de los Laboratorios de la Dirección de Calidad del CIGB.

A partir del análisis del objetivo general se derivaron los siguientes **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 alcanzar el **objetivo**, se llevarán a cabo las siguientes **tareas**:

- Análisis de los procesos y la documentación generada en el Grupo de Microbiología de la Dirección de Calidad del CIGB.
- Actualización sobre las últimas tendencias de la tecnología, métodos y herramientas en el diseño, implementación y validación de bases de datos.
- Estudio de los artefactos y actividades asociadas al rol de Diseñador de base de datos propuestos por el Proceso Unificado de Desarrollo.
- Estudio sobre la generación de la capa de mapeo Objeto-Relacional en Symfony.
- Diseño del diagrama de clases persistentes.
- Diseño del modelo físico de datos.
- Análisis de la integridad de la base de datos.
- Normalización de la base de datos.
- Implementación de disparadores y procedimientos almacenados de la base de datos.

- Análisis de la seguridad de la base de datos.
- Análisis de trazabilidad de las acciones.
- Búsqueda de herramientas para pruebas de carga intensiva.
- Análisis de optimización de consultas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

En el capítulo 1, se brinda información sobre las tendencias y tecnologías actuales relacionadas con el objeto de estudio, así como una descripción del sistema gestor de base de datos utilizado y de la herramienta CASE. Se realiza una valoración del Proceso Unificado de Desarrollo, se profundiza en las actividades y artefactos realizados por el rol diseñador de base de datos.

1.1 Proceso de gestión de la información en el laboratorio de Microbiología.

En el LM se reciben muestras de los diferentes productos que se producen en el CIGB y fuera del mismo, pueden venir de diferentes laboratorios y del Grupo de Liberación Analítica. En el LM se realizan diferentes tipos de ensayos a las muestras y se obtiene un control microbiológico de todos los procesos de producción, se controla la pureza y viabilidad de los bancos de células, la posible contaminación en la etapa fermentativa, purificación y control de los IFA. Lleva a cabo el monitoreo microbiológico ambiental, control microbiológico de las muestras procedentes de los sistemas auxiliares. Se procesan además muestras de productos finales estériles y no estériles. Todos los medios de cultivo utilizados son controlados para promoción del crecimiento y esterilidad.

En el LM, para llevar a cabo sus funciones, existen actualmente varios flujos de trabajo. Un grupo de personas se encargan de realizar todas las actividades necesarias de cada uno de los procesos. La realización de uno de estos procesos resulta muy tediosa, ya que todos los documentos a consultar, crear y actualizar se encuentran en planillas, libros y en informes. Toda la información de estos documentos a gestionar, se realiza con métodos manuales, que provoca una gran pérdida de tiempo a los trabajadores del laboratorio.

1.2 Bases de Datos.

Antes del surgimiento de la informática, los datos se almacenaban en archivos, planillas y libros. Con el desarrollo de las tecnologías informáticas y la necesidad de almacenar grandes cantidades de información surgen las Bases de Datos (BD) en los años 60, las cuales ofrecen un amplio rango de soluciones al problema de almacenar datos.

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas, es decir una estructura reconocible desde un programa informático, diseñado para satisfacer los requerimientos de información de una empresa u organización.

Las BD y su tecnología tienen un gran impacto en el desarrollo de la informática, desempeñan un papel decisivo en casi todas las áreas de aplicación de las computadoras, como los negocios, la ingeniería, la medicina y la educación.

Ventajas de los Sistemas de Bases de Datos.

- **Control de la redundancia:** En los SBD los ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos.
- **Consistencia de datos:** Al eliminar o controlar la redundancia de datos se reduce en gran medida el riesgo de que hayan varias copias del mismo dato. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantengan consistentes.
- **Más información sobre la misma cantidad de datos:** Al estar todos los datos integrados, se puede extraer información adicional sobre los mismos.
- **Compartición de datos:** En un sistema de BD la información puede ser compartida para todos los usuarios que estén autorizados para la gestión de los datos.
- **Mejora en la integridad de los datos:** Refiriéndose a la validez y la consistencia de los datos almacenados, la cual se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se encarga de mantenerlas.
- **Mejora en la seguridad:** La seguridad de la BD es la protección frente a usuarios no autorizados, los SGBD permiten mantener la seguridad mediante el establecimiento de claves para identificar al personal autorizado a la base de datos.
- **Mejora en la accesibilidad a los datos:** Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos. Los SGBD disponen de operaciones de importación y exportación de datos con otras aplicaciones.
- **Mejora en el mantenimiento gracias a la independencia de datos:** Los SGBD separan las descripciones de los datos de las aplicaciones. Esto se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

- **Aumento de la concurrencia:** La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas, ni se pierdan información.
- **Mejora en los servicios de copias de seguridad y de recuperación ante fallos:** Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos. En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

1.2.1 Bases de Datos y sus componentes principales.

Al conjunto formado por la base de datos y el software se le llama sistema de base de datos que está compuesto por los siguientes elementos:

- **Hardware:** Conjunto de dispositivos físicos en el que se almacenan las bases de datos, en el que se incorporan unidades de almacenamiento masivo para este fin.
- **Software:** Es el sistema gestor de base de datos y el encargado de almacenar los datos.
- **Datos:** Incluyen los datos que se necesitan almacenar y los metadatos que son datos que sirven para describir lo que se almacena en la base de datos.
- **Usuarios:** Personas que manipulan los datos del sistema, existen tres categorías:
 - **Usuarios finales:** Personas que utilizan datos de la base de datos para su trabajo cotidiano. No utilizan la base de datos directamente, utilizan aplicaciones con el propósito de facilitarles la manipulación de los datos.
 - **Desarrolladores:** Analistas y programadores encargados de generar aplicaciones para los usuarios finales.
 - **Administradores:** Son los encargados de gestionar las bases de datos.

1.2.2 Modelos de Bases de Datos.

“El conjunto de componentes o herramientas conceptuales que un SGBD proporciona para modelar recibe el nombre de modelo de BD.” (4)

Los modelos de BD proporcionan tres tipos de herramientas:

- Colección de estructuras de datos con las que se puede construir la BD: tablas, registros, árboles, etc.
- Diferentes tipos de restricciones (o reglas) de integridad que el SGBD tendrá que hacer cumplir a los datos: dominios, claves, etc.
- Una serie de operaciones para trabajar con los datos. Un ejemplo de ello, en el modelo relacional, es la operación SELECT, que sirve para seleccionar las tuplas que cumplen alguna condición. Un ejemplo de operación típica del modelo jerárquico y del modelo en red podría ser la que nos dice si un determinado registro tiene “hijos” o no.

Existen varios modelos de BD, destacándose el modelo jerárquico, red, relacional y orientado a objeto.

Bases de datos Jerárquica:

Son bases de datos que almacenan su información en una estructura jerárquica en forma de árboles. Utiliza dos conceptos principales de estructuración de datos: registros y vínculos padre-hijo (VPH). Un registro es una colección de valores de campos que proporcionan información sobre una entidad. Un tipo de registro corresponde a un nodo del árbol, y un tipo de VPH corresponde a una arista del árbol. Un tipo de vínculo padre-hijo es un vínculo 1: N, entre los tipos de registros. El tipo de registro del lado 1 se denomina tipo de registro padre, y el del lado N se llama tipo de registro hijo del tipo de VPH, un tipo de registro padre puede tener varios hijos y el tipo de registro que no tiene padre es llamado raíz, y a los tipos de registro que no tienen hijos se les conoce como hojas.

Las bases de datos jerárquicas son útiles en aplicaciones que manejan información, permiten crear estructuras estables. Las principales limitaciones de este modelo son la duplicidad de registros, integridad referencial y su incapacidad de representar eficientemente la redundancia de datos.

Bases de datos Red

Este modelo es ligeramente distinto del jerárquico; su diferencia fundamental es que un nodo no está limitado a ser hijo de un solo nodo, se permite que un mismo nodo tenga varios padres. Es una gran mejora con respecto al modelo jerárquico, ya que ofrece una solución eficiente al problema de redundancia de datos.

Bases de datos Relacional

El modelo de datos relacional surge en 1970 por Edgar Frank Codd, produciéndose un paso fundamental para la evolución de las Bases de Datos. Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. El principal objetivo del modelo de

datos relacional es facilitar que la base de datos sea percibida o vista por el usuario como una estructura lógica que consiste en un conjunto de relaciones y no como una estructura física de implementación, por lo que se logra un alto grado de independencia de los datos.

Las bases de datos relacionales están estructuradas por una o más tablas relacionadas que contienen la información ordenada de una forma organizada. Generalmente contendrán muchas tablas y una tabla sólo contiene un número fijo de campos. Los nombres de los campos de una tabla son distintos y el orden no está determinado y para cada campo existe un conjunto de valores posible.

El modelo relacional es un modelo de datos y, como tal, tiene en cuenta tres aspectos siguientes de los datos:

- La **estructura**, que debe permitir representar la información que nos interesa del mundo real.
- La **manipulación**, a la que da apoyo mediante las operaciones de actualización y consultas de los datos.
- La **integridad**, que es facilitada mediante el establecimiento de reglas de integridad; es decir, condiciones que los datos deben cumplir.

Bases de datos Orientadas a Objetos

Este modelo de datos es uno de los más reciente y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento). Una base de datos orientada a objetos permite crear objetos que existan permanentemente, o persistan.

Un objetivo de las bases de datos orientadas a objetos es mantener una correspondencia directa entre los objetos del mundo real y de la base de datos, de modo que dichos objetos no pierdan su integridad ni su identidad y se puedan identificar y manipular fácilmente.

Una base de datos orientada a objetos incorpora todos los conceptos importantes del paradigma de objetos:

Encapsulamiento: Propiedad que permite ocultar la información al resto de los objetos, lo que impide el accesos incorrectos o conflictos.

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.

El modelo de base de datos seleccionado fue el modelo de datos relacional, por las características que presenta: evita la duplicidad de registros, garantiza la integridad referencial, la independencia física de datos, requiere de menos espacio de almacenamiento de datos y minimiza la redundancia de datos. Además, de ser el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos.

1.2.3 Sistemas de Gestión de la Información de Laboratorios.

“Un LIMS o "Laboratory Information Management System" es un programa de gestión de laboratorios que permite recoger, almacenar, calcular y gestionar datos en una amplia variedad de formas. Los LIMS representan una importante herramienta para la gestión global de un laboratorio en un entorno de calidad, se agilizan temas de registro de datos primarios, archivo, trazabilidad, etc. y minimizan los errores debidos a la transferencia de información.” (5)

Las ventajas de los LIMS:

- **Reproducibilidad** en cualquier momento y de forma sencilla de toda la documentación generada con anterioridad.
- **Control** de cualquier cambio en todas aquellas zonas críticas de la información del laboratorio, para poder impedir modificaciones no autorizadas y además guardar una trazabilidad de los mismos.
- **Revisión y visualización** de datos más flexible y accesible.
- **Generación** más rápida y efectiva de informes.
- **Centralización de la información** en una única base de datos, lo que conlleva que la accesibilidad a los datos sea más segura, rápida, cómoda, sencilla.
- **Mantenimiento de estados** ligados a proyectos, muestras, pruebas, resultados, de manera automática por el Sistema.
- **Reducción de costos**, tanto en el laboratorio (porque simplifica la gestión del mismo) como en la planta de producción de aquellas empresas que la posean (porque permite la detección de partidas de productos en mal estado antes del empleo de los mismos en el proceso de fabricación).

Muchos laboratorios requieren el uso de herramientas informáticas para recopilar datos y analizar los resultados. Los laboratorios tienen que establecer y aplicar procedimientos para la protección de datos que garanticen su integridad, confidencialidad, almacenamiento, transmisión y proceso, características que los relacionan con el importante papel que juegan las bases de datos.

Bases de Datos en LIMS existentes en el mundo:

Debido al alto costo de los LIMS existentes en la actualidad y la necesidad de una herramienta para la gestión de la información de los laboratorios, la Dirección de Calidad del CIGB junto a la Dirección de la UCI decidieron desarrollar un LIMS para facilitar el trabajo y minimizar los posibles errores humanos en los resultados de las pruebas o análisis que se realizan en los laboratorios, producto al gran volumen de información que se manipulan diariamente y la necesidad de almacenarla en una BD. Actualmente existen en el mundo LIMS que cuentan con una base de datos, entre ellos se encuentran:

La corporación STARLIMS aporta soluciones LIMS en todas las organizaciones dentro de la salud pública, la industria farmacéutica, petroquímica, forense, alimenticia y bebida, ambiental y química.

El Sistema de Administración de Información en Laboratorios (StarLIMS) es una herramienta de alta tecnología de gestión de laboratorios innovadora, flexible y fácil de usar, diseñado para una amplia variedad de laboratorios que opera en muchas disciplinas científicas e industriales. Herramienta muy poderosa para gestionar procesos complejos, asegurar conformidad reguladora y promover la colaboración entre el laboratorio y la empresa. Con una documentación completa, legible y disponible, las organizaciones pueden asegurar la integridad y validez de los resultados y confiar completamente en el proceso de análisis.

STARLIMS permite utilizar una única base de datos para que operen los múltiples laboratorios funcionales en diferentes localizaciones geográficas y el acceso a los datos es definido para cada usuario. Además de su arquitectura multi-capa que separa completamente los componentes bases de datos, reglas del negocio y tecnología.

Se puede mencionar otro de los LIMS más utilizados actualmente y es el caso del **LabWare LIMS** que es un sistema de información de laboratorio con acceso semejante cliente/servidor y Web, con una amplia variedad de módulos que fácilmente se pueden incorporar a la aplicación para cubrir las necesidades de un segmento del mercado y permite que el software se integre dentro de cualquier entorno informático empresarial. Presenta una completa funcionalidad para seguimiento de muestras, certificación de usuarios, gestión de instrumentos, auditoría, y planificación de muestras e informes, así como muchas otras funciones. Como base de datos, se pueden usar todos los productos comerciales, desde ORACLE o SQL Server hasta DB2.

1.3 Sistemas Gestores de Bases de Datos.

Un sistema de gestión de la base de datos es denominado como: “Conjunto de elementos software con capacidad para definir, mantener y utilizar una base de datos.” (6)

Los SGBD están constituidos por un paquete de software cuya función es la gestión del acceso a la BD, las operaciones fundamentales son: crear, modificar, eliminar y obtener la estructura asociada al esquema lógico de una BD. El objetivo principal es proporcionar un entorno práctico y eficiente a la hora de almacenar y recuperar la información de la base de datos, lo que facilita la gestión de grandes volúmenes de datos.

Características de los Sistemas de Gestión de Base de Datos:

- **Control de la redundancia:** Los SGBD deberán ser capaz de controlar la redundancia de datos para que no haya inconsistencias entre los datos.
- **Restricción de los accesos no autorizados:** Los SGBD deben contar con un subsistema de seguridad y autorización que permitan al administrador de base de datos crear cuentas y especificar restricciones para las bases de datos; lo que obliga, automáticamente, al cumplimiento de dichas restricciones.
- **Cumplimiento de las restricciones de integridad:** La mayor parte de los SGBD tienen ciertas restricciones de integridad que deben cumplir los datos, estos deben definir recursos para definir tales restricciones y hacer que se cumplan.
- **Respaldo y recuperación:** Todos los SGBD deben contar con recursos para recuperarse de fallos de hardware o de software, de esto se encarga el subsistema de respaldo y recuperación del SGBD.

Desventajas del sistema de gestión de base de datos:

- **Complejidad:** Los SGBD son un conjunto de programas muy complejos con una gran funcionalidad.
- **Vulnerable a los fallos:** El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse.
- **Costo del equipamiento adicional:** Tanto los SGBD como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento y se necesite una máquina

más grande o una máquina que se dedique solamente al SGBD, esto provoca que un sistema de base de datos sea más caro.

- **Costo económico del SGBD:** Varía en dependencia del entorno y de la funcionalidad que ofrece.
- **Tamaño del SGBD:** Son programas complejos y muy extensos que requieren de gran cantidad de espacio en disco y de memoria para trabajar de forma eficiente.

1.3.1 PostgreSQL.

PostgreSQL es un potente Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) de código abierto. Tiene más de 15 años de activo proceso de desarrollo a nivel mundial y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad, integridad de los datos, y la corrección. Es multiplataforma y funciona en los principales sistemas operativos, como Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. También apoya el almacenamiento de grandes objetos binarios, imágenes, sonidos o vídeo.

Características de PostgreSQL:

- Objeto-relacionales: PostgreSQL en cada tabla define una clase que permite implementar la herencia entre tablas o clases, funciones y operadores polimórficos.
- Implementación del estándar SQL92/SQL99.
- Código abierto: Debido a la licencia liberal, PostgreSQL puede ser usado, modificado y distribuido por todo el mundo de forma gratuita para cualquier fin, ya sea de datos, comerciales o académicas.
- Múltiples lenguajes de procedimientos: Los disparadores y otros procedimientos pueden ser escritos en varios lenguajes de procedimientos. Código del lado del servidor es comúnmente escrito en PL / PostgreSQL, un lenguaje de procedimiento similar al de Oracle PL / SQL. También se puede desarrollar código del lado del servidor en Tcl, Perl, incluso bash (el de código abierto Linux / Unix shell).
- Múltiples-cliente API: PostgreSQL soporta el desarrollo de aplicaciones cliente en varios lenguajes, interfaz para PostgreSQL desde C, C + +, ODBC, Perl, PHP, Tcl / Tk, y Python.
- Tipos de datos: integer, string, numeric, boolean, char, varchar, date, interval, y timestamp, tipos geométrica, tipo de datos booleanos y tipos de datos diseñados específicamente para hacer frente a las direcciones de red.

- La extensibilidad: Es una de las características más importantes de PostgreSQL ya que puede ser ampliado, se pueden añadir nuevos tipos de datos, nuevas funciones y operadores, e incluso nuevos lenguajes de procedimiento y de cliente.
- Tiene un soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados.
- MVCC, o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios, es decir, mientras un proceso escribe no se bloquean las tablas, ni siquiera las tuplas. Mediante el uso de MVCC, PostgreSQL evita el problema de que procesos lectores esperen a que se termine de escribir por completo. MVCC está considerado mejor que el bloqueo a nivel de tupla porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.
- Integridad Referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

Luego de un estudio y comparación entre diferentes SGBD realizado anteriormente por los analistas del proyecto, se decidió utilizar una herramienta de código abierto y multiplataforma para el desarrollo de la BD del LM. Se seleccionó como SGBD PostgreSQL 8.2, ya que es uno de los más utilizados en la actualidad por su alto nivel de estabilidad, confiabilidad y escalabilidad.

1.4 Mapeo de Objetos a Bases de Datos.

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, al forzar al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (7)

Symfony es un framework para desarrollar aplicaciones web, está desarrollado con PHP 5 y es compatible con muchos SGBD tales como, PostgreSQL, MySQL y Oracle. Proporciona herramientas que simplifican el desarrollo de las aplicaciones web complejas, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web.

Symfony está basado en el patrón de diseño web conocido como arquitectura MVC (Modelo Vista Controlador), formado por tres niveles. El modelo representa la información con la que trabaja la aplicación, la 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), de obtiene así un mantenimiento más sencillo de las aplicaciones. En las aplicaciones Symfony, el acceso y la modificación de los datos almacenados en la base de datos se realiza mediante objetos; de esta forma nunca se accede de forma explícita a la base de datos.

La capa del modelo se puede dividir en la capa de acceso a los datos y en la capa de abstracción de la base de datos. Las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar las consultas.

PHP 5 y Symfony son orientados a objetos. La base de datos para el LM tiene una estructura relacional y para acceder a esta como si fuera orientada a objetos, es necesaria una interfaz que traduzca la lógica de los objetos a la lógica relacional. Esta interfaz se denomina “mapeo de objetos a bases de datos” (ORM, de sus siglas en inglés “object-relational mapping”). Un ORM consiste en una serie de objetos que permiten acceder a los datos y que contienen en su interior cierta lógica de negocio. La principal ventaja que aporta el ORM es la reutilización, lo que permite llamar a los métodos de un objeto de datos desde varias partes de la aplicación e incluso desde diferentes aplicaciones.

Una de las ventajas de utilizar estas capas de abstracción de objetos/relacional es que evita utilizar una sintaxis específica de un sistema de bases de datos concreto. Esta capa transforma automáticamente las llamadas a los objetos en consultas SQL optimizadas para el sistema gestor de bases de datos que se utiliza en cada momento.

Symfony utiliza Propel como ORM y Propel utiliza Creole como capa de abstracción de bases de datos y están completamente integrados en Symfony, utiliza una sintaxis específica para las consultas y a cambio optimiza y adapta el lenguaje SQL a la base de datos concreta que se utiliza, lo que evita reescribir parte de las consultas en caso de que sea necesario cambiar de gestor. Mediante Propel se puede crear, obtener, modificar o borrar datos de la base de datos sin utilizar instrucciones SQL.

1.5 Metodología de desarrollo de software: Proceso Unificado de Desarrollo de Software (RUP).

“El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software.” (8) El Proceso Unificado de Desarrollo es un proceso bien definido, estructurado y adaptable a las características y necesidades de cada proyecto específico.

“El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual para especificar, visualizar, construir y documentar artefactos de un sistema de software.” (9)

La metodología de desarrollo seleccionada por el equipo del proyecto fue Rational Unified Process (RUP). Está formado por cuatro fases y nueve flujos de trabajos, basándose en el Lenguaje Unificado de Modelado (UML) como herramienta principal.

RUP fue creado por Ivar Jacobson, Grady Booch y James Rumbaugh en 1998, su objetivo es producir software de alta calidad, que cumpla con los requerimientos de los usuarios dentro de la planificación y presupuesto establecidos. RUP es un proceso de desarrollo de software y junto a UML constituye la metodología estándar más utilizada para el desarrollo de un producto de software. El ciclo de vida de RUP se caracteriza por:

Dirigido por casos de uso: el proceso de desarrollo sigue un hilo, avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso son un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

Iterativo e incremental: en donde el trabajo se divide en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto.

RUP define cuatro elementos: los **roles**, que responden a la pregunta ¿Quién?, las **actividades** que responden a la pregunta ¿Cómo?, los **artefactos**, que responden a la pregunta ¿Qué? y los **flujos de trabajo** que responde a la pregunta ¿Cuándo?

Un **rol** define las responsabilidades de un individuo, o de un grupo de individuos de un equipo de trabajo. Una persona puede desempeñar diversos roles y un rol puede ser representado por varias

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

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

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

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

1.6 Rol diseñador de Base de Datos.

El diseñador de base de datos es responsable de diseñar el almacenamiento persistente de datos para ser usado por el sistema, define los detalles del diseño de la base de datos que incluye: tablas, índices, vistas, restricciones, disparadores, procedimientos almacenados y otros objetos específicos de la base de datos necesarios para almacenar, recuperar y eliminar objetos persistentes. Es responsable de la integridad del modelo de datos y asegura que el modelo de datos como un todo sea correcto, coherente, y comprensible. Debe tener conocimientos sólidos de modelados de datos, diseño de base de datos, técnicas de análisis y diseño orientado a objetos, arquitectura de sistemas, administración de base de datos, comprensión del entorno y lenguaje de implementación PHP 5.0.

1.7 Actividades del rol diseñador de Base de Datos.

Pasos para el Diseño de la Base de datos:

Desarrollar el modelo lógico de la BD (opcional)

Desarrollar el diseño físico de la BD.

- Definir Dominios.
- Crear los elementos iniciales del diseño físico de la BD.
- Definir las tablas de referencia.
- Crear llave 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.

Desarrollar el modelo lógico de la BD: Definir un modelo lógico del diseño de la BD, tiene como propósito proveer una vista lógica de los datos llaves de las entidades y sus relaciones que son independientes de cualquier software específico o implementación de la base de datos.

Desarrollar el diseño físico de la BD: Definir el diseño físico detallado de la base de datos incluye elementos del modelo (tablas, vistas y procedimientos almacenados) que representan el diseño subyacente de almacenamiento de datos. Colectivamente, estos elementos del modelo comprenden el Modelo Físico de Datos de la base de datos y está contenido en el artefacto Modelo de Datos.

Revisar los resultados: El propósito es asegurar la calidad e integridad del Modelo de Datos. El diseñador de la base de datos regularmente debe revisar la estructura implementada para asegurar que el Modelo de Datos es consistente con cualquier cambio que se ha hecho directamente en la base de datos. Los defectos identificados que no son corregidos en el mismo tiempo que han sido detectados deben estar documentados en Change Requests y eventualmente se debe asignar a alguien para la resolución.

1.8 Artefactos del rol diseñador de Base de Datos.

Modelo de Datos: Es el artefacto resultante de la actividad de diseño de base de datos y describe las representaciones lógicas y físicas de los datos persistentes utilizados por la aplicación. Este puede ser creado inicialmente mediante ingeniería inversa a partir de una BD existente o a partir de un conjunto de clases del diseño persistentes en el modelo de diseño. (Ver Anexo 2. Artefacto del rol diseñador de BD)

Los modelos de datos son específicamente necesarios donde la estructura de datos persistente no puede ser automáticamente y mecánicamente derivada de la estructura de clases persistentes en el modelo del diseño. Se utiliza para definir el mapeo entre las clases de diseños persistentes y estructuras de datos persistentes, y para definir la persistencia de estructuras de datos.

1.9 Herramientas CASE.

Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador): Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software y su principal objetivo es mejorar la calidad del producto junto con la eficiencia de los desarrolladores de software.

Objetivos

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto.
- Aumentar la biblioteca de conocimiento informático de una empresa lo que ayuda a la búsqueda de soluciones para los requisitos.
- Ayudar a la reutilización del software, portabilidad y estandarización de la documentación.

Existen varias aplicaciones CASE: Database Designer for MySQL, Eclipse, Embarcadero, ER/Studio, Enterprise Architect, MetaCASE, Rational Rose, System Architect, Umbrello, Visible Enterprise Products, Visual Paradigm for UML.

1.9.1 Visual Paradigm para UML.

La herramienta CASE seleccionada en el proyecto para realizar el modelado fue Visual Paradigm for UML 6.0 Enterprise Edition que utiliza el Lenguaje Unificado de Modelado (UML) como lenguaje de modelado. Está diseñada para distintos usuarios entre los que se incluyen ingenieros de software, analistas de sistemas, analistas de negocios, arquitectos y desarrolladores. Está orientada a la creación de diseños y se usa el paradigma de programación orientada a objetos. Visual Paradigm (VP) incluye una herramienta llamada Visual Architect que permite la generación de código para el manejo de la base de datos. Con esta herramienta se puede generar código para los lenguajes PHP, JAVA y C# y para los gestores de base de datos DB2, Informix, SQL Server, MySQL, Oracle y PostgreSQL.

Beneficios de Visual Paradigm para UML:

- Persistencia de forma fácil.
Los desarrolladores emplean mucho esfuerzo en salvar y cargar objetos entre la memoria y la base de datos lo que hace que el programa sea complicado y difícil de mantener. VP simplifica estas tareas mediante la generación de una capa de persistencia entre objeto y modelos de datos.
- Generador de mapeo objeto-relacional sofisticado.
La capa de mapeo objeto-relacional que se genera incorpora características como soporte de transacciones, capa de conectar en caché, agrupación de conexiones y personalización de sentencias SQL.

- Amplia cobertura para bases de datos.
Soporta una amplia gama de base de datos, incluidos Oracle, DB2, Cloudscape / Derby, Sybase Adaptive Server Enterprise, Sybase SQL Anywhere, Microsoft SQL Server, PostgreSQL, MySQL y otros.
- Base de datos de ingeniería inversa.
Permite la ingeniería inversa existente en una base de datos a través JDBC en el modelo entidad-relación. Los desarrolladores pueden transformar el modelo entidad-relación al modelo de objetos y rediseñar la base de datos para un mayor desarrollo.
- Integración con IDE.
VP-UML no sólo es una aplicación independiente, se puede integrar a los principales Integrated Development Environments (IDEs): Eclipse/WebSphere®, Borland JBuilder®, NetBeans/Sun™ ONE, IntelliJ IDEA™ Oracle JDeveloper, BEA WebLogic Workshop™ y Visual Studio .NET.

1.10 Herramientas de desarrollo.

1.10.1 EMS SQL Manager for PostgreSQL.

EMS SQL manager 2005 for PostgreSQL es una potente herramienta de alto nivel de rendimiento para la administración y desarrollo de los servidores de bases de datos para PostgreSQL. Funciona con cualquiera de las versiones de PostgreSQL hasta la 8.2 y soporta las últimas características de PostgreSQL. Ofrece una amplia gama de herramientas poderosas para los usuarios experimentados como Visual Database Designer, Visual Query Builder.

