

Universidad de las Ciencias Informáticas

Facultad 2



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

Título: *Componente para la gestión y monitoreo de datos de
ensayos clínicos en centros sin acceso al sistema Clínicas 1.0*

Autora: Arianne Sayú Ramirez

Tutor: Ing. Yanssel Urquijo Morales

Co-Tutora: Ing. Mayté Estevez Crespo

La Habana, junio de 2014

Año 56 de la Revolución

Declaración de Autoría

Declaro que soy lo única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Arianne Sayú Ramirez

Firma del Autor

Ing. Yanssel Urquijo Morales

Firma del Tutor

Ing. Mayté Estevez Crespo

Firma del Co-Tutor

Datos de Contacto

Tutor: Ing. Yanssel Urquijo Morales

Correo electrónico: yurquijo@uci.cu

Universidad de las Ciencias Informáticas

Co-Tutora: Ing. Mayté Estevez Crespo

Correo electrónico: maytee@uci.cu

Universidad de las Ciencias Informáticas

Agradecimientos

Quiero agradecerle a toda mi familia por apoyarme siempre y por brindarme su guía.

A mi mamá por ser la luz que me mostró el camino, por quererme, apoyarme y darme mucho amor. Sin ella no hubiera llegado hasta aquí. Por ser más que mi madre, mi amiga, por darme y enseñarme la vida.

A mi abuela Nena, por quererme tanto y siempre estar presente para mí. Por confiar siempre en mí y en mis decisiones.

A mi compañero de tesis Oscar por su apoyo incondicional y a Roanny por ser partícipe de que este momento existiese.

A mi novio Terry, por ayudarme a cumplir mi sueño, por acompañarme durante los 3 últimos años de la carrera y regalarme todos los momentos que compartimos.

A mis compañeras de apartamento Yisel, Annies, Ania, Yanet, Dayana, Yanei, Geydi y Tita. A mis amigas Yisel, Yainet y mi prima Yindra por nuestra hermosa amistad. A los varones de mi grupo Leodán, Maikel, Gianni, Cutiño, Palmero y en especial el loco de Alberto. A Yuset el gastronómico que se convirtió en mi tío universitario. A la locaria de Ary y su esposo Bencomo.

A mis tutores. A todas las personas de Guantánamo que me quieren y me han apoyado.

A todos los aquí presentes.

DEDICATORIA

A mi familia, por confiar en mí, por apoyarme con amor infinito en todos los retos que he afrontado en mi carrera. En especial a mi mamá Arais, que ha sido madre y padre durante estos 22 años de mi vida.

A todas aquellas personas que me quieren y me desean bien y a Oscar que siempre será mi compañero de tesis.

Resumen

Con el presente trabajo se tiene como objetivo diseñar e implementar las funcionalidades que se encargan de la gestión y el monitoreo de datos, durante la conducción de ensayos clínicos en las diferentes instituciones hospitalarias que no tienen acceso al sistema Clínica 1.0.

Para el desarrollo de las funcionalidades se utilizaron: como gestor de base de datos PostgreSQL, el lenguaje de programación Java, el entorno de desarrollo integrado NetBeans y como metodología de desarrollo de software el Proceso Unificado de Desarrollo (RUP). Para la creación de los artefactos se utilizó el Lenguaje Unificado de Modelado, Visual Paradigm como herramienta CASE y como arquitectura 3 capas.

Se prevé que el sistema permita eliminar el procesamiento manual del registro de todos los datos obtenidos en un estudio. Además de garantizar el registro y monitoreo de datos asociados a los sujetos y la creación de notas y discrepancias de un estudio, proporcionando así de una forma más rápida la creación de nuevos medicamentos destinados al tratamiento del cáncer y otras enfermedades incurables.

Palabras clave:

