

Universidad de las Ciencias Informáticas

Facultad 2



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

Desarrollo del componente para sincronizar la conducción de
ensayos clínicos en los centros sin acceso al sistema Clínicas 1.0

Autores: Yohana Rodríguez Balaguert

Oscar Leandro Guerra Santos

Tutores: MSc. Dunior Socarrás Benítez

Ing. Idayana Bastarreche Calistre

Co-tutor: Ing. Silvio Sadoth López Henquén

La Habana, junio 2014

“Año 56 de la Revolución”



*“Seamos realista y hagamos lo imposible”
Ernesto Che Guevara*

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los 23 días del mes de junio del año 2014.

Yohana Rodríguez Balaguert
Firma del autor

Oscar Leandro Guerra Santos
Firma del autor

Idayana Bastarrece Calistre
Firma del tutor

Dunior Socarrás Benítez
Firma del tutor

Silvio Sadoth López Henquén
Firma del Tutor

Datos de contacto

Ing. Idayana Bastarreche Calistre

Graduado de Ingeniero en Ciencias Informáticas, en el año 2011, en la Universidad de las Ciencias Informáticas. Ha sido cotutora de dos trabajos de diploma en el departamento. Actualmente se desempeña como analista principal en el proyecto SIGICEM.

Correo electrónico: ibastarreche@uci.cu

MSc. Dunior Socarrás Benítez

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2008. Posee la categoría docente de Instructor. Durante su trabajo como profesor ha impartido la asignatura Idioma Extranjero II y Segundo Perfil. En la vinculación con la producción pertenece al Departamento de Sistemas Especializados en Salud (SES) del Centro de Informática Médica (CESIM).

Correo electrónico: dsocarras@uci.cu

Ing. Silvio Sadoth López Henquén

Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el año 2012.

Correo electrónico: shenquen@uci.cu

Dedicatoria

De Yohana

A mis padres, a mi hermano y a mi tía por todo el apoyo y la ayuda que me han dado, por confiar en mí y por ser la base para llegar hasta donde estoy.

A mi novio Orlando por ser mi sostén en todo minuto, por todo el amor cariño que me ha dado y por estar a mi lado en los buenos y malos momentos.

Agradecimientos

Agradezco

A mi mamá por ser por ser todo, por darme la vida y todo el apoyo del mundo, por darme las fuerzas para seguir, por su cariño, su sacrificio y su amor incondicional, sin ti nada sería posible... Gracias por existir Mamá...

A mi tía querida que siempre lo ha dado todo por mí, que me ha acompañado, apoyado y ayudado en todo lo posible y que siempre ha estado en los momentos más importantes de mi vida, sin ti tampoco hubiera sido posible... Gracias

A mi hermano que siempre ha estado a mi lado, apoyándome y brindándome toda su ayuda y cariño, por ser el mejor amigo que he tenido y por hacerme saber que puedo contar con él en todo momento.

A mi padre por siempre ser el guía, el ejemplo, el maestro, el sacrificio, el que me ha enseñado como alcanzar todas mis metas con mi esfuerzo y sacrificio.

A mi novio Orlando por ayudarme, quererme, por su atención infinita, por apoyarme en todas mis dificultades y momentos difíciles. Gracias

A mi compañero de tesis Oscar Leandro que ha pasado conmigo estos momentos difíciles, que sabe todos los sustos y aprietos que hemos tenido, y que lo dio todo para que pudiéramos lograr esta victoria.

A todas mis amistades que me han acompañado en estos cinco años y a los que ya no están aquí pero que siempre me han apoyado y confiado en mí: Dignay, Maydel, Dayana, Ana Fé, María, Anakaena, Jorge, Reinaldo y a todos los compañeros que he tenido en el transcurso por la universidad.

A mis tutores Idayana y Duniór que me han ayudado tanto y me han guiado en la realización de este trabajo y a todos los que de una forma u otra han aportado su ayuda para hacer este sueño posible.

Yohana Rodríguez Balaguert

RESUMEN

El objetivo de la presente investigación se centra en la sincronización de los datos entre los sistemas Clínicas 1.0 y SIGECConductor, solucionando de este modo la necesidad del Centro de Inmunología Molecular (CIM) de mantener sincronizada la información referente a los ensayos clínicos (EC) almacenada en sus bases de datos y en los centros hospitalarios conductores de los estudios. Este sistema facilitará el manejo de la información asociada a los mismos, y permitirá la actualización rápida de los datos vinculados a los ensayos realizados en el CIM y demás centros conductores.

Para la realización del sistema se utilizó Proceso Unificado de Desarrollo de Software (RUP) como metodología para guiar el desarrollo del software, apoyándose en tecnologías y herramientas libres como el lenguaje de programación Java, el entorno de desarrollo NetBeans, la herramienta para la modelación Visual Paradigm y el gestor de bases de datos PostgreSQL. Se emplearon además el patrón arquitectónico Modelo-Vista-Controlador e Hibernate para el Mapeo Objeto-Relacional (ORM).

Las funcionalidades del sistema facilitarán la actualización de la información referente a los ensayos clínicos realizados, permitirá la sincronización de los datos en el CIM y los centros hospitalarios conductores de los estudios y que fluya de manera rápida, íntegra y segura la información referente a los EC que se realizan, garantizando la integridad de la información que se maneje.

Palabras claves: actualización, bases de datos, ensayos clínicos, sincronización.

INDICE

INTRODUCCIÓN - 1 -

Capítulo 1: Fundamentación Teórica - 5 -

 1.1. Ensayos clínicos (EC)..... - 5 -

 1.2. Ensayos clínicos en Cuba..... - 6 -

 1.3. Sincronización de datos..... - 7 -

 1.4. Replicación de datos..... - 7 -

 1.4.1 Servidor de replicación. - 8 -

 1.4.2 Tipos de replicación de datos - 8 -

 1.4.3 Entornos de replicación de datos..... - 9 -

 1.4.4 Dirección de transmisión de datos en la replicación..... - 9 -

 1.4.5 Modelos de distribución de cambios - 9 -

 1.4.6 Conflictos de la replicación de datos..... - 10 -

 1.5. Sistemas analizados para la sincronización y replicación de datos..... - 10 -

 1.5.1 SymmetricDS..... - 11 -

 1.5.2 CronSQL..... - 11 -

 1.5.3 DBMoto..... - 12 -

 1.5.4 SharePlex..... - 13 -

 1.5.5 Slony - 14 -

 1.5.6 Reko - 15 -

 1.5.7 Magic@ Data Replication eXtensible Solution - 16 -

 1.6. Resultado del análisis de las soluciones estudiadas - 17 -

 1.7. Metodología..... - 17 -

 1.8. Herramientas y tecnologías - 20 -

Capítulo 2. Características del sistema - 25 -

 2.1. Solución Propuesta..... - 25 -

 2.2. Modelo del dominio..... - 25 -

 2.3. Especificación de los requisitos de software - 27 -

 2.3.1 Requisitos funcionales - 27 -

 2.3.2 Requisitos no funcionales - 28 -

 2.4. Patrones de Casos de uso..... - 29 -

 2.5. Actores del sistema..... - 30 -

 2.6. Diagrama de casos de uso del sistema..... - 30 -

 2.7. Descripción de los casos de uso del sistema - 31 -

Capítulo 3. Análisis y diseño del sistema - 34 -

 3.1. Descripción de la arquitectura..... - 34 -

3.1.1 Patrón Arquitectónico.....	- 34 -
3.2. Modelo de Análisis.....	- 35 -
3.2.1 Diagramas de clases del análisis.....	- 35 -
3.3. Modelo de Diseño.....	- 36 -
3.3.1 Patrones de diseño.....	- 37 -
3.3.2 Patrones GRASP.....	- 38 -
3.3.3 Diagramas de interacción.....	- 39 -
3.3.4 Diagramas de clases del diseño.....	- 40 -
3.3.5 Descripción de las clases del sistema.....	- 41 -
Capítulo 4. Implementación del sistema.....	- 47 -
4.1. Modelo de datos.....	- 47 -
4.2. Diagrama de despliegue.....	- 50 -
4.3. Diagrama de componentes.....	- 50 -
4.4. Tratamiento de errores.....	- 51 -
4.5. Seguridad.....	- 52 -
4.6. Estrategias de codificación. Estándares y estilos a utilizar.....	- 52 -
Conclusiones.....	- 54 -
Recomendaciones.....	- 55 -
Referencias Bibliográficas.....	- 56 -
Bibliografía.....	- 60 -
Anexos.....	- 64 -
Anexo 1: Entrevista.....	- 64 -
Anexo 2: Descripción de Casos de Uso.....	- 64 -
Anexo 3: Diagramas de Clases del Análisis.....	- 73 -
Anexo 4: Diagramas de Secuencia.....	- 75 -
Anexo 5: Diagramas de clases del diseño.....	- 77 -
Glosario de términos.....	- 80 -

INTRODUCCIÓN

La informática en la actualidad ha alcanzado un acelerado y notable desarrollo, pudiendo ser de utilidad en cualquier rama o esfera de la sociedad, esto ha hecho conscientes a todos de la necesidad de avanzar a la par del mundo en cuanto a progreso tecnológico e informatización del entorno laboral. La aplicación de soluciones informáticas en los procesos empresariales favorece las condiciones de trabajo del personal y en ocasiones aumenta la productividad de la empresa, además de repercutir en la calidad de los productos y los servicios.

El uso de las Tecnologías de la Información y las Comunicaciones (TIC) en el mundo ha facilitado la cooperación, el intercambio de ideas entre los individuos de la sociedad y el desarrollo de las tareas que se realizan en las diferentes esferas sociales; tanto en la economía, educación, medicina, así como en la producción de toda clase de bienes y servicios. Su uso desarrolla la iniciativa, expresión, creatividad, motivación, intercambio e interés de las personas, además de facilitar el trabajo en equipos, brindar grandes beneficios y adelantos en la salud y la educación y permitir el acceso al flujo de conocimientos e información.

En la actualidad Cuba ha dedicado recursos y esfuerzos para no quedar rezagada en el mundo respecto al uso de las TIC, alcanzando logros en la mayoría de los sectores sociales, especialmente en el de la salud. Se presenta como un reto conducir, coordinar el desarrollo, la culminación y la implementación de aplicaciones informáticas para el sistema de salud (1). Con el objetivo de lograr la informatización del mismo se han desplegado varios sistemas informáticos en algunos centros médicos del país, mejorando así los procesos que se llevan a cabo en cada uno de ellos en beneficio de los especialistas médicos y los pacientes.

El Centro de Inmunología Molecular (CIM) es una de las instituciones que se ha favorecido con el uso de las TIC en Cuba. El objetivo principal de las investigaciones que se realizan en el CIM es la búsqueda de nuevos productos para el diagnóstico y tratamiento del cáncer y enfermedades relacionadas con el sistema inmune. Los proyectos de investigación básica están concentrados en la inmunoterapia del cáncer, especialmente en el desarrollo de "vacunas moleculares", ingeniería de anticuerpos, ingeniería celular, bioinformática y regulación de la respuesta inmune. Este centro realiza, en hospitales altamente especializados, ensayos clínicos para el tratamiento de cáncer de diferentes orígenes y otras enfermedades del sistema inmune. (2)

El CIM utiliza para la gestión de los ensayos clínicos el sistema Clínicas 1.0 desarrollado por el Centro de Informática Médica (CESIM) de la Universidad de las Ciencias Informáticas (UCI); este centro se encarga de desarrollar productos destinados al sector de la salud, y con la realización de este sistema informático cumplió el objetivo de facilitar al país y al CIM la gestión de los estudios realizados. Con esta aplicación se logran una serie de ventajas prácticas, como la reducción del tiempo, recursos y esfuerzos necesarios en la conducción de los estudios clínicos, disminuye la posibilidad de errores, gracias a que los datos son introducidos directamente por quienes los recogen, eliminado así la doble entrada de datos y facilitando el flujo de información entre las diferentes entidades conductoras de ensayos.

El CIM realiza a través del Clínicas 1.0 el diseño o cronograma de los ensayos clínicos que se efectúan en diferentes hospitales, no teniendo algunos de estos últimos formas para lograr la conexión a la aplicación, lo que imposibilita el acceso a la misma, por lo que necesitan un sistema capaz de gestionar la conducción de los estudios que realizan. Se encuentra actualmente en desarrollo el sistema para la conducción de ensayos clínicos SIGECConductor, que será desplegado en cada uno de los centros sin conexión al Clínicas 1.0, con el fin de facilitar la realización de los estudios que se efectúan en esos hospitales.

El SIGECConductor necesita mantener sincronizados sus datos con la información almacenada en el Clínicas 1.0, ya que este es quien especifica a través de los cronogramas que realiza, cómo y cuándo se debe realizar el ensayo clínico en cada hospital y actualmente no existe una manera eficiente para que los centros hospitalarios accedan a esta información. La falta de sincronización entre estos sistemas puede provocar dificultades en la actualización de los datos entre el CIM y los hospitales, imposibilita la actualización y gestión de los datos de forma centralizada, lo que trae como consecuencias que existan discrepancias o errores en la información almacenada en las distintas instituciones hospitalarias.

Además no existe una forma de importar hacia las bases de datos del CIM la información referente a la conducción de los ensayos clínicos realizados en los diferentes hospitales. De igual manera el Clínicas 1.0 propone un diseño para la gestión de los datos de los sujetos de cada centro, lo que debe ser de conocimiento de ellos para impedir posibles errores en la conducción de los EC. Estas dificultades pueden ocasionar daños a las instituciones que conducen los ensayos clínicos desde el punto de vista económico y uso de materiales y recursos innecesariamente.

Por lo expuesto anteriormente se plantea como **problema a resolver**: ¿Cómo lograr la actualización de los datos entre el Clínicas 1.0 y el SIGECConductor, de manera que se garantice la digitalización de la conducción del ensayo clínico en los centros sin conexión?

El **objeto de estudio** de la investigación es la sincronización de datos entre aplicaciones informáticas, enmarcado en el **campo de acción**: la sincronización de datos entre el Clínicas 1.0 y el SIGECConductor.

En aras de resolver el problema planteado se define como **objetivo general**: Desarrollar un componente para la sincronización de los datos entre el Clínicas 1.0 y el SIGECConductor.

Para dar cumplimiento al objetivo planteado se definieron las siguientes **tareas de investigación**:

- Asimilación de la metodología, las herramientas y tecnologías definidas por el proyecto que serán utilizadas en el desarrollo del componente para la sincronización de los datos entre el Clínicas 1.0 y el SIGECConductor.
- Caracterización de las particularidades del proceso de sincronización de datos.
- Diseño de los artefactos del flujo de trabajo definidos por la metodología RUP: Modelado de negocio, Requisitos, Análisis y Diseño e Implementación, para la obtención de la documentación necesaria.
- Implementación de las funcionalidades identificadas en el proceso, según lo determinado en la especificación de requisitos de software.

Con el desarrollo de las funcionalidades propuestas se proporcionarán los siguientes beneficios a los especialistas del CIM y de los centros sin acceso al sistema Clínicas 1.0:

- Facilitará la actualización de los datos referentes a los ensayos clínicos realizados.
- Permitirá optimizar el tiempo dedicado a la entrada y actualización de la información en las bases de datos.
- Permitirá la gestión de los datos de forma centralizada.

Los **métodos y técnicas de la investigación científica** utilizados para la realización de la presente investigación son:

Empíricos:

- Entrevista no estructurada: usado como técnica de recopilación de información para definir el alcance de la investigación e identificar las funcionalidades que debe tener el sistema. La entrevista realizada no contó con preguntas específicas previamente elaboradas y se realizó en forma de diálogo.

Teóricos:

- La modelación: a través de él se crearon abstracciones para explicar la realidad, realizando diferentes diagramas para facilitar la comprensión del equipo de desarrollo sobre cómo debería ser diseñado e implementado el sistema.
- Análisis Histórico-Lógico: se utilizó en la realización del estudio del estado del arte, además para la definición de las herramientas, tecnologías y metodología a utilizar en el proceso de desarrollo de software.
- Inductivo-Deductivo: es un procedimiento que permitió arribar a conclusiones lógicas generales a partir del estudio de hechos y del conocimiento adquirido en la investigación.

El presente trabajo está estructurado en 4 capítulos:

Capítulo 1. Fundamentación teórica: se incluye un estudio del estado del arte del tema tratado, a nivel internacional y nacional. Además se definen las tecnologías, metodología y herramientas que se utilizarán para el desarrollo de la solución propuesta.

Capítulo 2. Características del sistema: se describen los conceptos del dominio, se muestran los requisitos funcionales y no funcionales, se presentan los diagrama de casos de uso del sistema y sus descripciones.

Capítulo 3. Análisis y diseño del sistema: se describen los patrones de diseño y arquitectura a utilizar y define artefactos como diagrama de clases del diseño y los diagramas de interacción.

Capítulo 4 Implementación del sistema: describe el modelo de implementación, se realizan el modelo de los datos, los diagramas de componente y despliegue. Además queda plasmado en este capítulo como se hará el tratamiento de errores, los estilos y estándares a utilizar para la codificación.

Capítulo 1: Fundamentación Teórica

En este capítulo se aborda la fundamentación teórica de la investigación explicando algunas características de los ensayos clínicos, cómo funcionan estos en el Centro de Inmunología Molecular (CIM), algunos conceptos asociados a los procesos de sincronización y replicación de bases de datos, los diferentes sistemas existentes en el mundo que realizan estas funciones y la metodología, tecnologías y herramientas a utilizar en el desarrollo del software.

1.1. Ensayos clínicos (EC)

Un ensayo clínico es un estudio que ayuda a los médicos e investigadores a encontrar mejores métodos de prevención, tratamiento o diagnóstico de enfermedades como el cáncer. Para combatir y prevenir estos tipos de enfermedades se elaboran productos químicos que son convertidos posteriormente en medicamentos o tratamientos para los seres humanos, teniendo que - antes de la aplicación de estas sustancias- realizar estos exámenes para probar su efectividad en las personas. Existen diferentes tipos de EC para el cáncer, algunos evalúan nuevas formas de tratar el cáncer, otros prueban nuevos métodos de prevención, detección o diagnóstico. Algunos de ellos ayudan a evaluar y mejorar la calidad de vida de los pacientes con cáncer. (3)

El ensayo clínico se puede realizar en cuatro fases:

Fase I: Constituye el primer paso en la investigación de una sustancia o medicamento nuevo en el hombre. (4) Son estudios que brindan información sobre el efecto que puede provocar el producto en personas sanas y orientan las pautas de administración para casos futuros.

Fase II: Representa el segundo estado en la evaluación de un nuevo producto o medicamento en el ser humano. (4) Es realizado a pacientes que padecen una enfermedad de interés o pertenecen a una entidad clínica determinada y su objetivo es proporcionar información preliminar sobre la eficacia de la sustancia, establecer relaciones dosis-respuesta de la misma y ampliar los datos obtenidos en la fase I sobre la seguridad necesaria para aplicar estos fármacos.

Fase III: Los ensayos de esta fase están destinados a evaluar la eficacia y seguridad del tratamiento experimental intentando reproducir las condiciones de uso habituales y considerando las alternativas terapéuticas disponibles en la indicación estudiada. (4) La

muestra de pacientes que se toma es representativa de la población que utilizará el medicamento y es más amplia que en la fase II, por lo que estos estudios son controlados y se realizan de forma aleatoria.

Fase IV: este tipo de EC se realiza después de comercializado un medicamento y busca descubrir algún aspecto aún no valorado o condiciones de uso diferentes a las autorizadas.

1.2. Ensayos clínicos en Cuba

Con el objetivo de combatir enfermedades mortales como el cáncer o el SIDA, científicos cubanos se han dado a la tarea de crear productos químicos y convertirlos en medicamentos para tratamientos a seres humanos. Uno de los centros dedicados a esta labor es el CIM, institución biotecnológica cubana de ciclo cerrado que forma parte del Polo Científico del Oeste de Ciudad de La Habana, dedicada a la investigación básica, desarrollo y fabricación de productos a partir del cultivo de células superiores, las que deben desarrollarse de acuerdo con las regulaciones vigentes para este tipo de producto. (5)

En años anteriores el CIM realizaba la recogida de los datos de los EC de forma manual, e introduciéndola a una base de datos. Esto hacía el proceso poco eficiente, ya que no brindaba ningún tipo de información, solo se almacenaba la misma. Esto generó la necesidad para el CIM de tener un sistema para gestionar la conducción de los estudios realizados, desarrollando la UCI un sistema informático para mitigar estas necesidades, surgiendo así el sistema Clínicas 1.0.

La informatización de la gestión de los EC en Cuba y a nivel mundial trae consigo una serie de ventajas prácticas:

- Acelera el desarrollo de los productos y optimiza el proceso de monitoreo, ya que disminuye el tiempo de conducción del EC y el número de visitas a los sitios de inclusión (lugar donde se elige a la población de estudio), lo que reduce los costos de los estudios. (6)
- Reducir los costos en cuanto a transportación y material de oficina ya que se ocupará menos espacio físico al eliminar los CRD¹ impresos. (6)

¹ CRD: Cuaderno de Recogida de Datos.

- Permite una regularización mayor e inmediata de los datos recogidos, ya que el sistema chequea que no se introduzcan valores fuera de rango, en unidades erradas o que se omita o quede incompleta cierta información. (6)
- Respuesta rápida a problemas que en el modo tradicional tardan semanas y meses en conocerse y por consecuencia más tiempo en solucionarse. (6)
- Permite acceder de manera inmediata y actualizada al estado de desarrollo del EC, permitiendo un mejor control y análisis sobre los datos que se manejan en cada estudio.
- Puede simplificar el proceso de randomización² del ensayo. El software se ocupa de asignar el brazo del tratamiento en el que se incluye un paciente, justo en el momento de su inclusión. (6)

1.3. Sincronización de datos

La sincronización de datos es el proceso donde dos dispositivos se actualizan de manera que contengan los mismos datos, es mantener coherencia en la información contenida en distintas aplicaciones, bases de datos o sistemas. Si se agrega, modifica o elimina un archivo de una ubicación, el proceso de sincronización agregará, modificará o eliminará el mismo archivo en las otras ubicaciones. (7) Es muy frecuente que algunos sistemas de información no tengan sus datos centralizados, sino que los gestionen en múltiples aplicaciones, siendo necesario mantener la correspondencia entre las mismas.