Características:

- Excelente herramienta visual y de texto para la construcción de consultas.
- Interfaz gráfica de usuario bien amigable
- Rápida navegabilidad.
- Soporte completo de PostgreSQL hasta la versión 8.2.
- Eficiente gestión de seguridad.
- Impresionante capacidades de exportación e importación de los datos.
- Potente diseñador visual de bases de datos.
- Fácil uso del asistentes de PostgreSQL para la realización de las tareas de mantenimiento.
- El acceso al servidor PostgreSQL a través del protocolo HTTP.

1.10.2 EMS Data Generator 2005 for PostgreSQL.

Es un impresionante software para la generación de datos de pruebas a varias tablas para una base de datos PostgreSQL. El asistente de aplicación permite definir tablas y campos para la generación de datos, establecer rangos de valores, generar campos char por máscara, cargar valores desde archivos para campos de tipo objetos binarios (BLOB), obtener listas de valores desde consultas SQL y muchas funciones para generar los datos de pruebas de manera simple y directa.

Características:

- Amigable interfaz gráfica.
- Generación de datos para varias tablas de diferentes bases de datos en un host.
- Soporte para todos los tipos de datos PostgreSQL como array, dirección de red y geométricos.
- Generación de diferentes tipos de valores para cada campo, incluye lista, random, generación incremental de datos y más.
- Control automático sobre la integridad referencial para la generación de datos a tablas vinculadas.
- Amplia variedad de generación de parámetros para la de cada tipo de campo
- Capacidad de establecer valores NULL.

1.11 Conclusiones.

En este capítulo se introducen conceptos importantes de interés necesarios para el desarrollo de este trabajo. Se fundamenta la selección de las herramientas y tecnologías escogidas para desarrollar el diseño de una base de datos que permita el almacenamiento de datos del Módulo de Microbiología de la Dirección de Calidad del CIGB, se concluye que se va a utilizar:

- ✓ El modelo de base de datos relacional.
- ✓ Como metodología de desarrollo de software RUP.
- ✓ La herramienta CASE de modelado Visual Paradigm for UML 6.0 Enterprise Edition para la realización del diagrama de clases persistentes y para diseñar la base de datos.
- ✓ Como sistemas de gestión de base de datos PostgreSQL 8.2, se utiliza EMS SQL 2005 Manager 2005 for PostgreSQL para la administración de la BD y para las pruebas carga intensiva el EMS Data Generator 2005 for PostgreSQL.
- ✓ Para la generación de las clases de la capa de acceso a datos se determino utilizar el framework de desarrollo web Symfony 1.0.11.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

En el presente capítulo, se realiza una descripción y análisis de la solución propuesta para el trabajo de diploma; se traza una estrategia de integración de la solución con los otros módulos del proyecto, se describe la arquitectura del sistema. Se hace referencia al modelo de objetos del negocio, a los requisitos funcionales y no funcionales relacionados con las bases de datos. Se identifican las clases persistentes y se elabora el diagrama de clases persistentes y se realiza la descripción de las clases. Se describe el diseño físico de la base de datos al hacer una descripción de sus tablas.

2.1 Estrategia de integración de la solución con otros módulos o sistemas.

La Dirección de Calidad del CIGB está formada por dos Departamentos, Control de Calidad y Aseguramiento de la Calidad, estos están constituidos por varias secciones y grupos de trabajos. Con el objetivo de realizar una estrategia de integración de la solución entre todos los módulos para el almacenamiento de los datos, se les da seguimientos a varias series de actividades para la creación de la base de datos del sistema.

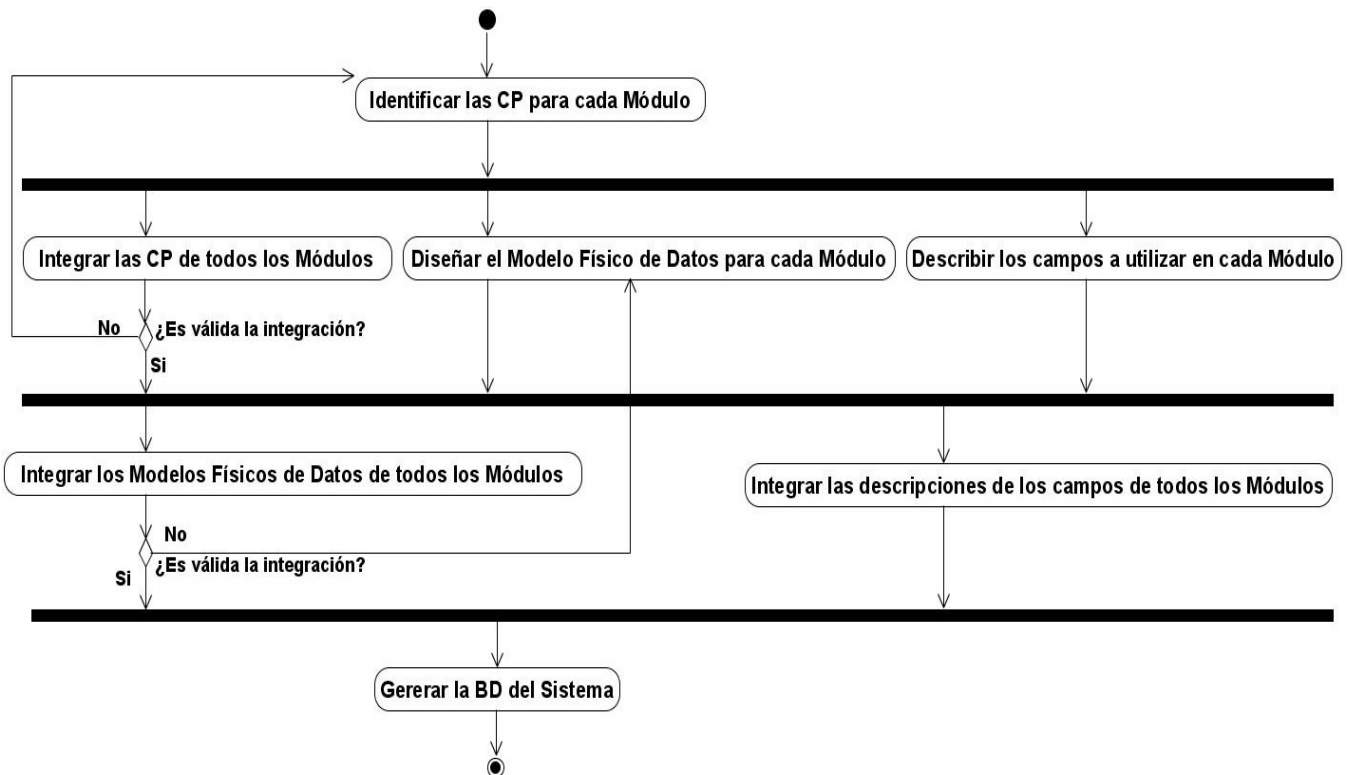


Figura 2.1 Integración de la solución con otros módulos.

Identificar las clases persistentes (CP) en cada Módulo: Identificar las clases persistentes para cada módulo a partir del diagrama de clases del diseño.

Integración de las clases persistentes de los Módulos: Integrar y relacionar las clases persistentes de todos los módulos en un diagrama de clases persistentes, validar la integración, y si no existe una correcta integración de las clases persistentes, entonces revisar reiteradamente los diagramas de clases persistentes de cada módulo hasta lograr una correcta integración.

Diseñar el Modelo Físico de Datos para cada Módulo: Realizar 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: Realizar una descripción de todos los campos que poseen las tablas para cada módulo.

Integrar los Modelos Físicos de Datos de todos los Módulos: Integrar los Modelos de Datos realizados en cada módulo en único modelo. Validar la integración, y si no existe una correcta integración de la estructura de la BD, revisar reiteradamente los modelos de cada módulo hasta lograr una correcta integración.

Integrar las descripciones de los campos de los Módulos: Conformar un diccionario de datos con las descripciones de todos los campos de las tablas de los diferentes módulos, así los desarrolladores de la aplicación y diseñadores de BD utilizan la misma nomenclatura.

2.2 Descripción de la arquitectura y fundamentación.

La base de datos propuesta para el laboratorio de microbiología contiene 55 tablas en el modelo físico, las principales tablas y más utilizadas son:

le_esterilidad, le_muestra, sic_0711_gchviabilidad, sic_0162_gchviabilidad, sic_0102_geesterilidad, sic_0198_geepantimicrobianos, sic_0199_gelmicrobiano, sic_0102_geesterilidad pues almacenan la mayor cantidad información manipulada en el laboratorio.

En el diagrama de despliegue que se muestra a continuación se modela la arquitectura en tiempo de ejecución del sistema, formado por las computadoras clientes mediante una conexión HTTPS con el servidor de aplicaciones, los dispositivos externos como impresora y scanner están conectados a las computadoras mediante el puerto USB y el servidor de base de datos con una conexión TCP/IP + SSL con el servidor de aplicaciones.

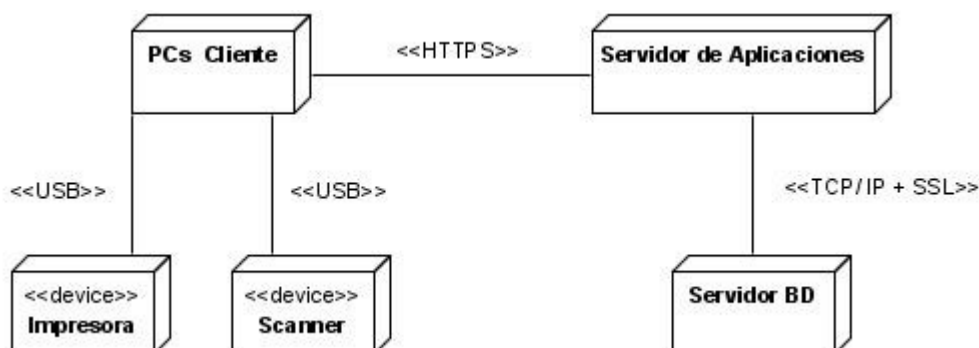


Figura 2.2 Diagrama de despliegue.

Para crear el modelo de objetos de datos que utiliza Symfony, se debe traducir el modelo relacional de la base de datos a un modelo de objetos de datos. Para realizar ese mapeo, el ORM necesita una descripción del modelo relacional, que se llama “esquema” (schema). En el esquema se definen las tablas, sus relaciones y las características de sus columnas.

Symfony puede utilizar la capa de acceso a bases de datos proporcionada por Creole para generar un archivo **schema.yml** a partir de una base de datos existente, mediante el comando *propel-build-schema* se genera un nuevo archivo `schema.yml` a partir de la estructura de la base de datos y se almacena en el directorio `config/` del proyecto, pudiéndose construir el modelo a partir de este esquema. En el archivo `databases.yml` se configura la conexión a la BD.

El esquema se utiliza para construir las clases del modelo que necesita la capa del ORM, estas clases se generan mediante una línea de comandos llamada *propel-build-model*. Las clases con nombre Base son las que se generan directamente a partir del esquema y se guardan en el directorio `lib/model/om/`. Esas clases no se modifican, porque cada vez que se genera el modelo, se borran todas.

Se utiliza la tabla `sic_0711_gchviabilidad` y al ejecutar este comando, se analiza el esquema y se generan las clases base del modelo, en el directorio `lib/model/om/`

- `Basesic_0711_gchviabilidad.php`
- `Basesic_0711_gchviabilidadPeer.php`

Se crean las clases de objetos propias que están en el directorio `lib/model` que heredan de las clases con nombre Base. Estas clases no se modifican cuando se ejecuta la tarea *propel-build-model*, por lo que son las clases en las que se añaden los métodos propios.

Las clases objeto que representan un registro de la base de datos, permiten acceder a las columnas de un registro y a los registros relacionados, mientras que las clases peer permiten obtener los registros de las tablas de la BD y tienen métodos estáticos para trabajar con las tablas. Sus métodos devuelven normalmente un objeto o una colección de objetos de la clase objeto relacionado.

De la tabla sic_0711_gchviabilidad se generó la clase objeto:

- sic_0711_gchviabilidad.php

Y la clase peer:

- sic_0711_gchviabilidadPeer.php

La combinación de las clases objeto y las clases “peer” y las versiones básicas y personalizadas de cada una hace que se generen cuatro clases por cada tabla del esquema.

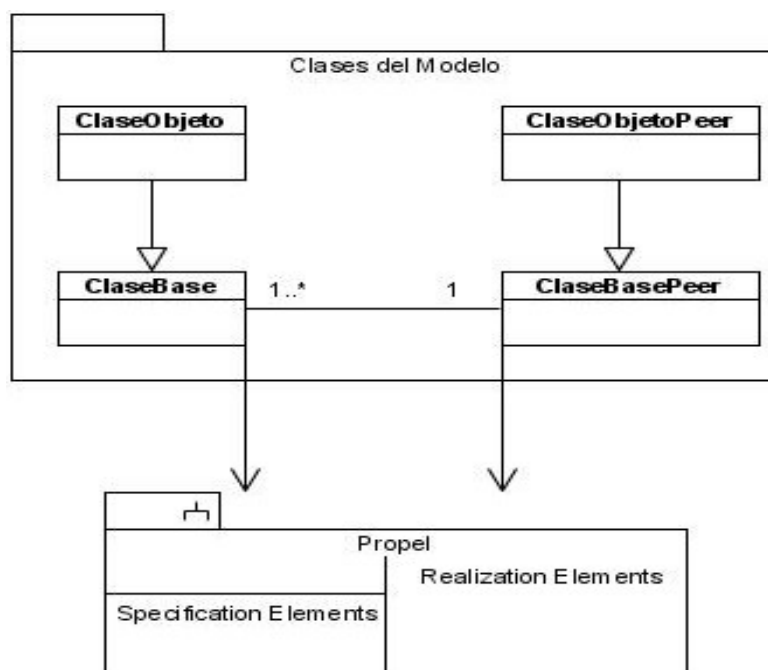


Figura 2.3 Diagrama de ORM implementada por Symfony

2.3 Modelo de objetos del negocio.

Un modelo de objetos describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso de negocio. El siguiente diagrama de objetos corresponde al caso de uso del negocio Realizar Ensayo Viabilidad. Para consultar los restantes modelos de objetos remitirse al Expediente del Proyecto.

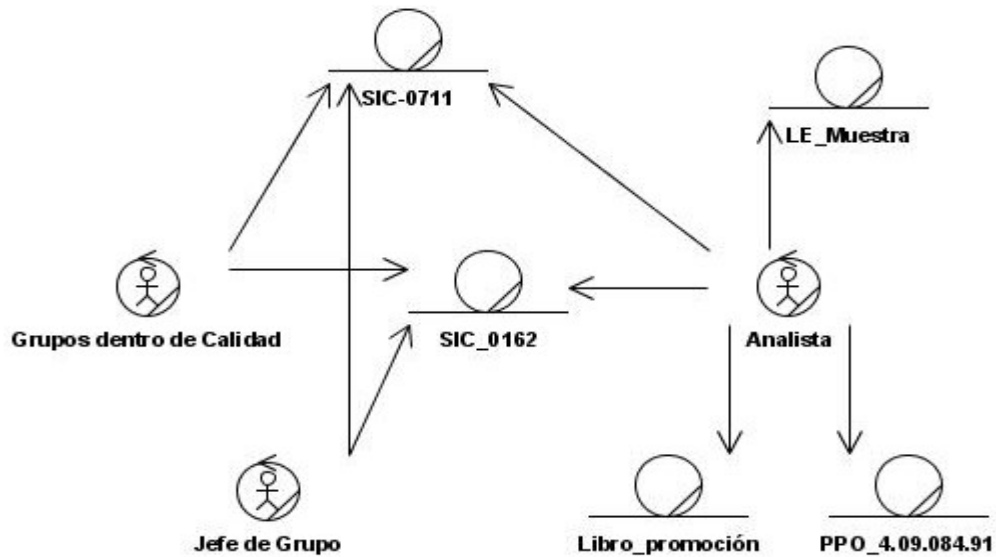


Figura 2.4 Realizar Ensayo Viabilidad

2.4 Requisitos Funcionales y No Funcionales.

Los requisitos son uno de los flujos de trabajo más importantes, porque en él se establece qué es lo que debe hacer el sistema, se establece un acuerdo entre el cliente y los desarrolladores. Los requisitos se dividen en dos grupos: funcionales y no funcionales. Los requisitos funcionales que son capacidades o condiciones que el sistema debe cumplir. Se modelan mediante diagramas de Casos de Uso. Los requisitos no funcionales son propiedades o cualidades que el producto debe tener, existen múltiples categorías para clasificarlos, tales como, requerimientos de software, hardware, apariencia o interfaz externa, seguridad, usabilidad, soporte, portabilidad, etc.

Requisitos Funcionales (10)

1. Gestionar Ensayo Control de Proceso:
 - 1.1. Crear nuevo registro.
 - 1.2. Buscar y visualizar registro.
 - 1.3. Modificar datos de un registro.
 - 1.3.1. Registrar traza.
 - 1.4. Generar reporte.
2. Gestionar Chequeo de Viabilidad:
 - 2.1. Crear nuevo registro.

- 2.2. Buscar y visualizar registro.
- 2.3. Modificar datos de un registro.
 - 2.3.1. Registrar traza.
- 2.4. Generar reporte.
- 3. Gestionar Control Microbiológico de las Aguas:
 - 3.1. Crear nuevo registro.
 - 3.2. Buscar y visualizar registro.
 - 3.3. Modificar datos de un registro.
 - 3.3.1. Registrar traza.
 - 3.4. Generar reporte.
- 4. Gestionar Control Ambiental mediante Placa Expuesta:
 - 4.1. Crear nuevo registro.
 - 4.2. Buscar y visualizar registro.
 - 4.3. Modificar datos de un registro.
 - 4.3.1. Registrar traza.
 - 4.4. Generar reporte.
- 5. Gestionar Chequeo Inicial de Viabilidad:
 - 5.1. Crear nuevo registro.
 - 5.2. Buscar y visualizar registro.
 - 5.3. Modificar datos de un registro.
 - 5.4. Generar reporte.
- 6. Gestionar Ensayo Efectividad de Preservos Antimicrobianos:
 - 6.1. Crear nuevo registro.
 - 6.2. Buscar y visualizar registro.
 - 6.3. Modificar datos de un registro.

- 6.4. Generar reporte.
- 7. Gestionar Ensayo Límite Microbiano:
 - 7.1. Crear nuevo registro.
 - 7.2. Buscar y visualizar registro.
 - 7.3. Modificar datos de un registro.
 - 7.3.1. Registrar traza.
 - 7.4. Generar reporte.
- 8. Gestionar Ensayo de Esterilidad:
 - 8.1. Crear nuevo registro.
 - 8.2. Buscar y visualizar registro.
 - 8.3. Modificar datos de un registro.
 - 8.3.1. Registrar traza.
 - 8.4. Generar reporte.
- 9. Gestionar Control Microbiológico de las Manos Enguantadas:
 - 9.1. Crear nuevo registro.
 - 9.2. Buscar y visualizar registro.
 - 9.3. Modificar datos de un registro.
 - 9.3.1. Registrar traza.
 - 9.4. Generar reporte.
- 10. Gestionar Ensayo de Preparación de Soluciones:
 - 10.1. Crear nuevo registro.
 - 10.2. Buscar y visualizar registro.
 - 10.3. Modificar datos de un registro.
 - 10.3.1. Registrar traza.
 - 10.4. Generar reporte.

11. Gestionar Control Muestreo de Superficies:

- 11.1. Crear nuevo registro.
- 11.2. Buscar y visualizar registro.
- 11.3. Modificar datos de un registro.
 - 11.3.1. Registrar traza.
- 11.4. Generar reporte.

12. Gestionar Control Microbiológico de Vapor Puro:

- 12.1. Crear nuevo registro.
- 12.2. Buscar y visualizar registro.
- 12.3. Modificar datos de un registro.
 - 12.3.1. Registrar traza.
- 12.4. Generar reporte.

13. Gestionar Control Muestreo con el Analizador Ambiental:

- 13.1. Crear nuevo registro.
- 13.2. Buscar y visualizar registro.
- 13.3. Modificar datos de un registro.
 - 13.3.1. Registrar traza.
- 13.4. Generar reporte.

Requisitos No Funcionales:

Usabilidad.

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

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.

Por su parte los clientes requerirán de un navegador capaz de interpretar secuencias de códigos JavaScript.

Portabilidad.

El sistema podrá ser usado sobre los sistemas operativos Windows y Linux.

Seguridad.

Garantizar que la información sensible solo pueda ser vista por los usuarios con el nivel de acceso adecuado. Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo. 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 registros modificados por los usuarios, para garantizar el control de las operaciones de este tipo en el sistema.

Se podrá acceder a algunas funcionalidades del sistema desde cualquier 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.

El sistema validará la captación de datos para evitar entradas inadecuadas.

Software.

Se deberá disponer para poder instalar la aplicación desarrollada, con Sistema Operativo Windows 98 o superior. Igual sistema operativo en las computadoras personales clientes de los usuarios, las cuales además accederán al sistema usando un navegador compatible o superior con Internet Explorer 5.5+, Netscape, Mozilla 1.7 o superior o FireFox 0.9.3 o superior.

En el servidor de la aplicación el software recomendado es Windows 2000 o superior.

Hardware.

Se deberá contar con impresora y escáner en las computadoras personales clientes que interactúen con la aplicación y vayan a imprimir algún documento generado por la misma.

Se deberán incluir más computadoras personales en los grupos de la dirección de Calidad.

PC servidor: Pentium IV de capacidad de disco duro superior a 80.0 GB, microprocesadores superior a 2.0 GHz, 1.0 GB mínimo de RAM.

Restricciones en el diseño y la implementación.

La lógica de presentación constituirá una capa independiente de la lógica de negocio, centrandose su función en la interfaz de usuario y validaciones simples de los datos de entrada. Debe ser una aplicación Web desarrollada con la tecnología para creación de páginas dinámicas PHP y base de datos en PostgreSQL. Se utilizarán herramientas de desarrollo que garanticen la calidad de todo el ciclo de desarrollo del producto.

Se usará el lenguaje de programación PHP 5.

2.5 Diagrama de clases del diseño.

Una clase de diseño es una construcción similar en la implementación del sistema. Esta describe un conjunto de objetos que comparten las mismas responsabilidades, las relaciones, las operaciones, atributos, y la semántica. Ver ejemplo de un diagrama de clases de diseño correspondiente al caso de uso del sistema Realizar Chequeo de Viabilidad (Ver Anexo 3. Diagrama de clases del diseño del CUS Realizar Chequeo de Viabilidad). Para consultar los restantes diagramas de clases de diseño remitirse al Expediente del Proyecto.

2.6 Identificación de las clases persistentes.

Todas las clases identificadas en el dominio del análisis no son persistentes, una clase persistente (persistent) es una clase entidad que tiene la capacidad de mantener su valor en el espacio y en el tiempo.

El diagrama de clases persistentes se utiliza para modelar la estructura lógica de la BD, las clases representan las tablas y los atributos las columnas, a partir de las clases del diseño realizadas por los analistas, se identificaron las clases persistentes y sus relaciones de: composición, agregación y asociación. Para consultar el diagrama de clases persistentes en forma ampliada (Ver Anexo 4. Diagrama de Clases Persistentes).

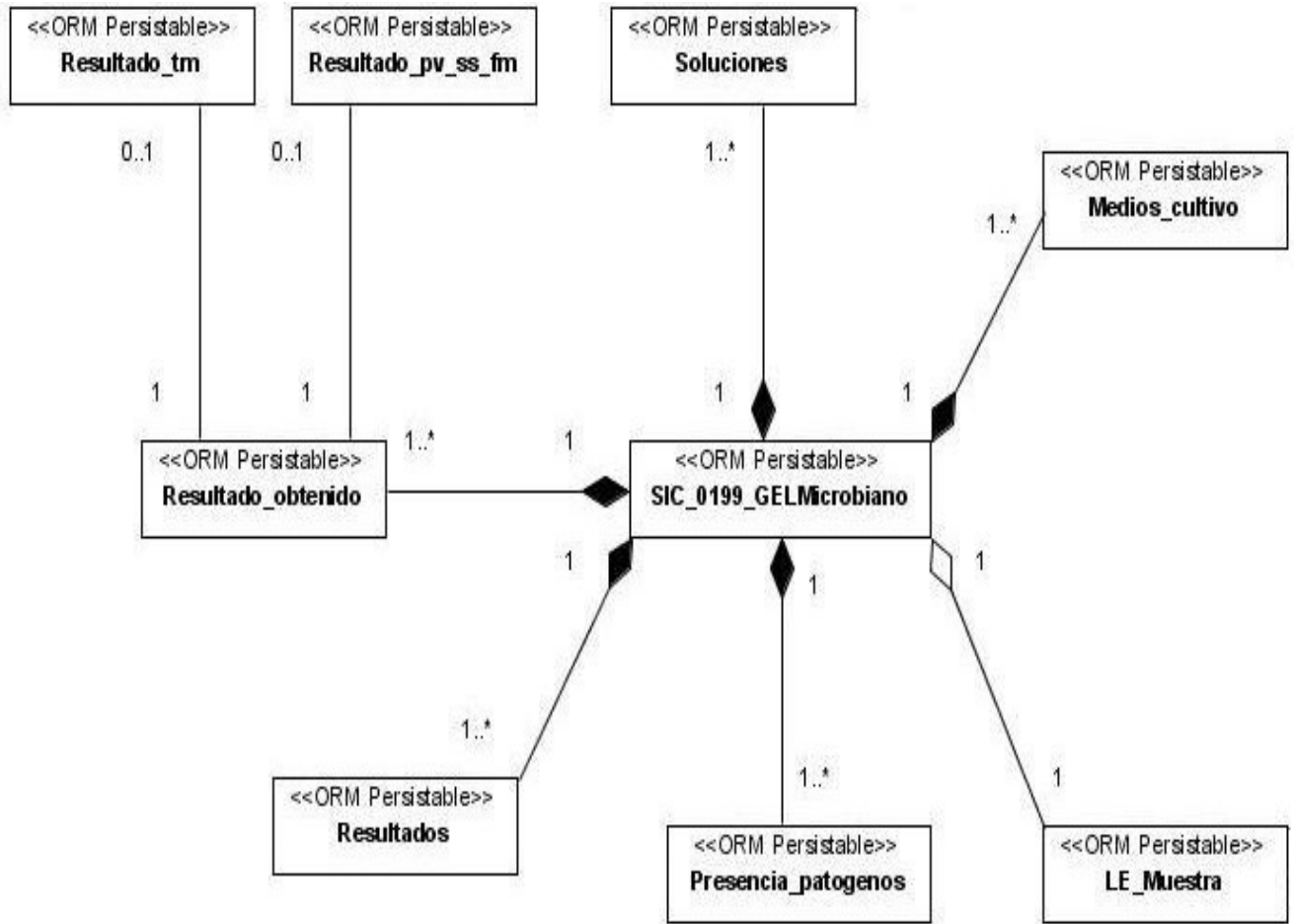


Figura 2.5 Diagrama de Clases Persistentes. (Parte 1)