Ensayos clínicos, gestión y monitoreo de datos, estudio

Índice

Introducción	1
Capítulo1: Fundamentación Teórica de la Investigación	6
1.1 Conceptos asociados al dominio del problema	6
1.2 Gestión de ensayos clínicos	6
1.3 Sistemas de Gestión de Ensayos Clínicos.....	7
1.3.1 Sistemas internacionales.....	7
1.3.2 Sistemas nacionales.....	9
1.4 Aplicación Desktop (también llamada de Escritorio)	10
1.5 Metodología, tecnologías y herramientas.....	11
1.5.1 Metodología de desarrollo	11
1.5.2 Lenguaje de programación	12
1.5.3 Entorno de Desarrollo Integrado (IDE): Netbeans versión 7.4.....	13
1.5.4 Librería Swing.....	13
1.5.5 Máquina Virtual de Java: Open-jdk-8.0.....	14
1.5.6 Herramienta de Mapeo objeto-relacional (ORM): Hibernate versión 3.6.10	14
1.5.7 Herramienta CASE: Visual Paradigm para UML versión 8.0.....	15
1.5.8 Lenguaje Unificado de Modelado: UML versión 2.0	15
1.5.9 Herramienta de Administración de la Base de Datos: PgAdmin3 versión 1.16.1.....	16
1.5.10 Sistema Gestor de Base de Datos: PostgreSQL versión 9.2.4.....	16
Capítulo 2: Características del Sistema	18
2.1 Objeto de automatización	18
2.2 Modelado del dominio.....	18

2.2.1	Descripción del flujo del diagrama del modelo de dominio.....	19
2.2.2	Definición de los conceptos del modelo del dominio	20
2.3	Propuesta del Sistema.....	20
2.3.1	Requisitos funcionales.....	21
2.3.2	Requisitos no funcionales	24
2.3.3	Actores del Sistema.....	25
2.3.4	Diagrama de Casos de Uso del Sistema.....	27
2.3.5	Descripción de los Casos de Uso del Sistema	28
Capítulo 3:	Diseño del Sistema	36
3.1	Descripción de la Arquitectura	36
3.1.1	Arquitectura en 3 capas.....	36
3.1.2	Patrones de diseño.....	37
3.2	Modelo de Diseño.....	41
3.2.1	Diagramas de clases del diseño	41
3.2.2	Diagramas de colaboración	48
Capítulo 4:	Implementación del Sistema	51
4.1	Modelo de datos	51
4.2	Implementación	55
4.2.1	Diagrama de Despliegue	55
4.2.2	Diagrama de Componentes	56
4.2.3	Estilos y estándares de codificación	57
4.2.4	Tratamiento de Excepciones.....	59
Conclusiones	60

Tabla de contenido

Recomendaciones	61
Referencias Bibliográficas.....	62
Bibliografía.....	66
Glosario de Términos.....	70
Anexo	73
Anexo1	73

Índice de Tablas

Tabla 1. Requisitos funcionales del sistema.....	24
Tabla 2. Descripción de los Actores del Sistema	27
Tabla 3. Caso de uso Mostrar Notas.....	31
Tabla 4. Caso de uso Crear Nota General	32
Tabla 5. Caso Uso Responder Nota General	34
Tabla 6. Descripción de las clases: Mostrar Notas.....	44
Tabla 7. Descripción de las clases: Crear Nota General	46
Tabla 8. Descripción de las clases: Responder Nota General.....	48
Tabla 9. Descripción de estereotipos	49
Tabla 10. Descripción de la entidad Nota de discrepancias	53
Tabla 11. Descripción de la entidad Tipo de notas de discrepancia	54
Tabla 12. Descripción de la entidad Estado de momentos de seguimiento.....	54
Tabla 13. Descripción de la entidad Estado de resolución	55

Índice de Ilustraciones

Ilustración 1. Modelo de dominio.....	19
Ilustración 2. Actores del Sistema	26
Ilustración 3. Diagrama de Casos de Usos del Sistema.....	28
Ilustración 4. Arquitectura en 3 capas	36
Ilustración 5. Uso del Patrón DAO	38
Ilustración 6. Uso del patrón Alta cohesión	39
Ilustración 7. Uso del patrón Bajo acoplamiento	39
Ilustración 8. Uso del patrón Experto	40
Ilustración 9. Uso del patrón Creador.....	40
Ilustración 10. Uso del patrón Controlador	41
Ilustración 11. Diagrama de clases del diseño Mostrar Notas	42
Ilustración 12. Diagrama de clases del diseño Crear Nota General	45
Ilustración 13. Diagrama de clases del diseño Responder Nota general.....	47
Ilustración 14. Diagrama de colaboración Mostrar Notas	49
Ilustración 15. Diagrama de colaboración Crear Nota General.....	49
Ilustración 16. Diagrama de colaboración Responder Nota General	50
Ilustración 17. Modelo de datos	52
Ilustración 18. Diagrama de despliegue	55
Ilustración 19. Diagrama de Componentes	57

Introducción

Desde sus orígenes, el hombre ha intentado mejorar sus procesos de vida, impulsando así el desarrollo de sus medios. Sin dudas desde su comienzo trató de utilizar los conocimientos obtenidos en la rama dedicada a la salud para su bienestar, promoviendo de esta forma la tecnología hacia el campo de la medicina. Desde finales del siglo XX el vertiginoso desarrollo alcanzado por las Tecnologías de la Información y las Comunicaciones (TIC) ha repercutido extraordinariamente en todos los aspectos de la sociedad.