La sincronización puede ser unidireccional, donde cada vez que se introduce o modifica información en una ubicación, se agrega o cambia en la otra, pero no en sentido contrario. En la bidireccional los archivos se copian en ambas ubicaciones, por lo que se mantiene la consistencia entre ellas.

1.4. Replicación de datos.

La replicación de datos es el proceso de copiar y mantener actualizados objetos de una base de datos, entendiéndose tablas, secuencias o disparadores (triggers³), de tal forma que los cambios realizados localmente, sean enviados a las demás bases de datos remotas para ser aplicados. Es un proceso utilizado para compartir objetos de una base de datos hacia múltiples

²Randomización es un procedimiento sistemático y reproducible por el cual los sujetos participantes en un ensayo clínico son distribuidos al azar en los distintos grupos de tratamiento.

³ Un trigger es una acción en cadena que empieza cuando un evento específico ocurre sobre una tabla específica.

locaciones, (8) consiste en copiar datos de un dispositivo de almacenamiento primario a un dispositivo secundario o de copia de seguridad. Si el dispositivo primario falla, los datos están disponibles en el secundario, lo que asegura alta disponibilidad y tolerancia ante errores graves. (9)

1.4.1 Servidor de replicación.

Un servidor de replicación coordina las actividades de replicación de datos entre servidores de base de datos o de intercambio de datos con otros servidores de replicación que se encuentren en otras regiones. La información que recibe un servidor de replicación puede provenir de:

- Transacciones de datos originadas en una base de datos primaria, para ser distribuidas a los sitios con suscripciones a ella.
- Transacciones enviadas por otro servidor de replicación para que ser aplicadas a sus bases de datos locales. (10)

1.4.2 Tipos de replicación de datos

La replicación de datos se puede clasificar en dos tipos dependiendo del modo en que se transmite la información y según las características que posee el lugar donde será empleada. La estabilidad de la red y la urgencia de la necesidad de la información son elementos fundamentales a tener en cuenta para clasificar la replicación, pudiendo ser utilizada la replicación en línea o fuera de línea.

Replicación de datos en línea: la réplica de un nodo a otro se realiza mediante una conexión de red, pudiendo ser programada la transmisión de datos por el usuario. Este tipo de replicación es recomendado en casos en que se considere una replicación en tiempo real. (10)

Replicación de datos fuera de línea: existen problemas de comunicación mediante la red o no existe tal comunicación. Admite cierto grado de desactualización en la información, ya que los nodos que replican datos no velan por el tiempo de latencia⁴ y utilizan ficheros portables para la actualización. (10)

⁴ Se denomina latencia a la suma de retardos temporales dentro de una red, es producido por la demora en la transmisión y propagación de paquetes.

1.4.3 Entornos de replicación de datos

Los entornos de replicación de datos dependen del gestor de base de datos y del sistema operativo que se emplee en cada nodo servidor involucrado en la replicación. Los entornos pueden ser los siguientes:

Entorno homogéneo: cuando la replicación se ejecuta entre nodos servidores de datos que contienen el mismo tipo de gestor de bases de datos y sistema operativo. Aunque este último puede variar.

Entorno heterogéneo: cuando la replicación de datos se ejecuta entre nodos servidores de datos que contienen diferentes gestores de base de datos y el mismo sistema operativo, aunque pueden no tener la misma plataforma. (10)

1.4.4 Dirección de transmisión de datos en la replicación

Las características del negocio y las necesidades de información son las que definen el sentido de transmisión de los datos en la replicación. Se debe definir uno de los siguientes sentidos:

Maestro–Esclavo (Master-Slave): es unidireccional, ya que los nodos esclavos solo ejecutan consultas de lectura sobre los datos del servidor maestro, realizando este último, consultas de lectura/escritura sobre los nodos restantes. En este tipo de dirección de transmisión el servidor maestro envía los cambios de datos al servidor esclavo de forma asincrónica, el esclavo responde a las consultas de sólo lectura, mientras que el servidor principal se está ejecutando. (11)

Multi–Maestro (Multi-Master): es bidireccional y en este tipo de dirección de transmisión, múltiples nodos maestros ejecutan entre ellos consultas de lectura/escritura, la información es actualizada en estos servidores maestros, manteniendo cada uno de estos un registro de cambios, para luego ser replicados en los demás servidores. (12)

1.4.5 Modelos de distribución de cambios

Los cambios producidos en los objetos de la base de datos que forman parte de la réplica se transmiten con uno de los siguientes modelos:

Sincrónico: es de consistencia fuerte, ya que asegura que todas las réplicas de datos estén actualizadas en línea. Los cambios que se ejecutan en un nodo maestro instantáneamente se

aplican en el nodo origen y en los destinos. Si se produjera algún fallo en el proceso de replicación se anula completamente la transacción de datos.

Asincrónico: es de consistencia débil y en este modelo los cambios realizados se acumulan en un nodo local, siendo enviados posteriormente a los restantes nodos. Funciona sin comunicación entre los nodos, pero necesita que los nodos servidores almacenen las modificaciones que se van generando en la replicación, para luego iniciar un proceso de reconciliación para determinar los valores actuales de cada nodo. (13)

1.4.6 Conflictos de la replicación de datos

- De actualización: una información determinada es actualizada simultáneamente por varias transacciones diferentes. (14)
- De eliminación: una transacción intenta eliminar datos que están siendo actualizados o eliminados por otra transacción en ese mismo momento. (14)
- De unicidad: ocurre cuando la réplica de un registro intenta violar una restricción de integridad, tal es el caso de la llave primaria. Un ejemplo es que se generen dos transacciones provenientes de dos nodos diferentes, que intentan insertar en su tabla, con el mismo valor de clave primaria. (14)
- De orden: acontece si se altera el orden de propagación a los nodos. Se produce cuando la replicación de datos posee un número superior a tres nodos servidores que actúan como maestros. En caso de que la transacción a un nodo determinado esté bloqueada o afectada, la replicación con los cambios en los datos se desarrolla a través de uno de los nodos maestros restantes, sin embargo luego de concluir el proceso, se evidencia que los datos que se debían transmitir hacia un nodo específico siguiendo un patrón o camino determinado, se realizó de forma diferente a lo planificado, por lo que puede producirse el conflicto. (14)

1.5. Sistemas analizados para la sincronización y replicación de datos

En la actualidad las instituciones poseen gran cantidad de información almacenada en diferentes bases de datos, aplicaciones y sistemas, por lo que necesitan el uso de software para la sincronización de datos y para mantener la coherencia en la información. A nivel internacional algunos de los sistemas existentes son:

1.5.1 SymmetricDS

La versión más actual de este sistema es la 3.5.17. Es un software replicador de datos con modelo de distribución asincrónico que permite subscriptores múltiples y sincronización bidireccional. Hace uso de tecnologías web y de bases de datos, realiza la réplica casi en tiempo real, trabaja con conexiones de bajo ancho de banda, y es resistente a períodos de inoperatividad de la red. (15)

El software puede ser instalado como una aplicación web dentro del servidor de aplicaciones Java, o puede ser incorporado a otra aplicación Java. Está escrito en Java 5 y requiere Java SE Runtime Environment (JRE) o Java SE Development Kit (JDK) versión 5.0 o superior.

Un Nodo es una única instalación de SymmetricDS, este se inicializa mediante un fichero con extensión .properties, y es configurado insertando datos de configuración en tablas específicas de la base de datos. (16) Seguidamente el Nodo crea disparadores en las tablas de aplicación especificadas, permitiendo así que los eventos sean capturados y entregados a otros Nodos SymmetricDS.

Soporta la sincronización entre diferentes plataformas de base de datos, entre las que están Oracle, SQL Server, MySQL, PostgreSQL, Microsoft Azure, Apache Derby, IBM DB2, Firebird, Informix, Greenplum. Está desarrollo con licencia GNU/LGPL, se puede utilizar tanto para Linux como para Windows, permite ser configurado para insertar o extraer datos, filtrar cuáles archivos incluir y excluir de la sincronización.

Algunos de los inconvenientes que presenta SymmetricDS son que no posee interfaz para administrar la sincronización lo cual dificulta el proceso para usuarios inexpertos, no limpia los archivos (no comprueba si los archivos fueron eliminados) lo que conduce a errores en los registros, el fondo inicial de carga puede quedar atrapado en un bucle y los procesos de extracción de bases de datos pueden quedar superpuestos después de un corte en la conexión ya que no detecta ni soluciona los conflictos de red.

1.5.2 CronSQL

CronSQL es una herramienta que permite realizar múltiples sincronizaciones entre varias bases de datos. Hace coincidir datos entre múltiples programas e incluso entre una página web, y permite mantener diferentes sistemas de gestión de información sincronizados total o parcialmente.

CronSQL soporta conexiones a cualquier tipo de base de datos compatible con Object Linking and Embedding for Databases (OLEDB, traducido al español como "Enlace e incrustación de objetos para bases de datos"), entre las que se encuentran Oracle, Microsoft SQL Server y Sybase Adaptive Server Enterprise. Esta herramienta es capaz de conectar a través de una interfaz desarrollada en PHP con bases de datos de servidores web con el acceso externo restringido. (17)

Tiene una interfaz visual, intuitiva y sencilla, pero para configurar la sincronización en CronSQL, es necesario tener conocimientos básicos en consultas SQL y sobre el modelo de datos de las bases de datos a sincronizar. Es por ello que esta es una herramienta muy adecuada para programadores, administradores de páginas web y administradores de bases de datos; también para empresas que dispongan de distintas soluciones de software heterogéneo sin ningún mecanismo de sincronización y que precisen centralizar la información de forma rápida y sencilla. (18) Otro de los inconvenientes que muestra CronSQL es que su licencia es privada, lo que supone gastos adicionales.

1.5.3 DBMoto

Herramienta que se utiliza para la replicación e integración de datos en tiempo real, que contiene una interfaz gráfica y amigable (Enterprise Manager). Su última versión es DBMoto 7 y entre las bases de datos que soporta están: Oracle, Microsoft SQL Server, MySQL, SQL Anywhere, y PostgreSQL. Esta herramienta posee distintas formas de replicación: Refresh (fotografía), Mirroring (transaccional unidireccional) y Synchronization (transaccional bidireccional), lo que permite que la entrega de datos sea fácil e instantánea y que el servidor de bases de datos mantenga la estabilidad. La replicación se realiza mediante reglas y funciones que deben ser definidas, almacenadas y ejecutadas. (19) Como recursos utiliza Windows Server 2012/2008/2003, Microsoft .NET Framework 4.0 o superior.

DBMoto crea reportes gráficos de actividades y accesos, lo que permite la realización de auditorías para el análisis de los datos, cuenta con una consola de errores para solucionar problemas que se presenten después de una replicación, realiza un grupo de replications, lo que reduce el tiempo de conectividad y tráfico entre el origen y el destino, posibilita la creación de tablas automáticas en el destino, garantizando un rápido despliegue de la información.

Algunas de las ventajas que ofrece DBMoto es que controla los datos mediante secuencias de comandos y eventos, sin tener en cuenta la base de datos asociada y se puede configurar para

que genere notificaciones vía correo electrónico si ocurre algún suceso crítico. Entre sus desventajas se encuentra que si un tiempo muerto de conexión se produce cuando se intenta conectar al servidor de base de datos para procesar la transacción, DBMoto reporta un error de conexión en los registros, e intenta volver a conectar en el próximo intervalo de tiempo. Un comando de error por tiempo muerto puede causar el fallo de la replicación. Otra de las desventajas que manifiesta es que representa un gasto enorme en el pago de una costosa licencia. (20)

1.5.4 SharePlex

SharePlex for Oracle es una solución de replicación y de integración de datos para Oracle 10g, 11g, 12c, y otras bases de datos. (21) Hace una copia casi en tiempo real de los datos.

Algunos de los beneficios que tiene el uso de SharePlex son: (22)

- Mejora el rendimiento de los sistemas de Procesamiento de Transacciones En Línea (OLTP por sus siglas en inglés On-Line Transaction Processing)
- Optimiza el uso de aplicaciones de inteligencia comercial
- Garantiza una alta disponibilidad y una rápida recuperación de desastres
- Aprovecha informes en tiempo real y almacenamiento de datos
- Crea informes de operación y archivos
- Elimina interrupciones planificadas al realizar ajustes, actualizaciones y migraciones
- Informes centralizados o consolidación
- Mejora el desempeño de sistemas transaccionales

SharePlex puede proporcionar integridad, monitoreo, comparación y reparación de los datos, todo en un paquete. Ofrece soporte para características avanzadas de seguridad Oracle, en las que se incluyen el Cifrado de Datos Transparente (TDE por sus siglas en inglés) y la Compresión de Columnas Híbridas (HCC por sus siglas en inglés). TDE protege mediante encriptación la información confidencial de las organizaciones, mientras HCC permite maximizar el espacio de almacenamiento y rendimiento, ya que se almacenan más datos en discos.

SharePlex 8.0 (versión más actual) permite la réplica de datos desde Oracle a varias bases de datos entre las que están: SQL Server, DB2, Sybase, Netezza o Teradata, Hadoop y Greenplum.

Requisitos que debe tener el sistema: (23)

- Memoria
 - Procesos SharePlex: 64 bits y puede superar los 4 GB.
 - Memoria de procesos: superior o igual a 256 MB.
- Software adicional
 - SQL * Plus: programa de línea de comandos de Oracle que ejecuta comandos SQL de forma interactiva mediante un script.
- Requisitos adicionales
 - Se recomienda descriptores de archivos del sistema de un mínimo de 1024 o superior, según lo permita el sistema.

SharePlex tiene un precio, por lo que son necesarias otras soluciones factibles que contribuyan a la independencia tecnológica. Shareplex no puede replicar ciertos componentes que son usados por Oracle, dentro de ellos se pueden mencionar los siguientes: Alter Table para borrar una columna, agregar una columna de objetos largos (LOB, LARGE OBJECTS); además las bases de datos origen y destino deben usar la misma versión de SharePlex. (24)

1.5.5 Slony

Slony-I es un sistema de replicación “maestro a múltiples esclavos” para PostgreSQL apoyado en réplica de cascadas. Incluye todos los rasgos y capacidades necesarios para reproducir grandes bases de datos a un número razonable y limitado de sistemas esclavos (aproximadamente 12 esclavos inmediatos). (25) No es recomendado para sistemas con problemas en la conexión de red, ya que necesita que todos los nodos estén disponibles todo el tiempo. Slony-I realiza una réplica de espejos, exactamente igual al origen de datos y posee una configuración avanzada.

Ventajas de Slony-I:

- Las actualizaciones se transmiten en tiempo real desde el nodo servidor maestro a las bases de datos esclavas y en caso de ocurra un fallo en el nodo origen, no se afecta el trabajo con los nodos destino. (26) (27)

Desventajas de Slony-I:

- Tiene una compleja instalación, configuración y administración. (28)

- El nodo maestro para escritura es único. (28)
- No es recomendado para sistemas con conexiones inestables o configuraciones variables. (28)
- Las bases de datos esclavas no se modifican.
- No detecta los fallos que puedan ocurrir en la red. (29)
- Sólo soporta el servidor de base de datos PostgreSQL, replica sólo entre bases de datos con estructuras iguales y no tiene la capacidad de replicar objetos de gran tamaño. (28)

Entre los sistemas existentes a nivel nacional están los siguientes:

1.5.6 Reko

Es un software para replicar información entre bases de datos que posean iguales o diferentes estructuras. Fue desarrollado en la UCI y cubre necesidades como la protección, recuperación, sincronización y transferencia de datos entre diversas localizaciones, o la centralización de la información en un único lugar. Reko 3.0, versión ya terminada de este sistema, soporta las bases de datos Oracle y PostgreSQL, mientras que a la versión 3.1 que está actualmente en desarrollo se le añade MySQLServer. Es un sistema de replicación multi-maestro con modelo de distribución de cambios asíncrono o síncrono, permite la réplica y sincronización de datos con conexión y sin conexión, la selección de los datos que se desean replicar y la definición de filtros a aplicar en la replicación. (30)

Su diseño e implementación fueron realizados utilizando herramientas Open Source y librerías de clases con licencia gratuita. Fue implementado en la plataforma Java Enterprise Edition (JEE), soporta la replicación de transacciones a través de Hibernate y la de acciones por vía de disparadores. Ha sido registrado como producto y se ha liberado por Calisoft⁵ UCI en dos ocasiones, constituye una solución soberana de alto impacto en la sustitución de importaciones dado que es una solución madura construida sobre una arquitectura en tecnologías modernas y libres.

Presenta las siguientes características y/o funciones:

⁵ Calisoft: Centro Nacional de Calidad de Software, el principal reto de sus acciones es el desarrollo del modelo cubano para el perfeccionamiento de aplicaciones informáticas en la industria nacional.

- Soporte para réplica de datos entre gestores diferentes (PostgreSQL-Oracle).
- Puede ser instalado en las plataformas Windows y Linux, para funcionar no necesita un ambiente gráfico.
- Permite conocer el estado de los datos de la réplica en tiempo real a través de una interfaz.
- Detecta los errores de conexión. En caso de ocurrir alguno, mantiene los datos de réplica en un estado estable, al restablecerse la conexión, sincroniza los datos entre las bases de datos automáticamente.
- La transferencia de los datos de replicación puede efectuarse sobre los protocolos TCP/IP, FTP, HTTP o por ficheros de forma manual.
- Los datos son enviados por la vía de los protocolos seguros HTTPS y SSL. (31)

1.5.7 Magic@ Data Replication eXtensible Solution

Es una herramienta desarrollada en la UCI con dirección de transmisión multi-maestra, modelo de distribución de cambios asincrónico, y entre bases de datos relacionales con gestores diferentes. Es un software multiplataforma, permite un gran aprovechamiento del ancho de banda, el mínimo impacto en el rendimiento, el intercambio bidireccional en la replicación de los datos, además admite configuraciones de sincronización entre los sistemas que no se conecten mediante una red física.

Ventajas: (16)

- Facilita el mapeo entre estructuras diferentes.
- Soporta la aplicación de filtros.
- Extensible para diversos gestores de bases de datos.
- Propicia la realización de transformaciones a los datos, mediante el proceso de (Extracción, Transformación y Carga).

Desventajas: (16)

- No permite la detección ni solución de conflictos.
- Las configuraciones realizadas no pueden ser guardadas para usos posteriores.

- Emplea un conjunto de herramientas propietarias para su implementación.

1.6. Resultado del análisis de las soluciones estudiadas

Después de realizado el análisis de los sistemas escogidos a nivel internacional y nacional, se puede apreciar que los mismos tienen características importantes para el proceso de sincronización y replicación de bases de datos. Se llega a la conclusión de que las soluciones robustas en esta temática son muy costosas y las que existen en software libre están incompletas y son difíciles de configurar. Otro de los inconvenientes encontrados es que no cumplen con características que son fundamentales para el CIM y los centros hospitalarios, entre las que se encuentran brindar la opción de importar y exportar los datos a un determinado formato, y serializar y decodificar la información para mantener la confidencialidad de la organización.

Uno de los sistemas analizados fue la herramienta Reko, que sin dudas fue la que mayor impacto causó por sus ventajas y excelentes funcionalidades; pero que no es recomendable usar para replicar la base de datos que posee el CIM, por problemas conocidos de inestabilidad con los identificadores y resolución de problemas de unicidad. Por tanto ninguno de los sistemas analizados ofrece una solución completa para satisfacer los aspectos necesarios para la institución, sin embargo pueden ser utilizadas algunas de las particularidades que presentan, como base o guía para desarrollar una solución propia que responda a las especificaciones del CIM y los centros hospitalarios.

A continuación se describen la metodología, tecnologías y herramientas utilizadas para dar solución al problema planteado, las cuales sirven de guía y apoyo para el proceso de desarrollo de software; estas fueron definidas por el arquitecto del proyecto Ensayos Clínicos y se encuentran precisadas en el documento “SES_SISEC_010216_5 ArqSWVtaEntorno_v2.0”.

1.7. Metodología

En el proceso de desarrollo de un software es indispensable la utilización de metodologías que sean una guía para el equipo de trabajo, que faciliten y garanticen un correcto y eficiente desarrollo del sistema. Para garantizar la excelencia de un producto de software es indispensable seguir un conjunto de procedimientos, técnicas, herramientas y documentación que ayuda a los desarrolladores a la hora de crear sus aplicaciones. Su denominación es

metodología de desarrollo y cumplir con los requisitos iniciales y minimizar las pérdidas de tiempo en el proceso de desarrollo, son sus principales objetivos.

La metodología a utilizar en el desarrollo del presente proyecto es **Rational Unified Process (RUP)**. Su objetivo es asegurar la producción de software de alta calidad que satisfaga la necesidad del usuario final dentro de un tiempo y presupuesto previsible.