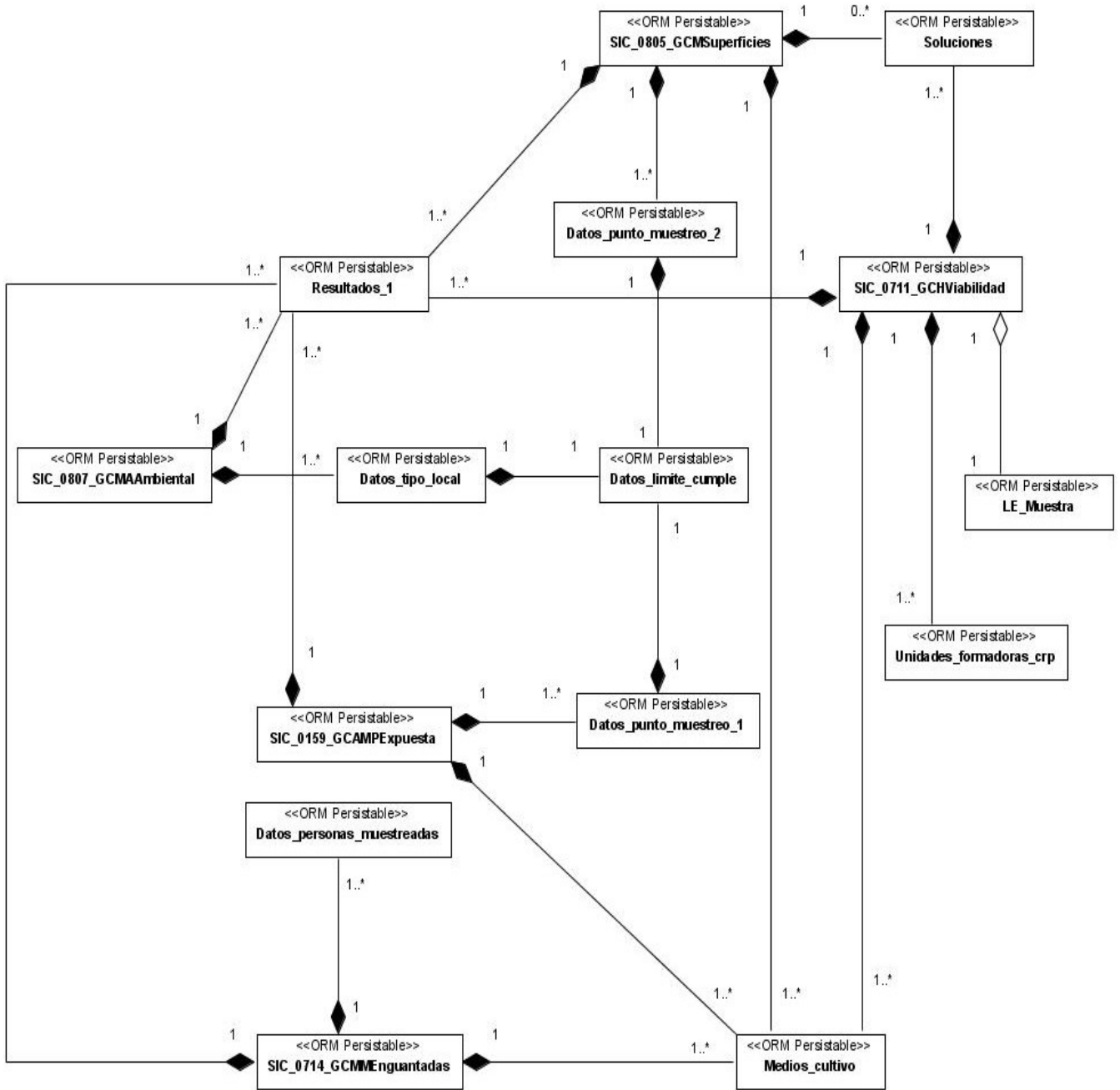


Figura 2.6 Diagrama de Clases Persistentes. (Parte 2)

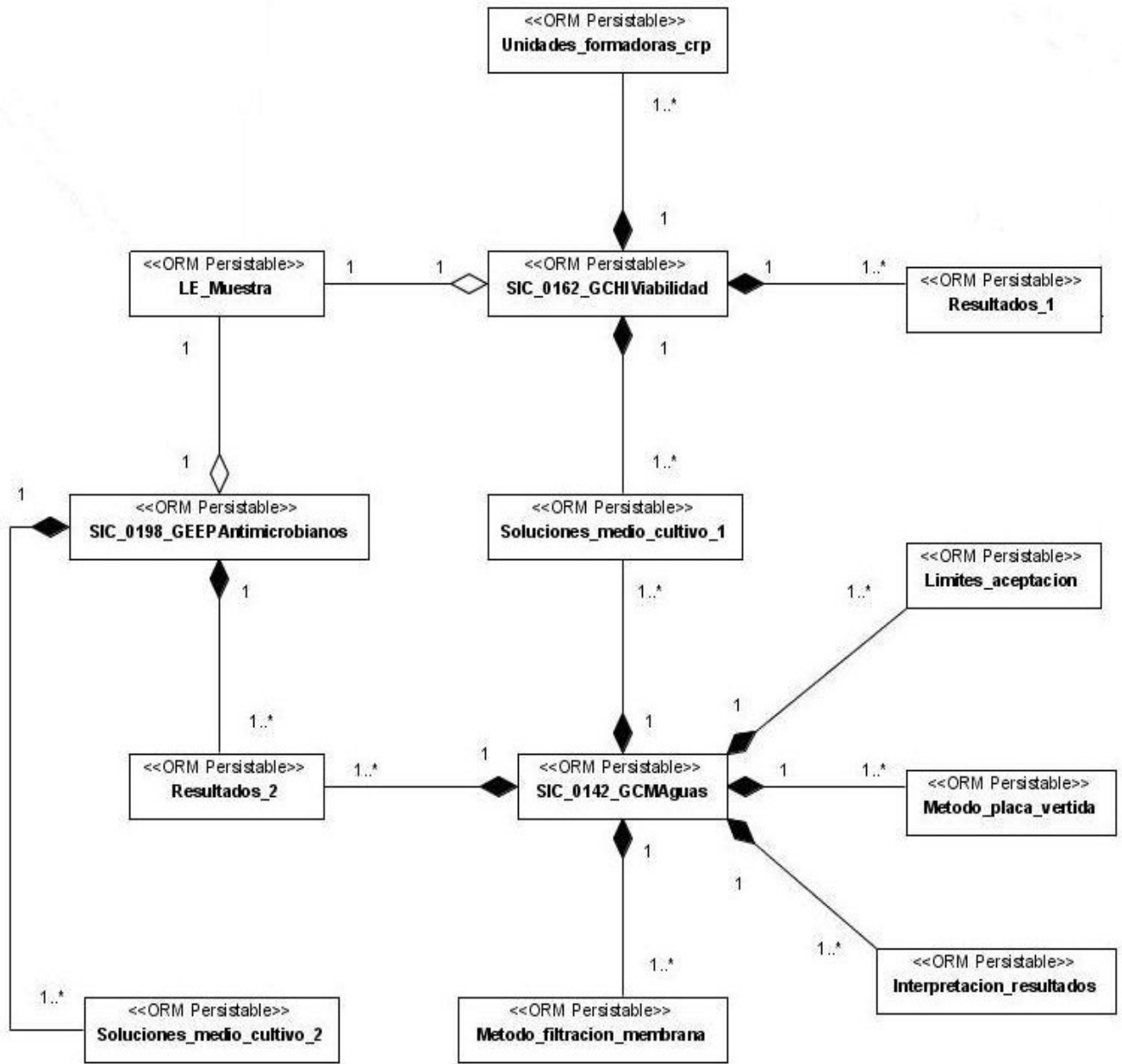


Figura 2.7 Diagrama de Clases Persistentes. (Parte 3)

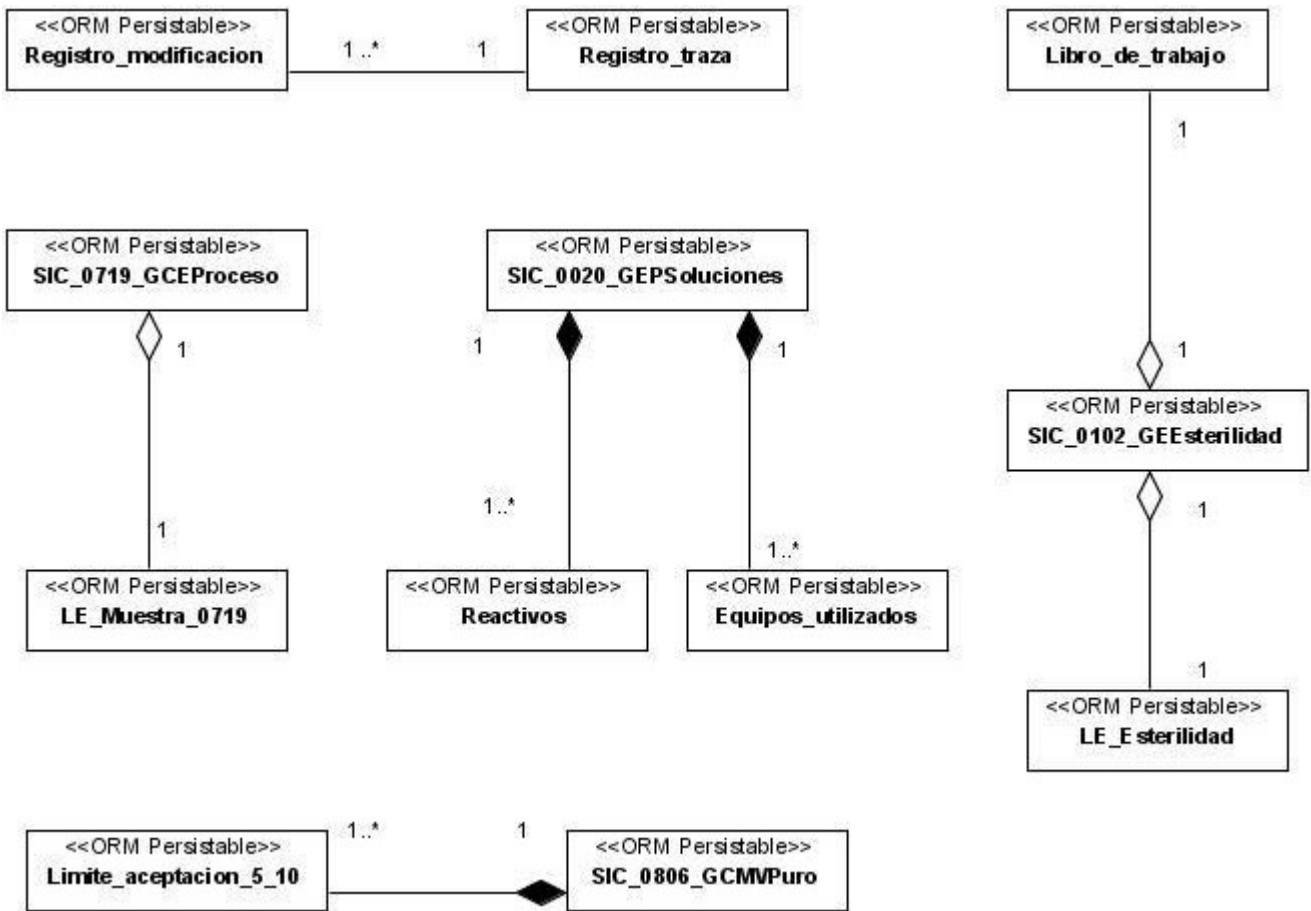


Figura 2.8 Diagrama de Clases Persistentes. (Parte 4)

2.7 Descripción de las clases persistentes.

En las siguientes tablas se incorporan las descripciones de todas las clases persistentes con sus atributos y tipos de datos, todas tienen implementados el constructor, las propiedades get y set de sus atributos.

Nombre: Datos_limite_cumple	
Tipo de clase: Entidad	
Atributo:	Tipo:
dlc_ID	long
limite_alerta	long
cumple_alerta	boolean
limite_accion	long
cumple_accion	boolean
Responsabilidad:	

Capítulo 2: Descripción y análisis de la solución propuesta

Nombre:	Datos_limite_cumple
Descripción:	Constructor

Nombre: Datos_personas_muestreadas	
Tipo de clase: Entidad	
Atributo:	Tipo:
dpm_ID	long
nomb_pers_muest	string
ufc_mano_izquierda	long
cumple_alerta_izq	boolean
cumple_accion_izq	boolean
ufc_mano_derecha	long
cumple_alerta_der	boolean
cumple_accion_der	boolean
Responsabilidad:	
Nombre:	Datos_de_personas_muestreadas
Descripción:	Constructor

Nombre: Datos_punto_muestreo_1	
Tipo de clase: Entidad	
Atributo:	Tipo:
dpmuestreo_1_ID	long
punto_muestreo	string
bacterias_hongos	double
Responsabilidad:	
Nombre:	Datos_punto_muestreo_1
Descripción:	Constructor

Nombre: Datos_punto_muestreo_2	
Tipo de clase: Entidad	
Atributo:	Tipo:
dpmuestreo_2_ID	long
punto_de_muestreo	string
b_h_UFC	double
b_h_UFC_25cm2	double
Responsabilidad:	
Nombre:	Datos_punto_muestreo_2
Descripción:	Constructor

Nombre: Datos_tipo_local	
Tipo de clase: Entidad	
Atributo:	Tipo:
datos_tipo_local_ID	long
locales	string
punto_muestreo	string

Capítulo 2: Descripción y análisis de la solución propuesta

vol_muestreado	string
bacterias_hongos	double
ufc	double
Responsabilidad:	
Nombre:	Datos_tipo_local
Descripción:	Constructor

Nombre: Equipos_utilizados	
Tipo de clase: Entidad	
Atributo:	Tipo:
equipos_utilizados_ID	long
nomb_equipos	string
cod_equipos	string
fecha_venc_calib_d	short
fecha_venc_calib_m	short
fecha_venc_calib_a	int
Responsabilidad:	
Nombre:	Equipos_utilizados
Descripción:	Constructor

Nombre: Interpretacion_resultados	
Tipo de clase: Entidad	
Atributo:	Tipo:
int_result_ID	long
punto_muestreo	string
no_control	long
lote_proc	string
ctam	double
mo_coliforme	double
e_coli	double
ps_aerug	double
cumple_alerta	boolean
cumple_accion	boolean
Responsabilidad:	
Nombre:	Interpretacion_resultados
Descripción:	Constructor

Nombre: LE_Esterilidad	
Tipo de clase: Entidad	
Atributo:	Tipo:
no_esterilidad_ID	long
folio	int
nombre_producto	string
no_entrada_lab	long
fecha_recepcion_d	short

Capítulo 2: Descripción y análisis de la solución propuesta

fecha_recepcion_m	short
fecha_recepcion_a	int
no_entrada_lab	int
toma_muestra	int
entregado_por	string
recibidor	string
Responsabilidad:	
Nombre:	LE_Esterilidad
Descripción:	Constructor

Nombre: LE_Muestra	
Tipo de clase: Entidad	
Atributo:	Tipo:
lote_muestra	string
tipo_ensayo	string
folio	int
muestra_de	string
fecha_recepcion_d	short
fecha_recepcion_m	short
fecha_recepcion_a	int
produccion	string
fecha_analisis_d	short
fecha_analisis_d	short
fecha_analisis_d	int
entregado	string
recibido	string
observ_muestra	string
Responsabilidad:	
Nombre:	LE_Muestra
Descripción:	Constructor

Nombre: LE_Muestra_0719	
Tipo de clase: Entidad	
Atributo:	Tipo:
codigo	string
no_lote	string
folio	int
descrip_muestra	string
entregado_por	string
recibidor	string
fecha_d	short
fecha_m	short
fecha_a	int
Responsabilidad:	
Nombre:	LE_Muestra_0719

Capítulo 2: Descripción y análisis de la solución propuesta

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

Nombre: Libro_de_trabajo	
Tipo de clase: Entidad	
Atributo:	Tipo:
lib_trab_ID	long
recogido_pag	long
no_no	long
folios	int
Responsabilidad:	
Nombre:	Libro_de_trabajo
Descripción:	Constructor

Nombre: Limite_aceptacion_5_10	
Tipo de clase: Entidad	
Atributo:	Tipo:
limite_acept_5_10_ID	long
punto_muestreo	string
ufc_200_mi	long
cumple	boolean
Responsabilidad:	
Nombre:	Limite_aceptacion_5_10
Descripción:	Constructor

Nombre: Limite_aceptacion	
Tipo de clase: Entidad	
Atributo:	Tipo:
limite_acept_ID	long
tipo_agua	string
limite_alerta	long
punto_1	long
limite_aceptacion	long
punto_2	long
Responsabilidad:	
Nombre:	Limite_aceptacion
Descripción:	Constructor

Nombre: Medios_cultivo	
Tipo de clase: Entidad	
Atributo:	Tipo:
medios_cultivo_ID	long
medio_cultivo	string
no_parte	long
no_lote	string
fecha_vencimiento_d	short

Capítulo 2: Descripción y análisis de la solución propuesta

fecha_vencimiento_m	short
fecha_vencimiento_a	int
Responsabilidad:	
Nombre:	Medios_cultivo
Descripción:	Constructor

Nombre: Metodo_filtracion_membrana	
Tipo de clase: Entidad	
Atributo:	Tipo:
metd_filt_membrana_ID	long
p_muestreo	string
v_filtrado	string
ctam_ufc	double
mocg_ucf	double
e_coli	boolean
ps_aeruginosa	boolean
Responsabilidad:	
Nombre:	Metodo_filtracion_membrana
Descripción:	Constructor

Nombre: Metodo_placa_vertida	
Tipo de clase: Entidad	
Atributo:	Tipo:
metd_placa_vertida_ID	long
puntos	long
ufc_placa_x1	long
ufc_placa_x2	long
Responsabilidad:	
Nombre:	Metodo_placa_vertida
Descripción:	Constructor
Nombre:	ValorMedio(int ufc_placa_x1, int ufc_placa_x2)
Descripción:	Calcula el valor medio
Nombre:	CoeficienteVariacion(float k,int d)
Descripción:	Calcula el coeficiente de variación, k es el Valor Medio y d la Desviacion Estandar.
Nombre:	DesviacionEstandar(float k, int ufc_placa_x1, int ufc_placa_x2)
Descripción:	Calcula la desviación estándar, k es el Valor Medio

Nombre: Presencia_patogenos	
Tipo de clase: Entidad	
Atributo:	Tipo:
presencia_patogenos_ID	long
muestra	string

Capítulo 2: Descripción y análisis de la solución propuesta

e_coli	boolean
ps_aeruginosa	boolean
s_aureus	boolean
salmonella_sp	boolean
c_albicans	boolean
entero_bacterias	boolean
hongos	boolean
Responsabilidad:	
Nombre:	Presencia_patogenos
Descripción:	Constructor

Nombre: Reactivos	
Tipo de clase: Entidad	
Atributo:	Tipo:
reactivos_ID	long
np	long
cantidad_usar	string
cantidad_utilizada	string
no_lote	string
nombre_reactivos	string
Responsabilidad:	
Nombre:	Reactivos
Descripción:	Constructor

Nombre: Registro_modificacion	
Tipo de clase: Entidad	
Atributo:	Tipo:
id_mod	long
valor_nuevo	string
valor_viejo	string
campo_modificado	string
Responsabilidad:	
Nombre:	Registro_modificacion
Descripción:	Constructor

Nombre: Registro_traza	
Tipo de clase: Entidad	
Atributo:	Tipo:
Id_traza	long
usuario	string
tabla_modificada	string
fecha_modificacion	Timestamp
id_tupla_modificada	string
Responsabilidad:	
Nombre:	Registro_traza

Capítulo 2: Descripción y análisis de la solución propuesta

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

Nombre: Resultado_obtenido	
Tipo de clase: Entidad	
Atributo:	Tipo:
resultado_obtenido_ID	long
muestra	string
metodo_employado	string
Responsabilidad:	
Nombre:	Resultado_obtenido
Descripción:	Constructor

Nombre: Resultado_pv_ss_fm	
Tipo de clase: Entidad	
Atributo:	Tipo:
resultado_pv_ss_fm_ID	long
ctmv_gom	long
ctml_gomL	long
Responsabilidad:	
Nombre:	Resultado_pv_ss_fm
Descripción:	Constructor

Nombre: Resultado_tm	
Tipo de clase: Entidad	
Atributo:	Tipo:
resultado_tm_ID	long
comb_tubos_crec	double
dilucion_central	double
nmp_de_m	long
Responsabilidad:	
Nombre:	Resultado_tm
Descripción:	Constructor

Nombre: Resultados	
Tipo de clase: Entidad	
Atributo:	Tipo:
resultado_ID	long
folios	int
pag_libro_result	long
Responsabilidad:	
Nombre:	Resultados
Descripción:	Constructor

Nombre: Resultados_1	
Tipo de clase: Entidad	

Capítulo 2: Descripción y análisis de la solución propuesta

Atributo:	Tipo:
result_1_ID	long
folios	int
pag_libro_result	long
libro_result_no	long
Responsabilidad:	
Nombre:	Resultados_1
Descripción:	Constructor

Nombre: Resultados_2	
Tipo de clase: Entidad	
Atributo:	Tipo:
result_2_ID	long
folios	int
pag_libro_result	long
lote_medio_cultivo	string
Responsabilidad:	
Nombre:	Resultados_2
Descripción:	Constructor

Nombre: SIC_0020_GEPSoluciones	
Tipo de clase: Entidad	
Atributo:	Tipo:
folio	long
no_lote	string
no_parte	long
vt	string
fecha_preparacion_d	short
fecha_preparacion_m	short
fecha_preparacion_a	int
nombre_solucion	string
solvente	string
PH	double
conductividad	string
ajuste_pH_sol	boolean
PH_deseado	double
sol_ajuste_pH	string
np	long
no_lotes	string
PH_final_solucion	double
esterilización	boolean
vapor_saturado	long
tiempo	int
no_frascos_almacen	long
volumen_frasco	string
filtracion	string

Capítulo 2: Descripción y análisis de la solución propuesta

almacenar_sol	string
fecha_vencimiento_d	short
fecha_vencimiento_m	short
fecha_vencimiento_a	int
recibido_por	string
fecha_recibido_d	short
fecha_recibido_m	short
fecha_recibido_a	int
limite_prom_crecimiento	int
limite_esterilidad	int
v_obt_prom_crecimiento	int
v_obt_esterilidad	int
filtracion	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0020_GEPSoluciones
Descripción:	Constructor

Nombre: SIC_0102_GEEsterilidad	
Tipo de clase: Entidad	
Atributo:	Tipo:
fecha_siembra_d	short
fecha_siembra_m	short
fecha_siembra_a	int
tipo_material	string
lote_materiales	string
resultado_materiales	boolean
libT_ambiente	int
libT_me	int
libT_superficie	int
pag_ambiente	int
pag_me	int
pag_superficie	int
granel	boolean
volumen_muestra	string
volumen_inoculado	string
no_frascos	long
vol_frasco	string
vol_inoculado_fm	string
medio_tioglicolato_3	boolean
medio_tioglicolato_5	boolean

Capítulo 2: Descripción y análisis de la solución propuesta

medio_tioglicolato_8	boolean
medio_tioglicolato_11	boolean
medio_tioglicolato_14	boolean
caldo_soya_3	boolean
caldo_soya_5	boolean
caldo_soya_8	boolean
caldo_soya_11	boolean
caldo_soya_14	boolean
em_satif	boolean
emc_satif	boolean
pcmc_satif	boolean
caae_satif	boolean
microorg_primer_ensayo	string
microorg_segundo_ensayo	string
repeticion	boolean
pe_no_valido	boolean
desviacion_proced	boolean
lista_desv	string
observ_proced	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0102_GEEsterilidad
Descripción:	Constructor
Nombre:	FechaVencimientoMc()
Descripción:	Calcular fecha de vencimiento de materiales críticos sumándole 7 días a la fecha de siembra
Nombre:	PagLTNo()
Descripción:	Calcular valor Página del libro de trabajo/número del libro producción
Nombre:	FechaVencimientoCs()
Descripción:	Calcular fecha de vencimiento de medios de cultivo y solución sumándole 15 días a la fecha de siembra.

Nombre: SIC_0142_GCMAguas	
Tipo de clase: Entidad	
Atributo:	Tipo:
folio	int
lote_proceso	string
f_siembra_d	short
f_siembra_m	short
f_siembra_a	int

Capítulo 2: Descripción y análisis de la solución propuesta

produccion	string
desviacion_proced	boolean
lista_desv	string
observ_proced	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0142_GCMAguas
Descripción:	Constructor

Nombre: SIC_0159_GCAMPExpuesta	
Tipo de clase: Entidad	
Atributo:	Tipo:
folio	int
lote_proceso	
fecha_muestreo_d	short
fecha_muestreo_m	short
fecha_muestreo_a	int
hora	string
no_control	long
cond_muestreo	string
cant_personas	long
limpieza_muestreo	boolean
observ_microscopico	string
area	string
higienizante_utilizado	string
produccion	string
desviacion_proced	boolean
lista_desv	string
observ_proced	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0159_GCAMPExpuesta
Descripción:	Constructor

Nombre: SIC_0162_GCHIViabilidad	
Tipo de clase: Entidad	

Capítulo 2: Descripción y análisis de la solución propuesta

Atributo:	Tipo:
cepa	string
no_entrada_lab	long
banco_homogeneo	boolean
repeticion	boolean
pe_no_valido	boolean
desviacion_proced	boolean
lista_desv	string
observ_proced	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0162_GCHIVIabilidad
Descripción:	Constructor
Nombre:	ConcentracionBanco()
Descripción:	Promedio de los resultados de los valores de Xm

Nombre: SIC_0198_GEEPAntimicrobianos	
Tipo de clase: Entidad	
Atributo:	Tipo:
preservo	string
concentracion	long
tiempo_almacenamiento	string
temperatura	float
no_entrada_lab	long
fecha_comienzo_ensayo_d	short
fecha_comienzo_ensayo_m	short
fecha_comienzo_ensayo_a	int
tamano_muestra	long
concent_medio_cultivo	double
e_coli_ci	string
ps_aerug_ci	string
st_ci	string
c_alb_ci	string
a_niger_ci	string
e_coli_volm	string
ps_aerug_volm	string
st_volm	string
c_alb_volm	string
a_niger_volm	string
ufc_e_coli	string

Capítulo 2: Descripción y análisis de la solución propuesta

ufc_ps_aerug	string
ufc_st	string
ufc_c_alb	string
ufc_a_niger	string
e_coli_0	int
e_coli_7	int
e_coli_14	int
e_coli_28	int
ps_aerug_0	int
ps_aerug_7	int
ps_aerug_14	int
ps_aerug_28	int
st_0	int
st_7	int
st_14	int
st_28	int
c_alb_0	int
c_alb_7	int
c_alb_14	int
c_alb_28	int
a_niger_0	int
a_niger_7	int
a_niger_14	int
a_niger_28	int
cond_muestra	string
repeticion	boolean
pe_no_valido	boolean
desviacion_proced	boolean
lista_desv	string
observ_proced	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0198_GEEPAntimicrobianos
Descripción:	Constructor

Nombre: SIC_0199_GELMicrobiano	
Tipo de clase: Entidad	
Atributo:	Tipo:
no_control	long
fecha_ensayo_d	short
fecha_ensayo_m	short

Capítulo 2: Descripción y análisis de la solución propuesta

fecha_ensayo_a	int
conteo_microorg	long
conteo_mohos_levaduras	long
ausencia_de	string
repeticion	boolean
pe_no_valido	boolean
desviacion_proced	boolean
lista_desv	string
observ_proced	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0199_GELMicrobiano
Descripción:	Constructor

Nombre: SIC_0711_GCHViabilidad	
Tipo de clase: Entidad	
Atributo:	Tipo:
cepa	string
no_entrada_lab	long
volumen_inoculo	long
diluciones_sembradas	double
concent_banco	string
concent_banco_ant	string
alerta	boolean
repeticion	boolean
pe_no_valido	boolean
desviacion_proced	boolean
lista_desv	string
observ_proced	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0711_GCHViabilidad
Descripción:	Constructor

Nombre: SIC_0714_GCMMEnguantadas	
Tipo de clase: Entidad	

Capítulo 2: Descripción y análisis de la solución propuesta

Atributo:	Tipo:
folio	long
lote_proceso	string
fecha_muestreo_d	short
fecha_muestreo_d	short
fecha_muestreo_d	int
num_control	long
area	string
produccion	string
desviacion_proced	boolean
lista_desv	string
observ_proced	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0714_GCMMEnguantadas
Descripción:	Constructor

Nombre: SIC_0719_GCEProceso	
Tipo de clase: Entidad	
Atributo:	Tipo:
datos_muestra_ID	Long
results_719	string
produccion	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0719_GCEProceso
Descripción:	Constructor

Nombre: SIC_0805_GCMSuperficies	
Tipo de clase: Entidad	
Atributo:	Tipo:
folio	int
metodo_utilizado	string
objeto_muestreo	string
fecha_muestreo_d	short
fecha_muestreo_m	short

Capítulo 2: Descripción y análisis de la solución propuesta

fecha_muestreo_a	int
area	string
produccion	string
desviacion_proced	boolean
lista_desv	string
observ_proced	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0805_GCMSuperficies
Descripción:	Constructor

Nombre: SIC_0806_GCMVPuro	
Tipo de clase: Entidad	
Atributo:	Tipo:
folio	int
no_lote	string
fecha_muestreo_d	short
fecha_muestreo_m	short
fecha_muestreo_a	int
f_vencimiento_d	short
f_vencimiento_m	short
f_vencimiento_a	int
result_pag	long
result_folio	long
produccion	string
desviacion_proced	boolean
lista_desv	string
observ_proced	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0806_GCMVPuro
Descripción:	Constructor

Nombre: SIC_0807_GCMAAmbiental	
Tipo de clase: Entidad	
Atributo:	Tipo:

Capítulo 2: Descripción y análisis de la solución propuesta

folio	int
fecha_muestreo_d	short
fecha_muestreo_d	short
fecha_muestreo_d	int
lotes_proceso	string
medio_cultivo	string
np	long
no_lote	string
personas_area	int
limpieza_muestreo	boolean
area	string
higienizante_utilizado	string
produccion	string
observaciones	string
realizado_por	string
fecha_realizado_d	short
fecha_realizado_m	short
fecha_realizado_a	int
terminado	boolean
Responsabilidad:	
Nombre:	SIC_0807_GCMAAmbiental
Descripción:	Constructor

Nombre: Soluciones	
Tipo de clase: Entidad	
Atributo:	Tipo:
soluciones_ID	long
solucion	string
no_parte	long
no_lote	string
fecha_vencimiento_d	short
fecha_vencimiento_m	short
fecha_vencimiento_a	int
Responsabilidad:	
Nombre:	Soluciones
Descripción:	Constructor

Nombre: Soluciones_medio_cultivo_1	
Tipo de clase: Entidad	
Atributo:	Tipo:
sol_mc_1_ID	long
sol_medio_cultivo	string
no_parte	long
no_de_lote	string
fecha_vencimiento_d	short
fecha_vencimiento_m	short

Capítulo 2: Descripción y análisis de la solución propuesta

fecha_vencimiento_a	int
Responsabilidad:	
Nombre:	Soluciones_medio_cultivo_1
Descripción:	Constructor

Nombre: Soluciones_medio_cultivo_2	
Tipo de clase: Entidad	
Atributo:	Tipo:
sol_mc_2_ID	long
sol_medio_cultivo	string
no_parte	long
no_de_lote	string
fecha_vencimiento_d	short
fecha_vencimiento_m	short
fecha_vencimiento_a	int
fecha_uso_d	short
fecha_uso_m	short
fecha_uso_a	int
Responsabilidad:	
Nombre:	Soluciones_medio_cultivo_2
Descripción:	Constructor

Nombre: Unidades_formadoras_crp	
Tipo de clase: Entidad	
Atributo:	Tipo:
colonias_replicas_1_ID	int
bulbo_no	int
x1	int
x2	int
x3	int
x4	int
DIL	float
Responsabilidad:	
Nombre:	Unidades_formadoras_crp
Descripción:	Constructor
Nombre:	ValorMedio (int x1, int x2, int x3, int x4)
Descripción:	Método para calcular el valor medio
Nombre:	CoeficienteVariacion(float k, int x1, int x2, int x3, int x4)
Descripción:	Método para calcular el coeficiente de variación, k es el valor Medio
Nombre:	ConcentracionBanco(int x1, int x2, int x3, int x4)
Descripción:	Promedio de los resultados de los valores de Xm
Nombre:	XmBulbo(int x1, int x2, int x3, int x4)
Descripción:	Promedio de los resultados de los valores de Xm

2.8 Diseño de la BD.

El modelo físico de datos se desarrolló a partir de la base del conjunto de clases persistentes y sus asociaciones en el modelo de diseño, la herramienta utilizada para la realización de este modelo fue Visual Paradigm for UML 6.0 Enterprise Edition generándose posteriormente para PostgreSQL. A continuación se representa el modelo físico de datos de la BD del Laboratorio de Microbiología, para consultar el modelo en forma ampliada (Ver Anexo 5. Modelo Físico de Datos).