Con el surgimiento de las TIC el campo de la medicina se ha dotado de nuevas formas para mejorar los procesos vitales, su uso ha dado paso a la obtención de mejores resultados en el diagnóstico y cura de enfermedades, mejorando la calidad de vida humana. Muchos países han optado por el desarrollo de las TIC para lograr una evolución en el desarrollo de herramientas dirigidas a automatizar procesos en áreas como la información, planificación, investigación, gestión, prevención y promoción en el diagnóstico o tratamiento de enfermedades.

Existen en el mundo numerosos centros de investigaciones científicas cuyo objetivo principal es la fabricación de medicamentos para el tratamiento de enfermedades incurables, situados principalmente en países con un alto nivel de desarrollo. Para lograr la creación de cada uno de estos fármacos se realizan los ensayos clínicos (EC) o estudios clínicos como también se les conoce, que no son más que la evaluación práctica de un producto, sustancia o medicamento prometedor, cuya aplicación en humanos pretende valorar su efectividad, así como su nivel de vida. Cada estudio responde preguntas científicas e intenta encontrar mejores formas de prevenir, explorar, diagnosticar o tratar una enfermedad. (1)

En la medida en que se avanza en el desarrollo de la tecnología, se avanza en la gestión de datos de los EC, esto trae aparejado el incremento del número de pacientes a estudiar y por consiguiente el número de hospitales que se incorporan en el reclutamiento de pacientes. Las compañías farmacéuticas y biotecnológicas, para lograr una gestión eficiente de los ensayos clínicos, requieren del uso de un sistema de gestión de ensayos clínicos (CTMS por sus siglas en inglés) confiable, que apoye a los científicos en el monitoreo de datos de ensayos clínicos y a la investigación de los efectos adversos de los medicamentos a partir de datos clínicos.

Cuba no está exenta de esta competencia, pues ha adquirido a lo largo de los años experiencia en el tema de los ensayos clínicos. En la actualidad existen en el país varios centros para la creación de fármacos,

entre ellos se encuentran el Centro de Investigación Instituto Finlay o Centro para la Investigación, Desarrollo y Producción de Vacunas y Sueros, el Centro de Ingeniería Genética y Biotecnología y el Centro de Inmunología Molecular (CIM), una institución biotecnológica cubana dedicada a la investigación básica, desarrollo y fabricación de productos. (2)

El objetivo principal de las investigaciones en el CIM es la búsqueda de nuevos productos para el diagnóstico y tratamiento del cáncer y enfermedades relacionadas con el sistema inmune. Dicho centro realiza, en hospitales altamente especializados, ensayos clínicos para el tratamiento de cáncer de diferentes orígenes y otras enfermedades. (2)

Es de vital importancia mantener un estricto control sobre toda la información almacenada de cada uno de los EC, teniendo en cuenta que son investigaciones efectuadas en seres humanos con el fin de determinar o confirmar efectos clínicos, farmacológicos y otros efectos, así como detectar las reacciones adversas y/o estudiar la absorción, distribución, metabolismo e interacción con otros medicamentos con el fin de determinar su inocuidad o eficacia. (3)

Cada ensayo realizado a uno o varios productos es almacenado, pero debe ser consultado y evaluado por diversos grupos de especialistas con el fin de mejorar los resultados o determinar si son efectivos. Por lo que es muy necesario que la información esté disponible y de forma legible para que pueda ser consultada en el momento necesario.

En la actualidad se hace engorroso el monitoreo y la comparación de resultados porque la realización del registro de todos los datos obtenidos para un estudio en específico es de forma manual, donde muchas veces tiene que ser buscada la información en archivos muy grandes, retrasando el proceso de aprobación de un ensayo clínico.

Una vez obtenidos todos los resultados, son almacenados en una base de datos, lo que trae consigo dificultades al transcribir la información de un formato duro a un formato electrónico, además, de incurrir en atraso y errores en el llenado de los modelos, falta de estandarización e interpretaciones erróneas. De forma similar, la información archivada en formato duro, se encuentra expuesta a sucesos impredecibles que pueden provocar la pérdida o deterioro de la misma.