Las características del ciclo de vida de RUP que serán utilizadas por el equipo de desarrollo son:

- Dirigido por casos de uso: las necesidades y deseos de los clientes se reflejan en los casos de uso. Estos son los que guían y dirigen el proceso de desarrollo del software en su totalidad, son la entrada fundamental cuando se identifican y especifican las clases, subsistemas e interfaces, y cuando se planifican las iteraciones del desarrollo del sistema. La mayoría de las actividades como el análisis, diseño y prueba son llevadas a cabo partiendo de los casos de uso. (32)
- Centrado en la arquitectura: La arquitectura es la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo o aceptar. Esta comprende decisiones sobre la organización del sistema, los elementos estructurales que lo compondrían, sus interfaces, sus colaboraciones y su composición. La arquitectura es necesaria para entender el sistema, organizar el desarrollo, fomentar la reutilización y hacer evolucionar el sistema. (32)

Una particularidad de este proceso es que se realiza mediante iteraciones y en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo de software. (33)

RUP divide en 4 fases el desarrollo del software:

- Inicio: se define el alcance del proyecto, trata de responder a las preguntas de si es factible construir un nuevo proyecto o comprarlo; además se establece el ámbito y los límites del proyecto, se estima los riesgos y los costes en términos de recursos y tiempos.

- Elaboración: el propósito de esta fase es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos. (34)
- Construcción: es la etapa de implementación, el objetivo principal de esta fase es lograr la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones.
- Transición: fin del proyecto, su finalidad es poner el software en manos de los usuarios finales.

Además establece nueve disciplinas o flujos de trabajo, seis de ingeniería y tres de apoyo a realizar en cada fase del proyecto, de las cuales son utilizadas las siguientes:

Disciplina de desarrollo: determina las etapas que se realizarán durante el desarrollo del proyecto.

- Ingeniería o modelado del negocio: se hace un análisis y entendimiento de las necesidades del negocio.
- Requerimientos: establece qué es exactamente lo que debe hacer el sistema a construir, se hace la especificación de requisitos, la cual debe ser entendida y aceptada por los usuarios finales.
- Análisis y Diseño: trasladando los requerimientos dentro de la arquitectura de software. (33) El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema. (34)
- Implementación: creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado, se implementan las clases en ficheros ejecutables.

Las características principales que presenta RUP son:

- Es una forma disciplinada para la asignación de tareas y responsabilidades.
- Implementación de las mejores prácticas en Ingeniería de Software.
- El desarrollo es iterativo, administra los requisitos, usa la arquitectura basada en componentes, se realiza un control de los cambios que se efectúan, el software es modelado de forma visual y se verifica la calidad del software.

1.8. Herramientas y tecnologías

Además se hace necesario el uso de herramientas y tecnologías que faciliten el desarrollo del sistema.

Lenguaje de programación

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principio de los años 90's. (35) Permite crear varios tipos de aplicaciones, entre las que se destacan las cliente-servidor y las Java Server Pages. Tiene Licencia Pública General de GNU, es multiplataforma y es un lenguaje de programación robusto. Tiene mecanismos de seguridad incorporados, los que controlan el acceso a recursos de las máquinas donde se está ejecutando, puede incorporar herramientas de documentación e impide el uso de técnicas inadecuadas de programación. (36)

Algunas de las características de este lenguaje son:

- Orientado a objetos: da soporte a las técnicas de desarrollo de la programación orientada a objetos y a la reutilización de componentes de software.
- Distribuido: diseñado para trabajar en ambiente de redes, posee una gran biblioteca de clases para la utilización los protocolos TCP/IP, HTTP y FTP.
- Interpretado: se traduce cada fichero fuente de clases a código de bytes, pudiendo ser interpretado este por todas las máquinas que den soporte a un visualizador de que funcione con Java.
- Sólido: no se quiebra fácilmente por errores de programación. (37)

Entorno de Desarrollo Integrado (IDE)

NetBeans 7.4 es un IDE de código abierto y gratuito para los desarrolladores de software que posee las herramientas necesarias para la creación de aplicaciones de escritorio profesionales, empresariales, web y móviles. Puede utilizar lenguajes como Java, JavaFX, C/C ++, PHP, JavaScript, Groovy. El proyecto NetBeans fue fundado en el año 2000 por Sun Microsystems, es renovado y mejorado constantemente ya que se considera el IDE estándar en el desarrollo con Java. (38)

Características de NetBeans:

- Es de fácil instalación, uso y actualización.
- Es licenciado bajo la Common Development and Distribution License ⁶(CDDL), una licencia basada en la Mozilla Public License ⁷(MPL).
- Posee mejoras para el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) y para la Arquitectura Orientada a Sistemas (SOA, por sus siglas en inglés).
- Brinda soporte para PHP y Symfony.
- Puede ser utilizado en múltiples plataformas como Windows, Linux, Mac OS X y Solaris.
- La programación se realiza mediante módulos. (39)

Lenguaje de Modelado

UML 2.0 es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y reglas para permitir la comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. Lo hace mediante diagramas y esquemas desde una perspectiva orientada a objetos. (40)

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones: (40)

- Visualizar: permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: permite especificar cuáles son las características de un sistema antes de su construcción
- Construir: a partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión. (40)

⁶ Common Development and Distribution License: Licencia Común de Desarrollo y Distribución es una licencia producida por Sun Microsystems, de código abierto y libre.

⁷ Mozilla Public License: Licencia Pública de Mozilla es una licencia desarrollada por Netscape Communications Corporation, de código abierto y de software libre, que fue posteriormente traspasada a la Fundación Mozilla.

Herramienta CASE

Las herramientas CASE (Computer Aided Software Engineering, o traducido al español Ingeniería de Software Asistida por Computación) son una forma para modelar los procesos de negocios.

Visual Paradigm for UML 8.0 Enterprise Edition es una herramienta concebida para soportar el ciclo de vida completo del proceso de desarrollo de software: análisis y diseño, construcción, pruebas y despliegue, admite la elaboración de todos los tipos de diagramas de clases, código inverso, permite la generación de código desde diagramas, importar y exportar para otras herramientas case y generar documentación de diferentes extensiones.

Facilita la realización de los procesos de ingeniería e ingeniería inversa, el modelado con Sistema de Versiones Concurrentes (CVS por sus siglas en inglés) y Subversión (control de versiones), y la generación de bases de datos. Soporta múltiples usuarios trabajando en el mismo proyecto, por lo que se dice que es una herramienta colaborativa. Tiene licencia gratuita y comercial. (41)

Gestor de Bases de Datos

PostgreSQL 9.2 es un sistema para la gestión de base de datos relacional orientada a objetos, publicado bajo la licencia Berkeley Software Distribution, es un proyecto de código abierto dirigido por una comunidad de desarrolladores.

Principales características:

- Alta concurrencia: permite el acceso y manipulación de una tabla por diferentes procesos sin necesidad de bloquear la misma, permitiendo que cada usuario tenga una visión consistente de lo último a lo que se le hizo commit⁸.
- Provee soporte para números de precisión arbitraria, texto de largo ilimitado, figuras geométricas, direcciones IP, direcciones MAC y arreglos.

Administrador de Bases de Datos

⁸ En la ciencia de la computación y la gestión de datos, commit (acción de comprometer) se refiere a establecer un conjunto de cambios, una sentencia commit en SQL se utiliza para finalizar una transacción de base de datos.

pgAdmin III 1.16.1 es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, posee la licencia Open Source, puede ser usada en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como otras de tipo comercial como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres. (42) Responde a todo tipo de necesidades que pueda presentar un usuario, desde realizar simples consultas SQL hasta desarrollar complejas bases de datos. La interfaz gráfica facilita la administración de PostgreSQL, la aplicación brinda soporte para el motor de replicación Slony-I, la conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets y puede encriptarse mediante SSL para mayor seguridad.

Herramientas de Mapeo de Objeto Relacional

Hibernate es un entorno de trabajo que facilita la consulta de bases de datos para obtener objetos y el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación. Es software libre, distribuido bajo los términos de la licencia GNU y Licencia Pública General Reducida, permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Ofrece además un lenguaje de consulta de datos llamado Hibernate Query Language (HQL), al mismo tiempo que una Application Programming Interface (Interfaz de Programación de Aplicaciones, API por sus siglas en inglés) para construir las consultas programáticamente. (43)

Entre las ventajas que posee incluye su total independencia con el motor de base de datos, ya que la capa de datos y la de lógica del negocio son totalmente independientes, lo cual propicia que para cambiar el motor de base de datos solo hay que cambiar una línea de un fichero de conexión, además evita mucho del código confuso de la capa de persistencia, permitiendo centrarse en la lógica de negocio. (44)

Librería de interfaz gráfica

Swing es una biblioteca gráfica para el lenguaje de programación Java que incluye widgets⁹ para interfaz gráfica de usuario tales como cajas de texto, botones, desplegables y tablas. Posee una amplia variedad de componentes, el aspecto de las interfaces y los diálogos pueden

⁹Widgets: pequeña aplicación o programa presentado en archivos o ficheros pequeños cuyos objetivos son dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.

ser personalizados, es posible crear escritorios virtuales o interfaz de múltiples documentos, se facilita la generación de interfaces que ayuden a la accesibilidad de discapacitados, por ejemplo en Braille, presenta componentes para tablas y árboles de datos, potentes manipuladores de texto, capacidad para "deshacer", una gestión mejorada de la entrada del usuario y cualquier componente puede estar anidado en otro.

En el capítulo se realizó un análisis de los principales antecedentes a nivel nacional e internacional de los sistemas de replicación y sincronización, lo cual permitió que se estudiaran los conceptos relacionados con el tema propuesto en la investigación. Con el estudio de los sistemas existentes se logró determinar si algunos de los sistemas es factible para dar solución al problema de la investigación, identificándose que ninguno de ellos cumple con todos los requisitos necesarios para dar cumplimiento a los objetivos planteados en la investigación, debido a que las aplicaciones robustas de replicación poseen licencia privativa y las herramientas libres no permiten sincronizar las bases de datos con las que se trabaja. Además se realizó una descripción de la metodología, herramientas y tecnologías que se utilizarán para dar respuesta al problema planteado.

Capítulo 2. Características del sistema

Este capítulo contiene las características del sistema a construir, la captura de los requisitos funcionales y no funcionales, el diagrama de casos de uso del sistema, así como las respectivas descripciones de los casos de uso considerados más significativos.

2.1. Solución Propuesta

Para dar solución a la problemática planteada, se propone desarrollar funcionalidades sobre tecnologías de escritorio, que permitan la sincronización de la información existente en las bases de datos de los sistemas Clínicas 1.0 y SIGECConductor. Se utilizarán las tablas de la base de datos relacionadas con los cronogramas, los estudios y la conducción de los EC. Los usuarios que interactúan con las funcionalidades serán: el gerente de datos, encargado de exportar los cronogramas asociados a cada centro y otros usuarios (preferentemente el informático de cada centro hospitalario), que se encargará de exportar la conducción de los estudios que se realicen en su institución hospitalaria.

2.2. Modelo del dominio

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés, muestra (a los modeladores) clases conceptuales significativas en un dominio del problema. (45) Consiste en una serie de diagramas de clases que representan los conceptos del entorno de un problema y sus relaciones.

En la Figura 1 se describe detalladamente los conceptos más importantes y sus relaciones en el Modelo de Dominio correspondiente al sistema a desarrollar.

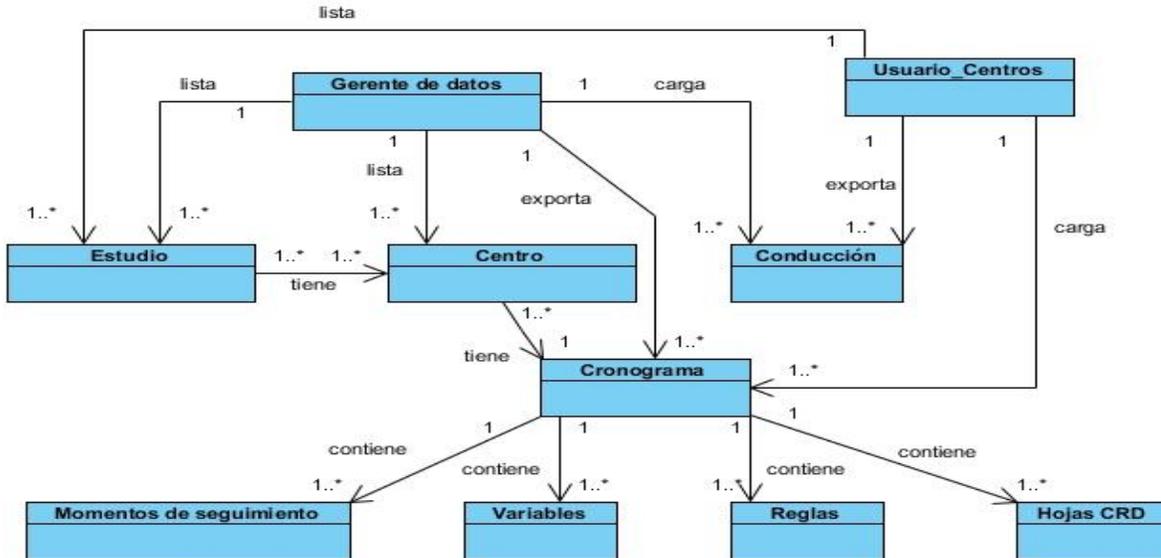


Figura.1 Modelo de Dominio.

2.2.1 Descripción de las clases

Gerente de datos: esta clase representa a la persona que gestiona la información que se encuentra en la parte del sistema que representa al sistema Clínicas 1.0. Es la encargada de exportar el cronograma y cargar la conducción de los estudios realizados en los hospitales.

Usuario_Centros: esta clase representa a la persona que gestiona la información que se encuentra en la parte del sistema que representa al SIGECConductor. Es la encargada de exportar la conducción y cargar el cronograma realizado en el CIM.

Estudio: representa la investigación, análisis o enfermedad por la cual se realizan los ensayos clínicos en los diferentes hospitales.

Centro: simboliza los hospitales o centros que se dedican a la realización de estudios sobre diferentes enfermedades y a la conducción de los ensayos clínicos.

Cronograma: constituye la asociación de una planificación general de un estudio determinado en un centro específico. Entre los datos que contiene están el estudio, centro, usuarios, momentos de seguimiento, variables, reglas y hojas CRD.

Conducción: representa la ejecución de ensayos clínicos en algún hospital o centro encargado de esta función.

2.3. Especificación de los requisitos de software

Los requisitos determinan cómo funcionará el sistema, son restricciones sobre su operación e implementación. Son “condiciones o capacidades que necesita el usuario para resolver un problema o conseguir un objetivo determinado”, también se aplica a las condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación. (46)

2.3.1 Requisitos funcionales

Los requisitos funcionales (RF) son una definición de los servicios que el sistema debe proporcionar, cómo debe reaccionar a una entrada particular y cómo se debe comportar ante situaciones particulares, describen el funcionamiento del sistema. (46)

A continuación se especifican los requisitos funcionales:

RF1 - Autenticar usuario

El sistema debe permitir que los usuarios se autenticuen, verificando los permisos que poseen para acceder al sistema y los que tienen sobre los centros y estudios, otorgando el acceso al software.

RF2 - Listar estudios

El sistema debe permitir al usuario ver la lista de los estudios que se están realizando.

RF3 - Listar centros a partir de un estudio

Permitirá visualizar la lista de los centros que realizan un estudio anteriormente seleccionado.

RF4 - Exportar cronograma

El sistema debe permitir exportar el cronograma asociado al centro y estudio en cuestión, con todos los datos vinculados a estos: estudio, centro, usuarios, roles, momentos de seguimiento, variables, reglas, hojas CRD.

RF5 - Cargar la conducción de los estudios realizados

El sistema debe permitir cargar la conducción de los estudios realizados en los diferentes centros hospitalarios.

RF6 - Cargar para la base de datos el cronograma

El sistema debe permitir cargar para la base de datos de los centros hospitalarios toda la información asociada al cronograma (estudio, centro, usuarios, roles, momentos de seguimiento, variables, reglas, hojas CRD).

RF7 - Exportar conducción

Permitirá exportar la conducción del estudio realizado en el centro y todos los datos asociados a esta.

RF8- Listar estudios del centro

Permite ver la lista de los estudios que se realizan en un centro determinado.

2.3.2 Requisitos no funcionales

Describen aspectos del sistema que son visibles por el usuario, no incluyen una relación directa con el comportamiento funcional del sistema. Los requerimientos no funcionales incluyen restricciones como el tiempo de respuesta (desempeño), la precisión, recursos consumidos, seguridad, etc. (47)

A continuación se listan los requisitos no funcionales del sistema que se encuentran reflejados en el documento "SES_SICEC_010113_ERS_v1.0".

✓ Usabilidad

Las personas que interactuarán con el software son informáticos del CIM y de los centros hospitalarios, la aplicación tendrá un ambiente sencillo, será fácil de manejar y sus funcionalidades serán accedidas de forma intuitiva, incluso para aquellos que no han tenido mucha experiencia en el trabajo con computadoras o con sistemas informáticos

✓ Seguridad

Todo acceso o manipulación del sistema estará subordinado a un proceso de autenticación del usuario, donde será especificado el usuario y contraseña. Las contraseñas deberán tener

más de siete caracteres de longitud y tener una fortaleza media, además las mismas serán encriptadas mediante el algoritmo MD5. Se verificarán los permisos que posee el usuario y las actividades que le son permitidas realizar. Los archivos serán guardados de forma serializada, proporcionando seguridad a la hora de transportarlos en algún dispositivo, siendo decodificados al ser importados nuevamente en la aplicación.

✓ **Extensibilidad**

Se debe lograr un diseño adaptable, con la capacidad de poder soportar funcionalidades adicionales o modificar las existentes sin impactar el resto de los requerimientos contemplados en el sistema.

✓ **Interfaz**

Las interfaces no tendrán muchas imágenes y poseerán pocos colores, cada módulo tendrá una interfaz con las funciones que le corresponden y contendrán bien estructurados los datos, permitiendo la interpretación correcta de la información a los usuarios. La entrada de datos incorrecta será detectada e informada al usuario.

✓ **Software**

Debe estar instalada una máquina virtual de Java, independientemente del sistema operativo que presente.

✓ **Hardware**

Procesador Celeron o PIII a 1.72 GHz o superior.

Memoria RAM de 128 MB o superior.

2.4. Patrones de Casos de uso

Los patrones de casos de uso ayudan a describir qué es lo que el sistema debe hacer, describen el uso del sistema y cómo este interactúa con los diferentes usuarios, capturan buenas prácticas para modelar casos de uso. El patrón de casos de uso a utilizar es el Múltiples Actores: Rol Común. Este se manifiesta cuando existe más de un actor que ingresa a un mismo caso de uso y estos tienen un rol común.

2.5. Actores del sistema

Un actor del sistema representa a la persona que interactúa con la aplicación, en la tabla 1 se plasman los actores del sistema con sus respectivas descripciones para lograr un mayor entendimiento acerca de las responsabilidades de cada uno.

Actores	Descripción
Usuario	Persona que interactúa con el software, puede ser el Gerente de datos o un usuario de los centros hospitalarios.
Gerente de datos	Es el encargado de administrar toda la información referente al sistema Clínicas 1.0, este actor pertenece al CIM y operará con el sistema que será desplegado en este centro.
Usuario_Centros	Es el encargado de administrar toda la información referente al SIGECConductor; este actor pertenece a los centros hospitalarios que realizan la conducción de los ensayos clínicos y operará con el sistema que será desplegado en estos centros.

Tabla 1 Descripción de los actores del sistema

2.6. Diagrama de casos de uso del sistema

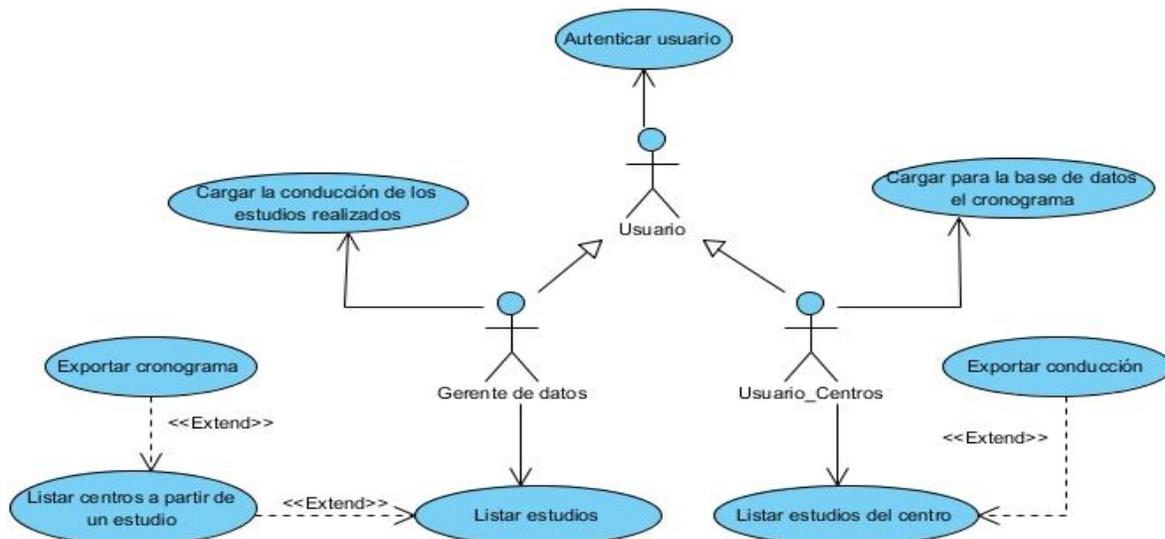


Figura.2 Diagrama de casos de uso del sistema

2.7. Descripción de los casos de uso del sistema

A continuación se describe el caso de uso Exportar conducción del sistema a desarrollar, detallando la secuencia de eventos que el actor lleva a cabo para completar un proceso determinado a través de la aplicación. Para dirigirse a los restantes casos de uso ver [Anexo 2](#).