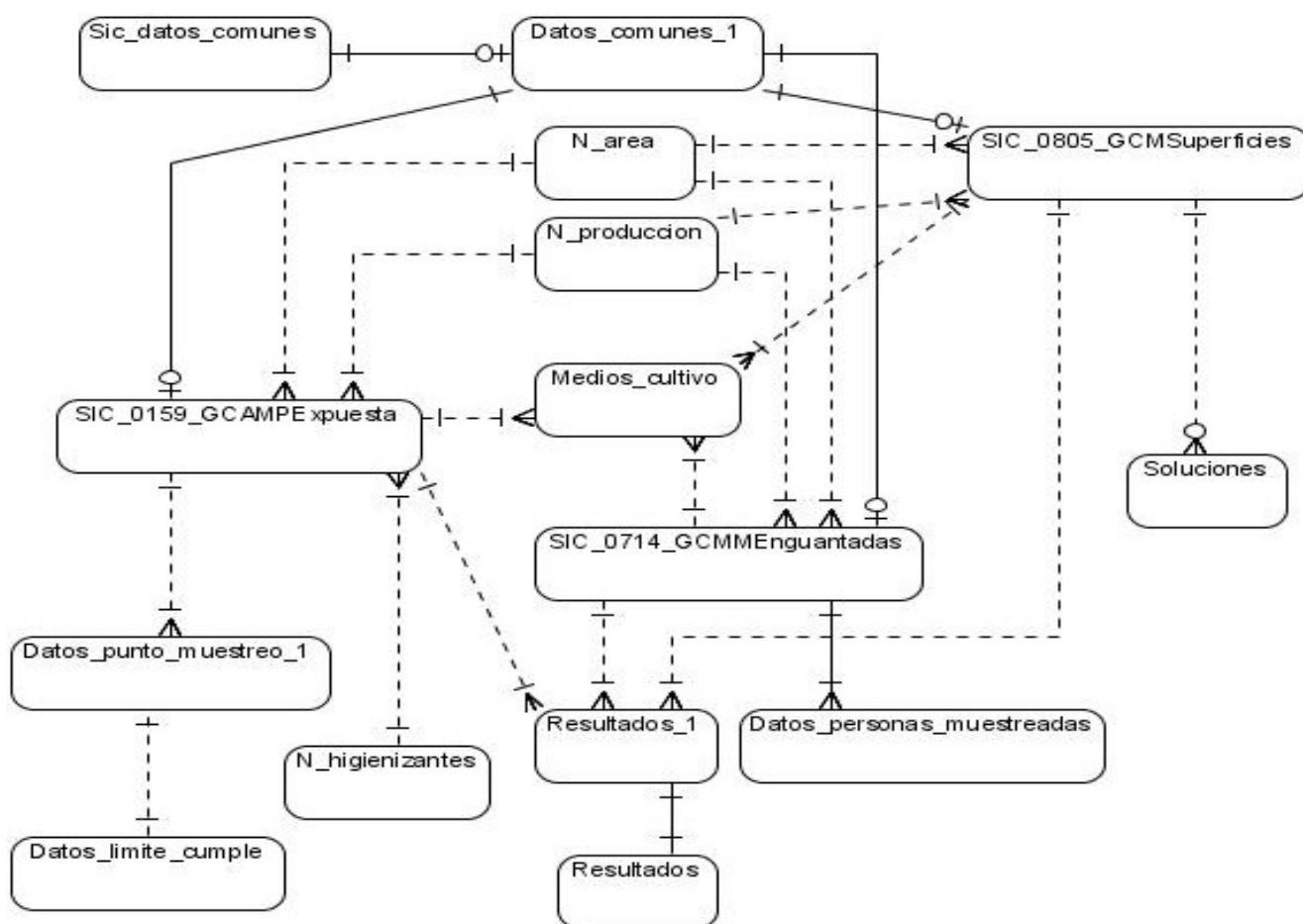


Figura 2.9 Modelo Físico de Datos. (Parte 1)

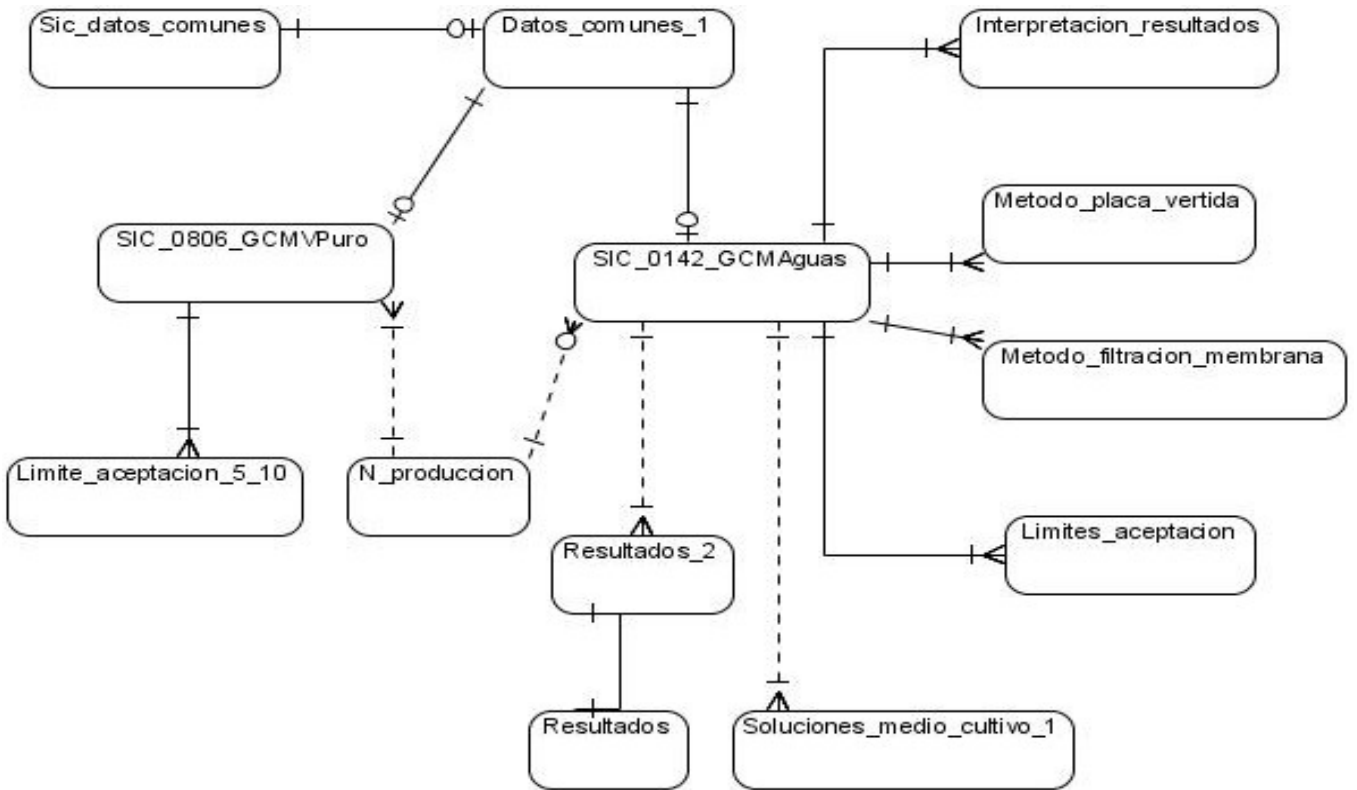


Figura 2.10 Modelo Físico de Datos. (Parte 2)

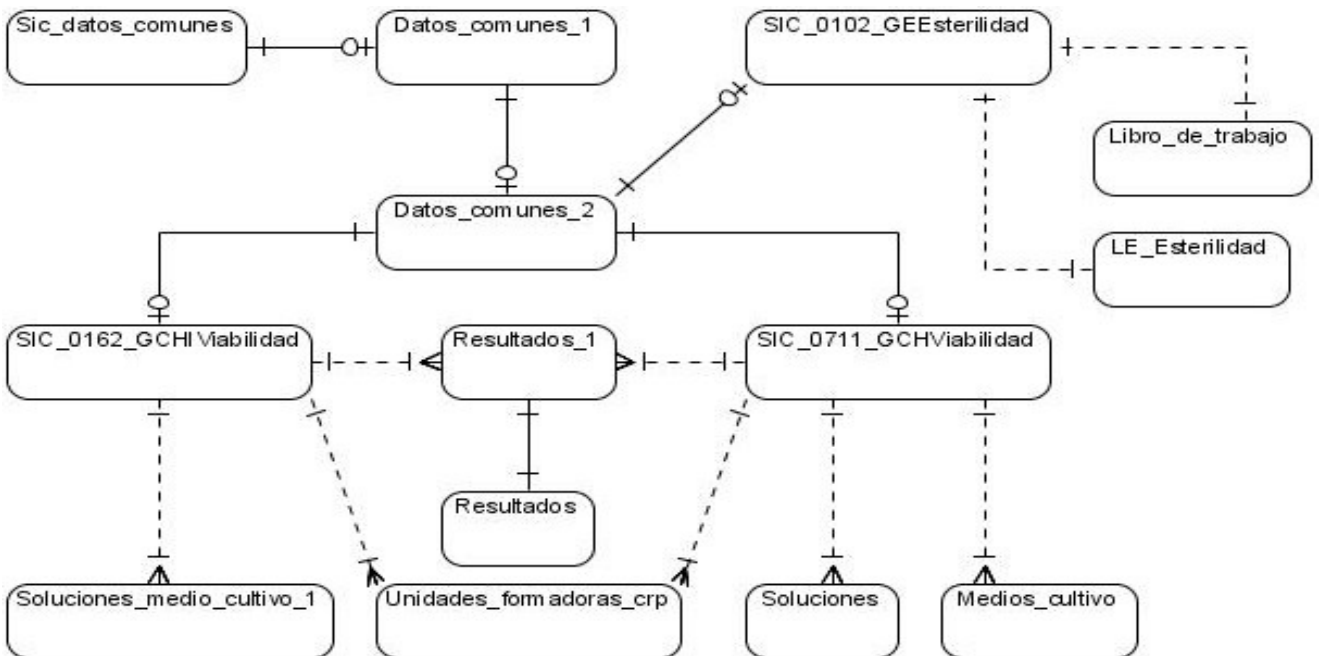


Figura 2.11 Modelo Físico de Datos. (Parte 3)

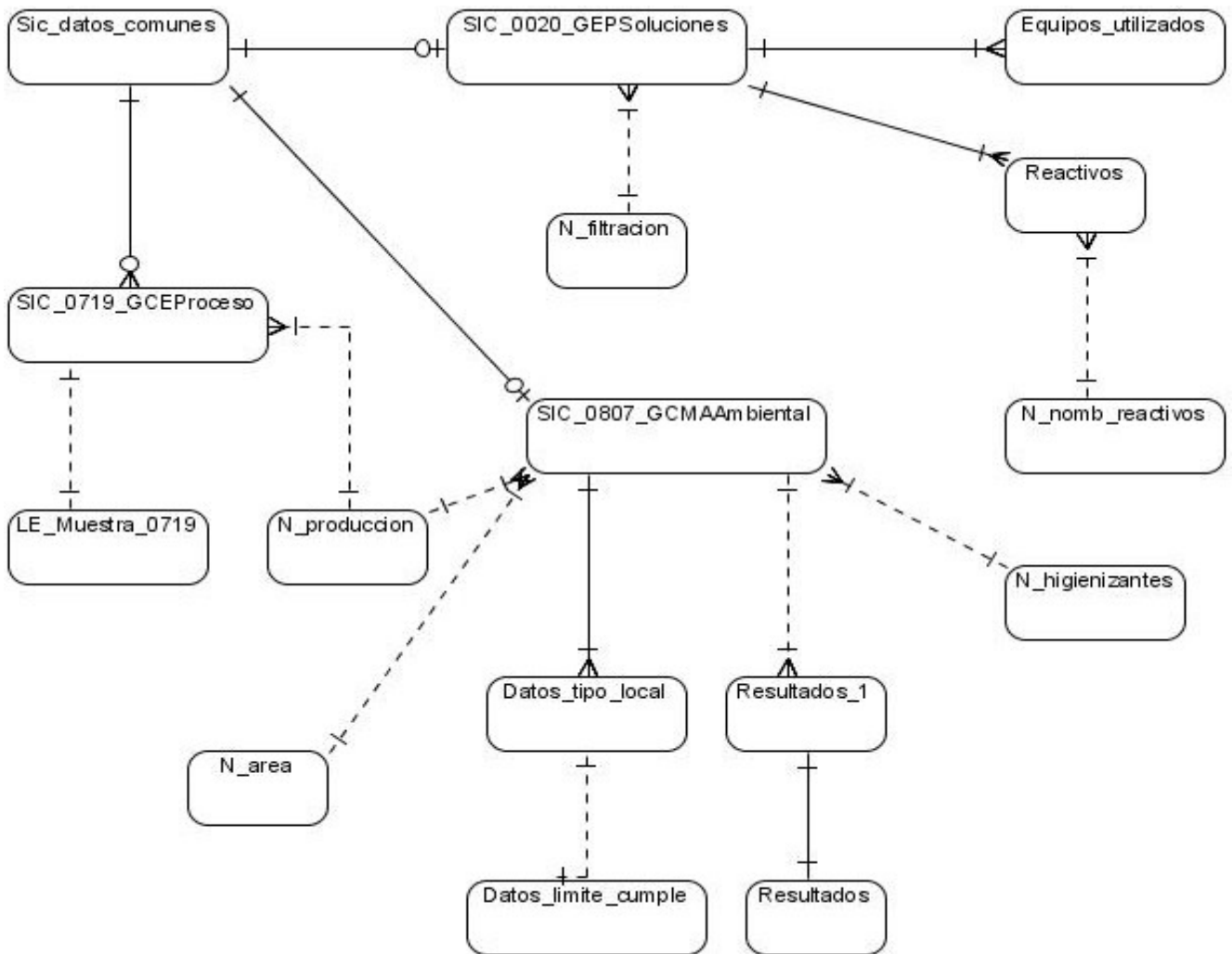


Figura 2.13 Modelo Físico de Datos. (Parte 5)

2.9 Descripción de las tablas.

Nombre: Datos_comunes_1		
Descripción: En esta tabla se agrupan atributos comunes de los SIC 0142,0159, 0714, 0805 y 0806.		
Atributo	Tipo	Descripción
+#Sic_datos_1_ID	int	Llave primaria foránea que la toma de la entidad Sic_datos_comunes.
Desviacion_proced	bool	Cumple o no.
Lista_desv	int	Lista de desviaciones.
Observ_proced	varchar	Observaciones de los procedimientos

Nombre: Datos_comunes_2		
Descripción: En esta tabla se agrupan atributos comunes de los SIC 0102, 0162, 0711, 0198 y 0199.		

Capítulo 2: Descripción y análisis de la solución propuesta

Atributo	Tipo	Descripción
+#Sic_datos_2_ID	int	Llave primaria foránea que la toma de la entidad Datos_comunes_1.
Repeticion	bool	Si es repetición o ensayo inicial.
Pe_no_valido	bool	Si el primer ensayo es no válido.
Pasa_prueba	bool	Pasa o no la prueba

Nombre: Datos_limite_cumple		
Descripción: : En esta tabla se agrupan atributos comunes de las tablas Datos_punto_muestreo_1, Datos_punto_muestreo_2 y Datos_tipo_local.		
Atributo	Tipo	Descripción
+Dlc_ID	int	Valor entero auto incremental. Llave primaria.
Limite_alerta	int	Valor limite alerta del punto de muestreo.
Cumple_alerta	bool	Cumple o no alerta.
Limite_accion	int	Valor limite acción del punto de muestreo.
Cumple_accion	bool	Cumple o no acción.

Nombre: Datos_de_personas_muestreadas		
Descripción: Datos de las personas muestreadas para el SIC 0714.		
Atributo	Tipo	Descripción
+Dpm_ID	int	Valor entero auto incremental. Llave primaria.
+#Dc1_Sic_datos_ID_714	int	Llave primaria foránea que la toma de la entidad SIC_0714_GCMMEnguantadas.
Nomb_pers_muest	varchar	Nombre de persona muestreada
Ufc_mano_izquierda	int	Valor de UFC izquierda.
Cumple_alerta_izq	bool	Cumple o no límite alerta.
Cumple_accion_izq	bool	Cumple o no límite de acción.
Ufc_mano_derecha	int	Valor de UFC derecha.
Cumple_alerta_der	bool	Cumple o no límite alerta.
Cumple_accion_der	bool	Cumple o no límite de acción

Nombre: Datos_punto_muestreo_1		
Descripción: Representa los datos de los punto de muestreo de las muestras para el SIC 0159.		
Atributo	Tipo	Descripción
+Dpmuestreo_1_ID	int	Valor entero auto incremental. Llave primaria.
Punto_muestreo	varchar	Puntos de muestreo de la muestra.
Bacterias_hongos	float	Valor de UFC para bacterias y hongos.
#Dc1_Sic_datos_ID_159	int	Llave foránea que la toma de la entidad SIC_0159_GCAMPExpuesta.
#Dlc_ID	int	Llave foránea que toma de la entidad Datos_limite_cumple.

Capítulo 2: Descripción y análisis de la solución propuesta

Nombre: Datos_punto_muestreo_2		
Descripción: Representa los datos de los puntos de muestreo de las muestras para el SIC 0805.		
Atributo	Tipo	Descripción
+Dpmuestreo_2_ID	int	Valor entero auto incremental. Llave primaria.
+#Dc1_Sic_datos_ID_805	int	Llave primaria foránea que la toma de la entidad SIC_0805_GCMSuperficies.
Punto_muestreo	int	Puntos de muestreo de la muestra.
B_h_ufc	float	Valor de UFC a 5 ml para Bacterias y hongos.
B_h_ufc_25cm2	float	Valor de UFC a 25 cm2 para bacterias y hongos.
#Dlc_ID	int	Llave foránea que toma de la entidad Datos_limite_cumple.

Nombre: Datos_tipo_local		
Descripción: Representa los datos de los locales relacionados con el SIC 0807.		
Atributo	Tipo	Descripción
+Datos_tipo_local_ID	int	Valor entero auto incremental. Llave primaria.
+#Sdc_Sic_datos_ID_807	varchar	Llave primaria foránea que la toma de la entidad SIC_0807_GCMAAmbiental.
Local	varchar	Nombre del Local.
Punto_muestreo	varchar	Puntos de muestreo de la muestra.
Vol_muestreado	varchar	Volumen muestreado.
Bacterias_hongos	float	Valor de UFC en placas para bacterias y hongos.
Ufc	float	Valor UFC en m3.
#Dlc_ID	int	Llave foránea que toma de la entidad Datos_limite_cumple.

Nombre: Equipos_utilizados		
Descripción: Nombre de los equipos utilizados para realizar el ensayo de preparación de soluciones.		
Atributo	Tipo	Descripción
+Equipos_utilizados_ID	int	Valor entero auto incremental. Llave primaria.
+#Sdc_Sic_datos_ID_20	int	Llave primaria foránea que la toma de la entidad SIC_0020_GEPSoluciones.
Nomb_equipos	varchar	Nombre de equipo en el ensayo.
Cod_equipos	varchar	Código de los equipos.
Fecha_venc_calib_d	int	Día de vencimiento de calibración.
Fecha_venc_calib_m	int	Mes de vencimiento de calibración.
Fecha_venc_calib_a	int	Año de vencimiento de calibración.

Nombre: Interpretacion_resultados		
Descripción: Interpretación de los resultados de la muestra analizada.		
Atributo	Tipo	Descripción
+Int_result_ID	int	Valor entero auto incremental. Llave primaria.
+#Dc1_Sic_datos_ID_142	int	Llave primaria foránea que la toma de la entidad

Capítulo 2: Descripción y análisis de la solución propuesta

		SIC_0142_GCMAguas.
Puntos_muestreo	varchar	Puntos de muestreo de la muestra
No_control	int	Número de control del punto de muestreo.
Lote_proc	varchar	Lote en proceso.
Ctam	float	Valor Ctam.
Mo_coliforme	float	Valor Mo_coliforme.
E_coli	float	Valor E_coli.
Ps_aerug	float	Valor Ps_aerug.
Cumple_alerta	bool	Cumple o no alerta.
Cumple_accion	bool	Cumple o no acción.

Nombre: LE_Esterilidad		
Descripción: Representa el libro de entrada de esterilidad.		
Atributo	Tipo	Descripción
+ No_esterilidad_ID	int	Valor entero auto incremental. Llave primaria.
+ No_lote	varchar	Lote de la muestra. Llave primaria.
Folio	int	Valor secuencial que se repite desde 1 hasta 999.
Nombre_producto	varchar	Nombre del producto
Muestra_de	varchar	Nombre de la muestra
No_entrada_lab	int	Número de entrada al laboratorio
Fecha_recepcion_d	int	Día de recepción
Fecha_recepcion_m	int	Mes de recepción
Fecha_recepcion_a	int	Año de recepción
Toma_muestra	varchar	Valor tomado de la muestra
Entregado_por	varchar	Muestra entregada por el distribuidor
Recibidor	varchar	Analista que recibe la muestra

Nombre: LE_Muestra		
Descripción: Representa el libro de entrada de muestra.		
Atributo	Tipo	Descripción
+ Muestra_de	varchar	Lote de la muestra. Llave primaria
+# N_tipo_ensayo_ID	int	Llave primaria foránea que la toma de la entidad N_tipo_ensayo
# N_produccion_ID	varchar	Producción de la muestra
Folio	int	Folio del libro entrada de muestra
Lote_muestra	varchar	Número de lote de la muestra
Fecha_recepcion_d	varchar	Día de recepción
Fecha_recepcion_m		Mes de recepción
Fecha_recepcion_a		Año de recepción
Fecha_analisis_d	int	Día del análisis
Fecha_analisis_m	int	Mes del análisis
Fecha_analisis_a	int	Año del análisis
Entregado	varchar	Muestra entregada por el distribuidor
Recibido	varchar	Analista que recibe la muestra
Observ_muestra	varchar	Observaciones del la muestra

Capítulo 2: Descripción y análisis de la solución propuesta

Nombre: LE_Muestra_0719		
Descripción: Representa el libro de entrada demuestra que se relaciona con el SIC 0719.		
Atributo	Tipo	Descripción
+Codigo	varchar	Código de la muestra. Llave primaria
No_lote	varchar	Lote de lo muestra
Folio	int	Folio del libro entrada de muestra 0179
Descrip_muestra	varchar	Descripción de la muestra
Entregado_por	varchar	Muestra entregada por el distribuidor
Recibidor	varchar	Analista que recibe la muestra
Fecha_d	int	Día
Fecha_m	int	Mes
Fecha_a	int	Año

Nombre: Libro_de_trabajo		
Descripción: Representa el libro de trabajo que se relaciona con el SIC 0102.		
Atributo	Tipo	Descripción
+Lib_trab_ID	int	Valor entero auto incremental. Llave primaria
Recogido_pag	int	Página del libro de trabajo
No_no	int	Número del libro de trabajo
Folios	int	Folios del libro de trabajo

Nombre: Limite_aceptacion_5_10		
Descripción: Límites de aceptación para el análisis microbiológico de vapor puro.		
Atributo	Tipo	Descripción
+Limite_acept_5_10_ID	int	Valor entero auto incremental. Llave primaria
+#Dc1_Sic_datos_ID_806	int	Llave primaria foránea que la toma de la entidad SIC_0806_GCMVPuro
Punto_muestreo	varchar	Puntos de muestreo de la muestra
UFC_200_MI	float	Valor para UFC de 200 mi
Cumple	bool	Cumple o no

Nombre: Limites_de_aceptacion		
Descripción: Límites de aceptación para el control microbiológico de las aguas.		
Atributo	Tipo	Descripción
+ Limite_acept_ID	int	Valor entero auto incremental. Llave primaria
+# Dc1_Sic_datos_ID_142	int	Llave primaria foránea que la toma de la entidad SIC_0142_GCMAguas
Tipo_agua	varchar	Tipo de agua para realizar en control microbiológico
Limite_alerta	int	Valor límite alerta del punto de muestreo
Punto_1	int	Límite de alerta para el agua
Limite_aceptacion	int	Valor límite acción del punto de muestreo
Punto_2	int	Límite de aceptación para el agua

Capítulo 2: Descripción y análisis de la solución propuesta

Nombre: Medios_cultivo		
Descripción: Recoge los datos de los medios de cultivos relacionados con los ensayos.		
Atributo	Tipo	Descripción
+ Medios_cultivo_ID	int	Valor entero auto incremental. Llave primaria
Medio_cultivo	varchar	Nombre de medio de cultivo
No_parte	int	Número de parte
Lote	varchar	Número de lote
Fecha_vencimiento_d	int	Día de vencimiento
Fecha_vencimiento_m	int	Mes de vencimiento
Fecha_vencimiento_a	int	Año de vencimiento
#Dc1_Sic_datos_ID_159	int	Llave foránea que la toma de la entidad SIC_0159_GCAMPExpuesta
#Dc2_Sic_datos_2_ID_711	int	Llave foránea que la toma de la entidad SIC_0711_GCHViabilidad
#Dc1_Sic_datos_ID_805	int	Llave foránea que la toma de la entidad SIC_0805_GCMSuperficies
#Dc1_Sic_datos_ID_714	int	Llave foránea que la toma de la entidad SIC_0714_GCMMEnguantadas
#Dc2_Sic_datos_2_ID_199	int	Llave foránea que la toma de la entidad SIC_0199_GELMicrobiano