Para facilitar la gestión de ensayos clínicos el CIM utiliza el sistema informático Clínicas 1.0, desarrollado por el departamento de Sistemas Especializados en Salud (SES) en la Universidad de las Ciencias

Informáticas (UCI). El sistema consta de cuatro módulos que permiten el diseño y conducción de los ensayos: Administrar Sistema, Gestionar Estudio, Extraer Datos y Enviar Datos.

Clínicas 1.0 es una aplicación web, donde el llenado de los datos se realiza desde las vistas proporcionadas por el mismo. En la actualidad la aplicación no está a disposición de cada institución hospitalaria del país, por lo que muchos hospitales continúan realizando todo el proceso de conducción de ensayos clínicos de forma manual. Para que un sistema de este tipo pueda ser desplegado en los centros que existen en la isla, sería necesaria una infraestructura tecnológica que aún no se ha alcanzado, teniendo como consecuencia que el acceso al mismo solo pueda ser desde un centro que contenga conexión disponible prácticamente todo el tiempo a la red nacional y al servidor central.

Para acceder a los módulos del sistema se deberían conectar especialistas desde los hospitales de todo el país lo cual se hace imposible, incurriendo en que no sea óptimo el proceso de monitoreo que se realiza en las diferentes provincias, donde los monitores tienen que revisar las hojas de cuaderno de recogida de datos (CRD¹) durante las visitas de monitorización.

Por otra parte, a pesar de las diversas funcionalidades que posee el sistema Clínicas 1.0, debido a que algunos hospitales especializados que realizan la gestión y monitoreo de los datos no tienen acceso al sistema, no es posible registrar y monitorear todos los datos de los sujetos asociados a los ensayos, esto trae consigo que no se gestione toda la información necesaria, además, de dificultades y demoras en el proceso de aprobación de un ensayo. De forma similar impide dar respuesta rápida a problemas que en el modo tradicional tardan semanas y meses en conocerse y en consecuencia, más tiempo en solucionarse. A raíz de lo anteriormente planteado, se identifica como **problema a resolver**:

¿Cómo facilitar la gestión y el monitoreo de datos de los ensayos clínicos en centros sin acceso al sistema Clínicas 1.0?

Basado en el problema planteado se define como **objeto de estudio**:

- Proceso de gestión de ensayos clínicos.

El **campo de acción** se enmarca en:

¹ Cuaderno de Recogida de Datos

- Gestión y monitoreo de datos de ensayos clínicos.

Para dar solución al problema planteado se propone como **objetivo general**:

- Desarrollar un componente que facilite la gestión y el monitoreo de datos de los ensayos clínicos en centros sin acceso al sistema Clínicas 1.0.

Para dar cumplimiento al objetivo de la investigación se desarrollaron las siguientes **tareas**:

- Establecer los fundamentos teóricos-metodológicos para el desarrollo de sistemas informáticos de gestión de ensayos clínicos.
- Analizar las funcionalidades del módulo Enviar Datos del sistema Clínicas 1.0.
- Describir las tecnologías, metodologías y herramientas para la solución del problema que se enfrenta.
- Generar los artefactos correspondientes a los flujos: Modelado del Negocio, Requisitos, Análisis y Diseño e Implementación, según la metodología RUP.
- Desarrollar el sistema informático para la gestión y el monitoreo de datos de ensayos clínicos en centros sin acceso al sistema Clínicas 1.0.

Beneficios

La utilización de la solución desarrollada traerá consigo los siguientes beneficios a los especialistas de los centros sin acceso al sistema Clínicas 1.0:

- Eliminará los problemas de dependencia de la red en las instituciones hospitalarias, contribuyendo a la informatización de la gestión y monitoreo de datos de ensayos clínicos realizada por los especialistas.
- Evitará la pérdida, duplicidad y deterioro de la información gestionada por los especialistas en los diferentes centros sin acceso al sistema Clínicas 1.0.

Se emplearon los siguientes métodos científicos:

Métodos Teóricos:

- **Analítico-sintético:** Se utiliza para el estudio de las diferentes bibliografías asociadas a la gestión de EC, de las cuales se tomaron los rasgos distintivos más importantes con el objetivo de arribar a conclusiones que sustentaran la necesidad de la investigación.
- **Histórico-lógico:** Se utiliza para analizar la evolución histórica de soluciones similares existentes, las tendencias más recientes del desarrollo de aplicaciones para los EC y complementar las características necesarias para la solución que se propone.
- **Modelado:** Se utiliza para realizar el modelado del sistema informático a través de los artefactos correspondientes a las fases: Modelo del Negocio, Requisitos, Análisis y Diseño e Implementación.