Tabla 2. Caso de Uso: Exportar conducción

Objetivo	Exportar la conducción asociada a un centro específico.	
Actores	Usuario_Centros	
Resumen	El caso de uso inicia cuando el actor estando en la interfaz principal del sistema selecciona la opción Exportar. El sistema muestra una nueva interfaz donde permite al usuario seleccionar el estudio deseado y ejecutar la opción Exportar Conducción. Permite guardar un archivo serializado dentro de una ubicación local especificada por el actor, este archivo contiene toda la información referente a la conducción de un estudio.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado en su módulo correspondiente.	
Postcondiciones	Se exporta el archivo perteneciente a la conducción de un estudio en un centro determinado.	
Flujo de eventos		
Flujo básico Exportar la conducción		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor estando en la interfaz principal del sistema selecciona la opción Exportar.	
2.		Abre una interfaz que muestra el listado de los estudios que se realizan en el centro; Ver caso de uso: Listar estudios del centro . Y permite: <ul style="list-style-type: none"> • Exportar Conducción.
3.	El usuario selecciona el estudio deseado.	
4.		Sombrea el estudio seleccionado.
5.	Selecciona la opción Exportar Conducción.	
6.		Abre una ventana que permite al usuario explorar los archivos y carpetas existentes en el ordenador y muestra los siguientes campos: <ul style="list-style-type: none"> • Lista de ubicaciones dentro del ordenador.

		<ul style="list-style-type: none"> • Lista de elementos dentro de una ubicación. • Nombre del archivo. • Archivos de tipo. Permite: <ul style="list-style-type: none"> • Guardar. • Cancelar.
7.	Selecciona la ubicación deseada dentro del ordenador.	
8.		Actualiza la lista de elementos disponibles dentro de la ubicación.
9.	Escribe el nombre del archivo a exportar y selecciona la opción Guardar.	
10		Verifica los datos entrados y el espacio disponible.
11		Guarda el archivo en la ubicación especificada por el usuario, finalizando así el caso de uso.
Flujos alternos		
9.a El usuario selecciona la opción Cancelar.		
	Actor	Sistema
1.		El sistema cierra la ventana de exploración retornando al usuario al paso 4 del Flujo Normal de Eventos.
9.b Existe un error en el nombre del archivo.		
	Actor	Sistema
1.		El sistema indica mediante una ventana de diálogo que existe un error al guardar el archivo porque el campo nombre está incorrecto. Permite: <ul style="list-style-type: none"> • Aceptar.
2.	Selecciona la opción Aceptar.	
3.		Cierra el diálogo retornando al paso 4 del Flujo Normal de Eventos.
10.a No existe espacio disponible en la ubicación seleccionada.		
	Actor	Sistema
1.		El sistema indica mediante una ventana de diálogo que existe un error al guardar el archivo porque no se dispone de espacio en disco para ello. Permite: <ul style="list-style-type: none"> • Aceptar.
2.	Selecciona la opción Aceptar.	

3.		Cierra el diálogo retornando al paso 8 del Flujo Normal de Eventos.
Relaciones	CU incluidos	No existe
	CU extendidos	No existe

A partir del resultado del análisis de la gestión de los datos en la conducción de ensayos clínicos en el CIM y en los diferentes centros hospitalarios que realizan estos estudios, se pudieron identificar las necesidades del cliente para simplificar esta gestión. Una vez obtenidas estas necesidades se identificaron las funcionalidades y cualidades con que debe contar el componente de sincronización, las cuales quedaron descritas en la realización de este capítulo.

Capítulo 3. Análisis y diseño del sistema

Con el desarrollo del presente capítulo se efectuará el análisis y diseño de la propuesta de solución para el componente de sincronización, se puntualizará la arquitectura y los patrones a utilizar para el diseño e implementación del mismo, obteniéndose los principales artefactos como son los diagramas de clases del análisis, del diseño y de interacción, con sus respectivas descripciones.

3.1. Descripción de la arquitectura.

La arquitectura del software de un programa o sistema de cómputo es la estructura o las estructuras del sistema, que incluyen los componentes del software, las propiedades visibles externamente de esos componentes y las relaciones entre ellos. (48) Es una representación que permite que un ingeniero del software analice la efectividad del diseño para completar los requisitos establecidos y minimice los riesgos coligados al desarrollo del software.

3.1.1 Patrón Arquitectónico

Para el diseño arquitectónico del sistema a desarrollar se utilizó el patrón arquitectónico Modelo Vista Controlador (MVC), este separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

- **Modelo:** administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).
- **Vista:** maneja la visualización de la información.
- **Controlador:** interpreta las acciones del ratón y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado. (49)

Las clases vista y controlador dependen de la clase modelo, no dependiendo esta última de ninguna de las clases, lo cual permite construir y comprobar el modelo de forma independiente a la representación visual.

De forma general la clase modelo agrupa los estados de la aplicación, responde a los requerimientos, muestra la funcionalidad de la aplicación y notifica los cambios a la Vista; la

clase controlador define el comportamiento de la aplicación, mapea las acciones del usuario a actualizaciones del Modelo y selecciona la Vista de respuesta; por su parte la clase Vista interpreta el modelo, solicita actualizaciones del modelo, envía las acciones del usuario al Controlador y permite a este último seleccionar las Vistas. (50)

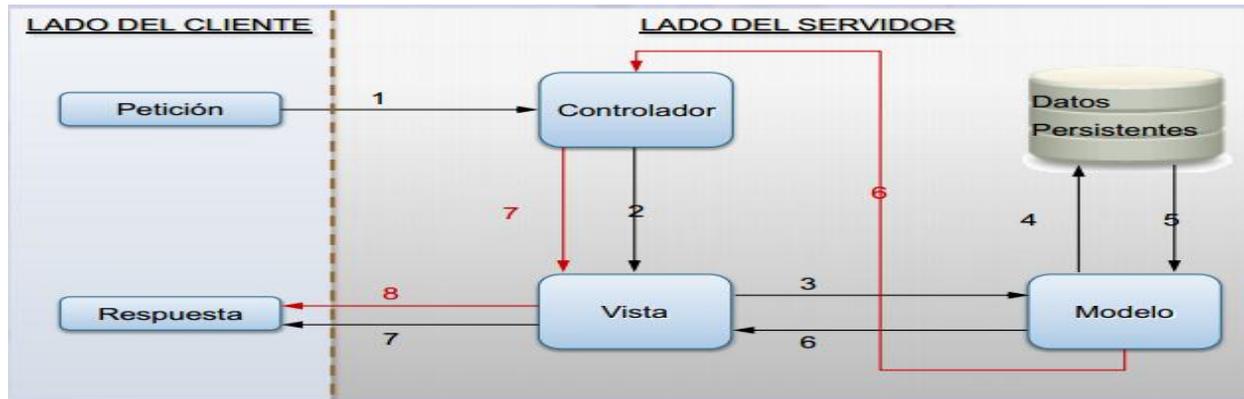


Figura.3 Funcionamiento del patrón Modelo Vista Controlador.

3.2. Modelo de Análisis

En el modelo de análisis se perfeccionan los requisitos identificados previamente, para obtener una visión más precisa de ellos y una representación de los mismos que ayude a organizar el sistema. Su función es facilitar al equipo de desarrollo el proceso de comprender cómo debería ser diseñado e implementado el sistema.

3.2.1 Diagramas de clases del análisis

Para la elaboración de los diagramas de clases del análisis han sido utilizados tres tipos de clases con los siguientes estereotipos:

Entidad: utilizado para modelar información que posee larga vida y que es a menudo persistente.



Figura.4 Clase Entidad

Interfaz: se utiliza para modelar la interacción entre el sistema y sus actores.

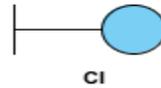


Figura.5 Clase Interfaz

Control: usado para coordinar la realización de los casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.



Figura.6 Clase Controladora

A continuación se muestra el diagrama de clases del análisis del caso de uso Exportar conducción, para ver los restantes dirigirse a los anexos sección [Anexo 3 Diagramas de Clases del Análisis.](#)

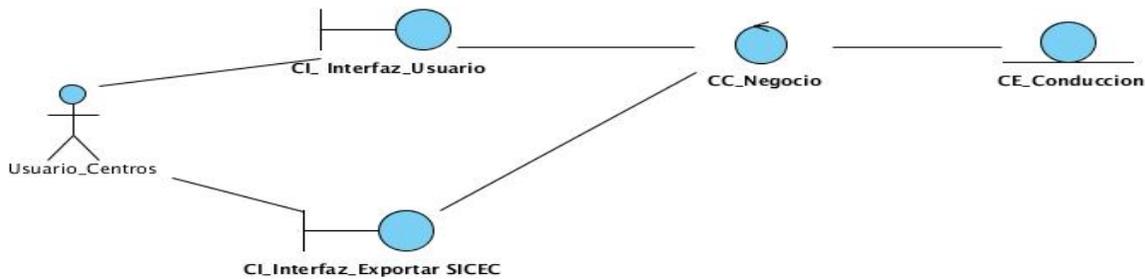


Figura.7 Caso de uso: Exportar conducción.

3.3. Modelo de Diseño

Es una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas con la implementación. Representa a los casos de uso en el dominio de la solución. Puede contener: los diagramas, las clases, paquetes, subsistemas, cápsulas, protocolos, interfaces, relaciones, colaboraciones, atributos, las realizaciones de los casos de uso, entre otros que se puedan considerar para el sistema en desarrollo. (51)

3.3.1 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos que se comunican entre sí, que es adaptada para solucionar un problema de diseño general en un contexto particular, o sea resulta ser una solución a un problema, que se puede aplicar muchas veces, en distintas situaciones. Su objetivo es reutilizar no sólo código, sino diseño de objetos, diseño de sistemas, modelos de análisis, etc., que se sabe que funcionan: reutilización del conocimiento.

Para el desarrollo del componente de sincronización se utilizan distintos patrones de diseño entre los que se encuentran:

Objeto de Acceso a Datos: DAO (por sus siglas en inglés) es un componente de software que proporciona una interfaz común entre la aplicación y uno o varios dispositivos de almacenamiento de información, tales como una base de datos, un archivo, servicios externos, etc. Es utilizado para abstraer y encapsular los accesos, gestionar las conexiones a la fuente de datos y obtener la información almacenada, (52) de esta manera los objetos tienen una forma unificada de acceder a esta.

Ventajas:

- No requiere que todos los objetos tengan conocimiento del destino final de la información que se manipula.
- Reduce el nivel de acoplamiento entre clases, disminuyendo la complejidad de realizar cambios.
- Se aíslan las conexiones a la fuente de datos en una capa fácilmente identificable y mantenible. (52)

En la siguiente figura se muestra la estructura del patrón DAO:

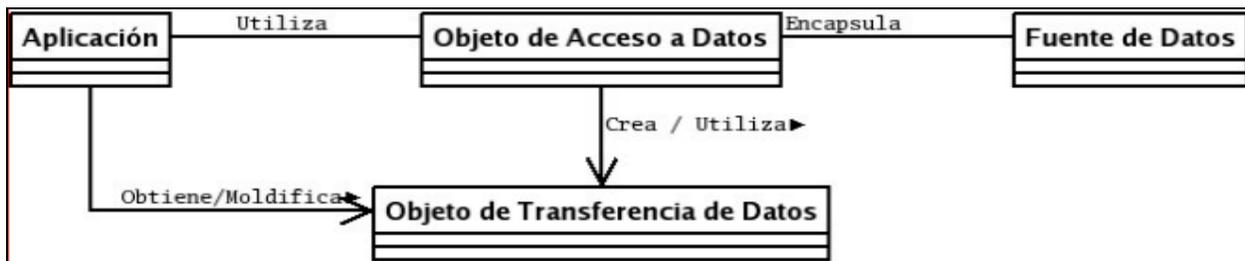


Figura.8 Estructura del patrón DAO.

Este patrón es utilizado en distintas clases del sistema para que accedan a las diferentes bases de datos, se utilizó para acceder a la información del CIM y a la base de datos de los centros. Permite manipular y obtener información como los cronogramas, estudios, usuarios, sujetos que se relacionan en las fuentes de datos, aislando a las clases modelo del trabajo directo con estas.

3.3.2 Patrones GRASP

GRASP: General Responsibility Assignment Software Paterns (patrones de los principios generales para asignar responsabilidades)

Los patrones son parejas de problema/solución, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. (53)

Experto: Este patrón se centra en asignar una responsabilidad a la clase que contiene los datos necesarios para desempeñarla, los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. El comportamiento se distribuye entre las clases que poseen la información solicitada, incitando de esta manera la utilización de clases que soporten una alta cohesión. La utilización de este patrón en el sistema se ejemplifica en las clases modelo del patrón DAO, ya que son ellas las que poseen las responsabilidades relacionadas con crear, guardar y actualizar los elementos de las bases datos.

Creador: responde al problema de quién debería ser el responsable de crear nuevas instancias de una clase. La creación de instancias es una actividad bastante común en los sistemas de software, por lo que es ventajoso contar con un responsable para la asignación de las responsabilidades de creación, lo que puede proporcionar mayor claridad, encapsulación, reutilización y brinda soporte a un bajo acoplamiento. Una ejemplificación del uso de este patrón en el sistema se encuentra en la clase controladorCIM, la cual posee la responsabilidad de crear instancias de diferentes clases.

Bajo Acoplamiento: intenta asignar responsabilidades de manera que el acoplamiento permanezca bajo. El acoplamiento es una medida de la fuerza con que un elemento está conectado a, tiene conocimiento de, confía en, otros elementos. Un elemento con bajo (o débil)

acoplamiento no depende de demasiados otros elementos, ya sean clases, subsistemas, sistemas, etc. (54) El Bajo Acoplamiento soporta el diseño de clases que son más independientes, lo posibilita minimizar el impacto que puede producir un cambio en alguna clase, además cada una de ellas es fácil de entender por separado y puede ser reutilizada. El uso de este patrón en el sistema se evidencia en que el mismo está implementado de forma que poca cantidad de clases tengan conocimiento o acceso a la información de las bases de datos, encargándose de esta función únicamente las clases correspondientes al patrón DAO.

Alta Cohesión: responde al problema de cómo mantener la complejidad dentro de los límites manejables. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Una clase con baja cohesión hace muchas cosas no afines o realiza trabajo excesivo. (53) Se aprecia en la clase controladora Autenticar del sistema, que maneja solamente la información referente a los usuarios y contraseñas del sistema.

Controlador: es un patrón de asignación de responsabilidades que indica quién debería encargarse de atender un evento del sistema, asigna la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase específica. (54) Un Controlador es un objeto que no pertenece a la interfaz de usuario y es responsable de definir el método para la operación del sistema. En el sistema es usado en la implementación de la clase controladora controladorCentro, la cual se encarga de llamar a las clases entidades y atender los eventos del sistema.

3.3.3 Diagramas de interacción

Los diagramas de interacción reflejan cómo interactúan los objetos durante la ejecución del sistema, describiendo el comportamiento del mismo; expresan un conjunto de objetos y sus relaciones, incluyendo mensajes que puedan realizarse entre ellos. Pueden ser utilizados para la visualización, especificación, construcción y documentación de los aspectos dinámicos de objetos, o para la modelación del flujo particular de control de un caso de uso. (55)

Los diagramas de interacción están conformados por los diagramas de secuencia y los de colaboración, de los cuales se escogieron los de secuencia para la modelación del componente de sincronización, ya que muestran la interacción de un conjunto de objetos a través del

tiempo. Los diagramas de secuencia representan la existencia de un objeto durante un cierto período de tiempo (línea de vida) y el instante de tiempo en el cual un objeto se encuentra desarrollando alguna acción (activación).

A continuación se muestra el diagrama de secuencia de caso de uso Exportar conducción, para ver los restantes dirigirse al [Anexo 4 Diagramas de Secuencia](#).

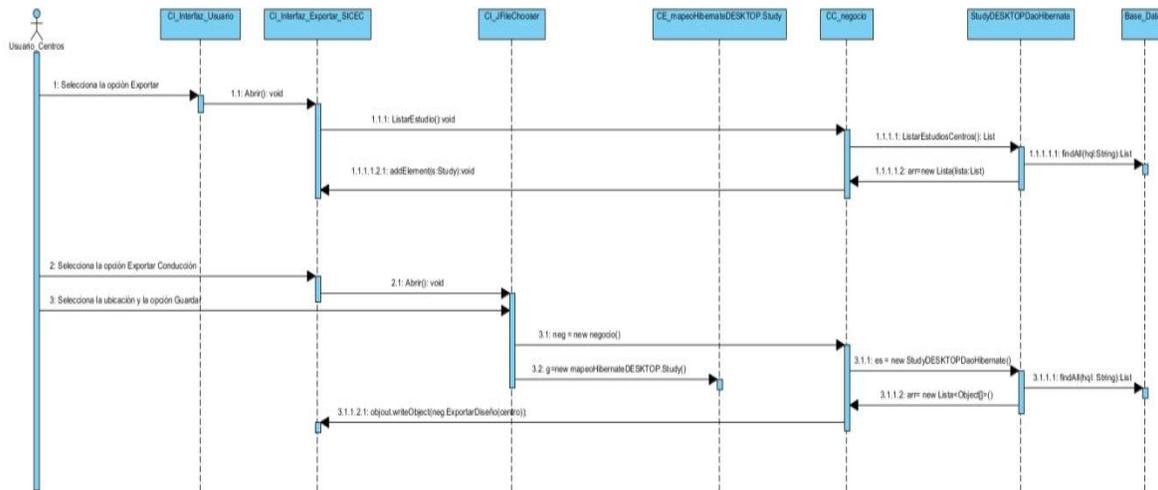


Figura.9 Caso de uso: Exportar conducción

3.3.4 Diagramas de clases del diseño

Son una representación visual del interior del sistema y permite visualizar las relaciones entre las clases que lo componen, se centran en los elementos del sistema de forma independiente del tiempo. A continuación se muestra el diagrama de clases del diseño del caso de uso Exportar conducción, para ver los restantes dirigirse al [Anexo 5 Diagramas de clases del diseño](#).

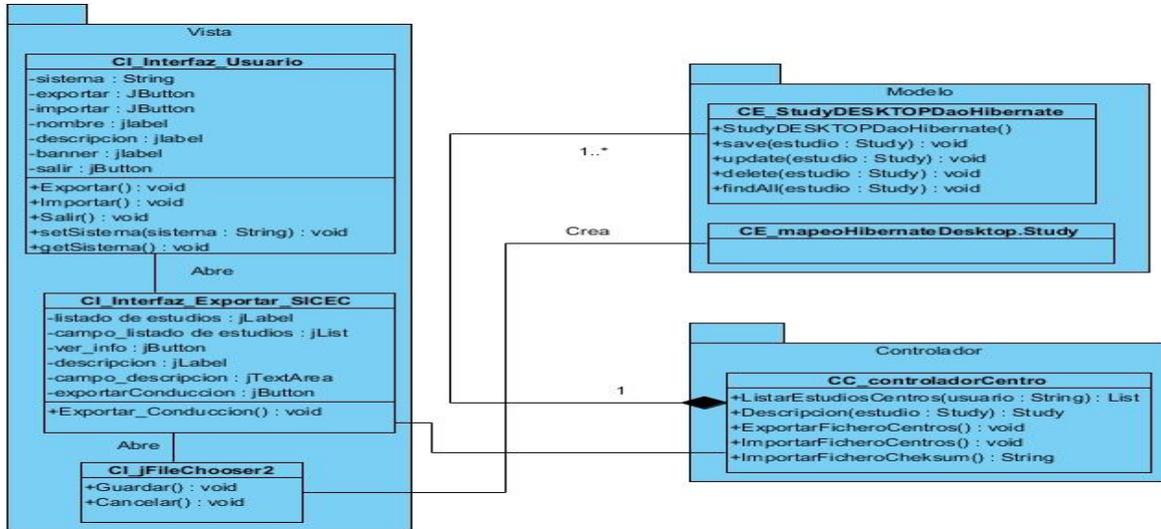


Figura.10 Caso de uso: Exportar conducción

3.3.5 Descripción de las clases del sistema

Nombre: controladorCIM	
Tipo de clase: Controladora	
Atributo	Tipo
No tiene	
Nombre:	ImportarFicheroCim ()
Descripción:	Importa hacia la base de datos el archivo que representa la conducción de los ensayos clínicos en algún hospital.
Nombre:	ExportarFicheroCim()
Descripción:	Exporta el cronograma asociado a un centro y estudio determinado.
Nombre:	ListarEstudiosCIM(usuario)
Descripción:	Busca en la base de datos del CIM todos los estudios existentes a los que tiene permiso acceder el usuario entrado como parámetro.
Nombre:	BuscarCentros(estudio)
Descripción:	Lista los centros que realizan un estudio determinado del CIM.

Tabla 3 Descripción de la clase controladora controladorCIM

Nombre: login	
Tipo de clase: Interfaz	
Atributo	Tipo
usuario	jLabel
contraseña	jLabel
sistema	jLabel
campo_usuario	TextField
campo_contraseña	PasswordField
campo_sistema	ComboBox
aceptar	Button
cancelar	Button
Nombre:	Aceptar()
Descripción:	Llama al método Autenticar_Usuario_CIM (usuario, contraseña) o Autenticar_Usuario_Centros (usuario, contraseña) de la clase controladora Autenticar dependiendo del valor que tome el campo_contraseña.
Nombre:	Cancelar()
Descripción:	Cierra la aplicación terminando el proceso.

Tabla 4 Descripción de la clase interfaz login

Nombre: Interfaz_Usuario	
Tipo de clase: Interfaz	
Atributo	Tipo
sistema	String
exportar	Button
importar	Button
nombre	jLabel
descripción	jLabel
banner	jLabel
salir	Button
Nombre:	Exportar()
Descripción:	Llama a la clase interfaz Interfaz_Exportar_ALAS o Interfaz_Exportar_SICEC.