Nombre: Metodo_filtracion_membrana		
Descripción: Recoge los datos del método de filtración por membrana aplicado a la muestra.		
Atributo	Tipo	Descripción
+Metd_filt_membrana_ID	int	Valor entero auto incremental. Llave primaria
+#Dc1_Sic_datos_ID_142	int	Llave primaria foránea que la toma de la entidad SIC_0142_GCMAguas
P_muestreo	varchar	Puntos de muestreo para filtración por membrana
V_filtrado	varchar	Volumen filtrado
Ctam_ufc	float	Conteo total de aerobios mesófilos ufc/placa
Mocg_ucf	float	M.o del grupo coliforme ufc/placa
E_coli	bool	Determinación de Patógenos ufc características de E_coli
Ps_aeruginosa	bool	Determinación de Patógenos ufc características de Ps_aeruginosa

Nombre: Metodo_placa_vertida		
Descripción: Recoge los datos del método placa vertida aplicado a la muestra.		
Atributo	Tipo	Descripción
+ Metd_placa_vertida_ID	int	Valor entero auto incremental. Llave primaria
+# Dc1_Sic_datos_ID_142	int	Llave primaria foránea que la toma de la entidad SIC_0142_GCMAguas

Capítulo 2: Descripción y análisis de la solución propuesta

Puntos	int	Puntos de la placa vertida
Ufc_placa_x1	int	Valor UFC para placa x1
Ufc_placa_x2	int	Valor UFC para placa x2

Nombre: N_area		
Descripción: Nomenclador para las áreas existentes.		
Atributo	Tipo	Descripción
+ Area_ID	int	Valor entero auto incremental. Llave primaria
Descripcion	varchar	Nombre de las aéreas de donde proviene la muestra

Nombre: N_filtracion		
Descripción: Nomenclador para los tipos de filtraciones.		
Atributo	Tipo	Descripción
+ Filtracion_ID	int	Valor entero auto incremental. Llave primaria
Descripcion	varchar	Tipo de filtración de la muestra

Nombre: N_higienizantes		
Descripción: Nomenclador para los higienizantes utilizados.		
Atributo	Tipo	Descripción
+Higt_ID	int	Valor entero auto incremental. Llave primaria
Descripcion	varchar	Higienizantes utilizados en una muestra

Nombre: N_metodo_empleado		
Descripción: Nomenclador para los métodos empleados en el conteo total de microorganismos viables.		
Atributo	Tipo	Descripción
+N_metodo_empleado_ID	int	Valor entero auto incremental. Llave primaria
Descripcion	varchar	Método empleado para los ensayos de las muestras

Nombre: N_muestras		
Descripción: Nomenclador para la condiciones de las muestras.		
Atributo	Tipo	Descripción
+Cond_muestra_ID	int	Valor entero auto incremental. Llave primaria
Descripcion	varchar	Condiciones de las muestras

Nombre: N_neutralizante		
Descripción: Nomenclador para los neutralizantes utilizados en los ensayos.		
Atributo	Tipo	Descripción
+Neutralizante_ID	int	Valor entero auto incremental. Llave primaria
Descripcion	varchar	Neutralizantes utilizados para los ensayos de las muestras

Capítulo 2: Descripción y análisis de la solución propuesta

Nombre: N_nomb_reactivos		
Descripción: Nomenclador para los reactivos utilizados en los ensayos.		
Atributo	Tipo	Descripción
+React_ID	int	Valor entero auto incremental. Llave primaria
Descripcion	varchar	Nombre de los reactivos empleados en los ensayos

Nombre: N_produccion		
Descripción: Nomenclador para las producciones.		
Atributo	Tipo	Descripción
+Produccion_ID	int	Valor entero auto incremental. Llave primaria
Descripcion	varchar	Tipos de producciones

Nombre: N_tipo_ensayo		
Descripción: Nomenclador para el tipo de ensayo utilizado.		
Atributo	Tipo	Descripción
+Tipo_ensayo_ID	int	Valor entero auto incremental. Llave primaria
Descripcion	varchar	Tipo de ensayo

Nombre: Presencia_patogenos		
Descripción: Almacena los datos de presencia de patógenos en 10 g o mL de muestra.		
Atributo	Tipo	Descripción
+Presencia_patogenos_ID	int	Valor entero auto incremental. Llave primaria
+#Dc2_Sic_datos_2_ID_199	int	Llave primaria foránea que la toma de la entidad SIC_0199_GELMicrobiano
Muestra	varchar	Nombre de la muestra
E_coli	bool	Cumple o no
Ps_aeruginosa	bool	Cumple o no
S_aureus	bool	Cumple o no
Salmonella_sp	bool	Cumple o no
C_albicans	bool	Cumple o no
Entero_bacterias	bool	Cumple o no
Hongos	bool	Cumple o no

Nombre: Reactivos		
Descripción: Reactivos utilizados en el ensayo.		
Atributo	Tipo	Descripción
+Reactivos_ID	int	Valor entero auto incremental. Llave primaria de los reactivos
+#Sdc_Sic_datos_ID_20	int	Llave primaria foránea que la toma de la entidad SIC_0020_GEPSoluciones
Np	int	Número parte de los reactivos
Cantidad_usar	varchar	Cantidad de reactivos a usar
Cantidad_utilizada	varchar	Cantidad de reactivos a utilizar
No_Lote	varchar	Lote de los reactivos
#React_ID	int	Llave foránea que la toma de la entidad N_nomb_reactivos

Capítulo 2: Descripción y análisis de la solución propuesta

Nombre: Registro_modificacion		
Descripción: Tabla utilizada para almacenar los valores de los datos modificados.		
Atributo	Tipo	Descripción
+Id_mod	int	Valor entero auto incremental. Llave primaria
+#Idtraza		Llave primaria foránea que la toma de la entidad Registro_trazas
Valor_viejo	varchar	Valor antes de la modificación
Valor_nuevo	timestamp	Valor después de la modificación
Campo_modificado	varchar	Nombre del campo modificado

Nombre: Registro_traza		
Descripción: Tabla utilizada para mantener el registro de las trazas de modificación de los datos de las tablas.		
Atributo	Tipo	Descripción
+Id_traza	int	Valor entero auto incremental. Llave primaria
Usuario	varchar	Nombre de usuario
Tabla_modificada	varchar	Nombre de tabla modificada
Fecha_modificacion	timestamp	Fecha de modificación de los datos
Id_tupla_modificada	varchar	Identificador de las tuplas modificadas

Nombre: Resultados		
Descripción: Datos relacionados con el libro de resultados		
Atributo	Tipo	Descripción
+Resultado_ID	int	Valor entero auto incremental. Llave primaria
Folios	int	Número de folio del libro de resultado
Pag_libro_result	int	Número de página del libro de resultados
# Dc2_Sic_datos_2_ID_199	int	Llave foránea que la toma de la entidad SIC_0199_GELMicrobiano

Nombre: Resultado_obtenido		
Descripción: Resultados obtenidos para conteo total de microorganismos viables		
Atributo	Tipo	Descripción
+Resultado_obtenido_ID	int	Valor entero auto incremental. Llave primaria
Muestra	varchar	Nombre de la muestra
#N_metodo_empleado_ID	int	Llave foránea que la toma de la entidad N_metodo_empleado
#Dc2_Sic_datos_2_ID_199	int	Llave foránea que la toma de la entidad SIC_0199_GELMicrobiano

Nombre: Resultado_pv_ss_fm		
Descripción: Resultados por placa vertida, siembra en superficie y filtración por membrana.		
Atributo	Tipo	Descripción
+Resultado_pv_ss_fm_ID	int	Valor entero auto incremental. Llave primaria

Capítulo 2: Descripción y análisis de la solución propuesta

+#Resultado_obtenido_pv_ID	int	Llave primaria foránea que la toma de la entidad Resultado_obtenido
Ctmv_goml	int	Valor c.t.m.v/g ó mL
Ctml_gomL	int	Valor c.t.m.l/g ó mL

Nombre: Resultado_tm		
Descripción: Resultados por tubos múltiples.		
Atributo	Tipo	Descripción
+Resultado_tm_ID	int	Valor entero auto incremental. Llave primaria
+#Resultado_obtenido_tm_ID	int	Llave primaria foránea que la toma de la entidad Resultado_obtenido
Comb_tubos_crec	float	Valor de combinación de tubos con crecimiento
Dilucion_central	float	Valor dilución central
Nmp_de_m	int	Valor NMP de m.o viables/g ó mL

Nombre: Resultados_1		
Descripción: Datos relacionados con el libro de resultados.		
Atributo	Tipo	Descripción
+#Resultado_1_ID	int	Llave primaria foránea que la toma de la entidad Resultado
Libro_result_no	int	Número de libro de resultados
#Dc1_Sic_datos_ID_159	int	Llave foránea que la toma de la entidad SIC_0159_GCAMPExpuesta
#Dc2_Sic_datos_2_ID_162	int	Llave foránea que la toma de la entidad SIC_0162_GCHIViabilidad
#Dc2_Sic_datos_2_ID_711	int	Llave foránea que la toma de la entidad SIC_0711_GCHViabilidad
#Sdc_Sic_datos_ID_807	int	Llave foránea que la toma de la entidad SIC_0807_GCMAAmbiental
#Dc1_Sic_datos_ID_805	int	Llave foránea que la toma de la entidad SIC_0805_GCMSuperficies
#Dc1_Sic_datos_ID_714	int	Llave foránea que la toma de la entidad SIC_0714_GCMMEnguantadas

Nombre: Resultados_2		
Descripción: Datos relacionados con el libro de resultados.		
Atributo	Tipo	Descripción
+#Resultado_2_ID	int	Llave primaria foránea que la toma de la entidad Resultado
Lote_medio_cultivo	varchar	Lote de medio de cultivo
#Dc1_Sic_datos_ID_142	int	Llave foránea que la toma de la entidad SIC_0142_GCMAguas
#Dc2_Sic_datos_2_ID_198	int	Llave foránea que la toma de la entidad SIC_0198_GEEPAntimicrobianos

Capítulo 2: Descripción y análisis de la solución propuesta

Nombre: SIC_0020_GEPSoluciones		
Descripción: Planilla SIC-0020 ensayo de preparación de soluciones.		
Atributo	Tipo	Descripción
+#Sdc_Sic_datos_ID	int	Llave primaria foránea que la toma de la entidad Sic_datos_comunes
Folio	int	Valor secuencial que se repite desde 1 hasta 999.
No_lote	varchar	Número de lote
No_parte	int	Número de parte
Vt	varchar	Volumen total de la muestra
Fecha_preparacion_d	int	Día de preparación
Fecha_preparacion_m	int	Mes de preparación
Fecha_preparacion_a	int	Año de preparación
Nombre_solucion	varchar	Nombre de solución
Solvente	varchar	Valor del solvente
PH	float	Valor PH
Conductividad	varchar	Conductividad
Ajuste_pH_sol	bool	Cumple o no
PH_deseado	float	Valor PH deseado
Sol_ajuste_pH	varchar	Solución pH
Np	int	Valor número de parte
No_lote	varchar	Número de lote
PH_final_sol	float	PH final
Esterilizacion	bool	Cumple o no
Vapor_saturado	int	Vapor saturado
Tiempo	int	Tiempo
No_frascos_almacen	int	Número de frascos
Volumen_frasco	varchar	Volumen obtenido por frasco
Almacenar_sol	varchar	Almacenar la solución
Fecha_vencimiento_d	int	Día de vencimiento
Fecha_vencimiento_m	int	Mes de vencimiento
Fecha_vencimiento_a	int	Año de vencimiento
Recibido_por	varchar	Recibido por
Fecha_recibido_d	int	Día de recibido
Fecha_recibido_m	int	Mes de recibido
Fecha_recibido_a	int	Año de recibido
Limite_prom_crecimiento	int	Límite de promoción del crecimiento
Limite_esterilidad	int	Límite de esterilidad
V_obt_prom_crecimiento	int	Valor obtenido de promoción del crecimiento
V_obt_esterilidad	int	Valor obtenido de esterilidad
#Filtracion_ID	int	Llave foránea que la toma de la entidad N_filtracion

Nombre: SIC_0102_GEEsterilidad		
Descripción: Planilla SIC-0102 ensayo de esterilidad.		
Atributo	Tipo	Descripción

Capítulo 2: Descripción y análisis de la solución propuesta

+#Dc2_Sic_datos_2_ID	int	Llave primaria foránea que la toma de la entidad Datos_comunes_2
Tec_siembra_directa	bool	Técnica de siembra empleada.
Fecha_siembra_d	int	Día de siembra
Fecha_siembra_m	int	Mes de siembra
Fecha_siembra_a	int	Año de siembra
Tipo_material	varchar	Tipo de material
Lote_materiales	varchar	Lote de los materiales
Resultado_materiales	bool	Resultado de los materiales
LibT_ambiente	int	Número del libro de trabajo para el objeto de muestreo ambiente
LibT_me	int	Número del libro de trabajo para el objeto de muestreo manos enguantadas
LibT_superficie	int	Número del libro de trabajo para el objeto de muestreo superficies
Pag_ambiente	int	Páginas del libro de trabajo para el objeto de muestreo ambiente
Pag_me	int	Páginas del libro de trabajo para el objeto de muestreo manos enguantadas
Pag_superficie	int	Páginas del libro de trabajo para el objeto de muestreo superficies
Granel	bool	Cumple o no
Volumen_muestra	varchar	Volumen de muestra
Volumen_inoculado	varchar	Valor volumen inoculado/medio
No_frascos	int	Número de frascos
Vol_frasco	varchar	Volumen del frasco
Vol_inoculado_fm	varchar	Volumen del inoculado
Medio_tioglicolato_3	bool	3 días de crecimiento del medio de fluido de tioglicolato
Medio_tioglicolato_5	bool	5 días de crecimiento del medio de fluido de tioglicolato
Medio_tioglicolato_8	bool	8 días de crecimiento del medio de fluido de tioglicolato
Medio_tioglicolato_11	bool	11 días de crecimiento del medio de fluido de tioglicolato
Medio_tioglicolato_14	bool	14 días de crecimiento del medio de fluido de tioglicolato
Caldo_soya_3	bool	3 días de crecimiento del medio caldo triptona soya
Caldo_soya_5	bool	5 días de crecimiento del medio caldo triptona soya
Caldo_soya_8	bool	8 días de crecimiento del medio caldo triptona soya
Caldo_soya_11	bool	11 días de crecimiento del medio caldo triptona soya
Caldo_soya_14	bool	14 días de crecimiento del medio caldo triptona soya

Capítulo 2: Descripción y análisis de la solución propuesta

Em_satif	bool	Cumple o no, esterilidad de los materiales
Emc_satif	bool	Cumple o no, esterilidad de los medios de cultivos
Pcmc_satif	bool	Cumple o no, promoción del crecimiento de los medios de cultivo
Caae_satif	bool	Cumple o no, control ambiental del área de ensayo
Microorg_primer_ensayo	varchar	Primer ensayo de microorganismos aislados
Microorg_segundo_ensayo	varchar	Segundo ensayo de microorganismos aislados
Invalida_ens_pasa_rep	bool	Se invalida el ensayo y pasa a repetición
# Le_no_esterilidad_ID	varchar	Llave foránea que la toma de la entidad LE_Esterilidad
# LE_EsterilidadNo_lote	varchar	Llave foránea que la toma de la entidad LE_Esterilidad
# Lt_lib_trab_ID	int	Llave foránea que la toma de la entidad Libro_de_trabajo

Nombre: SIC_0142_GCMAguas		
Descripción: Planilla SIC-0142 ensayo en sistemas auxiliares para el control microbiológico de las aguas.		
Atributo	Tipo	Descripción
+#Dc1_Sic_datos_ID	int	Llave primaria foránea que la toma de la entidad Datos_comunes_1
Folio	int	Valor secuencial que se repite desde 1 hasta 999.
Lote_proceso	varchar	Lotes en proceso
F_siembra_d	int	Día de siembra
F_siembra_m	int	Mes de siembra
F_siembra_a	int	Año de siembra
#N_produccion_ID	int	Llave foránea que la toma de la entidad N_procedencia

Nombre: SIC_0159_GCAMPEpuesta		
Descripción: Planilla SIC-0159 programa de monitoreo ambiental para el control ambiental mediante placa expuesta.		
Atributo	Tipo	Descripción
+#Dc1_Sic_datos_ID	int	Llave primaria foránea que la toma de la entidad Datos_comunes_1
Folio	int	Valor secuencial que se repite desde 1 hasta 999.
Lotes_proceso	varchar	Lotes en proceso
Fecha_muestreo_d	int	Día de muestreo
Fecha_muestreo_m	int	Mes de muestreo

Capítulo 2: Descripción y análisis de la solución propuesta

Fecha_muestreo_a	int	Año de muestreo
Hora	varchar	Hora
No_control	int	Número de control
Cond_muestreo	varchar	Condiciones de muestreo
Cant_personas	int	Cantidad de personas
Limpieza_muestreo	bool	Cumple o no, limpieza del muestreo
Higienizante_utilizado	varchar	Higienizantes utilizados
Observ_microscopico	varchar	Observaciones microscópico
#Area_ID	int	Llave foránea que la toma de la entidad N_area
#Produccion_ID	int	Llave foránea que la toma de la entidad N_produccion
#Higt_ID_159		Llave foránea que la toma de la entidad N_higienizantes

Nombre: SIC_0162_GCHIVIabilidad		
Descripción: Planilla SIC-0162 ensayo de viabilidad para el chequeo inicial de viabilidad a los bancos de células.		
Atributo	Tipo	Descripción
+#Dc2_Sic_datos_2_ID	int	Llave primaria foránea que la toma de la entidad Datos_comunes_2
Cepa	varchar	Microorganismo utilizado
No_entrada_lab	int	Número de entrada al laboratorio
Banco_homogeneo	bool	Cumple o no, banco homogéneo
#LE_Muestramuestra	varchar	Llave foránea que la toma de la entidad LE_Muestra
#LE_MuestraN_tipo_ensayo_ID	int	Llave foránea que la toma de la entidad LE_Muestra

Nombre: SIC_0198_GEEPAntimicrobianos		
Descripción: Planilla SIC-0198 ensayo de efectividad de preserva antimicrobianos.		
Atributo	Tipo	Descripción
+#Dc2_Sic_datos_2_ID	int	Llave primaria foránea que la toma de la entidad Datos_comunes_2
Preservo	varchar	Preservo
Concentracion	int	Concentración
Cond_muestra	varchar	Condiciones de la muestra
Tiempo_almacenamiento	varchar	Tiempo de almacenamiento
Temperatura	float	Temperatura
No_entrada_lab	int	Número de entrada al laboratorio
Fecha_comienzo_ensayo_d	int	Día de comienzo del ensayo
Fecha_comienzo_ensayo_m	int	Mes de comienzo del ensayo
Fecha_comienzo_ensayo_a	int	Año de comienzo del ensayo
Tamano_muestra	int	Tamaño de la muestra
Concent_medio_cultivo	float	Concentración
E_coli_ci	varchar	Microorganismos de prueba empleado
Ps_aerug_ci	varchar	Microorganismos de prueba empleado

Capítulo 2: Descripción y análisis de la solución propuesta

St_ci	varchar	Microorganismos de prueba empleado
C_alb_ci	varchar	Microorganismos de prueba empleado
A_niger_ci	varchar	Microorganismos de prueba empleado
E_coli_volm	varchar	Microorganismos de prueba empleado
Ps_aerug_volm	varchar	Microorganismos de prueba empleado
St_volm	varchar	Microorganismos de prueba empleado
C_alb_volm	varchar	Microorganismos de prueba empleado
A_niger_volm	varchar	Microorganismos de prueba empleado
Ufc_e_coli	varchar	Valor UFC/mL en control Microorganismos de prueba empleado
Ufc_ps_aerug	varchar	Valor UFC/mL en control Microorganismos de prueba empleado
Ufc_st	varchar	Valor UFC/mL en control Microorganismos de prueba empleado
Ufc_c_alb	varchar	Valor UFC/mL en control Microorganismos de prueba empleado
Ufc_a_nige	varchar	Valor UFC/mL en control Microorganismos de prueba empleado
E_coli_0	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
E_coli_7	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
E_coli_14	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
E_coli_28	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
Ps_aerug_0	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
Ps_aerug_7	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
Ps_aerug_14	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
Ps_aerug_28	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
St_0	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
St_7	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
St_14	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
St_28	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
C_alb_0	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
C_alb_7	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
C_alb_14	int	Valor UFC/mL en la muestra de

Capítulo 2: Descripción y análisis de la solución propuesta

		Microorganismos de prueba empleado
C_alb_28	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
A_niger_0	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
A_niger_7	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
A_niger_14	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
A_niger_28	int	Valor UFC/mL en la muestra de Microorganismos de prueba empleado
#Neutralizante_ID	int	Llave foránea que la toma de la entidad N_neutralizante
#LE_Muestramuestra	varchar	Llave foránea que la toma de la entidad LE_Muestra
#LE_MuestraN_tipo_ensayo_ID	int	Llave foránea que la toma de la entidad LE_Muestra
#Cond_muestra_ID	int	Llave foránea que la toma de la entidad N_muestras

Nombre: SIC_0199_GELMicrobiano		
Descripción: Planilla SIC-0199 ensayo límite microbiano.		
Atributo	Tipo	Descripción
+#Dc2_Sic_datos_2_ID	int	Llave primaria foránea que la toma de la entidad Datos_comunes_2
No_control	int	Numero de control
Fecha_ensayo_d	int	Día del ensayo
Fecha_ensayo_m	int	Mes del ensayo
Fecha_ensayo_a	int	Año del ensayo
Conteo_microorg	int	Conteo total de microorganismos viables
Conteo_mohos_levaduras	int	Conteo total de mohos y levaduras
Ausencia_de	varchar	Ausencia
#LE_Muestramuestra	varchar	Llave foránea que la toma de la entidad LE_Muestra
#LE_MuestraN_tipo_ensayo_ID	int	Llave foránea que la toma de la entidad LE_Muestra

Nombre: SIC_0711_GCHViabilidad		
Descripción: Planilla SIC-0711 ensayo de viabilidad para el chequeo de viabilidad a los bancos de células.		
Atributo	Tipo	Descripción
+#Dc2_Sic_datos_2_ID	int	Llave primaria foránea que la toma de la entidad Datos_comunes_2
Cepa	varchar	Microorganismo utilizado
No_entrada_lab	int	Número de entrada al laboratorio
Volumen_inoculo	varchar	Volumen de inóculo en placa
Diluciones_sembradas	float	Diluciones sembradas

Capítulo 2: Descripción y análisis de la solución propuesta

Concent_banco	varchar	Concentración del banco
Concent_banco_an	varchar	Concentración del banco anterior
Alerta	bool	Cumple o no
#LE_Muestramuestra	varchar	Llave foránea que la toma de la entidad LE_Muestra
#LE_MuestraN_tipo_ensayo_ID	int	Llave foránea que la toma de la entidad LE_Muestra

Nombre: SIC_0714_GCMMEnguantadas		
Descripción: Planilla SIC-0159 programa de monitoreo ambiental para el control microbiológico de las manos enguantadas.		
Atributo	Tipo	Descripción
+#Dc1_Sic_datos_ID	int	Llave primaria foránea que la toma de la entidad Datos_comunes_1
Folio	int	Valor secuencial que se repite desde 1 hasta 999.
Lote_proceso	varchar	Lote en proceso
Producción	varchar	Producción
Fecha_muestreo_d	int	Día del muestreo
Fecha_muestreo_m	int	Mes del muestreo
Fecha_muestreo_a	int	Año del muestreo
Numero_control	int	Número de control
#Area_ID	int	Llave foránea que la toma de la entidad N_area
#Produccion_ID	int	Llave foránea que la toma de la entidad N_produccion

Nombre: SIC_0719_GCEProceso		
Descripción: Planilla SIC-0719 ensayo control de proceso.		
Atributo	Tipo	Descripción
+Datos_muestra_id	int	Valor entero auto incremental. Llave primaria
+#Sdc_Sic_datos_ID_719	int	Llave primaria foránea que la toma de la entidad Sic_datos_comunes
Results_719	varchar	Resultados de la muestra
#Codigo_719	varchar	Llave foránea que la toma de la entidad LE_Muestra_0719
#Produccion_ID	int	Llave foránea que la toma de la entidad N_produccion

Nombre: SIC_0805_GCMSuperficies		
Descripción: Planilla SIC-0805 programa de monitoreo ambiental para el control de muestreo de superficies.		
Atributo	Tipo	Descripción
+#Dc1_Sic_datos_ID	int	Llave primaria foránea que la toma de la entidad Datos_comunes_1
Folio	int	Valor secuencial que se repite desde 1

Capítulo 2: Descripción y análisis de la solución propuesta

		hasta 999.
Metodo_utilizado	varchar	Método utilizado
Objeto_muestreo	varchar	Objeto de muestreo
Fecha_muestreo_d	int	Día del muestreo
Fecha_muestreo_m	int	Mes del muestreo
Fecha_muestreo_a	int	Año del muestreo
#Area_ID	int	Llave foránea que la toma de la entidad N_area
#Produccion_ID	int	Llave foránea que la toma de la entidad N_produccion

Nombre: SIC_0806_GCMVPuro		
Descripción: : Planilla SIC-0806 ensayo en sistemas auxiliares para el control microbiológico de vapor puro.		
Atributo	Tipo	Descripción
+#Dc1_Sic_datos_ID	int	Llave primaria foránea que la toma de la entidad Datos_comunes_1
Folio	int	Valor secuencial que se repite desde 1 hasta 999.
Lote	varchar	Número de lote
F_muestreo_d	int	Día del muestreo
F_muestreo_m	int	Mes del muestreo
F_muestreo_a	int	Año del muestreo
F_vencimiento_d	int	Día de vencimiento
F_vencimiento_m	int	Mes de vencimiento
F_vencimiento_a	int	Año de vencimiento
Result_pag	int	Páginas del libro de resultado
Result_folio	int	Folio del libro de resultado
#Produccion_ID	int	Llave foránea que la toma de la entidad N_produccion

Nombre: SIC_0807_GCMAAmbiental		
Descripción: Planilla SIC-0807 programa de monitoreo ambiental para el control de muestreo con el analizador ambiente.		
Atributo	Tipo	Descripción
+#Sdc_Sic_datos_ID	int	Llave primaria foránea que la toma de la entidad Sic_datos_comunes
Folio	int	Valor secuencial que se repite desde 1 hasta 999.
Fecha_muestreo_d	int	Día del muestreo
Fecha_muestreo_m	int	Mes del muestreo
Fecha_muestreo_a	int	Año del muestreo
Lotes_proceso	varchar	Nombre de lotes en proceso
Medio_cultivo	varchar	Medio de cultivo
Np	int	Número de parte
Lote	varchar	Lote
Personas_area	int	Personas del área

Capítulo 2: Descripción y análisis de la solución propuesta

Limpieza_muestreo	bool	Cumple o no
#Higt_ID	int	Llave foránea que la toma de la entidad N_higienizantes
#Area_ID	int	Llave foránea que la toma de la entidad N_area
#Produccion_ID	int	Llave foránea que la toma de la entidad N_produccion

Nombre: Sic_datos_comunes		
Descripción: En esta tabla se agrupan atributos comunes de los SIC 807, 719 y 0020		
Atributo	Tipo	Descripción
+#Sic_datos_1_ID	int	Valor entero auto incremental. Llave primaria
Observaciones	varchar	Observaciones
Realizado_por	varchar	Realizado por
Fecha_realizado_d	int	Día de realizado
Fecha_realizado_m	int	Mes de realizado
Fecha_realizado_a	int	Año de realizado
Terminado	bool	Terminado

Nombre: Soluciones		
Descripción: Recoge los datos de las soluciones de los medios de cultivos		
Atributo	Tipo	Descripción
+Soluciones_ID	int	Valor entero auto incremental. Llave primaria
Solucion	varchar	Soluciones
No_parte	int	Número de parte de la solución
No_lote	varchar	Número de lote de la solución
Fecha_vencimiento_d	int	Día de vencimiento
Fecha_vencimiento_m	int	Mes de vencimiento
Fecha_vencimiento_a	int	Año de vencimiento
#Dc2_Sic_datos_2_ID_711	int	Llave foránea que la toma de la entidad SIC_0711_GCHViabilidad
#Dc1_Sic_datos_ID_805	int	Llave foránea que la toma de la entidad SIC_0805_GCMSuperficies
#Dc2_Sic_datos_2_ID_199	int	Llave foránea que la toma de la entidad SIC_0199_GELMicrobiano

Nombre: Soluciones_medio_cultivo_1		
Descripción: Recoge los datos de las soluciones o medios de cultivos.		
Atributo	Tipo	Descripción
+Sol_mc_1_ID	int	Valor entero auto incremental. Llave primaria
Sol_medio_cultivo	varchar	Soluciones del medio de cultivo
No_parte	int	Número de parte del medio de cultivo
No_lote	varchar	Número de lote del medio de cultivo
Fecha_vencimiento_d	int	Día de vencimiento

Capítulo 2: Descripción y análisis de la solución propuesta

Fecha_vencimiento_m	int	Mes de vencimiento
Fecha_vencimiento_a	int	Año de vencimiento
#Dc1_Sic_datos_ID_142	int	Llave foránea que la toma de la entidad SIC_0142_GCMaguas
#Dc2_Sic_datos_2_ID_162	int	Llave foránea que la toma de la entidad SIC_0162_GCHIViabilidad

Nombre: Soluciones_medio_cultivo_2		
Descripción: Recoge los datos de las soluciones o medios de cultivos.		
Atributo	Tipo	Descripción
+#Sol_mc_1_ID_smc1	int	Llave primaria foránea que la toma de la entidad Soluciones_medio_cultivo_1
+#Dc2_Sic_datos_2_ID_198	int	Llave primaria foránea que la toma de la entidad SIC_0198_GEEPAntimicrobianos
Fecha_uso_d	int	Día de uso
Fecha_uso_m	int	Mes de uso
Fecha_uso_a	int	Año de uso

Nombre: Unidades_formadoras_crp		
Descripción: Recoge los datos de las unidades formadoras de colonias y réplicas.		
Atributo	Tipo	Descripción
+Unidades_cr_ID	int	Valor entero auto incremental. Llave primaria
Bulbo_no	int	Nombre de los bulbos
x1	int	Valor x1
x2	int	Valor x2
x3	int	Valor x3
x4	int	Valor x4
DIL	float	Valor Dilución
#Dc2_Sic_datos_2_ID_162	int	Llave foránea que la toma de la entidad SIC_0162_GCHIViabilidad
#Dc2_Sic_datos_2_ID_711	int	Llave foránea que la toma de la entidad SIC_0711_GCHViabilidad

2.10 Conclusiones.

En este capítulo se precisó la estrategia a seguir para la integración de la solución con los otros módulos.

Se ha representado el diagrama de clases persistentes a partir de las clases del diseño, identificándose un total de 44 clases persistentes relacionadas con sus atributos y métodos, realizándose una descripción para cada una.

Además se ha mostrado el modelo físico de datos de la base de datos para el Laboratorio Microbiología, en el cual se definieron 55 entidades y posteriormente se explican cada una de las entidades y sus atributos.

CAPÍTULO 3: VALIDACIÓN DEL DISEÑO REALIZADO.

En el presente capítulo se muestra la validación teórica y funcional realizada al diseño descrito en el capítulo anterior. Se muestran las reglas a tener en cuenta para un diseño de óptima calidad y se describen los detalles para probar la funcionalidad de la BD a través de la selección de consultas y herramientas de generación de datos de pruebas para un llenado voluminoso de la BD.

3.1 Validación teórica del diseño.

Para obtener un buen diseño de la BD se debe tener en cuenta algunos aspectos importantes que no deben ser violados, tales como: la integridad de los datos, la normalización del diseño, el análisis de la seguridad permitiendo controlar el acceso a los datos y que los mismos no se dañen como resultado de acciones no controladas y la trazabilidad de las acciones que se realizan en la BD.

3.1.1 Integridad.

La integridad de la BD se refiere a la validez, corrección y la consistencia de los datos almacenados en una BD. Una de las funciones más importantes de un sistema de base de datos relacional es preservar la integridad de la información almacenada en la mayor medida posible. La integridad se expresa mediante restricciones o reglas que no se pueden violar y es el SGBD quien se debe encargar de mantenerlas.

Para lograr la integridad de la BD fueron aplicadas las restricciones de integridad siguientes para garantizar que los datos fuesen válidos.

Integridad de dominio: Todos los valores no nulos que contiene la base de datos para un determinado atributo deben pertenecer al dominio declarado para dicho atributo, que permite restringir los valores de los atributos.

Ejemplo de este caso en la BD se definió la fecha por los campos: día, mes y año, en el que los días tienen que ser > 0 y < 32 , donde se restringe su dominio en int2, en el caso del mes sucede algo parecido el cual puede tomar valores entre 1 y 12 con un dominio int2, y en el caso del año que su dominio sería int4 entre 1000 y 9999, rango que aunque es amplio, valida cualquier fecha permisible de la realidad. El atributo folio es un entero autoincremental que sólo acepta valores desde 1 hasta 999 volviendo a empezar desde uno. Otro ejemplo sería la cantidad de caracteres válidos para una cadena, el atributo no_lote es un tipo de dato varchar de longitud 15 y tiene un dominio varchar(n) (15) que solamente puede aceptar letras y números en cualquier tabla que aparezca.

Para realizar estas validaciones PostgreSQL presenta:

Las restricciones (constraints) se aplican a columnas o tuplas de una tabla, algunos tipos especiales de estos son: unique, not null, references, check; los constraints actúan cuando se viola lo que ellos indican, pero el usuario no puede especificar acción alguna con esa violación, sino que el DBMS simplemente envía un mensaje de error e impide que se realice la operación que produjo la violación.

```
CONSTRAINT "sic_0714_gcmmenguantadas_dia" CHECK ((fecha_muestreo_d > 0) AND
(fecha_muestreo_d < 32).
```

```
CONSTRAINT "sic_0714_gcmmenguantadas_mes" CHECK ((fecha_muestreo_m > 0) AND
(fecha_muestreo_m < 13).
```

```
CONSTRAINT "sic_0714_gcmmenguantadas_anno" CHECK ((fecha_muestreo_m > 1000) AND
(fecha_muestreo_a < 9999).
```

```
CONSTRAINT "FOLIO" CHECK ((folio > 0) AND (folio < 1000)
```

```
CONSTRAINT "lote" CHECK ((lote)::text ~ similar_escape('([a-zA-Z][0-9])'*)::text, NULL::text)
```

Integridad referencial: En una relación todos los valores que toma una clave foránea deben ser nulos o deben coincidir con el valor de la clave primaria a la que hace referencia y garantiza que los valores de clave sean coherentes en las distintas tablas. Por ejemplo, en las tablas equipos_utilizados y sic_0020_GEPSoluciones de la base de datos del laboratorio de microbiología, la integridad referencial se basa en la relación entre la clave foránea (sic_20_ID) de la tabla equipos_utilizados y la clave primaria (sic_20_ID) de la tabla sic_0020_GEPSoluciones. Esta relación garantiza que los Equipos no puedan nunca hacer referencia a la tabla sic_0020_GEPSoluciones sin antes haber validado la existencia del SIC-0020.

PostgreSQL valida que se cumpla la integridad referencial, en la BD se utilizan los disparadores para el tratamiento de errores e insertar datos validos. En el siguiente ejemplo se utiliza un disparador que permite validar la existencia previa de la llave foránea sic_20_ID de la tabla equipos_utilizados:

```
declare
```

```
sic sic_0020_GEPSoluciones%rowtype;
```

```
BEGIN
```

```
SELECT into sic * from sic_0020_GEPSoluciones where sic_0020_GEPSoluciones.sic_20_ID = new.
sic_20_ID;
```

```
if not found THEN
```

```
raise exception 'Error de llave';  
END IF;  
RETURN NEW;  
END;
```

Integridad de entidad: La llave primaria de una tabla debe contener un valor único por cada tupla de la tabla, sino la base de datos perdería su integridad. En la BD de microbiología cada entidad tiene definida una llave primaria, que no admite valores nulos y siempre es única.

Chequeo de validez: Cuando se crea una tabla cada columna tiene un tipo de dato y el SGBD asegura que solamente los datos del tipo especificado sean ingresados en la tabla.

Datos requeridos: Establece que una columna que tenga un valor no nulo. Se define al efectuar la declaración de una columna como NOT NULL cuando la tabla que contiene la columna se crea por primera vez. Un ejemplo de este tipo sería el campo folio es un dato obligatorio y se debe declarar como NOT NULL.

3.1.2 Normalización de la Base de Datos.

La normalización se define como una técnica de análisis utilizado para producir un correcto diseño de bases de datos relacionales, está creada alrededor del concepto de formas normales, basada en una serie de reglas y cada una está basada en la que le antecede que sirven para ayudar a los diseñadores de bases de datos a minimizar los problemas de lógica.

Es un procedimiento para eliminar la redundancia en el modelo de datos en las tablas y ayuda a cumplir la integridad referencial de los datos en la base de datos. Esta se realiza sobre el modelo de datos después de una primera versión de las tablas. Con la realización de un buen diseño de la BD se logró minimizar la redundancia de los datos, se elimina la duplicidad de datos y la incorporación de atributos calculables, la cual genera inconsistencia en la BD, en grandes bases de datos no se puede afirmar que la redundancia es eliminada completamente que es lo deseado para permitir un fácil acceso a la información.

La BD presenta poca redundancia por ejemplo, se analizaron los SIC y se agruparon todos los datos comunes en tres tablas Sic_datos_comunes, Datos_comunes_1, Datos_comunes_2 que se relacionan entre si para evitar la repetición.

Para un buen diseño de una BD se debe cumplir con las siguientes formas normales:

- ✓ Primera Forma Normal(1FN)

- ✓ Segunda Forma Normal(2FN)
- ✓ Tercera Forma Normal(3FN)

Además de estas reglas para la normalización existen otras como: Forma Normal Boyce-Codd, Cuarta Forma Normal, Quinta Forma Normal o Forma Normal de Proyección-Unión, Forma Normal de Proyección-Unión Fuerte, Forma Normal de Proyección-Unión Extra Fuerte y Forma Normal de Clave de Dominio.

Primera Forma Normal: Prohíbe los atributos multievaluados y compuestos, solo admiten atributos con valores atómicos, cada tupla contiene un solo valor para cada atributo.

Se puede afirmar que el esquema relacional de la BD está en 1FN ya que cada tupla contiene exactamente un valor para cada atributo de las tablas de la BD, es decir, no existen atributos de sus tablas que tomen en algún momento un conjunto de valores.

Segunda Forma Normal: Debe estar en 1FN y todos los atributos no llaves son completamente dependientes de la llave primaria.

También se cumple con la 2FN porque primeramente se encuentran en 1FN, y además todos los atributos no llaves en las tablas, dependen totalmente de los atributos que componen la llave primaria y no solo de una parte de ella.

Tercera Forma Normal: La tabla se encuentra en 3FN si es está en 2FN y cada atributo que no forma parte de ninguna llave, depende directamente y no transitivamente, de la llave primaria.

La BD para el Laboratorio de Microbiología ha sido normalizada hasta la 3FN, resolviéndose de esta manera los problemas de inconsistencia y redundancia en sus tablas. No se cumplió con las demás reglas de normalización ya que mientras más reglas se cumplan aumenta el grado de normalización del esquema relacional de la BD obteniéndose un esquema mucho más complejo, y esto dificultaría el acceso a los datos desde el punto de vista orientado a objetos que desarrolla Symfony, en la capa ORM se generarían clases que no representan entidades significativas del negocio lo que dificulta el acceso a los datos.

3.1.3 Análisis de la seguridad de la Base de Datos.

La información almacenada en una BD está expuesta a sufrir múltiples riesgos de seguridad debido a los distintos ataques informáticos, el acceso no autorizado para borrar, modificar y propagar información de gran importancia, la pérdida de integridad, accesibilidad y la confidencialidad en los datos. El administrador de BD es el máximo responsable de controlar y garantizar la seguridad global

del sistema de BD, tienen la obligación de otorgar o revocar privilegios a cuentas individuales, usuarios o grupo de usuarios y efectuar los siguientes tipos de acciones como la creación de cuentas y asignación de niveles de seguridad, así como realizar salvadas para proteger los datos almacenados en la BD.

PostgreSQL presenta una serie de características que nos facilita el trabajo para tener un alto nivel de seguridad. Primeramente se deben controlar los permisos a los usuarios lo cual se conforma en PostgreSQL y utiliza tres niveles de acceso:

El nivel 0 es el que se encarga sobre las máquinas (host) y los usuarios. Es decir es el que permite configurar que máquina/s y/o usuario/s se pueden conectar a la BD y el modo en que pueden conectarse, utilizan las opciones del fichero de configuración `pg_hba.conf`.

El nivel 1 es el que se encarga de los usuarios y las BDs, permite configurar a cuál BD se pueden conectar determinados usuarios al emplear las opciones del fichero de configuración `pg_ident.conf`.

El nivel 2 es el que se encarga de las tablas, permite configurar a cuáles tablas pueden acceder, al utilizar los comandos GRANT para dar permisos y REVOKE para denegar los permisos.

Existe un módulo que se encargará de la integración de las bases de datos de los módulos y de la seguridad de la BD general. Para garantizar la seguridad en la BD propuesta para el LM se establecen controles de seguridad para los datos, solo pueden acceder los usuarios autorizados y garantiza el acceso y la realización de operaciones sobre algunas tablas o toda la BD según el rol que desempeñan, para dar cumplimiento a los RF se denegó el permiso de eliminar datos en todas las tablas. Se configuró los permisos de conexión para los host y los usuarios puedan acceder a la BD desde un rango de direcciones IP determinada.

Para garantizar que no exista pérdida de información en caso de que ocurra algún error en el servidor de la BD o alguna falla de energía, deben realizarse copias de seguridad de la BD regularmente. Para la realización de salvadas a la BD PostgreSQL utiliza la forma de volcado (dump) que cuenta con un comando denominado `pg_dump` que se encarga de hacerle un backup a la BD por ejemplo `pg_dump -U (usuario administrador) (nombre BD) > (nombre del backup)`, luego por medio del comando `psql` se restauran las salvadas por ejemplo `psql -U (usuario administrador) (nombre BD) < (nombre del backup)`. Estos métodos tienen ventajas como la compatibilidad entre versiones, se puede hacer con el servidor corriendo y se puede realizar a una o varias BD.

3.1.4 Trazabilidad de las acciones.

La trazabilidad es un conjunto de medidas, acciones y procedimientos que permiten registrar e identificar cada consulta desde su origen hasta su destino final. Es la capacidad para reconstruir la historia, recorrido o aplicación de una determinada consulta.

Para controlar las acciones realizadas por los usuarios que interactúan con la BD se crearon las tablas registro_trazas y registro_modificacion, en las cuales se almacena el historial de las actividades de los usuarios en todos los momentos, es decir, se recogerán los datos necesarios como momento en que se llevó a cabo, la acción realizada, usuario que inició la acción y el resultado de la misma, o sea, los valores de los datos antes de la actualización y después de la actualización.

Se empleó este método de registrar los eventos para poder conservar los valores que tenían los datos antes de realizar una actualización de la BD, y así dar cumplimiento a los requisitos funcionales relacionados con registros de trazas, pues el framework de desarrollo web Symfony guarda mucha información sobre los eventos realizados en archivos log 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 pero no controla lo relacionado con los valores de los datos antes de una actualización.

3.1.5 Disparadores.

Los disparadores (triggers) son parecidos a los constraints, pero proveen mayor flexibilidad pues el usuario puede especificar un conjunto de acciones complejas a ejecutarse cuando el trigger se dispare. Un disparador es una operación que se ejecuta automáticamente como consecuencia de la realización de una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE) de los datos en la base de datos.

Para dar cumplimiento a las reglas del negocio relacionadas con la BD del LM se diseñaron los siguientes disparadores:

Regla 1: Registrar trazas de los usuarios después de haber realizado una modificación en cualquier tabla de la BD.

Se diseñó un disparador para guardar el identificador de la traza, nombre del usuario, fecha de modificación, identificador de la tupla modificada, valor viejo del campo, valor nuevo del campo y nombre del campo modificado, garantizando el cumplimiento de los requisitos funcionales.

```
CREATE TRIGGER "sic_0162_gchiviabilidad_tr" AFTER UPDATE
```

ON "public"."sic_0162_gchiviabilidad" FOR EACH ROW

EXECUTE PROCEDURE "public"."Traza162"();

La función Traza162 sería:

declare

id integer;

BEGIN

LOCK TABLE registro_trazas IN EXCLUSIVE MODE;

IF((OLD.dc2_sic_datos_d_id != NEW.dc2_sic_datos_d_id) OR (OLD.cepa != NEW.cepa) OR (OLD.no_entrada_lab != NEW.no_entrada_lab) OR (OLD.banco_homogeneo != NEW.banco_homogeneo) OR (OLD.le_muestramuestra != NEW.le_muestramuestra) OR (OLD.le_muestran_tipo_ensayo_id != NEW.le_muestran_tipo_ensayo_id)) THEN

INSERT INTO registro_trazas (usuario, tabla_modificada, fecha_modificacion, id_tupla_modificada) VALUES (CURRENT_USER, TG_RELNAME, CURRENT_TIMESTAMP, OLD.dc2_sic_datos_d_id);

SELECT INTO id MAX (idtraza) FROM registro_trazas;

IF (OLD.cepa != NEW.cepa) THEN

INSERT INTO registro_modificacion (idtraza, valor_nuevo, valor_viejo, campo_modificado) VALUES (id, NEW.cepa, OLD.cepa, 'cepa');

END IF; IF (OLD.no_entrada_lab != NEW.no_entrada_lab) THEN

INSERT INTO registro_modificacion (idtraza, valor_nuevo, valor_viejo, campo_modificado) VALUES (id, NEW.no_entrada_lab, OLD.no_entrada_lab, 'no_entrada_lab');

END IF; IF ((OLD.banco_homogeneo != NEW.banco_homogeneo) AND (NEW.banco_homogeneo != TRUE)) THEN

INSERT INTO registro_modificacion (idtraza, valor_nuevo, valor_viejo, campo_modificado) VALUES (id, 'false', 'true', 'banco_homogeneo');

END IF; IF ((OLD.banco_homogeneo != NEW.banco_homogeneo) AND (NEW.banco_homogeneo != FALSE)) THEN

INSERT INTO registro_modificacion (idtraza, valor_nuevo, valor_viejo, campo_modificado) VALUES (id, 'true', 'false', 'banco_homogeneo');

END IF;

IF (OLD.le_muestramuestra != NEW.le_muestramuestra) THEN

INSERT INTO registro_modificacion (idtraza, valor_nuevo, valor_viejo,campo_modificado) VALUES (id, NEW.le_muestramuestra, OLD.le_muestramuestra, 'le_muestramuestra');

END IF; IF (OLD.le_muestran_tipo_ensayo_id != NEW.le_muestran_tipo_ensayo_id) THEN


```
INSERT INTO registro_modificacion (idtraza, valor_nuevo, valor_viejo, campo_modificado) VALUES
(id,NEW.le_muestran_tipo_ensayo_id,OLD.le_muestran_tipo_ensayo_id,
'le_muestran_tipo_ensayo_id');
END IF; END IF; RETURN NEW; END;
```

Regla 2: Para el caso de uso del sistema Gestionar Ensayo de Esterilidad, enviar los resultados de los ensayos de esterilidad realizados a las muestras enviadas por el Grupo de Liberación Analítica al Laboratorio de Microbiología.

Se diseñó un disparador para la tabla sic_0102_geesterilidad para enviar los siguientes datos: número de lote, conforme o no, identificador del SIC, nombre del analista, técnica utilizada, fecha de recepción y fecha de salida del resultado, a la tabla resultado del Grupo de Liberación Analítica.

```
CREATE TRIGGER "sic_0102_geesterilidad_tr1" AFTER INSERT
ON "public"."sic_0102_geesterilidad" FOR EACH ROW
EXECUTE PROCEDURE "public"."A"();
BEGIN
INSERT INTO resultado (lotee, nombre_analista, tecnica, fecha_recepcion_lab_d,
fecha_recepcion_lab_m, fecha_recepcion_lab_a, fecha_salida_resultado, conformeono, resultadoo)
SELECT le_esterilidad.no_lote, le_esterilidad.recibidor, sic_0102_geesterilidad.tecnica,
le_esterilidad.fecha_recepcion_d, le_esterilidad.fecha_recepcion_m, le_esterilidad.fecha_recepcion_a,
CURRENT_TIMESTAMP, datos_comunes_2.pasa_prueba,
sic_0102_geesterilidad.dc2_sic_datos_2_id FROM sic_0102_geesterilidad JOIN le_esterilidad
ON (NEW."Le_no_esterilidad_ID" = le_esterilidad.no_esterilidad_id)
INNER JOIN public.datos_comunes_2 ON (public.sic_0102_geesterilidad.dc2_sic_datos_2_id =
public.datos_comunes_2.sic_datos_2_id)
WHERE sic_0102_geesterilidad."Le_no_esterilidad_ID" = NEW."Le_no_esterilidad_ID";
RETURN NEW;
END;
```

3.2 Validación funcional.

3.2.1 Generación de datos para el llenado voluminoso de la BD.

Para la realización de pruebas de un llenado voluminoso a la BD, sin violar la integridad de la misma, se utilizó EMS Data Generator 2005 for PostgreSQL, la cual constituye una herramienta poderosa en la generación de datos de pruebas para bases de datos construidas en PostgreSQL. Permite seleccionar tablas para generar datos, define para cada campo el rango de valores admisibles y permite llenar la base de datos de forma eficiente y rápida.

Se realizaron pruebas de carga intensiva a la BD generándole un volumen de información estimado a 5 años de trabajo a las siguientes tablas: le_muestra, le_esterilidad, libro_de_trabajo se generaron 30000 datos, para sic_datos_comunes, datos_comunes_1, datos_comunes_2 se generaron 25000 datos y 20000 datos para sic_0162_gchiviabilidad, sic_0711_gchiviabilidad, sic_0102_geesterilidad, sic_0199_gelmicrobiano y sic_0198_geepantimicrobianos.

Las consultas generadas para la realización de pruebas, se escogen de acuerdo a las operaciones que se esperan sean las más utilizadas una vez implantada la BD. Con el objetivo de validar funcionalmente el diseño de la BD se muestran dos consultas realizadas a la BD y el tiempo de ejecución de las mismas.

Generar informe de ensayos de esterilidad por técnica de siembra directa que hayan pasado la prueba a partir del año 2005.

SELECT

```
public.sic_0102_geesterilidad.tec_siembra_directa,  
public.datos_comunes_2.pasa_prueba,  
public.le_esterilidad.folio,  
public.le_esterilidad.no_lote,  
public.le_esterilidad.muestra_de,  
public.le_esterilidad.fecha_recepcion_d,  
public.le_esterilidad.fecha_recepcion_m,  
public.le_esterilidad.fecha_recepcion_a,  
public.sic_datos_comunes.realizado_por,  
public.sic_datos_comunes.fecha_realizado_a
```

FROM

```
public.sic_0102_geesterilidad
```

INNER JOIN public.le_esterilidad **ON**

(public.sic_0102_geesterilidad."Le_no_esterilidad_ID"=public.le_esterilidad.no_esterilidad_id)

AND (public.sic_0102_geesterilidad.le_no_lote=public.le_esterilidad.no_lote)

INNER JOIN public.datos_comunes_2 **ON**

(public.sic_0102_geesterilidad.dc2_sic_datos_2_id=public.datos_comunes_2.sic_datos_2_id)

INNER JOIN public.datos_comunes_1 **ON**

(public.datos_comunes_2.sic_datos_2_id=public.datos_comunes_1.sic_datos_1_id)

INNER JOIN public.sic_datos_comunes **ON**

(public.datos_comunes_1.sic_datos_1_id=public.sic_datos_comunes.sic_datos_id)

WHERE

(public.datos_comunes_2.pasa_prueba = true) **AND**

(public.sic_0102_geesterilidad.tec_siembra_directa = true) **AND**

(public.sic_datos_comunes.fecha_realizado_a >= 2005)

ORDER BY

public.le_esterilidad.no_lote,

public.le_esterilidad.muestra_de

El resultado de esta consulta fue de 536 tuplas y se realizó en un tiempo de 4,1 segundos antes de ser indexadas las tablas, luego de ser indexados los campos no_lote, folio, muestra_de, fecha_recepcion_d, fecha_recepcion_m, fecha_recepcion_a de la tabla le_esterilidad, pasa_prueba de la tabla datos_comunes_2, fecha_realizado_a, realizado_por de la tabla sic_datos_comunes y tec_siembra_directa de la tabla sic_0102_geesterilidad de acuerdo a los parámetros de búsquedas, disminuyo el tiempo de respuesta a 1,3 segundos, utilizando el tipo de índice B-Tree pues incluye múltiples columnas.

Generar un informe de ensayos de chequeo inicial de viabilidad a los bancos de células que hayan sido repetidos, a partir del año 2005.

SELECT

public.sic_datos_comunes.realizado_por,

public.sic_datos_comunes.fecha_realizado_d,

public.sic_datos_comunes.fecha_realizado_m,

public.sic_datos_comunes.fecha_realizado_a,

public.le_muestra.lote_muestra,

public.le_muestra.muestra_de,

```
public.le_muestra.folio,  
public.sic_0162_gchiviabilidad.cepa,  
public.sic_0162_gchiviabilidad.no_entrada_lab  
FROM  
public.le_muestra  
INNER JOIN public.sic_0162_gchiviabilidad ON  
(public.le_muestra.muestra_de=public.sic_0162_gchiviabilidad.le_muestramuestra) AND  
(public.le_muestra.n_tipo_ensayo_id=public.sic_0162_gchiviabilidad.le_muestran_tipo_ensayo_id)  
INNER JOIN public.datos_comunes_2 ON  
(public.sic_0162_gchiviabilidad.dc2_sic_datos_d_id=public.datos_comunes_2.sic_datos_2_id)  
INNER JOIN public.sic_datos_comunes ON  
(public.sic_datos_comunes.sic_datos_id=public.datos_comunes_2.sic_datos_2_id)  
INNER JOIN public.datos_comunes_1 ON  
(public.datos_comunes_2.sic_datos_2_id=public.datos_comunes_1.sic_datos_1_id)  
WHERE  
(public.datos_comunes_2.repeticion = true) AND  
(public.sic_datos_comunes.fecha_realizado_a >= 2005)
```

El resultado de esta consulta fue de 284 tuplas y se realizó en un tiempo de 2,4 segundos antes de ser indexadas las tablas, luego de ser indexados los campos realizado_por, fecha_realizado_d, fecha_realizado_m, fecha_realizado_a de la tabla sic_datos_comunes, lote_muestra, muestra_de, folio de la tabla le_muestra, cepa, no_entrada_lab de la tabla sic_0162_gchiviabilidad, de acuerdo a los parámetros de búsquedas, disminuyo el tiempo de respuesta a 302 mili segundos, al utilizar el tipo de índice B-Tree pues incluye múltiples columnas.

La generación de datos de pruebas y la consulta realizada a la BD se ejecutaron en un tiempo óptimo de acuerdo a las características que presenta la computadora utilizada (PC Pentium (R) 4 CPU 3.00 GHz, 512 MB de RAM); este tiempo de ejecución se puede optimizar más, teniendo en cuenta las propiedades que presenta el servidor de BD del CIGB.

Cuando se realizan pruebas a una BD se obtienen resultados aproximados a los reales, ya que esta no asegura la ausencia de errores. La mejor prueba a realizar es la interacción directa de los usuarios con la aplicación, la eficiencia de esta depende de los siguientes factores: cantidad de usuarios conectados y el grado de concurrencia de las solicitudes efectuadas. Para desarrollar un buen software es necesario poner en práctica los conceptos de estabilidad, escalabilidad, eficiencia y seguridad; el

mantenimiento del software es fundamental para lograr nuevas versiones del mismo para erradicar los problemas detectados.

Luego de las pruebas realizadas se obtuvo un resultado satisfactorio al dar cumplimiento a los requisitos funcionales propuesto por los analistas.

3.2.2 Análisis de optimización de consultas.

Las consultas son peticiones o métodos para acceder los datos en una BD que permiten gestionar información. En la BD para el LIMS el acceso a los datos se realiza orientado a objetos, Symfony utiliza la capa ORM realizada en Propel que está formada por objetos que permiten acceder a los datos y que contienen en si mismo el código necesario para hacerlo. Propel utiliza Creole como la capa de abstracción que trae como beneficios permitir escribir consultas mediante una sintaxis independiente de la BD traducirlas las al lenguaje SQL concreto de la BD, pudiéndose cambiar fácilmente de una BD a otra. Esta capa obliga a utilizar una sintaxis específica para las consultas y a cambio optimiza y adapta el lenguaje SQL a la BD concreta que se utiliza.

Symfony genera una clase objeto base para cada una de las tablas, contiene una serie de constructores y accesores por defecto en función de la definición de cada columna: los métodos `new()`, `getColumna()` y `setColumna()` permiten crear y obtener las propiedades de los objetos. Los métodos `get` y `set` utilizan una variante del estándar de codificación `camelCase` aplicada al nombre de las columnas. Para guardar los datos en la base de datos, se debe invocar el método `save()` del objeto, si en la BD existe el objeto este método se traduce en una sentencia `UPDATE` o en `INSERT` si no existe el objeto.