Nombre:	Importar()
Descripción:	Llama a la clase interfaz Interfaz_Importar_ALAS o Interfaz_Importar_SICEC.
Nombre:	Salir()
Descripción:	Cierra la aplicación, terminando el proceso.
Nombre:	setSistema(sistema)
Descripción:	Permite realizar cambios en las características de la entidad sistema.
Nombre:	getSistema()
Descripción:	Devuelve los diferentes tipos de sistemas que pueden existir.

Tabla 5 Descripción de la clase interfaz Interfaz_Usuario

Nombre: Interfaz_Importar_SICEC	
Tipo de clase: Interfaz	
Atributo	Tipo
texto	jLabel
importarCronograma	jButton
area_de_texto	jTextArea
Nombre:	ImportarDiseño()
Descripción:	Llama a la clase interfaz jFileChooser.

Tabla 6 Descripción de la clase interfaz Interfaz_Importar_SICEC

Nombre: Interfaz_Importar_ALAS	
Tipo de clase: Interfaz	
Atributo	Tipo
texto	jLabel
importarConduccion	jButton
area_de_texto	jTextArea
Nombre:	Importar_Conduccion()
Descripción:	Llama a la clase interfaz jFileChooser.

Tabla 7 Descripción de la clase interfaz Interfaz_Importar_ALAS

Nombre: jFileChooser

Tipo de clase: Interfaz	
Atributo	Tipo
Ninguno	
Nombre:	Abrir()
Descripción:	Guarda el objeto a importar.
Nombre:	Cancelar()
Descripción:	Cierra la clase interfaz jFileChooser.

Tabla 8 Descripción de la clase interfaz jFileChooser

Nombre: Interfaz_Exportar_ALAS	
Tipo de clase: Interfaz	
Atributo	Tipo
listado de estudios	jLabel
campo_listado de estudios	jList
verCentros	jButton
centros	jLabel
campoCentros	jList
exportarDiseño	jButton
Nombre:	Ver_Centros()
Descripción:	Llama al método BuscarCentros(estudio) de la clase controladora.
Nombre:	Exportar_Diseño()
Descripción:	Llama a la clase interfaz jFileChooser.

Tabla 9 Descripción de la clase interfaz Interfaz_Exportar_ALAS

Nombre: Interfaz_Exportar_SICEC	
Tipo de clase: Interfaz	
Atributo	Tipo
listado de estudios	jLabel
campo_listado de estudios	jList
descripcion	jLabel
campo_descripcion	jTextArea
exportarConduccion	jButton

Nombre:	Exportar_Conduccion()
Descripción:	Llama a la clase interfaz jFileChooser.

Tabla 10 Descripción de la clase interfaz Interfaz_Exportar_SICEC

Nombre: Autenticar	
Tipo de clase: Controladora	
Atributo	Tipo
usuario	String
contraseña	String
Nombre:	Autenticar_Usuario_CIM(usuario, contraseña)
Descripción:	Busca en la base de datos el usuario que se corresponde con el usuario y contraseña entrados por parámetros, y que pertenezca al módulo del sistema Clínicas 1.0, verificando que tenga los permisos para acceder al sistema.
Nombre:	Autenticar_Usuario_Centros(usuario, contraseña)
Descripción:	Busca en la base de datos el usuario que se corresponde con el usuario y contraseña entrados por parámetros, y que pertenezca al módulo del sistema SIGEC Conductor, verificando que tenga los permisos para acceder al sistema.
Nombre:	md5(clear)
Descripción:	Aplica el algoritmo de encriptación md5 a las contraseñas.

Tabla 11 Descripción de la clase controladora Autenticar

Nombre: controladorCentro	
Tipo de clase: Controladora	
Atributo	Tipo
No tiene	
Nombre:	ExportarFicheroCentros ()
Descripción:	Exporta el archivo que representa la conducción de los ensayos clínicos

	en algún hospital.
Nombre:	ImportarFicheroCentros ()
Descripción:	Importa el archivo que representa el cronograma de los ensayos clínicos que deben realizar en el hospital.
Nombre:	ListarEstudiosCentros()
Descripción:	Lista los estudios realizado en el centro.
Nombre:	Descripción(estudio)
Descripción:	Devuelve una descripción del estudio realizado en el centro.

Tabla 12 Descripción de la clase controladora controladorCentro

El modelado de los diagramas de clases del análisis realizado en este capítulo ayudó a tener un mayor entendimiento y refinamiento del sistema, lo cual abrió el paso para la realización de los diagramas del diseño, artefactos importantes de salida de la fase del análisis y diseño. Al iniciar la fase de implementación los artefactos generados en este capítulo muestran al equipo de desarrollo una visión más clara de lo que se desea realizar en el sistema, permitiendo que pueda estar más enfocado al cumplimiento de los requerimientos funcionales del sistema.

Capítulo 4: Implementación del sistema

En la siguiente tabla se muestra cada uno de los nombres de las entidades reflejadas en el modelo de datos, especificando además la llave primaria (Primary Key, PK), el número de campos y una descripción de cada entidad.

No	Entidad	PK	Tipo de dato	Número de campos	Descripción de la entidad
1.	study	study_id	int	58	Entidad que contiene la información de los estudios y centros existentes
2.	Cronograma	cronograma_id	int	5	Entidad que contiene el cronograma general del estudio.
3.	Etapas	etapas_id	int	6	Entidad que contiene las etapas del cronograma general del estudio.
4.	Etapas_Study_Event_Definition	etapas_study_event_definition_id	int	3	Entidad que contiene la etapa y el momento de seguimiento del cronograma general.
5.	Study_Event_Definition	study_event_definition_id	int	14	Entidad que contiene las definiciones de momentos de seguimientos que aún no han sido asociados a "pacientes y hojas de crd" y pertenecen al cronograma general pero no al cronograma específico.
6.	Study_Event	study_event_id	int	16	Entidad que contiene los momentos de seguimiento existentes para un estudio.
7.	Study_Subject	study_subject_id	int	17	Entidad que contiene todos los "sujetos" que se deciden estudiar en un Ensayo y pasan a ser pacientes.
8.	Subject	subject_id	int	12	Entidad que define todos los sujetos asociados al estudio, dígame padre, madre, fecha de nacimiento, etc.
9.	Event_Crf	event_crf_id	int	20	Entidad que contiene las hojas de crd asociadas a un "momento de

Capítulo 4: Implementación del sistema

					seguimiento”.
10.	Crf_Version	crf_versio n_id	int	11	Entidad que contiene la versión de la hoja crd.
11.	Crf	crf_id	int	9	Entidad que contiene las hojas crd existentes.
12.	Event_Definition_Crf	event_defi nition_crf_ id	int	17	Entidad que contiene las hojas de momento de seguimiento del cronograma general.
13.	Val_Case,	case_id	int	11	Contiene los casos de validación o derivación.
14.	Item	ítem_id	int	13	Entidad que contiene las variables existentes en una hoja crd.
15.	Item_Data	ítem_data _id	int	10	Entidad que contiene los valores asociados a las variables de una hoja crd.
16.	Item_Data_Type	ítem_data _type_id	int	5	Contiene el tipo de dato de una variable de hoja crd.
17.	Item_Form_Metadata	ítem_form _metadata _id	int	21	Contiene los datos de la variable para mostrarla en un formulario.
18.	Item_Group_Metadata	ítem_grou p_metadat a_id	int	13	Contiene los datos del grupo para mostrar en un formulario.
19.	Section	section_id	int	15	Contiene la sección de hoja crd.
20.	Item_Group	ítem_grou p_id	int	9	Grupo de variables de hoja crd.
21.	Val_Dependency_Item	id	int	10	Contiene las variables de dependencia.
22.	Versioning_Map	versioning _map_id	int	3	Versión de hoja crd y variable de hoja crd.
23.	Response_Set	response_ set_id	int	6	Contiene las formas de mostrar la variable.
24.	Response_Type	response_ type_id	int	3	Contiene los tipos de formas de mostrar la variable.
25.	Status	status_id	int	3	Entidad que contiene los tipos de estados definidos para pacientes, momentos de seguimientos y hojas de crd.
26.	User_Account	user_id	int	19	Usuario o cuenta de usuario.
27.	Item_Reference_Type	ítem_refer	int	3	Tipo de referencia de la

		ence_type _id			variable.
--	--	------------------	--	--	-----------

Tabla 13 Descripción de las entidades del modelo de datos

4.2. Diagrama de despliegue

Los elementos del diseño al nivel del despliegue indican cómo se ubicarán la funcionalidad y los subsistemas dentro del entorno computacional físico que soportará al software. (56) El siguiente diagrama de despliegue muestra las relaciones físicas entre los componentes de hardware y de software utilizando nodos.

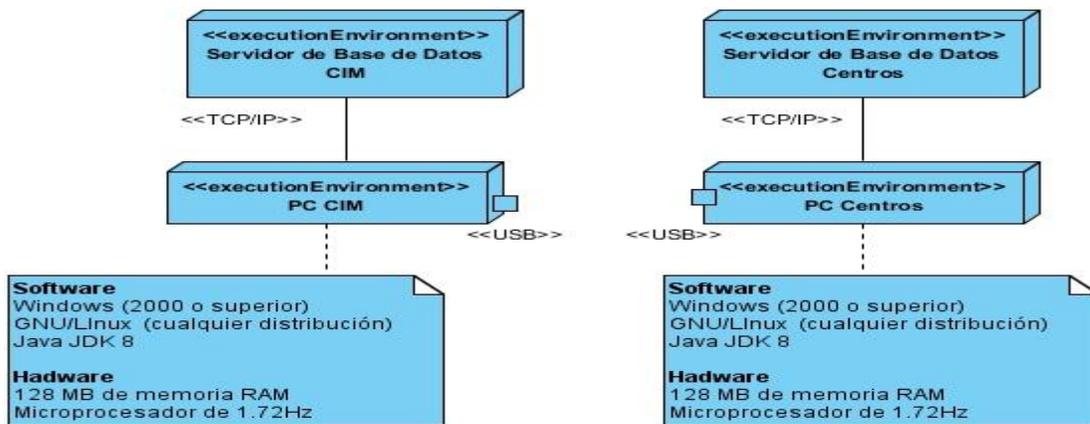


Figura.12 Diagrama de despliegue.

TCP/IP son las siglas de Protocolo de Control de Transmisión/Protocolo de Internet (en inglés Transmission Control Protocol/Internet Protocol), es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN)

4.3. Diagrama de componentes

El diseño a nivel de componentes define las estructuras de datos, los algoritmos, las características de la interfaz y los mecanismos de comunicación asignados a cada componente de software. (57)

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, representa las dependencias e interacciones entre los mismos, incluyendo componentes de

código fuente, de código binario, y ejecutables. A continuación se muestra el diagrama de componentes del sistema.

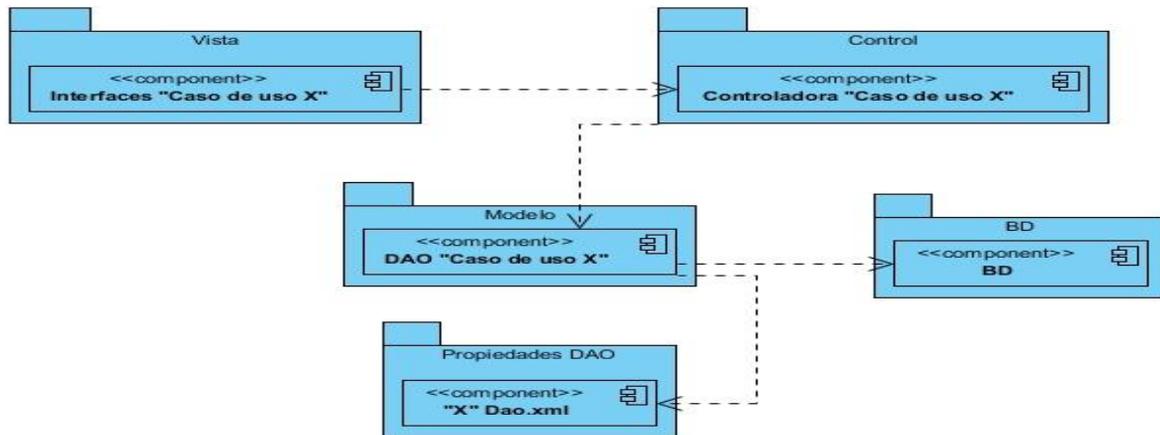


Figura.13 Diagrama de componentes del sistema.

En el paquete **Vista** se incluyen componentes formados por las clases interfaces que crean la vista para los diferentes casos de uso; el paquete **Control** contiene componentes formados por las distintas clases controladoras que posee el sistema y que intervienen en el caso de uso; mientras que en el paquete **Modelo** se encuentran componentes formados por clases entidades, incluyendo las del modelo de acceso a datos (DAO) que acceden a las bases de datos. Otros de los paquetes reflejados en el diagrama son el de **Base de Datos** y el de las **Propiedades DAO** que contiene las consultas en formatos XML¹⁰ que hacen la búsqueda de las entidades de las bases de datos.

4.4. Tratamiento de errores

El sistema posee funcionalidades que tienen en cuenta el tratamiento de los posibles errores que puedan ocurrir en el manejo de la aplicación. Se muestran mensajes de confirmación y de errores que son de fácil entendimiento para el usuario. Se realizan validaciones en el código para que no existan incorrecciones en el momento de importar los archivos, mostrando mensajes de error en caso de existir.

A continuación se muestran algunos ejemplos de mensajes de errores que muestra el sistema:

¹⁰ XML, siglas en inglés de eXtensible Markup Language (lenguaje de marcas extensible)

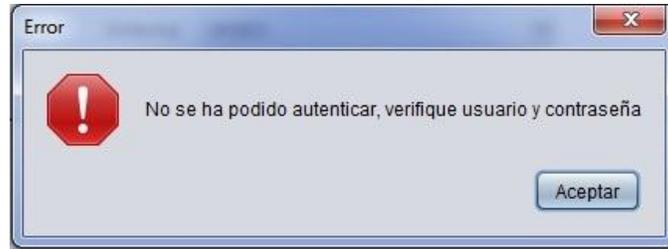


Figura.14 Mensaje de error en la autenticación.

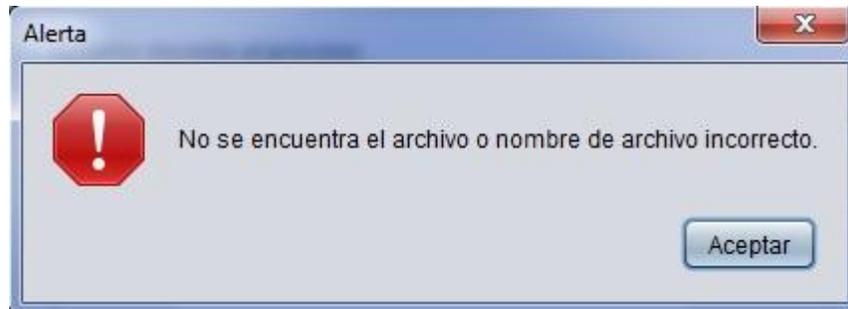


Figura.15 Mensaje de error importando la conducción.

4.5. Seguridad

La seguridad de un sistema es de gran importancia, ya que se debe garantizar la integridad, confiabilidad y autenticidad de la información que se maneja. Para garantizar esto, todo acceso al sistema es sometido a un proceso de autenticación, donde se requiere un nombre de usuario y la contraseña asociada a este nombre, estando esta última encriptada con el algoritmo MD5. Todos los archivos que se manejan en la aplicación, ya sean los cronogramas o los datos de las conducciones realizadas, serán guardados en forma serializada, haciéndolos ilegibles si se encuentran en algún dispositivo de almacenamiento.

4.6. Estrategias de codificación. Estándares y estilos a utilizar

Para el desarrollo del sistema propuesto se utilizan varios estándares de codificación, tales como:

Camel-Case: es la nomenclatura por excelencia en el mundo Java, consistente en utilizar las mayúsculas como separadores de palabras. Si comienza con mayúscula, se denomina *UpperCamelCase* y, si no, *lowerCamelCase*. (58) Los nombres de las clases deben estar escritos en *UpperCamelCase*, al igual que los nombres de los variables y métodos, a excepción

de que el identificador sea una palabra simple, caso en el cual se escribe todo con minúscula. A continuación se muestra la utilización de esta notación en el código del sistema.

```
public List ListarEstudiosCIM()
{
    List nueva = new ArrayList();
    StudyDESKTOPDaoHibernate es = new StudyDESKTOPDaoHibernate();
    nueva = es.findAll("select st.name from Study as st where st.parentStudyId is null");
    return nueva;
}
```

Figura.16 Ejemplo de notación Camel-Case en el sistema

Las palabras reservadas del lenguaje así como las construcciones que se asimilan a funciones (Array(), List(), etc.) se escribirán todas en mayúsculas. La indentación o sangrado debe ser hecha mediante tabulaciones y no se deben insertar espacios, el tamaño de las mismas queda a opción del programador. Los comentarios estarán ubicados al inicio de cada clase o función y al final de cada bloque de código. Las líneas en blanco se emplean antes y después de métodos, clases y estructuras. La estructura para las llaves que definen el cuerpo de una función se muestra en el siguiente ejemplo.

Ejemplo: void Funcion ()

```
{
//Instrucciones de la función
}
```

Con el desarrollo del capítulo y la realización de los diferentes artefactos que se generan en la implementación del sistema, se evidencian la disposición física del sistema y su estructura interna.

Conclusiones

Con la culminación del presente trabajo se concluye que:

- La realización de un estudio del estado del arte de los sistemas para sincronización y replicación de datos a nivel nacional e internacional, arrojó como resultado que ninguno permite dar una solución completa al problema planteado, siendo necesario la creación de un componente para cumplir estas funciones.
- La descripción de la metodología, las herramientas y las tecnologías definidas por el arquitecto principal del proyecto Ensayos Clínicos, ratificó que estas poseen las características necesarias para desarrollar el componente de sincronización.
- La elaboración de los artefactos correspondientes a la disciplina de desarrollo de RUP permitió crear la documentación necesaria de la investigación, sirviendo como guía al equipo de trabajo para implementar la solución obtenida.
- La implementación de las funcionalidades para sincronizar los datos de los sistemas Clínicas 1.0 y SIGECConductor brindó a los especialistas del CIM y de los centros conductores de los ensayos clínicos, un mejor manejo de los datos asociados a estos estudios.

Recomendaciones

Teniendo como base los resultados de esta investigación y la experiencia adquirida durante el desarrollo de la misma, se recomienda:

- Utilizar algoritmos como SHA-1 o SHA-2 para la encriptación de las contraseñas de los usuarios del sistema, ya que estos brindan un mayor nivel de seguridad y un menor riesgo ante ataques.
- Utilizar un algoritmo de encriptación potente como Triple DES, AES, RSA, que permita cifrar, haciendo ilegible la información almacenada en los archivos para personas ajenas a las instituciones relacionadas con la conducción de los ensayos clínicos, pudiendo ser legible solamente por personas que conozcan el algoritmo y la clave de encriptación.

Referencias Bibliográficas

1. **Díaz, Lic. Alfredo Rodríguez.** Retos de la Informatización del Sector de la Salud en Cuba. [En línea] [Citado el: 07 de febrero de 2014.] <http://www.cais.org.ar/sites/default/files/RetosInformatizacionSaludCuba2013.pdf>.
2. CIMAB.SA. *Centro de Inmunología Molecular.* [En línea] [Citado el: 11 de febrero de 2014.] <http://www.cimab-sa.com/index.php?action=cim>.
3. Eastern Cooperative Oncology Group (ECOG). *clin_trial_guide_spanish.pdf.* [En línea] [Citado el: 07 de febrero de 2014.] http://ecog.dfci.harvard.edu/general/clin_trial_guide_spanish.pdf.
4. Legislación Vigente de Medicamentos Huérfanos. *Ensayos Clínicos.* [En línea] Instituto de Salud Carlos III. [Citado el: 11 de febrero de 2014.] <http://www.ub.edu/legmh/ereensay.htm>.
5. CIMAher. [En línea] 2009. [Citado el: 22 de mayo de 2014.] <http://www.cimaher.com/index.php?action=cim>.
6. Revista Cubana de Informática. *"ALASCLÍNICAS": Sistema de gestión de ensayos clínicos.* [En línea] 25 de diciembre de 2011. [Citado el: 22 de mayo de 2014.] http://scielo.sld.cu/scielo.php?pid=S1684-18592012000100002&script=sci_arttext.
7. Diccionario de Informática. [En línea] [Citado el: 22 de mayo de 2014.] <http://www.alegsa.com.ar/Dic/sincronizar%20archivos.php>.
8. Replicación de Bases de Datos. [En línea] [Citado el: 17 de febrero de 2014.] <http://www.slideshare.net/luisfe/replicacin-base-de-datos-488210>.
9. **Oracle.** Oracle Guía de administración del sistema de Oracle Solaris Cluster. [En línea] 2010. [Citado el: 17 de febrero de 2014.]
10. **Runceanu, Adrian, Popescu, Marian y Runceanu, Mihaela.** Academia.edu. *TECHNIQUES FOR DATA REPLICATION ON DISTRIBUTED DATABASES.* [En línea] 08 de Noviembre de 2008. [Citado el: 11 de 06 de 2014.] http://www.academia.edu/2949400/TECHNIQUES_FOR_DATA_REPLICATION_ON_DISTRIBUTED_DATABASES.
11. mongoDB. *Master Slave.* [En línea] [Citado el: 17 de febrero de 2014.] <http://docs.mongodb.org/manual/core/master-slave/>.
12. Oracle. [En línea] [Citado el: 17 de febrero de 2014.] <http://docs.oracle.com/cd/E19656-01/821-1502/6nmfv3a55/index.html>.

13. Search Data Center. [En línea] [Citado el: 23 de enero de 2014.] <http://searchdatacenter.techtarget.com/es/cronica/Buenas-practicas-de-replicacion-de-datos-para-safeguard>.
14. **Alonzo, Luis Ignacio, y otros.** Instituto de Computación-Facultad de Ingeniería-Universidad de la República Oriental del Uruguay. [En línea] 2003. [Citado el: 10 de febrero de 2014.] <http://www.fing.edu.uy/inco/cursos/tagsi/Trabajos/2003/TAGSI03-TareaTecnSI-grupo4.pdf>.
15. Monografias.com. *SymmetricDs,_Herramienta_de_Replicación_de_Código_Abierto_*-. [En línea] [Citado el: 10 de febrero de 2014.] <http://www.monografias.com/trabajos83/symmetricds-herramienta-replicacion-codigo-abierto/symmetricds-herramienta-replicacion-codigo-abierto.shtml>.
16. **Camejo Suazo, Leandro Willian.** *Réplica de datos para el Sistema Informativo Automatizado de los Cuadros*. La Habana : s.n., 2013.
17. SPUNCH. *Soluciones Informáticas para la Gestión de la Calidad*. [En línea] [Citado el: 11 de febrero de 2014.] <http://www.spuch.com/productos/cronsql/cronsql.htm>.
18. SPUNCH. *CronSQL Herramienta de sincronización de bases de datos*. [En línea] [Citado el: 18 de 02 de 2014.] <http://www.spuch.com/descargas/Manual de usuario CronSQL.pdf>.
19. CIS. *DBMoto*. [En línea] [Citado el: 11 de febrero de 2014.] <http://www.cis.mx/dbmoto.html>.
20. *DBMoto_Managing Timeout Errors During Replication*. [En línea] [Citado el: 24 de febrero de 2014.] http://www.hitsw.com/support/kbase/DBMoto/DBMoto_1639_TimeoutErrors.htm.
21. Dell Software. *SharePlex*. [En línea] [Citado el: 11 de febrero de 2014.] <http://www.quest.com/shareplex-for-oracle/>.
22. *SharePlex_para_Oracle_Simple,_costeable,_cero_impacto_y_en_línea*. [En línea] [Citado el: 12 de febrero de 2014.] <http://www.reingtec.com/SharePlex/index.html>.
23. *SharePlex for Oracle - Replication Tools for 9i-10g-11g Databases*. [En línea] [Citado el: 11 de febrero de 2014.] <http://www.quest.com/shareplex-for-oracle/>.
24. **López Lemus, Edgar Felipe Alejandro.** *¿Solución de alta disponibilidad de base de datos por hardware o por software?* Facultad de Ingeniería, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA. Guatemala : s.n., 2005. pág. 90.
25. Slony-I. [En línea] [Citado el: 18 de febrero de 2014.] <http://slony.info/>.
26. *Introducing Slony - O'Reilly Media*. [En línea] [Citado el: 18 de febrero de 2014.] <http://www.onlamp.com/pub/a/onlamp/2004/11/18/slony.html>.
27. *Replicacion con postgresql y slony*. [En línea] [Citado el: 18 de febrero de 2014.] <http://www.slideshare.net/JohannaMendez2/replicacion-con-postgresql-y-slony-8437330>.

28. **Mariano Reingart.** PyReplica- Sistema de replicación simple para PostgreSQL programado en Python. [En línea] 2009. [Citado el: 22 de mayo de 2014.] http://www.arpug.com.ar/trac/raw-attachment/wiki/PgDayUnnoba/PDF_PyReplica.pdf.
29. hpfloresj. [En línea] [Citado el: 18 de febrero de 2014.] http://hpfloresj.blogspot.com/2012_02_01_archive.html.
30. **González, Mr Luis Alberto Pimentel.** Informática 2013. *Reko Replicator*. [En línea] [Citado el: 13 de enero de 2013.] <http://www.informaticahabana.cu/node/1870>.
31. **Gomez Peña, Angela Gloria y Alfonso Valdés, Javier.** *Desarrollo del módulo para la transmisión de datos de gran tamaño para el sistema Reko*. La Habana : s.n., 2010.
32. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. Madrid : Pearson Education, S. A., 2000.
33. **Mendoza Sanchez, Maria A.** informatizate. *Metodologías de Desarrollo de Software*. [En línea] 07 de junio de 2004. [Citado el: 19 de febrero de 2014.] <http://www.willydev.net/Descargas/cualmetodologia.pdf>.
34. **Martínez, Alejandro y Martínez, Raúl.** *Guía a Rational Unified Process*. Escuela Politécnica Superior de Albacete – Universidad de Castilla la Mancha.
35. Historia_del_lenguaje_Java. [En línea] [Citado el: 19 de febrero de 2014.] http://www.cad.com.mx/historia_del_lenguaje_java.htm.
36. Principales características de Java. [En línea] [Citado el: 05 de 01 de 2014.] <http://personales.upv.es/rmartin/cursoJava/Java/Introduccion/PrincipalesCaracteristicas.htm>.
37. JAVA. [En línea] [Citado el: 25 de abril de 2014.] <http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html>.
38. GENBETA: dev. *desarrollo y software*. [En línea] [Citado el: 22 de mayo de 2014.] <http://www.genbetadev.com/herramientas/netbeans-1>.
39. NetBeans. [En línea] Oracle Corporation, 2013. [Citado el: 22 de mayo de 2014.] <https://netbeans.org/cddl.html>.
40. **Orallo, Enrique Hernández.** El Lenguaje Unificado de Modelado (UML). [En línea] [Citado el: 23 de febrero de 2014.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
41. **López Boullón, Carlos Javier y Ortíz Noriega, Dresky.** *Módulo de Reko para la resolución de conflictos*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2010. pág. 13.
42. pgadmin. [En línea] [Citado el: 25 de abril de 2014.] http://www.guia-ubuntu.com/index.php/PgAdmin_III.
43. Hibernate ORM. [En línea] [Citado el: 24 de abril de 2014.] <http://hibernate.org/orm/>.

44. La importancia de java para el futuro. [En línea] [Citado el: 22 de mayo de 2014.] <http://carlos-henriquez.blogspot.com/2010/07/hibernate-framework-para-el-mapeo-de.html>.
45. **Larman, Craig**. Departamento de Lenguajes y Sistemas Informáticos. [En línea] [Citado el: 14 de marzo de 2014.] <http://lsi.ugr.es/~ig1/isoo/larman/Modelo%20del%20dominio.pdf>.
46. **Laguna, Miguel A.** Departamento de Informática - Universidad Valladolid. [En línea] [Citado el: 18 de marzo de 2014.] <http://www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf>.
47. **Quiroga, Juan Pablo**. Universidad de los Andes- Facultad de Ingeniería. [En línea] [Citado el: 18 de marzo de 2014.] <http://sistemas.uniandes.edu.co/~csof5101/dokuwiki/lib/exe/fetch.php?media=principal:csof5101-requerimientos.pdf>.
48. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico*. 6ta Edición. pág. 3. Vol. Capítulo 10 Diseño Arquitectónico.
49. **Reynoso, Carlos y Kicillof, Nicolás**. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Universidad de Buenos Aires. 2004. pág. 73.
50. **Quintero, Juan Bernardo**. *Arquitectura de Software- Patrones en la Arquitectura*.
51. MeRinde_-_Modelo_de_Diseño. [En línea] [Citado el: 25 de marzo de 2014.] http://merinde.net/index.php?option=com_content&task=view&id=92&Itemid=296.
52. **Mesa, Jose Luis, y otros**. Patrón DAO. [En línea] [Citado el: 26 de marzo de 2014.] http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/expoDAO.pdf.
53. **Larman, Craig**. *UML y Patrones: Introducción al análisis y diseño orientado a objetos*.
54. —. *UML y Patrones Segunda Edición*. Segunda Edición. Canadá : s.n. pág. 172.
55. DIAGRAMAS_DE_INTERACCION. [En línea] [Citado el: 01 de abril de 2014.] <http://uxmcc1.iimas.unam.mx/~cursos/Objetos/Cap18/cap18.html>.
56. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico. Capítulo 9: Ingeniería del diseño*. Sexta edición.
57. —. *Ingeniería de Software. Un enfoque práctico. Capítulo 11: Diseño de Componentes Parte 1*. Sexta edición.
58. Blog optymith. . *Estandares de codificación y buenas prácticas*. [En línea] 26 de noviembre de 2003. [Citado el: 30 de abril de 2014.] <http://blog.optimyth.com/es/2013/11/estandares-de-codificacion-y-buenas-practicas>.

Bibliografía

Alés Martínez, José Enrique, y otros. Guías operativas para los CEIC-Evaluación de Ensayos Clínicos. Madrid: Fundación AstraZeneca, 2006. pág. 294.

Alonzo, Luis Ignacio, y otros. Instituto de Computación- Facultad de Ingeniería- Universidad de la República Oriental del Uruguay. [En línea] 2003. [Citado el: 10 de febrero de 2014.] <http://www.fing.edu.uy/inco/cursos/tagsi/Trabajos/2003/TAGSI03-TareaTecnSI-grupo4.pdf>.

Alvarez de Zayas, Carlos. *Metodología de la Investigación Científica*. Santiago de Cuba : s.n., 1995.

Bles, Yordel Mesa y Pérez Boniche, Ainel. alasClínicas: Adaptación del sistema OpenClinica al control de procesos de los Ensayos Clínicos cubanos. Universidad de las Ciencias Informáticas. La Habana: s.n., 2009. Tesis de grado.

Blog optymith. . Estándares de codificación y buenas prácticas. [En línea] 26 de noviembre de 2003. [Citado el: 30 de abril de 2014.] <http://blog.optimyth.com/es/2013/11/estandares-de-codificacion-y-buenas-practicas>.

Cabalé Rondón, Dianne María y Hernández Reinoso, Alexey. Desarrollo del módulo Enviar Datos del sistema alasClinicas 2.0. La Habana: s.n., 2012.

Calisoft. *Centro Nacional de Calidad de Software*. [En línea] 2014. [Citado el: 11 de 06 de 2014.] https://calisoft.uci.cu/index.php?option=com_content&view=article&id=19:balance-calisoft-2014&catid=9&Itemid=157.

CIMAB.SA. Centro de Inmunología Molecular. [En línea] [Citado el: 11 de febrero de 2014.] <http://www.cimab-sa.com/index.php?action=cim>.

CIS. DBMoto. [En línea] [Citado el: 11 de febrero de 2014.] <http://www.cis.mx/dbmoto.html>.

Dell Software mejora su solución SharePlex para optimizar la integración y el análisis de los datos. [En línea] [Citado el: 12 de febrero de 2014.] http://www.esecuritymagazine.com/index.php?option=com_content&view=article&id=2377:dell&catid=44:general&Itemid=53.

Dell Software. SharePlex. [En línea] [Citado el: 11 de febrero de 2014.] <http://www.quest.com/shareplex-for-oracle/>.

Díaz, Dianet Berroa. Diseño del módulo Visualizar Información del sistema alasClinicas v2.o. La Habana: s.n., 2012.

Díaz, Lic. Alfredo Rodríguez. Retos de la Informatización del Sector de la Salud en Cuba. [En línea] [Citado el: 07 de febrero de 2014.] <http://www.cais.org.ar/sites/default/files/RetosInformatizacionSaludCuba2013.pdf>.

Eastern Cooperative Oncology Group (ECOG). clin_trial_guide_spanish.pdf. [En línea] [Citado el: 07 de febrero de 2014.] http://ecog.dfci.harvard.edu/general/clin_trial_guide_spanish.pdf.

Geek | Dell Software amplía SharePlex para optimizar la integración y análisis de datos - Geek. [En línea] [Citado el: 12 de febrero de 2014.] http://www.geek.com.mx/2013/05/dell-software-amplia-shareplex-para-optimizar-la-integracion-y-analisis-de-datos/.Historia_del_lenguaje_Java.

Hernández León, Rolando Alfredo y Coello González, Sayda. EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA. Ciudad de la Habana: EDUNIV Editorial Universitaria, 2002. ISBN: 959-16-0343-6.

—. El proceso de investigación científica. Ciudad de La Habana: Editorial Universitaria, 2011. ISBN 978-959-16-1307-3.

Historia_del_lenguaje_Java. [En línea] [Citado el: 19 de febrero de 2014.] http://www.cad.com.mx/historia_del_lenguaje_java.htm.

hpfloresj . [En línea] [Citado el: 18 de febrero de 2014.] http://hpfloresj.blogspot.com/2012_02_01_archive.html.

Jera, Erilán Martínez y Gonzáles Eríquez, Landy. alasClínicas: Desarrollo de la Gestión de Datos de Ensayos Clínicos a partir del sistema OpenClinica. La Habana : s.n., 2009.

Larman, Craig. UML y Patrones: Introducción al análisis y diseño orientado a objetos.

Legislación Vigente de Medicamentos Huérfanos. Ensayos Clínicos. [En línea] Instituto de Salud Carlos III. [Citado el: 11 de febrero de 2014.] <http://www.ub.edu/legmh/ereensay.htm>.

López Boullón, Carlos Javier y Ortíz Noriega, Dresky. Módulo de Reko para la resolución de conflictos. Universidad de las Ciencias Informáticas. La Habana : s.n., 2010. pág. 13.

MSc. Rodríguez García, Lucía, Ing. Martínez Jera, Erilán y MSc. Hernández Ramírez, Martha Denia. "ALASCLÍNICAS": SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS. La Habana : s.n., 2012. pág. 4.

Mendoza Sánchez, Maria A. informatizate. Metodologías de Desarrollo de Software. [En línea] 07 de junio de 2004. [Citado el: 19 de febrero de 2014.] <http://www.willydev.net/Descargas/cualmetodologia.pdf>.

Monografias.com. SymmetricDs,_Herramienta_de_Replicación_de_Código_Abierto_-. [En línea] [Citado el: 10 de febrero de 2014.] <http://www.monografias.com/trabajos83/symmetriccds-herramienta-replicacion-codigo-abierto/symmetriccds-herramienta-replicacion-codigo-abierto.shtml>.

Multi Master Replication (Oracle Fusion Middleware Deploymen Planning Guide for Oracle Directory Server Enterprise Edition). [En línea] [Citado el: 17 de febrero de 2014.] <http://docs.oracle.com/cd/E19656-01/821-1502/6nmfv3a55/index.html>.

Oracle. Oracle Guía de administración del sistema de Oracle Solaris Cluster . [En línea] 2010. [Citado el: 17 de febrero de 2014.]

Orallo, Enrique Hernández. El Lenguaje Unificado de Modelado (UML). [En línea] [Citado el: 23 de febrero de 2014.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.

PostgreSQL: Documentation: 8.4: High Availability, Load Balancing, and Replication. [En línea] [Citado el: 17 de febrero de 2014.] <http://www.postgresql.org/docs/8.4/static/high-availability.html>. Pressman, Roger S. Ingeniería de Software. Un enfoque práctico.

Replicación de Bases de Datos. [En línea] [Citado el: 17 de febrero de 2014.] <http://www.slideshare.net/luisfe/replicacin-base-de-datos-488210>.

Replicación de Datos - DBMoto. [En línea] [Citado el: 11 de febrero de 2014.] http://www.hitsw.com/localized/spanish/products_services/dbmoto/dbmoto.html.

Replicación de Datos - DBMoto 7. [En línea] [Citado el: 11 de febrero de 2014.] http://www.hitsw.com/localized/spanish/products_services/dbmoto/dbmoto_7.html.

Reynoso, Carlos y Kicillof, Nicolás. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Universidad de Buenos Aires. 2004

Rodríguez, Alejandro Fabelo. Guía para la administración y configuración en la sincronización de datos del Sistema de Gestión para la Ingeniería Clínica y Electromedicina. Universidad de las Ciencias Informáticas. La Habana : s.n., 2012. pág. 49, Tesis de grado.

Rondón Regalado, Martha María y Cabeza Beltran, Madiela. Sistema para la administración y supervisión del sistema de réplica "Magic@ Data Replication eXtensible Solution. La Habana: s.n., 2012.

RUP vs XP. [En línea] [Citado el: 18 de febrero de 2014.] www.usmp.edu.pe/publicaciones/boletin/fia/.../RUP%20vs.%20XP.pd.

SharePlex for Oracle - Replication Tools for 9i-10g-11g Databases. [En línea] [Citado el: 11 de febrero de 2014.] <http://www.quest.com/shareplex-for-oracle/>.

SharePlex_para_Oracle_Simple,_costeable,_cero_impacto_y_en_línea. [En línea] [Citado el: 12 de febrero de 2014.] <http://www.reingtec.com/SharePlex/index.html>.

Slony-I. Slony-I. [En línea] [Citado el: 18 de febrero de 2014.] <http://slony.info/>.

SPUNCH. CronSQL Herramienta de sincronización de bases de datos. [En línea] [Citado el: 18 de febrero de 2014.] [http://www.spuch.com/descargas/Manual de usuario CronSQL.pdf](http://www.spuch.com/descargas/Manual%20de%20usuario%20CronSQL.pdf).

SPUNCH. Soluciones Informáticas para la Gestión de la Calidad. [En línea] [Citado el: 11 de febrero de 2014.] <http://www.spuch.com/productos/cronsql/cronsql.htm>.

Valdés, Rodney Velázquez. Portal de soporte de los productos del CEIGE. Universidad de las Ciencias Informáticas. La Habana: s.n., 2012. pág. 87.

Ventajas y Desventajas - Metodología XP. [En línea] [Citado el: 13 de febrero de 2014.] <https://sites.google.com/site/xpmetodologia/marco-teorico/ventajas>.

Visconcti, Marcello y Astudillo, Hernán. Fundamentos de Ingeniería de Software.

Anexos

Anexo 1: Entrevista

Objetivo de la entrevista: Definir los requisitos funcionales y no funcionales del sistema.

Día de la entrevista: 10 de febrero de 2014

Persona entrevistada: Erislán Martínez Jera (Jefe de Proyecto).

Lugar de la entrevista: Laboratorio 205, Docente 6, UCI (Puesto de trabajo).

Modo de realización: Diálogo.

Aspectos a tener en cuenta:

1. Procesos que intervienen en la conducción de ensayos clínicos en el CIM y los centros hospitalarios.
2. Estructura de las bases de datos existentes que puedan ser utilizadas.
3. Pautas a seguir en el diseño de las interfaces del sistema.
4. Seguridad de la información que se maneje.
5. Características del sistema a desarrollar.
6. Cualidades del sistema a desarrollar.

Anexo 2: Descripción de Casos de Uso

Tabla 14. Caso de uso: Listar estudios

Objetivo	Ver listado de los estudios que se están realizando.	
Actores	Gerente de datos	
Resumen	El caso de uso inicia cuando el actor ingresa en el módulo SIGEC. Permite ver la lista de los estudios existentes.	
Complejidad	Baja	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado en su módulo correspondiente.	
Postcondiciones	Se muestra el listado de los estudios que se realizan.	
Flujo de eventos		
Flujo básico Listar estudios		
	Actor	Sistema

1.	El usuario ingresa en el módulo SIGEC.	
2.		Muestra el listado de los estudios existentes en ese momento. Y permite: <ul style="list-style-type: none"> • Ver centros • Exportar Diseño Finaliza así el caso de uso.
Relaciones	CU incluidos	No existe
	CU extendidos	Listar centros a partir de un estudio; ver caso de uso Listar centros a partir de un estudio.