Para 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` para definir consultas simples sin utilizar SQL, para conseguir la abstracción de la base de datos.

La invocación del método `doSelect()` es más potente que una consulta SQL debido a la optimización del código SQL, todos los valores que son pasados a `Criteria` se escapan antes de insertarlos en el código SQL para poder evitar los problemas de inyección SQL. El método devuelve un arreglo de objetos y no un result set. La capa ORM crea y rellena de forma automática los objetos en función del result set de la BD, este proceso es conocido 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 al utilizar un paginador.

A veces son necesarias realizar consultas compuestas por campos de varias tablas, donde la cantidad de tuplas a recuperar sea grande, para estos casos es conveniente obtener directamente el resultado y no obtener los objetos porque los tiempos de respuestas serían demasiado altos debido al proceso de hidratación. Luego 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 al posibilitar la 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 para acceder a la BD en Symfony las realizaron los programadores del proyecto, a partir de la capa ORM generada.

3.3 Conclusiones.

Para la validación del diseño se aplican los aspectos fundamentales que conforman y garantizan las cuestiones teóricas y funcionales del mismo:

Se garantizó la integridad de los datos, la normalización del diseño de la BD hasta la tercera forma normal, se establecieron controles de seguridad para el acceso a los datos y se crearon tablas para controlar la trazabilidad de las acciones realizadas por los usuarios que interactúan con la BD.

Se realizó un llenado voluminoso de la BD con la herramienta EMS Data Generator 2005 for PostgreSQL para la realización de pruebas de carga intensiva y se probaron algunas consultas SQL obteniéndose tiempos de respuesta satisfactorios, de esta forma se le da cumplimiento a los requisitos funcionales propuesto por los analistas.

CONCLUSIONES

Se desarrolló una Base de Datos para el Módulo de Microbiología del Sistema de Gestión de la Información de los Laboratorios en la Dirección de Calidad del Centro de Ingeniería Genética y Biotecnología (CIGB) que garantiza un adecuado almacenamiento, procesamiento y gestión de la información, cumple con los requisitos que validan su funcionabilidad y permite un mejor control de la calidad y manejo de la información con fines investigativos para todas las áreas de investigación y desarrollo de la producción, es significativo su aporte en el desarrollo científico tecnológico y la excelencia en los servicios que presta.

Mediante este trabajo se dio cumplimiento a los objetivos propuestos:

- Se diseñó una BD relacional, la cual contiene 55 tablas. Se seleccionó como herramienta de modelado el Visual Paradigm for UML 6.0 Enterprise Edition y el SGBD PostgreSQL 8.2 para garantizar la gestión de la BD.
- Se implementaron las funciones de la BD que garantizan su funcionamiento.
- Se realizó una validación teórica del diseño de la BD, se analizó la integridad, normalización, redundancia de la información, la seguridad y trazabilidad de las acciones.
- Se realizó una validación funcional al realizar pruebas de carga intensiva a la BD, se utiliza la herramienta EMS Data Generator 2005 for PostgreSQL para un llenado voluminoso de la BD y se obtiene como resultado el cumplimiento de todos los requisitos funcionales planteados por los analistas.

RECOMENDACIONES

Se recomienda dar continuidad al desarrollo y mejoramiento de esta versión de la Base de Datos para el módulo de Microbiología, al tener en cuenta que las planillas que utilizan los trabajadores del Laboratorio pueden sufrir cambios. Se recomienda la creación de un Data Warehouse, donde se almacene toda la información que esté guardada en la Base de Datos luego de 5 años de estar vigente en esta, para una posterior investigación o estudio que se desee realizar.

REFERENCIAS BIBLIOGRÁFICAS

1. **Sarmiento Almenares, Alieski y Cutiño Díaz, Elian.** LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis del Grupo de Recepción de Muestras y Manipulación de Expedientes. 2006.
2. Centro de Ingeniería Genética y Biotecnología. [En línea] [Citado el: 28 de Enero de 2008.] <http://www.cigb.edu.cu/pages/>
3. Centro de Ingeniería Genética y Biotecnología. Departamento de Control de la Calidad. [En línea] [Citado el: 29 de Enero de 2008.] <http://www.cigb.edu.cu/pages/ccalidad.htm>.
4. **Camps Paré, Rafael, y otros.** *Bases de datos*. s.l. : Eureka Media, SL, 2005. ISBN: 84-9788-269-5.
5. <http://www.onlabweb.com/lims.htm>. [En línea]
6. **Pons Capote, Olga, y otros.** *Introducción a las Bases de Datos. El Modelo Relacional*. s.l. : Thomson Learning Ibero, 2005. ISBN 8497323963.
7. **Potencier, Fabien y Zaninotto, François.** *Symfony, la guía definitiva*. [En línea] 2008. [Citado el: 6 de mayo de 2008.] <http://www.librosweb.es/symfony/index.html>.
8. **Booch, G., Rumbaugh, J. y Jacobson, I.** *El Lenguaje Unificado de Modelado*. 2000.
9. **Jacobson, I. y Booch, G. y Rumbaugh, J.** *El Proceso Unificado de Desarrollo de software*. s.l. : Addison-Wesley, 2000.
10. **Lacoste Ricardo, Yoendris y Sánchez Luna, Jorge Daniel.** *LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis del Laboratorio de Microbiología*. 2007.

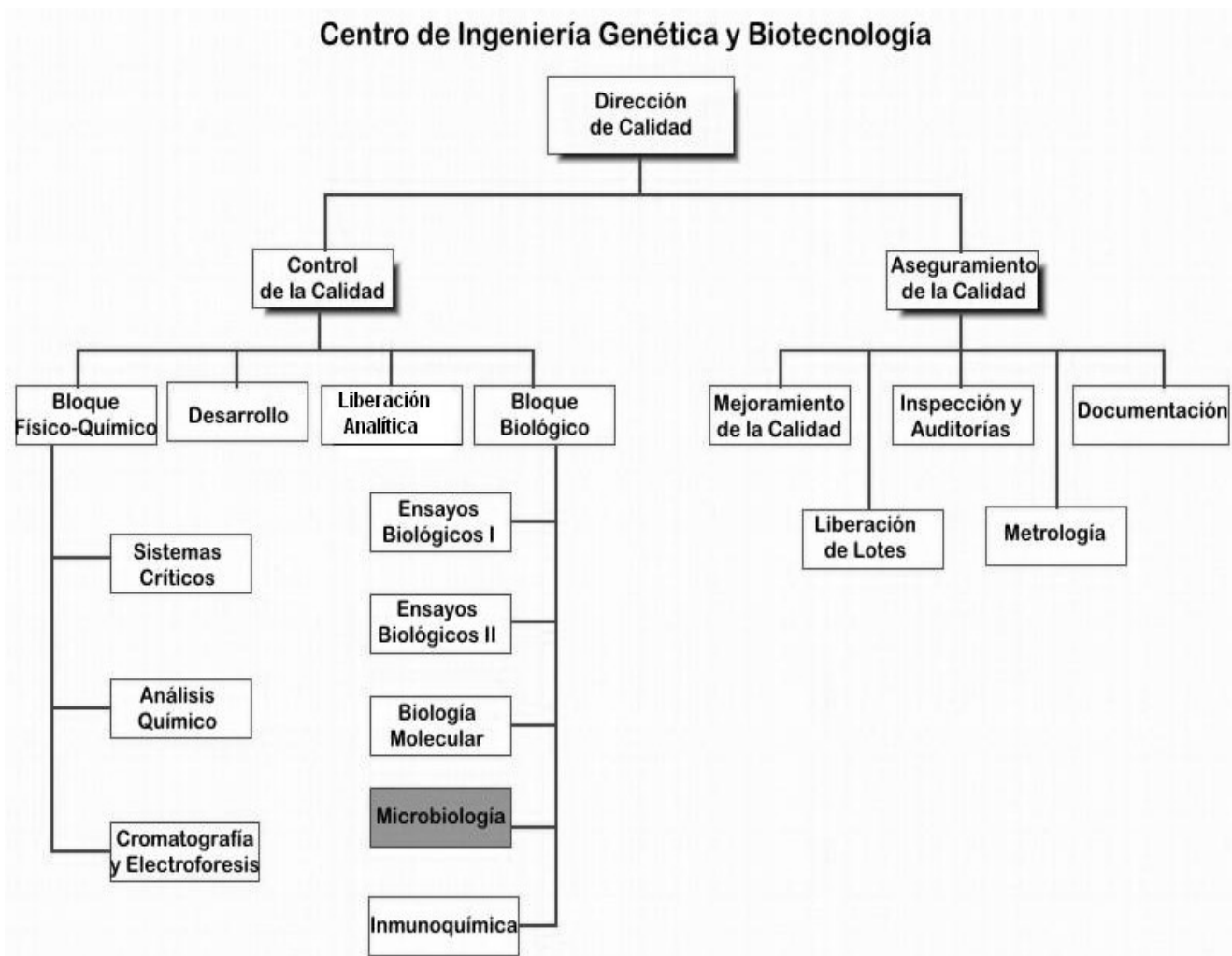
BIBLIOGRAFÍA

- Apuntes de ficheros y base de datos. [En línea] [Citado el: 28 de Enero de 2008.] <http://www3.uji.es/~mmarques/f47/apun/apun.html>
- Ayuda extendida del Rational Rose Enterprise Edition 2003.
- **Booch, G., Rumbaugh, J. y Jacobson, I.** El Lenguaje Unificado de Modelado. 2000.
- **Camps Paré, Rafael, y otros.** Bases de datos. s.l. : Eureka Media, SL, 2005. ISBN: 84-9788-269-5.
- Centro de Ingeniería Genética y Biotecnología. [En línea] [Citado el: 30 de Enero de 2008.] <http://www.cigb.edu.cu/pages/>
- **Desongles Corrales, Juan.** Técnicos Auxiliares de Informática. Temario parte II. s.l.: MAD-Eduforma, 2006. ISBN 8466554068.
- **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.
- **Douglas, Korry y Douglas, Susan.** PostgreSQL: The comprehensive guide to building, programming, and administering PostgreSQL databases, Second Edition. s.l. : Sams Publishing, 2005. 0-672-32756-2.
- EMS Data Generator for PostgreSQL. [En línea] [Citado el: 5 de Marzo de 2008.] <http://www.sqlmanager.net/products/postgresql/datagenerator>
- EMS SQL Manager for PostgreSQL. [En línea] [Citado el: 4 de Marzo de 2008.] <http://www.sqlmanager.net/en/products/postgresql/manager>
- **Hernández, Gonzáles Anaisa.** Un método para el diseño de la base de datos a partir del modelo orientado a objetos. [En línea] [Citado el: 10 de Mayo de 2008.] <http://redalyc.uaemex.mx/redalyc/pdf/615/61570402.pdf>
- **Jacobson, I. y Booch, G. y Rumbaugh, J.** El Proceso Unificado de Desarrollo de software.
- **Lacoste Ricardo, Yoendris y Sánchez Luna, Jorge Daniel.** LIMS de Calidad del Centro de Ingeniería Genética y Biotecnología: Análisis del Laboratorio de Microbiología. 2007.
- LIMS. [En línea] [Citado el: 14 de Abril de 2008.] <http://www.onlabweb.com/lims.htm>
- **Pons Capote, Olga, y otros.** Introducción a las Bases de Datos. El Modelo Relacional. s.l. : Thomson Learning Ibero, 2005. ISBN 8497323963.

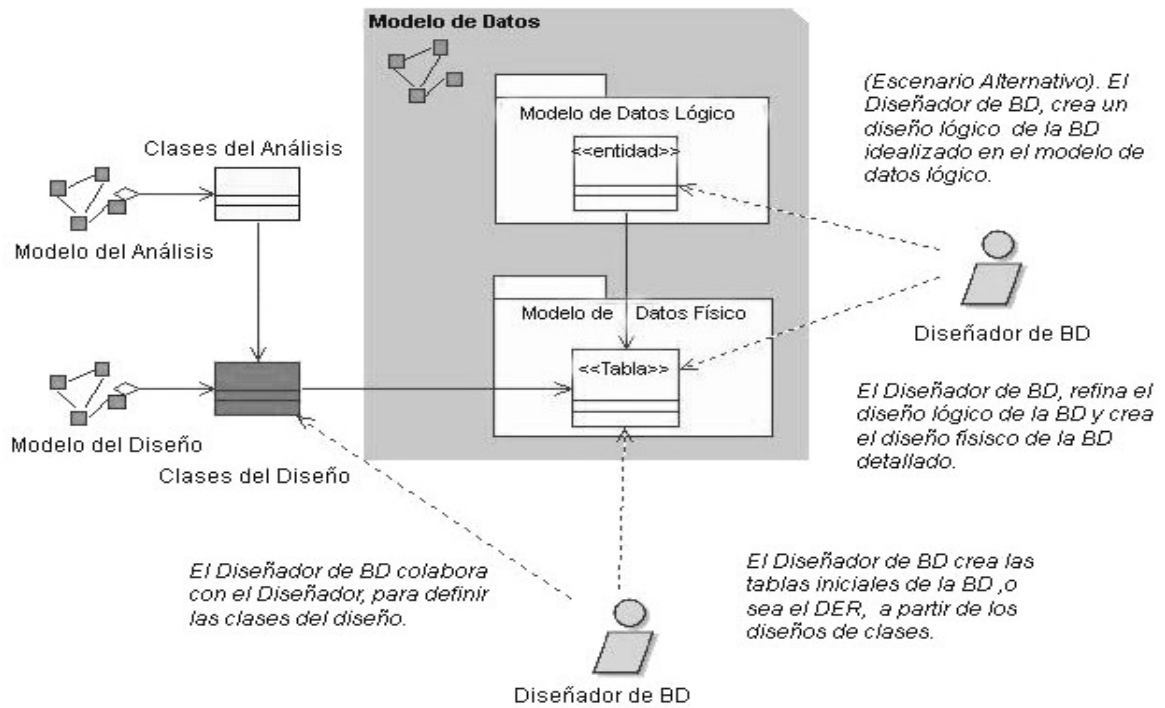
- **Potencier, Fabien y Zaninotto, François.** Symfony, la guía definitiva. [En línea] 2008. [Citado el: 25 de Abril de 2008.] <http://www.librosweb.es/symfony/index.html>.
- **Ramez A., Elmasri y Shamkant B., Navathe.** Sistema de Bases de Datos. Conceptos Fundamentales. s.l. : Addison-Wesley Iberoamericana, 2007.
- **Ruffino, Juan y Cordero, Robertodel.** Starlims. [En línea] [Citado el: 17 de Abril de 2008.] <http://www.inboxsa.com/starlims.htm>
- Starlims OOTB (OUT – OF – THE - BOX). [En línea] [Citado el: 18 de Abril de 2008.] <http://www.innovatek.com.co/ootb.htm>
- Starlims Versión 10. LIMS configurable para la colaboración entre el laboratorio y la empresa. [En línea] [Citado el: 20 de Abril de 2008.] http://www.starlims.com/es/STARLIMS_V10_Brochure_ES.pdf
- **Tejedores del Web.** Base de datos. [En línea] [Citado el: 25 de Mayo de 2008.] http://www.tejedoresdelweb.com/w/Bases_de_datos
- **The PostgreSQL Global Development Group.** About [En línea] [Citado el: 2 de febrero de 2008.] <http://www.postgresql.org/about/>
- **The PostgreSQL Global Development Group.** PostgreSQL 8.3.1 Documentation [En línea] [Citado el: 4 de febrero de 2008.] <http://www.postgresql.org/docs/8.3/interactive/index.html>
- [En línea] [Citado el: 30 de Abril de 2008.] <http://www.visual-paradigm.com>

ANEXOS

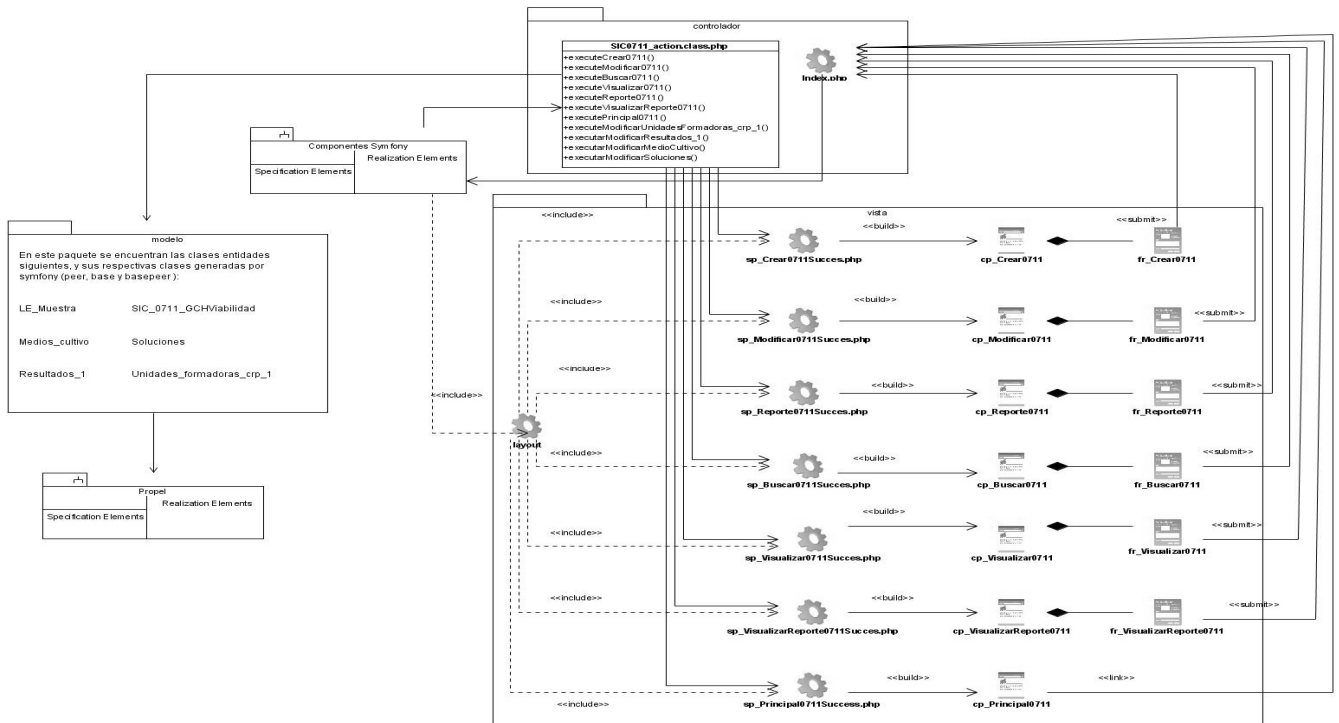
Anexo 1. Organigrama del Departamento de Calidad del CIGB.



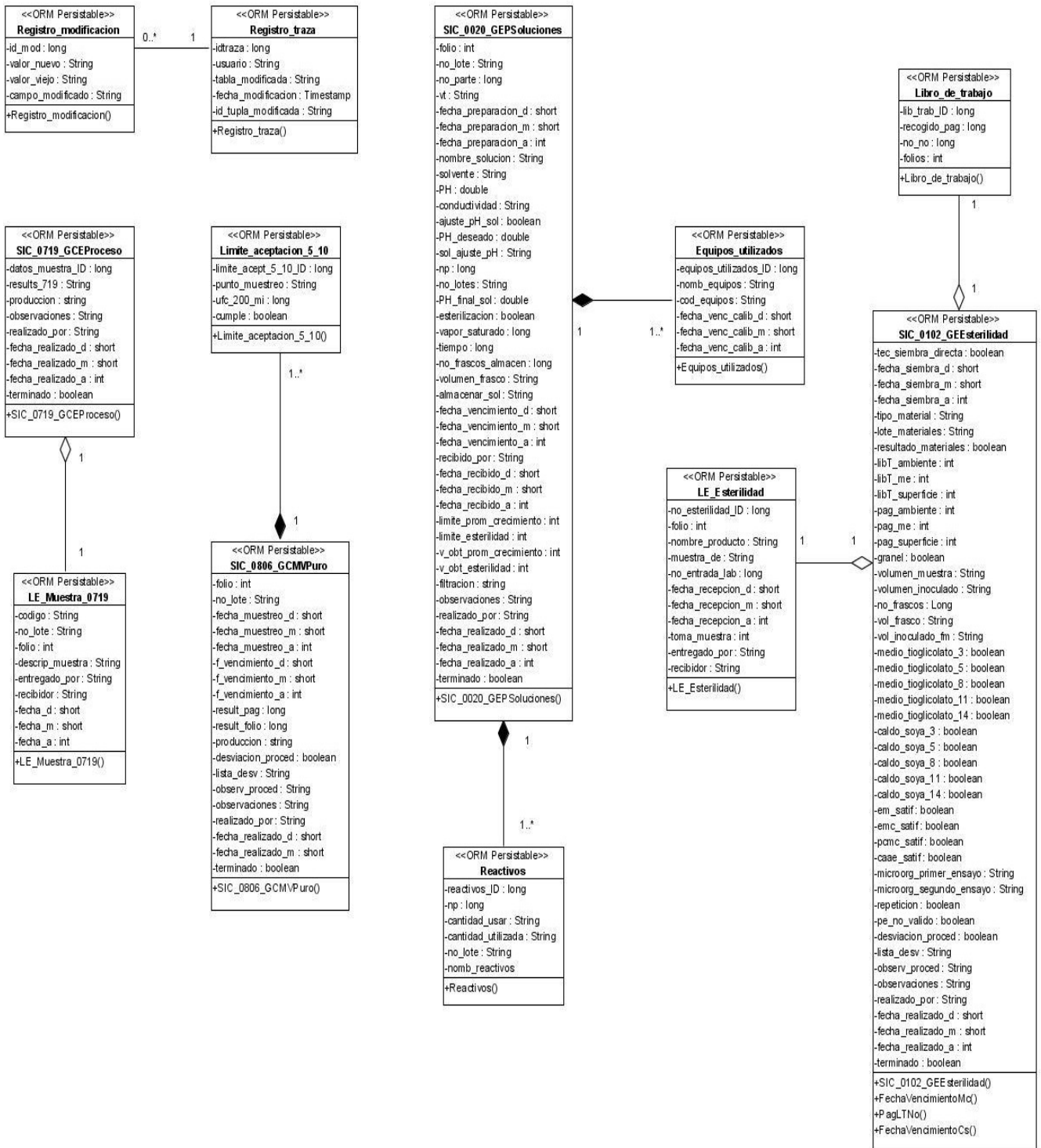
Anexo 2. Artefacto del rol diseñador de BD.

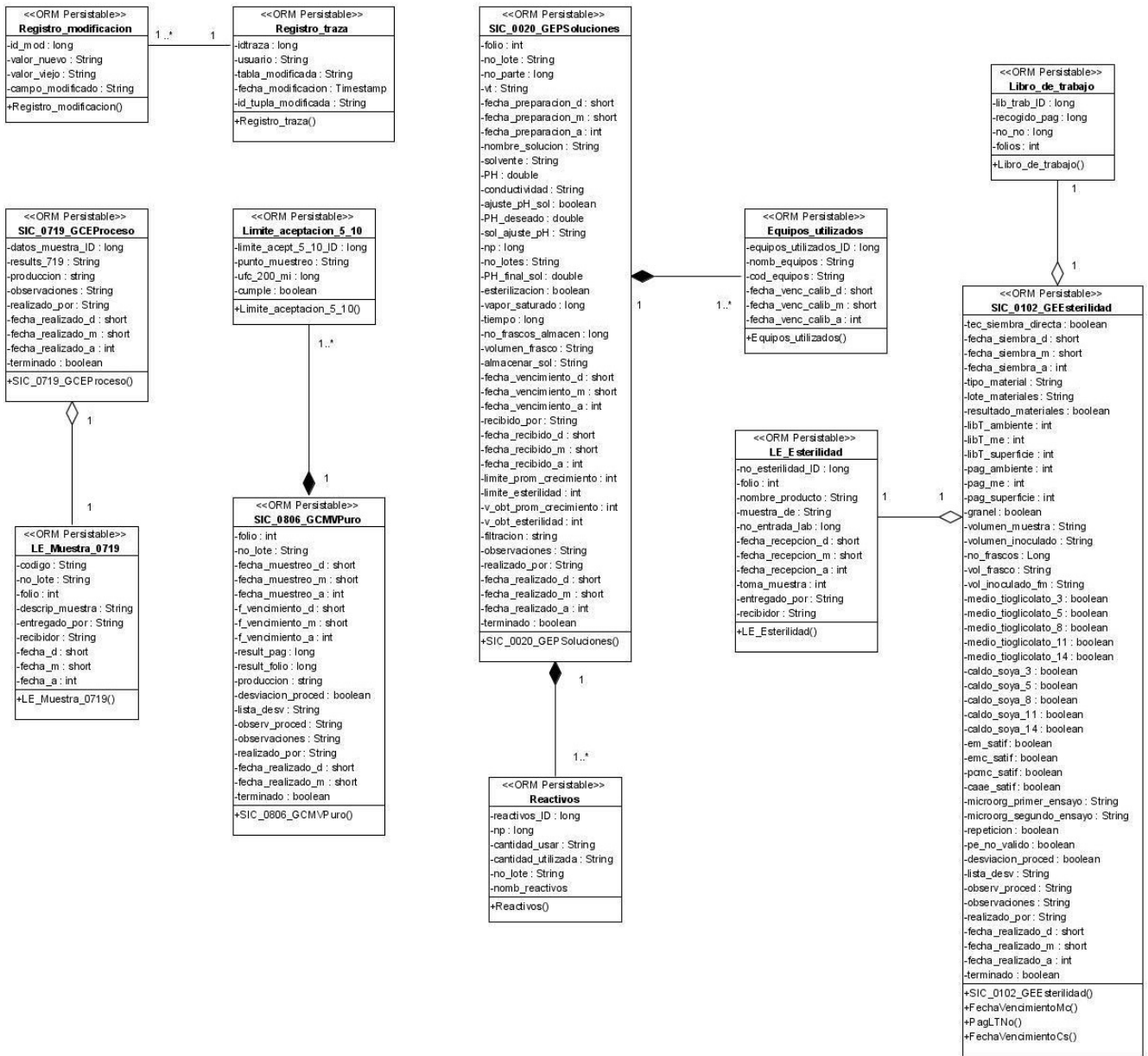


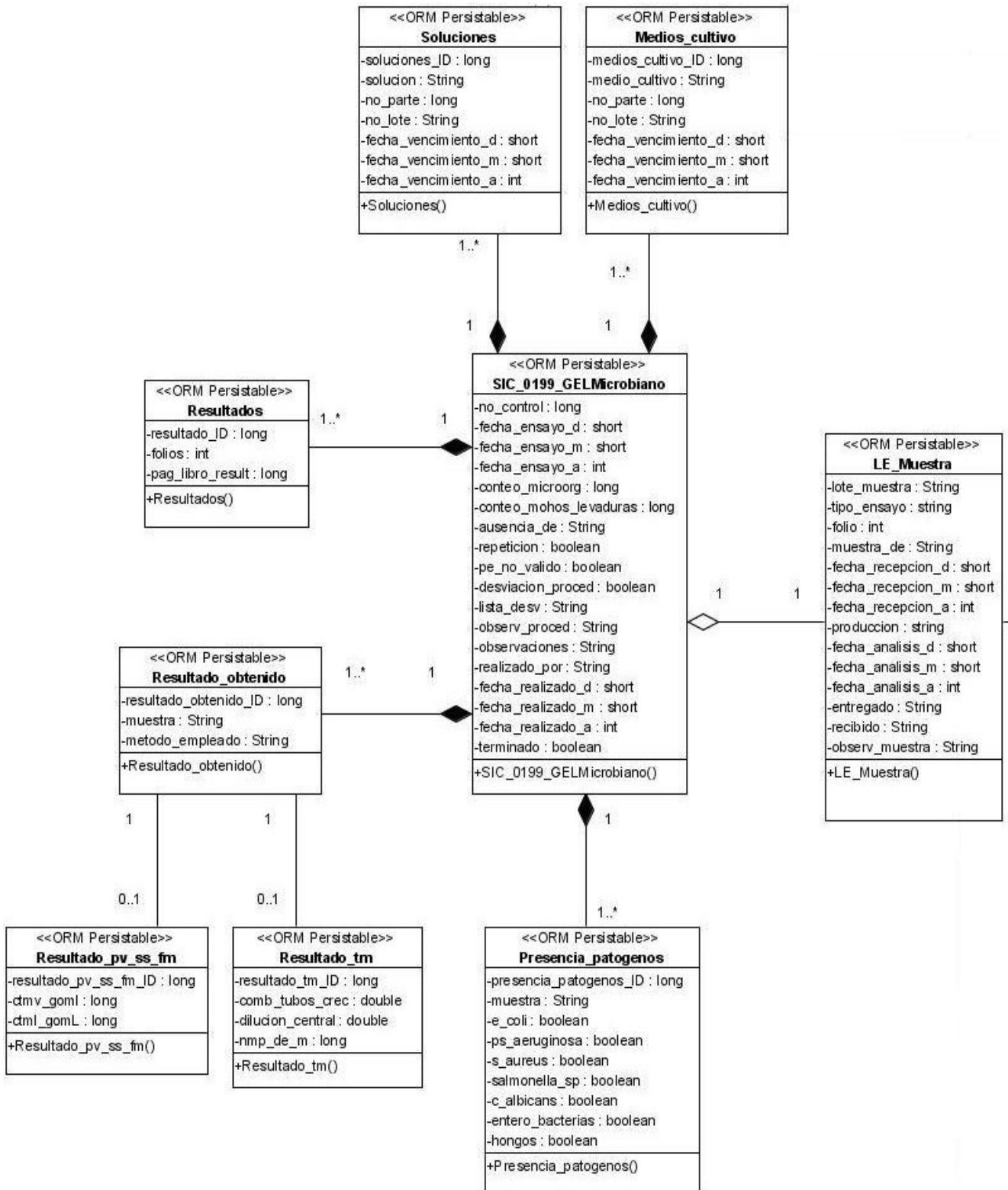
Anexo 3. Diagrama de clases del diseño del CUS Realizar Chequeo de Viabilidad.