Tabla 15. Caso de Uso: Listar centros a partir de un estudio

Objetivo	Ver la lista de los centros que realizan un estudio determinado.	
Actores	Gerente de datos	
Resumen	El caso de uso inicia cuando el actor luego de haber seleccionado un estudio de la lista de estudios, selecciona la opción Ver Centros. Permite observar la lista de los centros que realizan el estudio seleccionado por el usuario.	
Complejidad	Baja	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado en su módulo correspondiente.	
Postcondiciones	Se muestra la lista de los centros asociados al estudio.	
Flujo de eventos		
Flujo básico Listar centros a partir de un estudio		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor estando en la página principal selecciona la opción Exportar.	
2.		Muestra un listado de los estudios. Ver caso de uso: Listar estudios.
3.	El usuario selecciona el estudio al que desea ver los centros asociados.	
4.		Sombrea el estudio seleccionado.
5.	Selecciona la opción Ver Centros.	
6.		Muestra la lista de los centros asociados al estudio seleccionado por el usuario, finalizando así el caso de uso.
Flujos alternos		
1.a El usuario no selecciona ningún estudio.		
	Actor	Sistema
1.		Muestra un mensaje de error indicando que se debe seleccionar un estudio. Y permite: <ul style="list-style-type: none"> • Aceptar

2.	Selecciona la opción Aceptar.	
3.		Cierra el diálogo retornando al paso 1 del Flujo Normal de Eventos.
Relaciones	CU incluidos	Ninguno
	CU extendidos	Exportar Cronograma; ver caso de uso Exportar Cronograma

Tabla 16. Caso de uso: Autenticar usuario

Objetivo	Ingresar al sistema.	
Actores	Usuario	
Resumen	El caso de uso inicia cuando el actor inicia la aplicación. Permite al usuario entrar al sistema.	
Complejidad	Baja	
Prioridad	Alta	
Precondiciones	El sistema debe estar en ejecución.	
Postcondiciones	El usuario ingresa al módulo seleccionado.	
Flujo de eventos		
Flujo básico Autenticar usuario		
	Actor	Sistema
1	El caso de uso inicia cuando el actor inicia la aplicación.	
2		Muestra la interfaz de autenticación con los siguientes campos: <ul style="list-style-type: none"> • Usuario • Contraseña • Sistema Y permite: <ul style="list-style-type: none"> • Aceptar • Cancelar
3	Llena los campos usuario, contraseña y selecciona el sistema correspondiente y la opción Aceptar.	
4		Verifica los datos del usuario y contraseña en la base de datos.
5		Muestra una ventana de diálogo con el mensaje: "Se ha autenticado correctamente". Y permite: <ul style="list-style-type: none"> • Aceptar
6	Selecciona la opción Aceptar.	
7		Muestra la interfaz principal del sistema correspondiente al que pertenece el usuario. Finaliza el caso de uso.
Flujos alternos		
3.a El usuario no introduce los campos usuario o contraseña.		
	Actor	Sistema

1.		El sistema muestra una ventana de diálogo con un mensaje indicando que existen campos vacíos. Y permite:
2.	Selecciona la opción Aceptar.	
3.		Cierra la ventana de diálogo retornando al paso 2 del Flujo Normal de Eventos.
5.a Existe un error en el nombre de usuario o en la contraseña.		
	Actor	Sistema
1.		El sistema muestra mediante una ventana de diálogo que existe un error en los datos entrados porque el usuario o la contraseña son incorrectos. Y permite:
2.	Selecciona la opción Aceptar.	
3.		Cierra el dialogo retornando al paso 2 del Flujo Normal de Eventos.
4.a El usuario selecciona la opción Cancelar.		
	Actor	Sistema
1.		Cierra la aplicación, terminado el proceso, finalizando así el caso de uso.
Relaciones	CU incluidos	No existe
	CU extendidos	No existe

Tabla 17. Caso de uso: Exportar Cronograma

Objetivo	Exportar el cronograma asociado a un centro específico.
Actores	Gerente de datos
Resumen	El caso de uso inicia cuando luego de que el actor estando en la página principal seleccione la opción Exportar. El sistema cambia a una nueva interfaz donde el actor luego de seleccionar el estudio y el centro al que desea exportarle el cronograma, ejecuta la opción Exportar Cronograma. Permite guardar un archivo serializado dentro de una ubicación local especificada por el actor, este archivo contiene toda la información referente al cronograma de los estudios que se deben realizar en los hospitales.
Complejidad	Media
Prioridad	Alta
Precondiciones	El usuario debe estar autenticado en su módulo correspondiente.
Postcondiciones	Se exporta el archivo perteneciente al cronograma seleccionado.
Flujo de eventos	
Flujo básico Exportar Cronograma	

	Actor	Sistema
1.	El caso de uso inicia cuando el actor estando en la página principal selecciona la opción Exportar.	
2.		Abre una nueva interfaz mostrando una lista de estudios y centros. Ver caso de uso Listar estudios y Listar centros a partir de un estudio .
3.	Selecciona un centro.	
4.		Sombrea el centro seleccionado.
5.	Selecciona la opción Exportar Diseño.	
6.		Abre una ventana que permite al usuario explorar los archivos y carpetas existentes en el ordenador y muestra los siguientes campos: <ul style="list-style-type: none"> • Lista de ubicaciones dentro del ordenador. • Lista de elementos dentro de una ubicación. • Nombre del archivo. • Archivos de tipo. Y permite: <ul style="list-style-type: none"> • Guardar. • Cancelar.
7.	Selecciona la ubicación deseada dentro del ordenador.	
8.		Actualiza la lista de elementos disponibles dentro de la ubicación.
9.	Escribe el nombre del archivo a exportar y selecciona la opción Guardar.	
10		Verifica los datos entrados y el espacio disponible.
11		Guarda el archivo en la ubicación especificada por el usuario, finalizando así el caso de uso.
Flujos alternos		
7.a El usuario selecciona la opción Cancelar.		
	Actor	Sistema
1.		El sistema cierra la ventana de exploración retornando al usuario al paso 4 del Flujo Normal de Eventos.
9.a Existe un error en el nombre del archivo.		
	Actor	Sistema
1.		El sistema indica mediante una ventana de diálogo que existe un error al guardar el archivo porque el campo nombre está incorrecto. Brinda la opción:

		<ul style="list-style-type: none"> • Aceptar.
2.	Selecciona la opción Aceptar.	
3.		Cierra el diálogo retornando al paso 9 del Flujo Normal de Eventos.
10.b No existe espacio disponible en la ubicación seleccionada.		
	Actor	Sistema
1.		El sistema indica mediante una ventana de diálogo que existe un error al guardar el archivo porque no se dispone de espacio en disco para ello. Y permite: <ul style="list-style-type: none"> • Aceptar.
2.	Selecciona la opción Aceptar.	
3.		Cierra el diálogo retornando al paso 7 del Flujo Normal de Eventos.
Relaciones	CU incluidos	No existe
	CU extendidos	No existe

Tabla 18. Caso de uso: Cargar para la base de datos el cronograma

Objetivo	Cargar el cronograma asociado a un centro específico para la base de datos.	
Actores	Usuario_Centros	
Resumen	El caso de uso inicia cuando el actor estando en la interfaz del módulo SIGEC Conductor ejecuta la opción Importar, luego al estar en la nueva interfaz selecciona la opción Cargar Cronograma. Permite cargar al sistema un archivo serializado que representa el diseño de los ensayos clínicos realizados.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado en su módulo correspondiente.	
Postcondiciones	Se carga el archivo perteneciente al cronograma seleccionado.	
Flujo de eventos		
Flujo básico Cargar para la base de datos el cronograma		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor estando en la página principal selecciona la opción Importar.	
2.		Abre una nueva interfaz que muestra la opción: <ul style="list-style-type: none"> • Importar Diseño.
3.	Selecciona la opción Importar Diseño.	
4.		Abre una ventana que permite al usuario explorar los archivos y carpetas existentes en el ordenador y muestra los siguientes campos: <ul style="list-style-type: none"> • Lista de ubicaciones dentro del

		ordenador. <ul style="list-style-type: none"> • Lista de elementos dentro de una ubicación. • Nombre del archivo. Y permite: <ul style="list-style-type: none"> • Abrir. • Cancelar.
5.	Selecciona la ubicación deseada dentro del ordenador.	
6.		Actualiza la lista de elementos disponibles dentro de la ubicación.
7.	Selecciona el elemento que desea cargar al sistema.	
8.		Sombrea el archivo seleccionado y escribe el nombre del mismo en el campo Nombre del archivo.
9.	Ejecuta la opción Abrir.	
10		Carga el archivo seleccionado en el sistema, finalizando así el caso de uso.
Flujos alternos		
9.a El usuario selecciona la opción Cancelar.		
	Actor	Sistema
1.		El sistema cierra la ventana de exploración retornando al usuario al paso 2 del Flujo Normal de Eventos.
9.b Existen errores en el archivo a importar		
	Actor	Sistema
1.		Muestra un mensaje de error indicándole al usuario que el archivo posee defectos o está dañado y es imposible cargarlo. Y permite: <ul style="list-style-type: none"> • Aceptar.
2.	Selecciona la opción Aceptar.	
3.		Cierra la ventana de diálogo, retornando al usuario al paso 6 del Flujo Normal de Eventos.
Relaciones	CU incluidos	No existe
	CU extendidos	No existe

Tabla 19. Caso de uso: Cargar la conducción de los estudios realizados

Objetivo	Cargar hacia la base de datos la conducción de los estudios realizados.
Actores	Gerente de datos
Resumen	El caso de uso inicia cuando el actor estando en el módulo

	ejecuta la opción Importar. El sistema abre una interfaz que permite cargar estudio. Permite cargar un archivo serializado que representa los estudios de los ensayos clínicos realizados.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado en su módulo correspondiente.	
Postcondiciones	Se exporta el archivo perteneciente al cronograma seleccionado.	
Flujo de eventos		
Flujo básico Cargar la conducción de los estudios realizados		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor estando en el módulo ejecuta la opción Importar.	
2.		Permite: <ul style="list-style-type: none"> • Importar Conducción
3.	El usuario selecciona la opción Importar Conducción.	
4.		Abre una ventana que permite al usuario explorar los archivos y carpetas existentes en el ordenador y muestra los siguientes campos: <ul style="list-style-type: none"> • Lista de ubicaciones dentro del ordenador. • Lista de elementos dentro de una ubicación. • Nombre del archivo. Y permite: <ul style="list-style-type: none"> • Abrir. • Cancelar.
5.	Selecciona la ubicación deseada dentro del ordenador.	
6.		Actualiza la lista de elementos disponibles dentro de la ubicación.
7.	Selecciona el elemento que desea cargar al sistema.	
8.		Sombrea el archivo seleccionado y escribe el nombre del mismo en el campo Nombre del archivo.
9.	Ejecuta la opción Abrir.	
10.		Carga el archivo seleccionado en el sistema, finalizando así el caso de uso.
Flujos alternos		
9.a El usuario selecciona la opción Cancelar.		
	Actor	Sistema
1.		El sistema cierra la ventana de exploración retornando al usuario al paso 2 del Flujo Normal de Eventos.

9.b Existen errores en el archivo a importar.		
	Actor	Sistema
1.		Muestra un mensaje de error indicándole al usuario que el archivo posee defectos o está dañado y es imposible cargarlo. Y permite: <ul style="list-style-type: none"> • Aceptar
2.	Selecciona la opción Aceptar.	
3.		Cierra la ventana de diálogo, retornando al usuario al paso 6 del Flujo Normal de Eventos.
13.b No existe espacio disponible en la ubicación seleccionada.		
	Actor	Sistema
1.		El sistema indica mediante una ventana de diálogo que existe un error al guardar el archivo porque no se dispone de espacio en disco para ello. Brinda la opción: <ul style="list-style-type: none"> • Aceptar.
2.	Selecciona la opción Aceptar.	
3.		Cierra el diálogo retornando al paso 10 del Flujo Normal de Eventos.
Relaciones	CU incluidos	No existe
	CU extendidos	No existe

Tabla 20. Caso de uso: Listar estudios del centro

Objetivo	Ver listado de los estudios que se están realizando en el centro.	
Actores	Usuario_Centros	
Resumen	El caso de uso inicia cuando el actor ingresa en el módulo SIGEC Conductor. Permite ver la lista de los estudios existentes.	
Complejidad	Baja	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado en su módulo correspondiente.	
Postcondiciones	Se muestra el listado de los estudios que se realizan.	
Flujo de eventos		
Flujo básico Listar estudios		
	Actor	Sistema
1.	Ingresa en el módulo SIGEC Conductor	

2.		<p>Muestra el listado de los estudios existentes en ese momento. Permite:</p> <ul style="list-style-type: none"> • Exportar Conducción. <p>Finaliza así el caso de uso.</p>
----	--	---

Anexo 3: Diagramas de Clases del Análisis

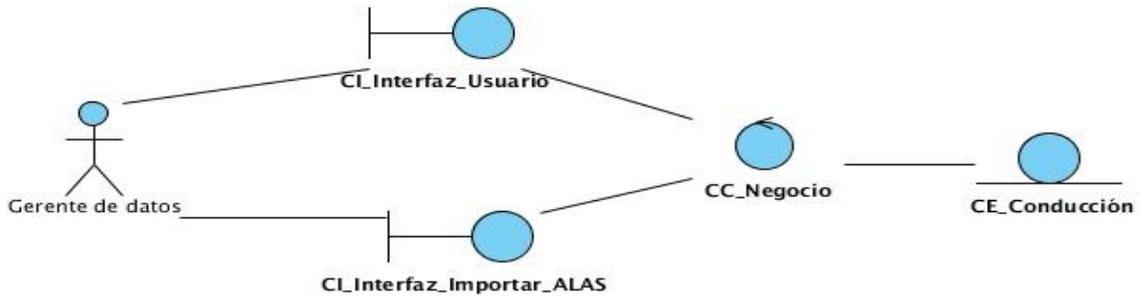


Figura.17 Caso de uso Cargar la conducción de los estudios realizados

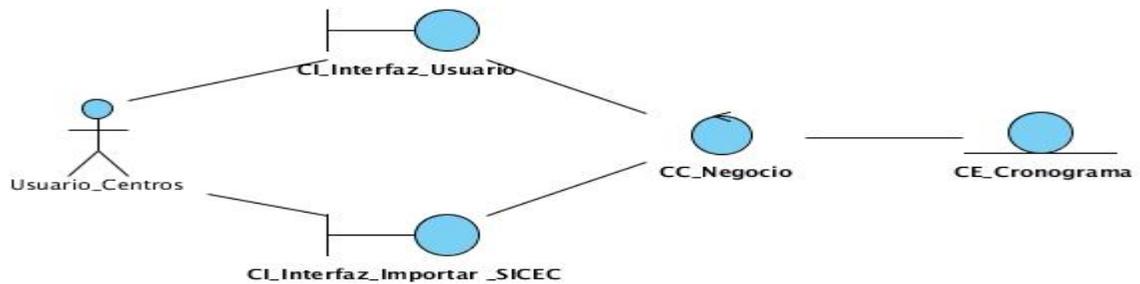


Figura.18 Caso de uso: Cargar para la base de datos el cronograma

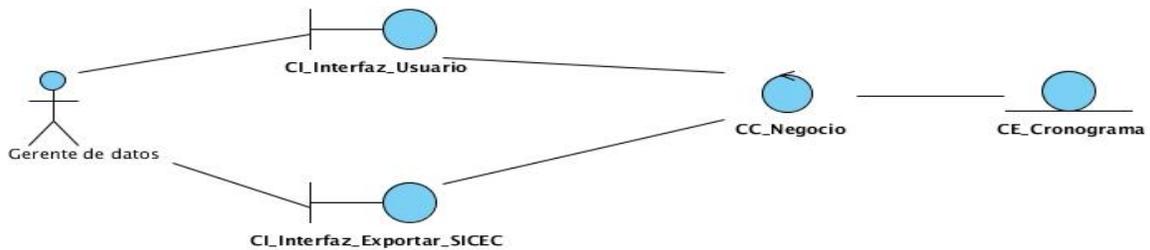


Figura. 19 Caso de uso: Exportar cronograma

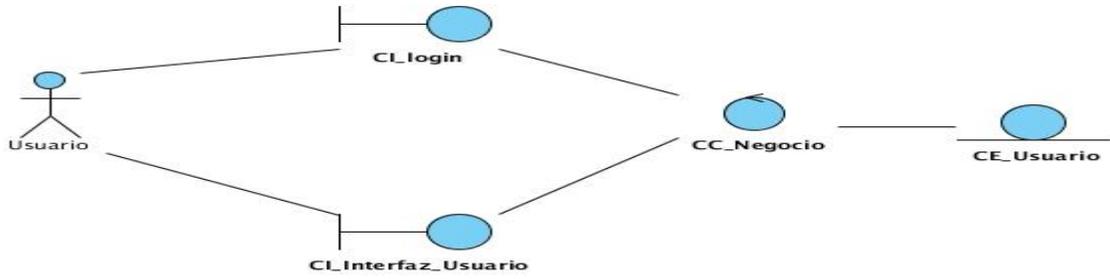


Figura.20 Caso de uso: Autenticar usuario

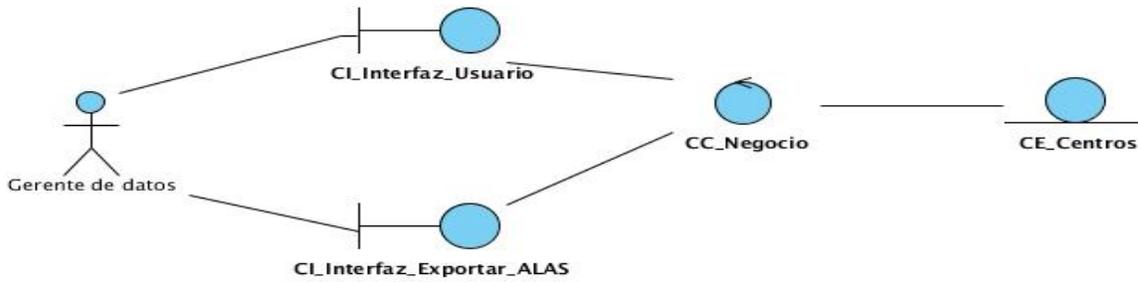


Figura.21 Caso de uso: Listar centros a partir de un estudio

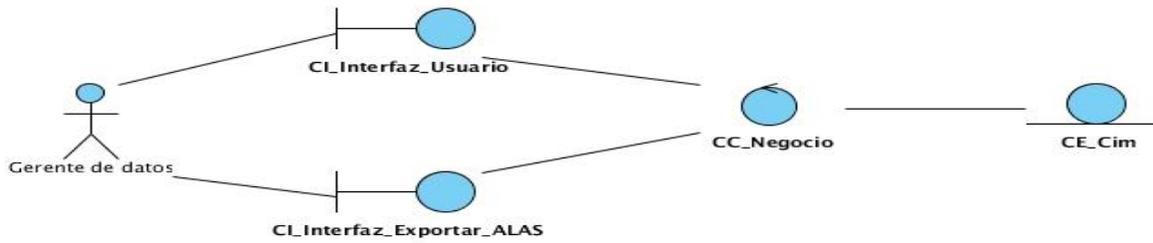
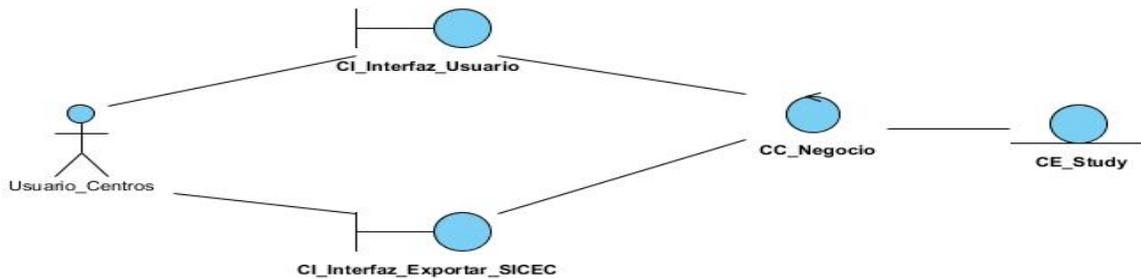


Figura.22 Caso de uso: Listar estudios



a.23 Caso de uso: Listar estudios del centro

Figur

Anexo 4: Diagramas de Secuencia

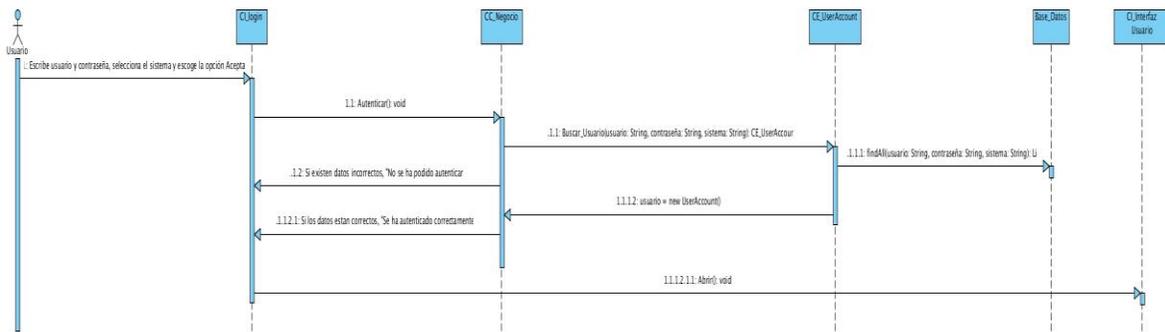


Figura.24 Caso de uso Autenticar usuario

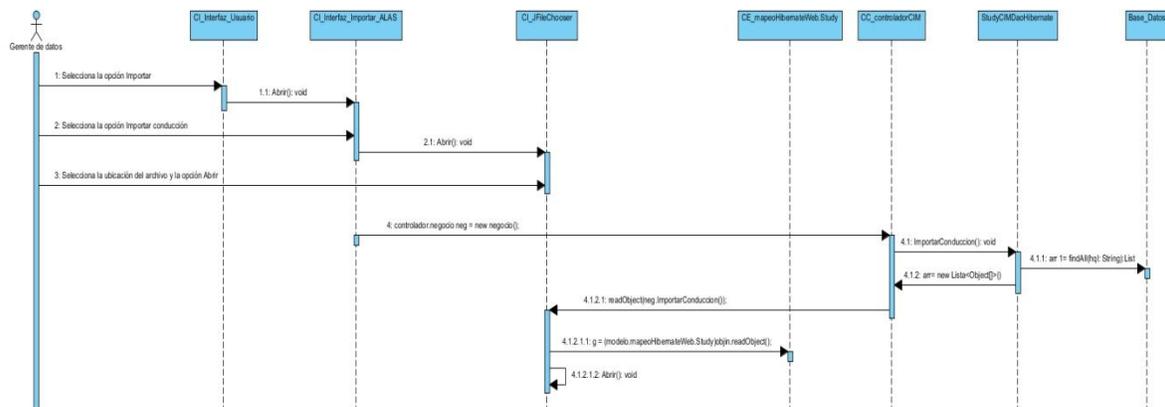


Figura.25 Caso de uso Cargar la conducción de los estudios realizados.

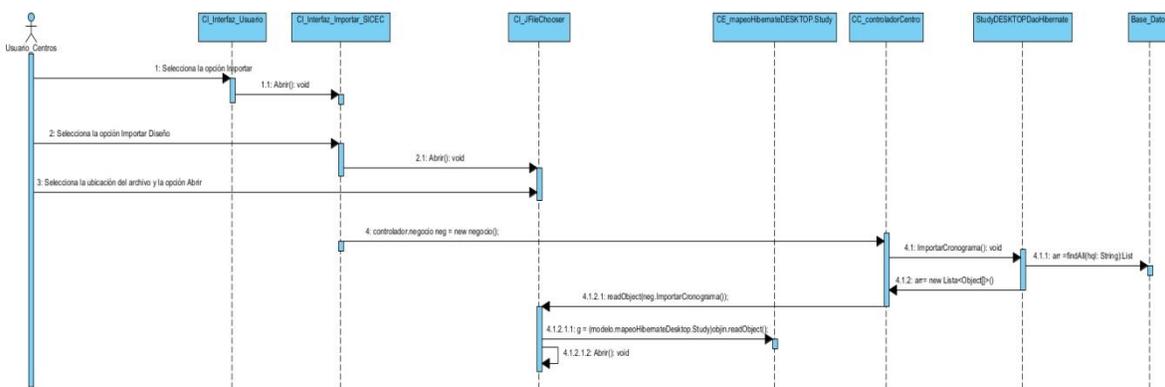


Figura.26 Caso de uso Cargar para la base de datos el cronograma.