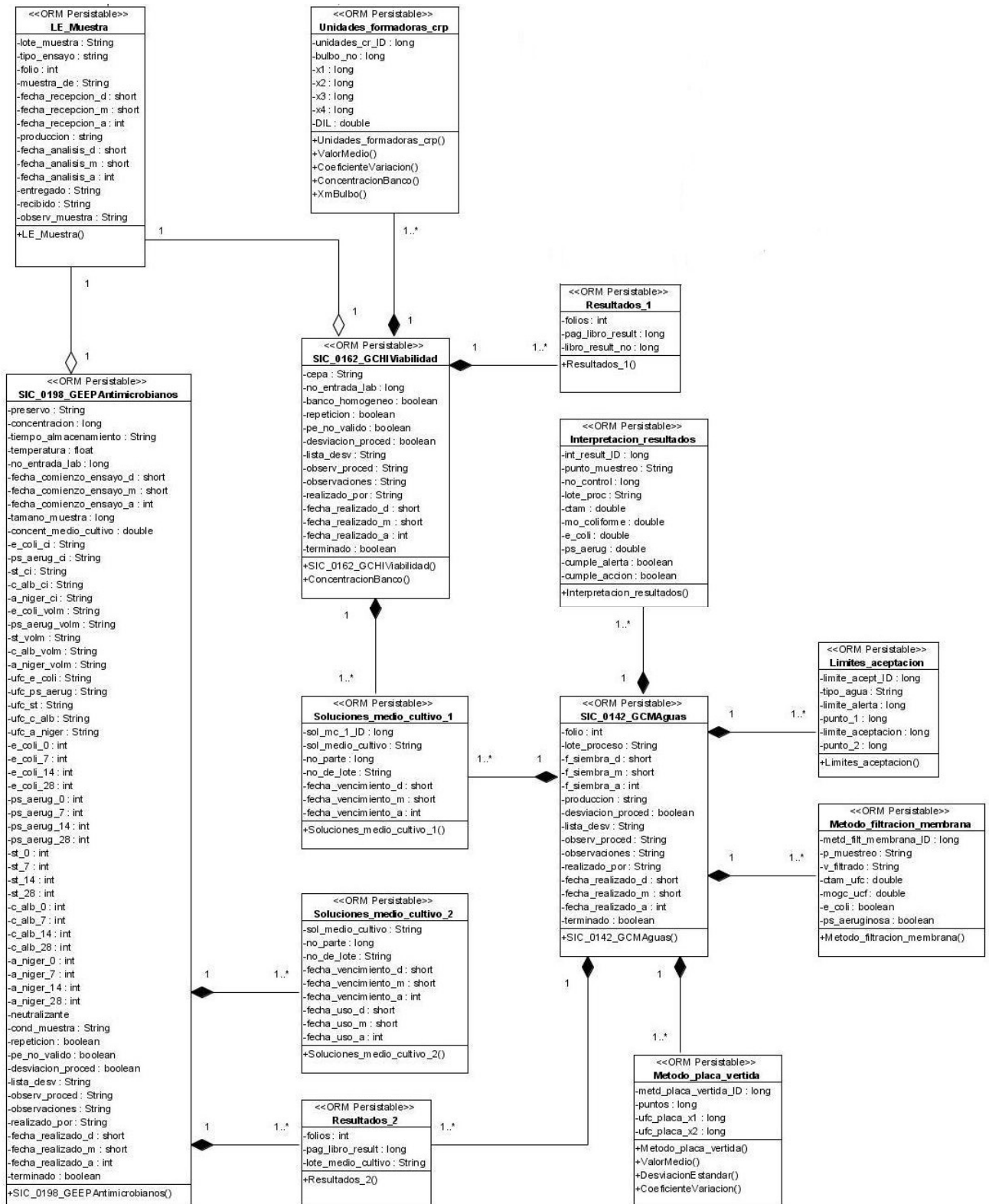


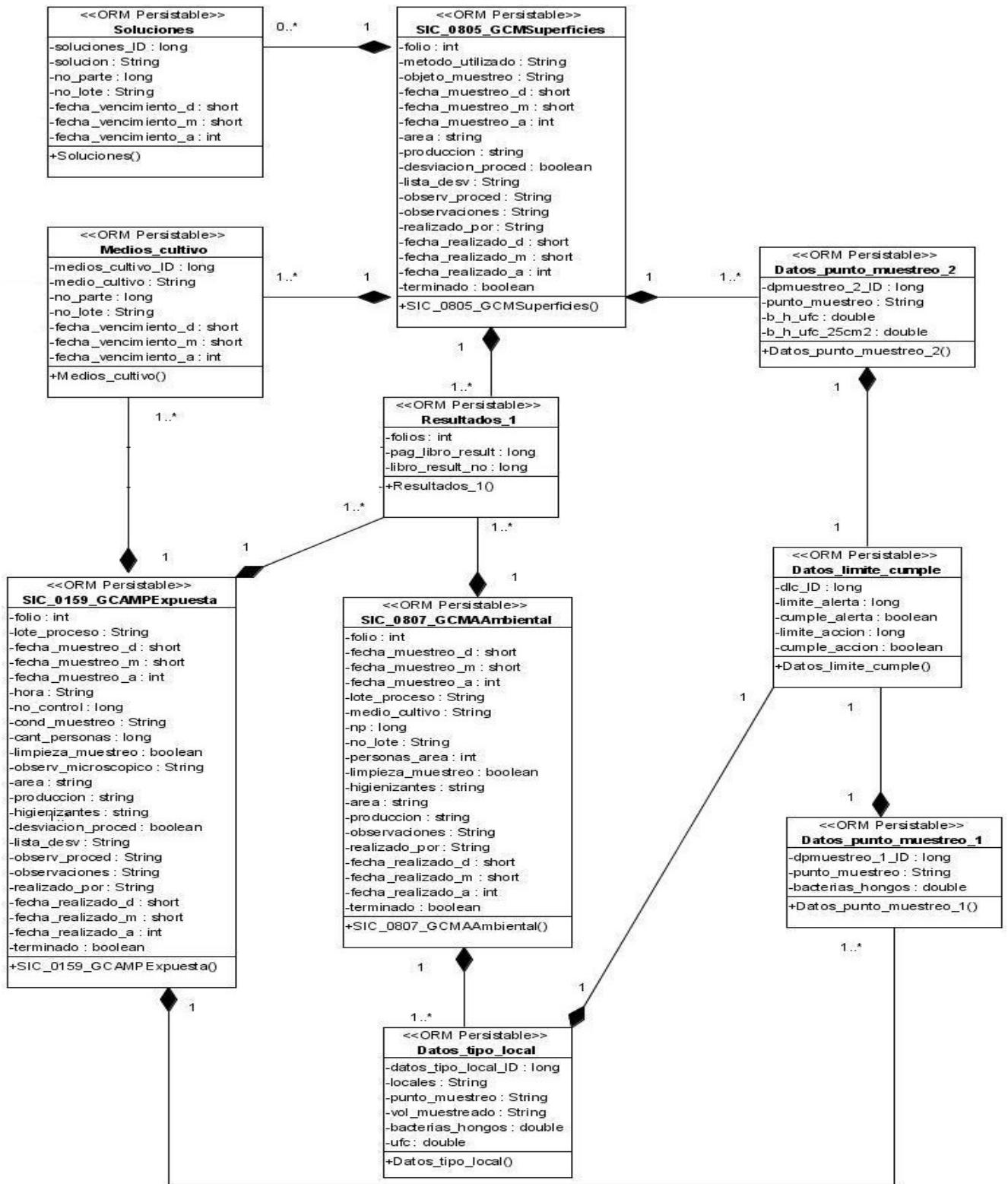
Anexo 4. Diagrama de Clases Persistentes.

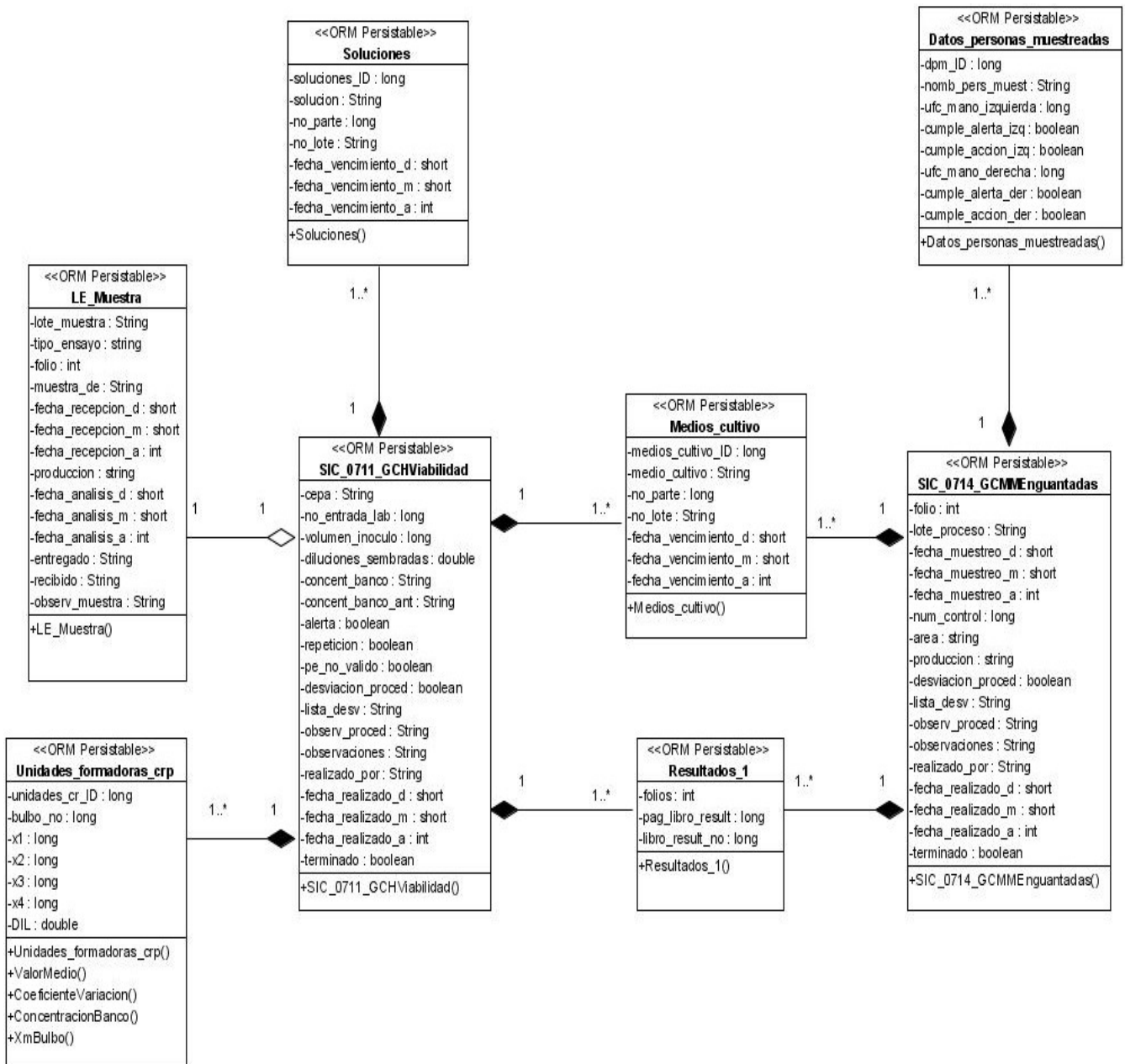


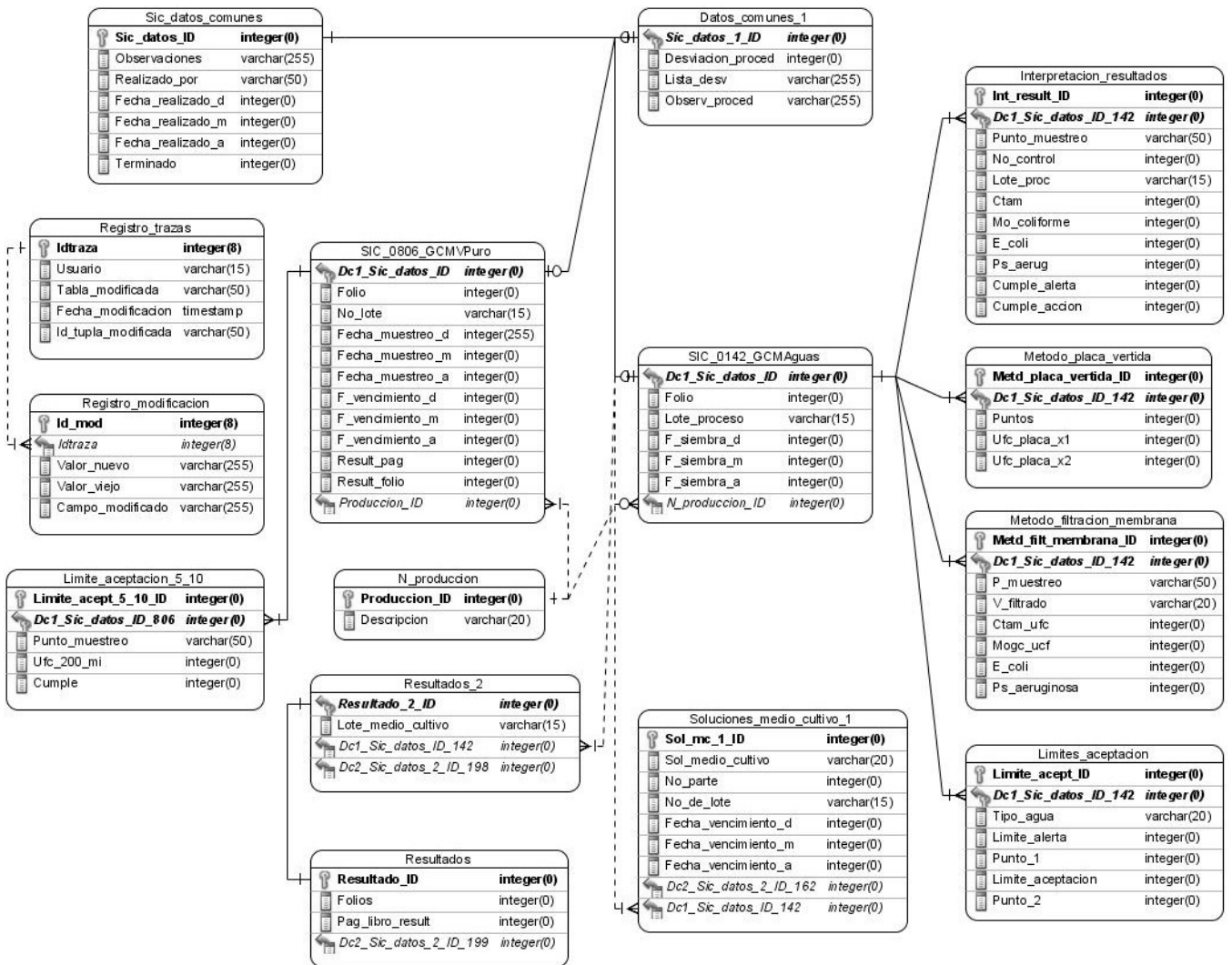


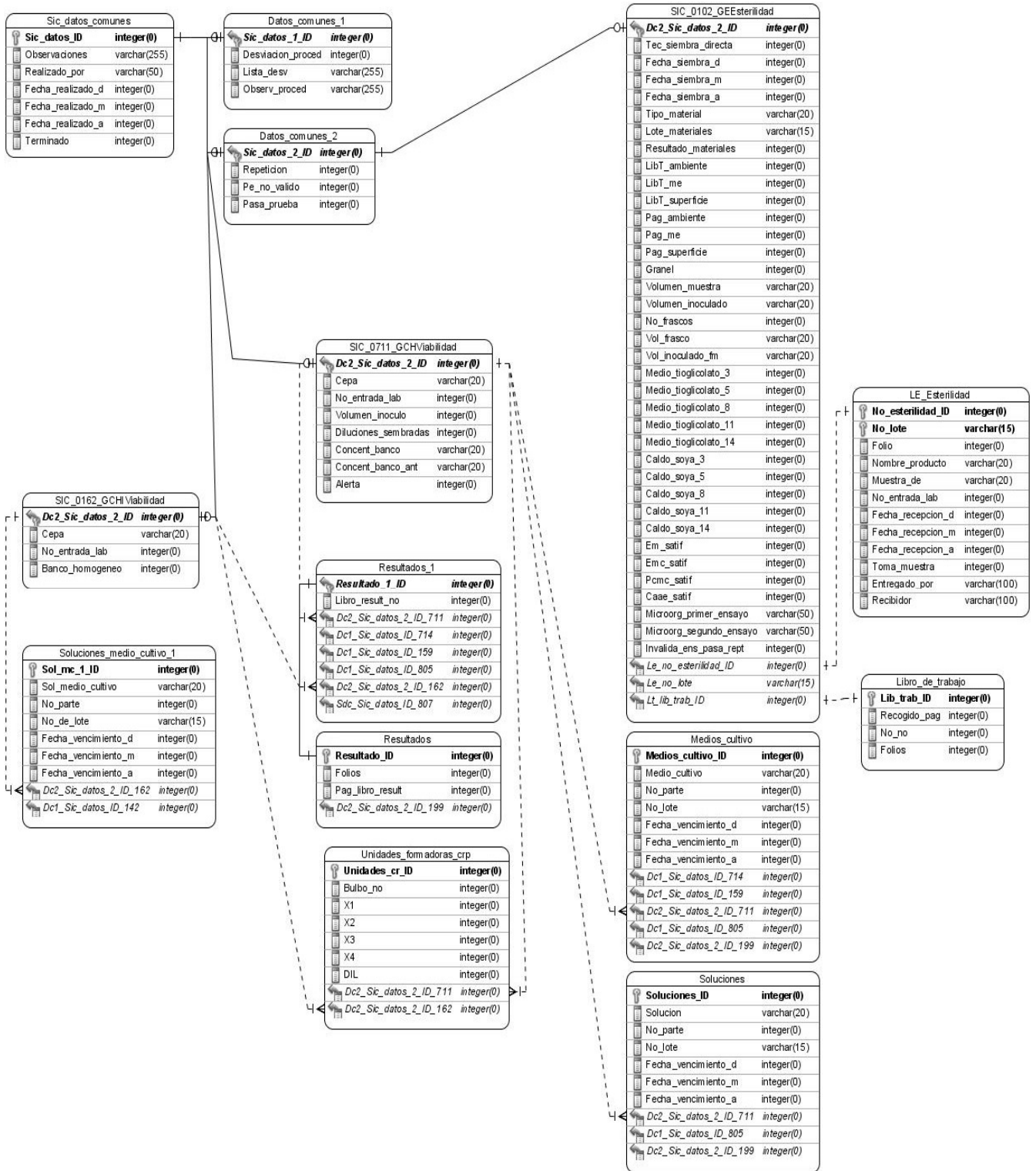


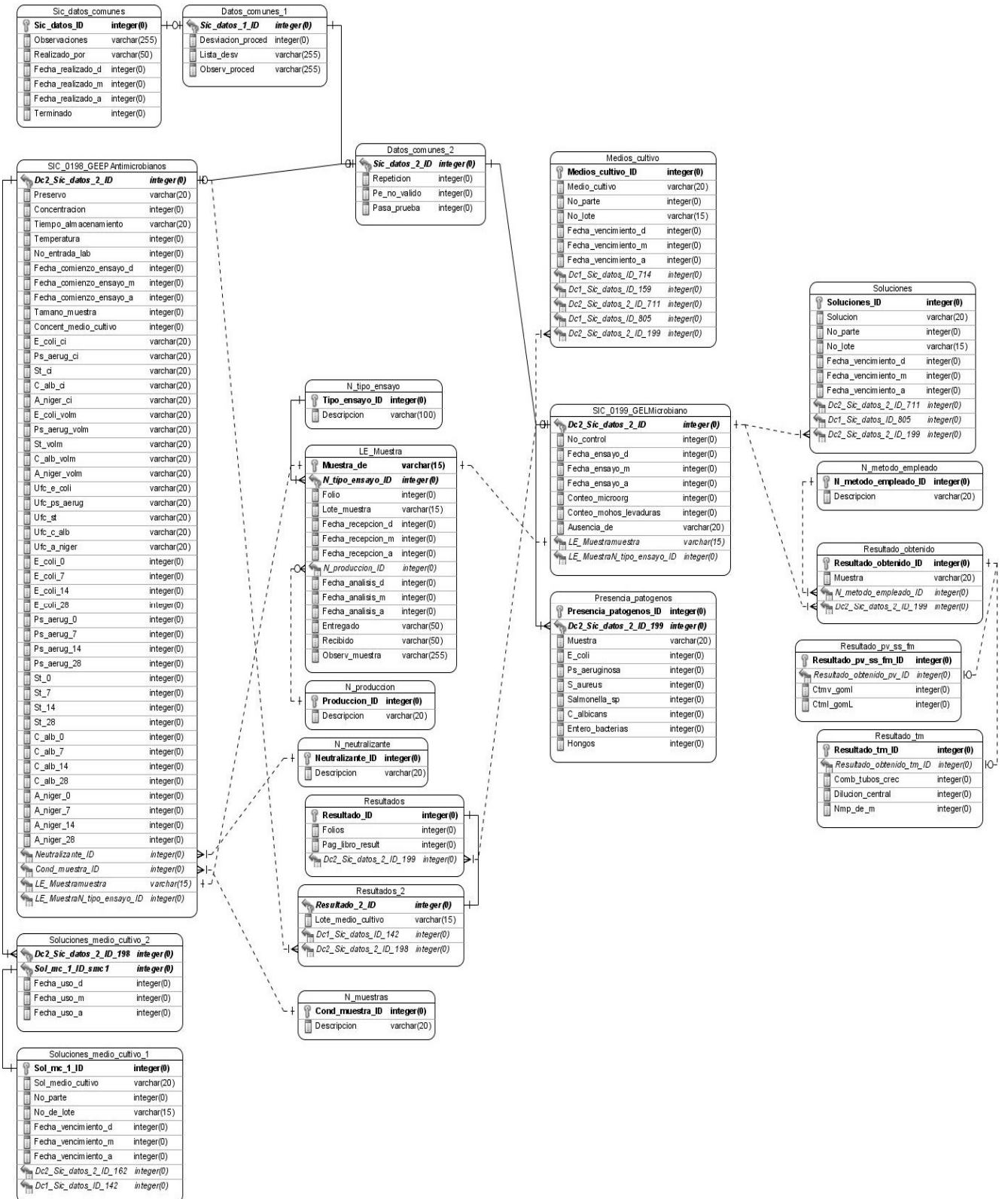


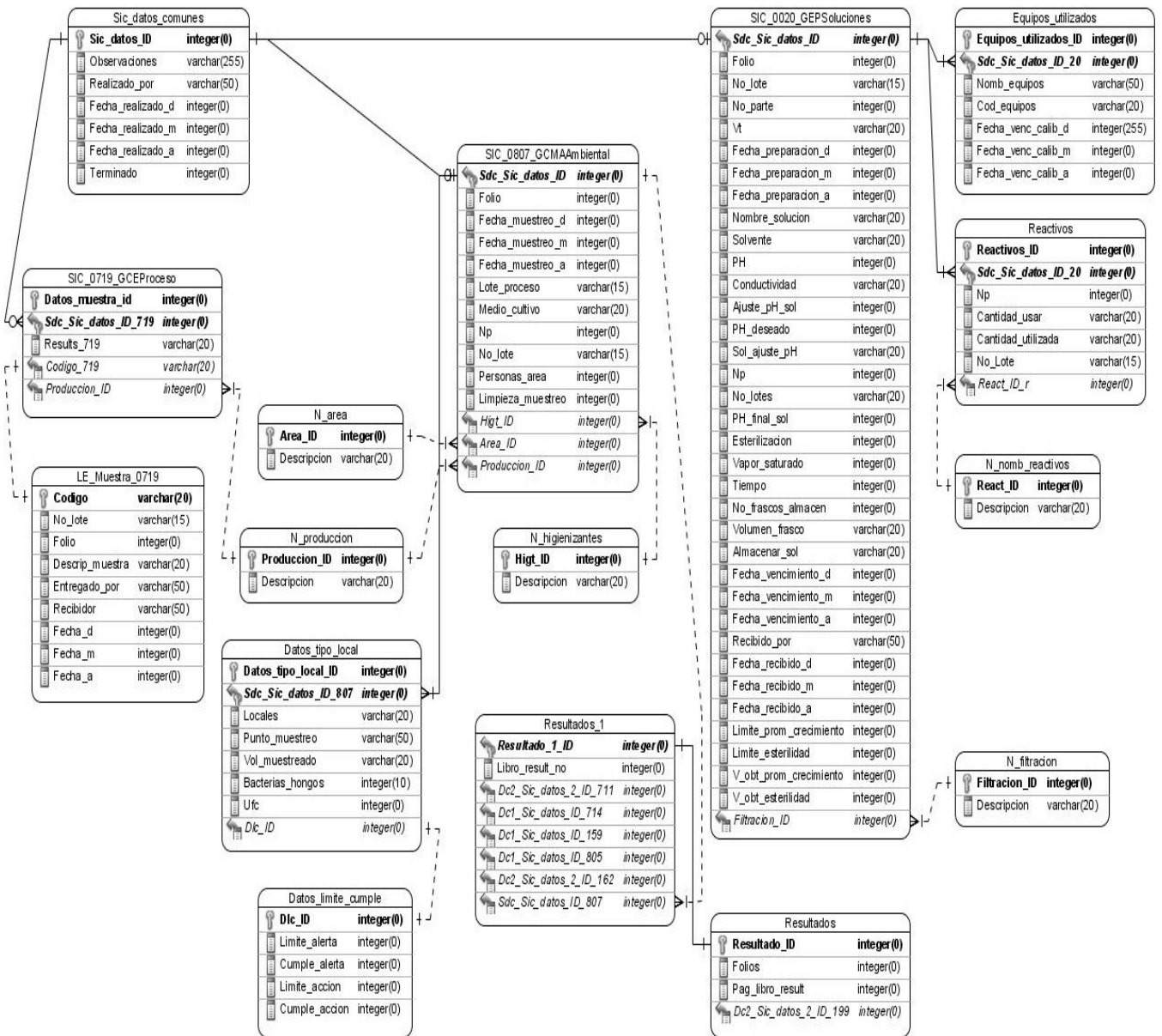












GLOSARIO DE TÉRMINOS

Backup: Copia de seguridad o de respaldo (backup en inglés). Copias de datos de tal forma que estas copias adicionales puedan restaurar un sistema después de una pérdida de información.

BD: Un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un y un conjunto de programas que manipulan esos datos.

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

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

MVC: (*Modelo Vista Controlador*). Es un patrón de arquitectura de software compuesto de tres componentes distintos: datos, interfaz de usuario, y lógica del negocio.

MVCC: Acceso concurrente multiversión. Es un sistema que ofrece el SGBD Postgre para controlar la concurrencia

ORM: (*Object-Relational Mapping*) El mapeo de objetos-relacional 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, creando una base de datos orientada a objetos virtual, por encima de la base de datos relacional.

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.

RUP: (*Rational Unified Process*) Proceso unificado de desarrollo de software.

SIC: Sistema de Información y Control

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