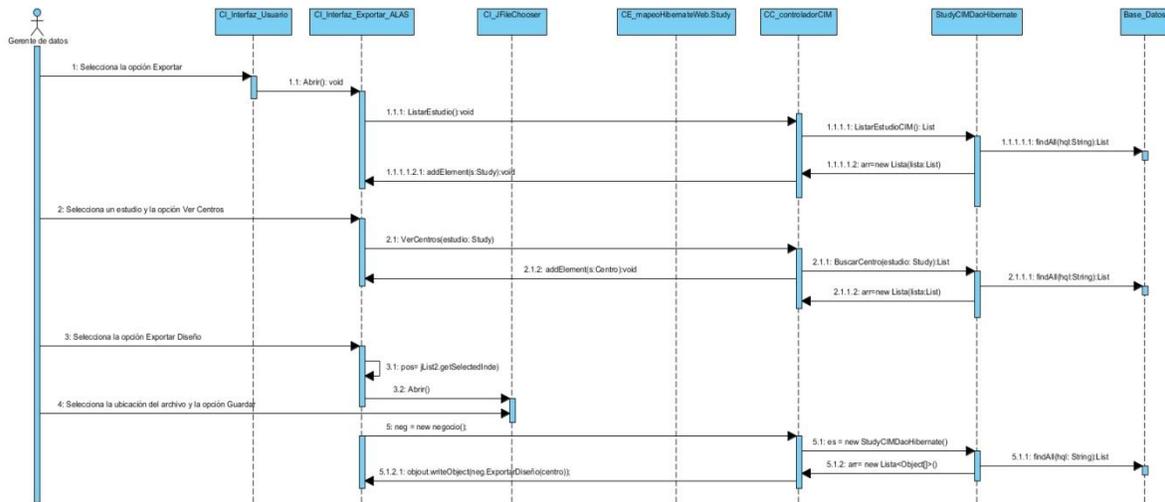


Figura.27 Caso de uso Exportar cronograma.

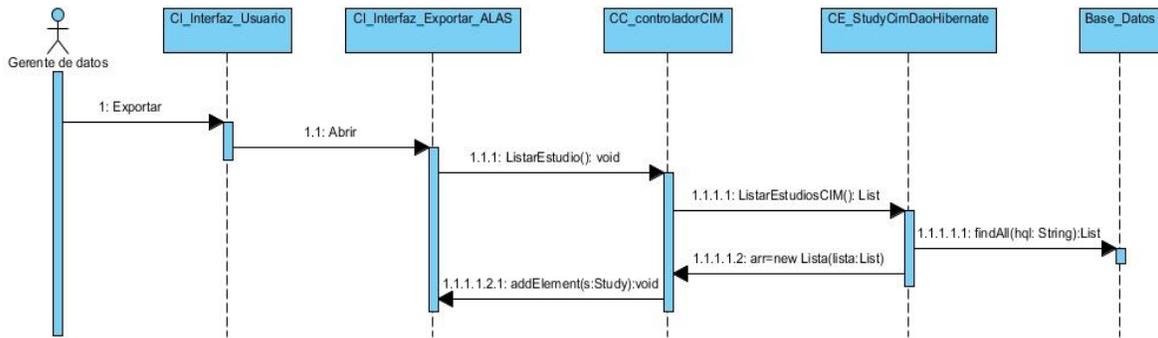


Figura.28 Caso de uso Listar estudios

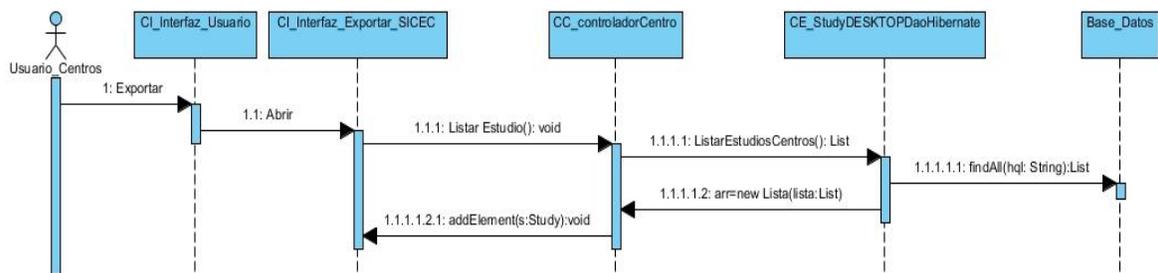


Figura.29 Caso de uso Listar estudios del centro

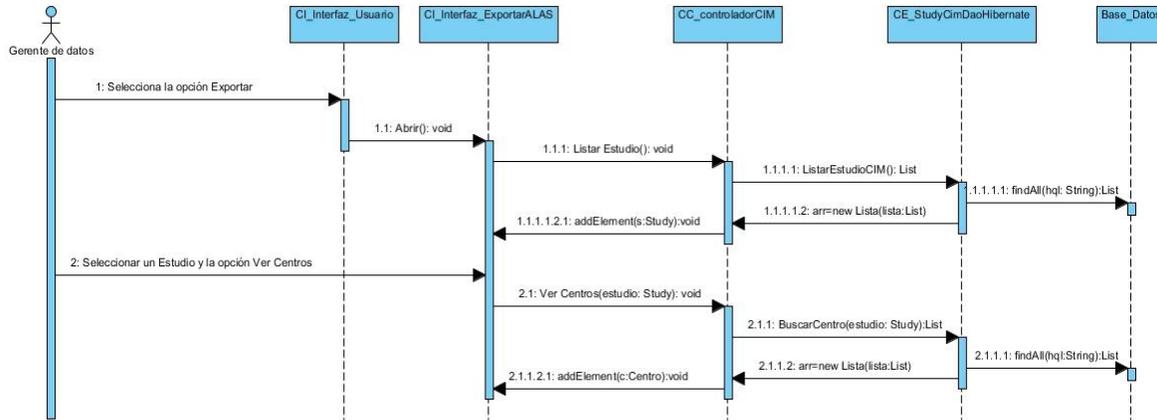


Figura.30 Caso de uso Listar centros a partir de un estudio

Anexo 5: Diagramas de clases del diseño

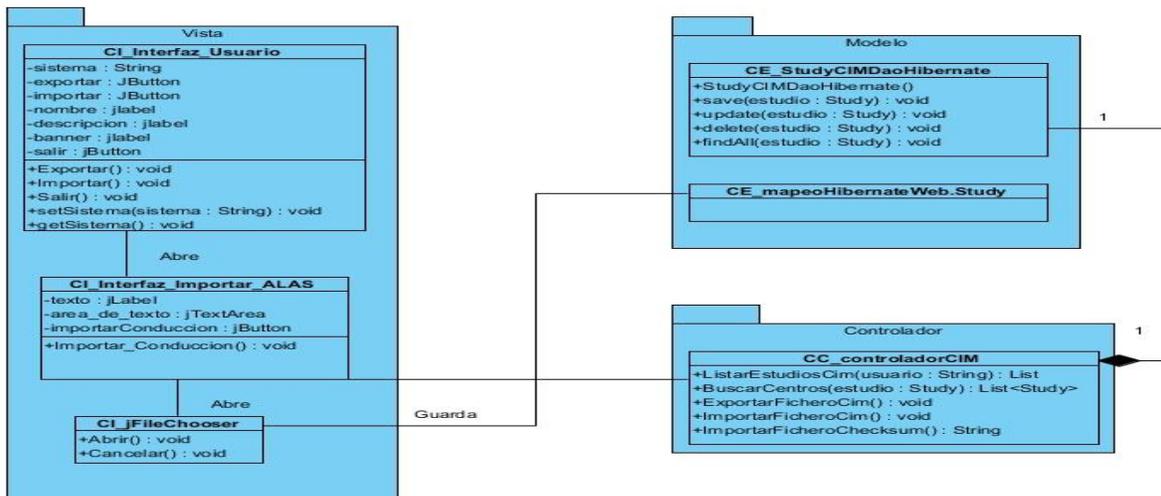


Figura.31 Caso de uso Cargar la conducción de los estudios realizados

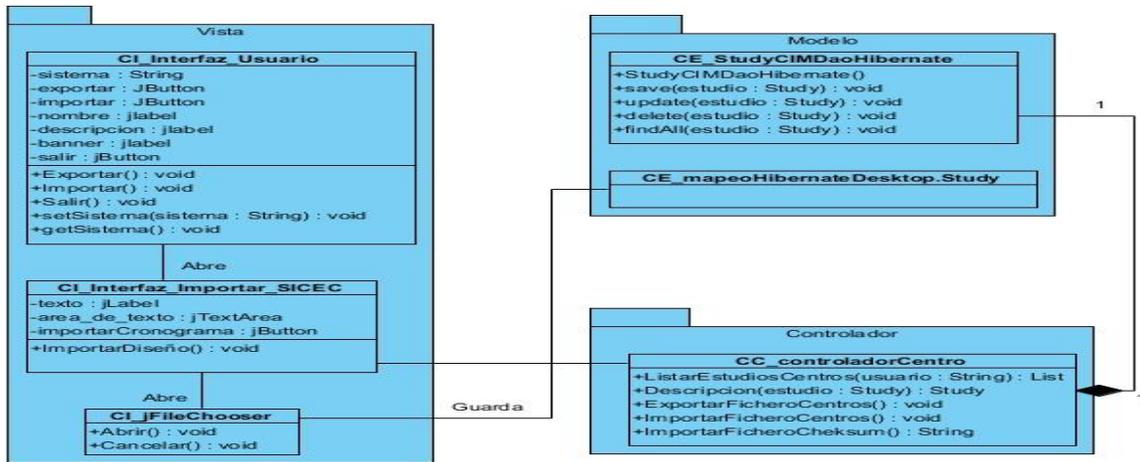


Figura.32 Caso de uso Cargar para la base de datos el cronograma

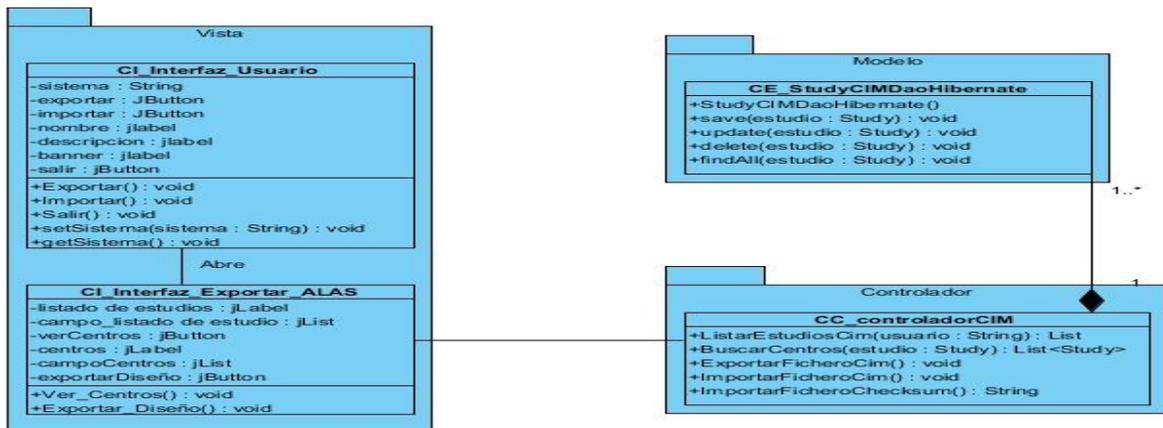


Figura.33 Caso de uso Listar estudios.

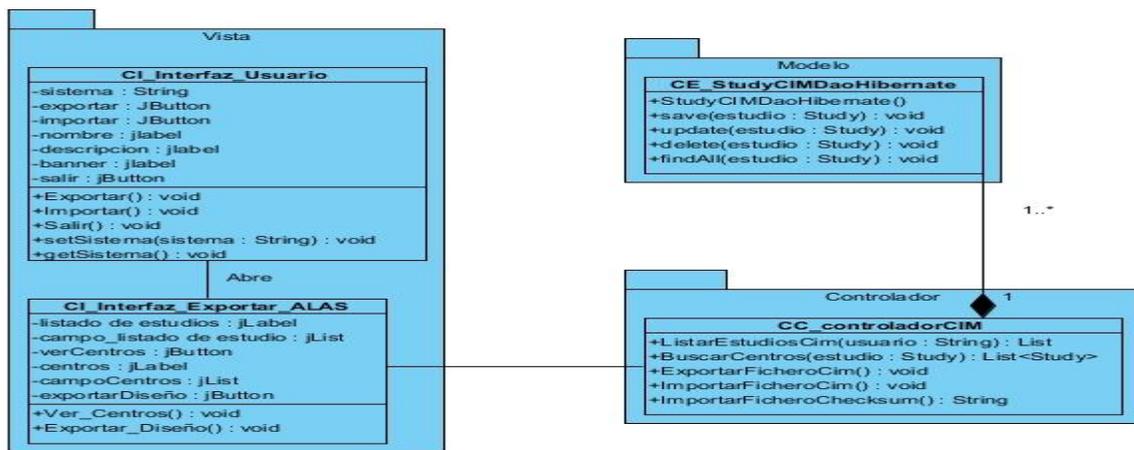


Figura.34 Caso de uso Listar centros a partir de un estudio

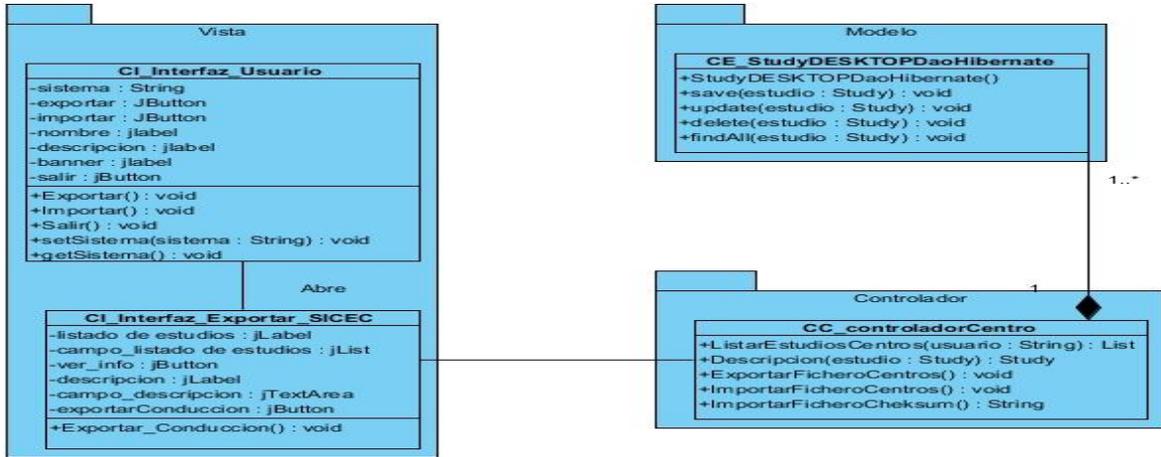


Figura.35 Caso de uso Listar estudios del centro

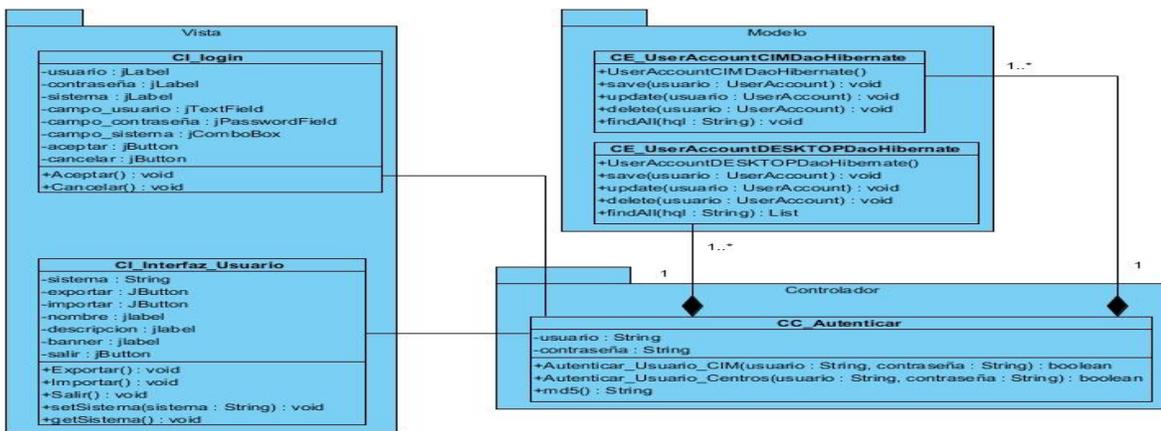


Figura.36 Caso de uso Autenticar usuario

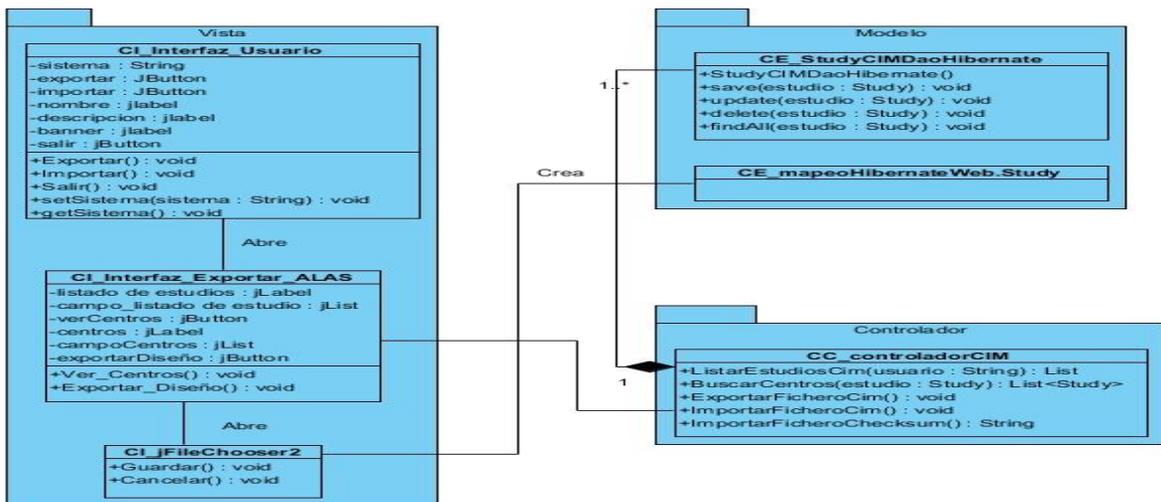


Figura.37 Caso de uso Exportar cronograma

Glosario de términos

BSD: Licencia Distribución de Software Berkeley.

CASE: Ingeniería de Software Asistida por Computadora.

Caso de Uso: Secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias.

Centro: recoge información relacionada a los sitios u hospitales tales como: Nombre del centro, ciudad, provincia, país, código postal, teléfono de contacto y correo electrónico.

Cronograma: representa la planificación de un estudio determinado en un centro específico, constituye la programación de un conjunto de Momentos de Seguimientos a los cuales estarán asociados uno o un conjunto de Hojas CRD, el cual será aplicado a un cierto número de sujetos. Entre los datos que posee están: la fecha en que se iniciará el estudio, la fecha de fin, entre otros.

Estudio: recoge información como: nombre del estudio, fecha de creación, investigador principal, descripción, estado del estudio y otras que pueden ser especificadas en el momento que se crea el mismo.

Expediente de proyecto: Conjunto de documentos y plantillas estructura siguiendo una jerarquía que constituyen referencia para la documentación de los proyectos de la Universidad de las Ciencias Informáticas.

GPL: Licencia Pública General.

Hardware: es el conjunto de los componentes que integran la parte material de una computadora.

Hibernate: librerías de mapeo objeto-relacional (ORM) utilizado en las aplicaciones Java. ORM es una técnica de programación para convertir datos entre el lenguaje de programación orientado a objetos y el sistema de base de datos relacional.

Hojas CRD: describe cómo se recoge toda la información relacionada a un sujeto con un determinado estudio. Esta recoge varios datos como el nombre del CRD, la fecha de inicio, descripción, entre otros.

LGPL: Licencia Pública General Reducida

Miembros del proyecto: personal interno del proyecto.

Réplica: término general para una copia de una recopilación de datos

Requisito: (1) Una condición o capacidad necesitada por el usuario para resolver un problema o lograr un objetivo. (2) Una condición o capacidad que debe poseer un producto o componente de producto para satisfacer un contrato, estándar, especificación, u otros documentos obligatorios formales. (3) Una representación documentada de una condición o capacidad como en (1) o (2).

Sincronización: se refiere al proceso de propagación de los cambios en los datos y el esquema entre la fuente de datos y los destinos después de haber aplicado la instantánea inicial en el destino.

Sistema: en informática es el conjunto de partes interrelacionadas, hardware, software y de recurso humano que permite almacenar y procesar información.

Software: es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

SQL: lenguaje estructurado de consulta, compuesto por comandos, cláusulas, operadores y funciones, que se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

Stakeholder: En la suite de productos de CMMI, un grupo o individuo que es afectado por o es de alguna forma responsable del resultado de la tarea. Stakeholders pueden incluir miembros del proyecto, proveedores, clientes, usuarios finales, entre otros.

Triggers: procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación sobre una base de datos. Es conocido como disparador en algunos contextos.