Método Empírico:

- **Entrevista:** La entrevista es una técnica de interrogación donde se desarrolla una conversación planificada con el sujeto entrevistado. Como tipo de entrevista se utiliza la **no estructurada** o **no estandarizada**, para obtener diferentes puntos de vistas respecto al problema a resolver así como la identificación de las funcionalidades a implementar.

El presente trabajo está estructurado en 4 capítulos:

CAPÍTULO 1: Fundamentación Teórica de la Investigación se brinda el estado del arte del proceso de gestión de ensayos clínicos; se describen la metodología de desarrollo a utilizar, las tecnologías y herramientas utilizadas para la solución propuesta.

CAPÍTULO 2: Características del Sistema se describe el funcionamiento del negocio a través del modelo de dominio; se muestran los requisitos funcionales y no funcionales y el diagrama de casos de uso con sus respectivas descripciones.

CAPÍTULO 3: Diseño del Sistema se describen los patrones de diseño a utilizar y se explican los aspectos referentes a la arquitectura de software, definiéndose el patrón arquitectónico a aplicar, además, comprende elementos importantes del diseño tales como diagramas y descripción de las clases.

CAPÍTULO 4: Implementación del Sistema se realiza la descripción de la relación del sistema con la base de datos y sus tablas, el diagrama de despliegue y los de componentes, se describe cómo se implementa el sistema, logrando de esta manera alcanzar el objetivo general del trabajo y darle solución al problema a resolver.

Capítulo1: Fundamentación Teórica de la Investigación

El presente capítulo contiene los conceptos fundamentales referentes al objeto de estudio de la investigación, además, se realiza un estudio de los Sistemas de Gestión de Ensayos Clínicos existentes tanto a nivel nacional como internacional. De igual manera se hace una descripción de la metodología, las tecnologías y herramientas utilizadas para el desarrollo de la solución.

1.1 Conceptos asociados al dominio del problema

Con el propósito de que el lector pueda alcanzar una mejor comprensión de los temas que serán tratados en este capítulo, se describen a continuación los conceptos más significativos, relacionados con el dominio del problema de la investigación.

Gestión de Datos de Ensayos Clínicos

La Gestión de Datos de Ensayos Clínicos asegura la calidad de la recopilación de datos recogidos en los ensayos clínicos, datos asociados a cada hoja CRD y de los pacientes por estudios, esto contiene actividades como: registrar pacientes, chequear sus datos y monitorear todos los datos de los sujetos.(4)

Monitoreo de Datos de Ensayos Clínicos

El Monitoreo de Datos de Ensayos Clínicos es un proceso que permite detectar y reportar las dificultades que no permitan la inserción o la evaluación adecuada de los pacientes, mediante este proceso se originan notas de discrepancias a partir de los CRD llenados en los hospitales, con el objetivo de facilitar el análisis final de los resultados de cada ensayo. (5)

Cuaderno de recogida de datos (CRD)

Los cuadernos de recogida de datos son formularios diseñados para anotar toda la información recogida durante un ensayo clínico. (6)

1.2 Gestión de ensayos clínicos

El proceso de gestión de ensayos clínicos se divide en dos etapas (7):

- **Diseño de los estudios.** En esta etapa se crea el estudio definiéndose los centros donde se conducirá el mismo; de igual forma se crean los CRD y se definen sus variables y un conjunto de reglas de validación y derivación que deben cumplirse en el llenado de los datos de cada una de las hojas CRD.

- **Conducción de los estudios.** En esta etapa se incluyen los sujetos a los cuales se les realizarán los estudios, además se gestionan sus datos personales para ser ubicados en un grupo según sus características, enfermedades o tratamientos. Por otra parte se realiza el monitoreo de los datos asociados a los sujetos y se procede a realizar el análisis de los mismos.

1.3 Sistemas de Gestión de Ensayos Clínicos

Fueron analizados como parte de la investigación los sistemas de gestión de ensayos clínicos que no han sido desarrollados sobre tecnologías web, los cuales apoyan el manejo electrónico de estudios clínicos en una forma más fácil y rápida. Entre ellos se encuentran el *Entrypoint Plus*, *Oracle Mobile Clinical Research Associate* y *BCX Clinical*.

1.3.1 Sistemas internacionales

Entrypoint Plus (de la compañía Phoenix, situada en los Estados Unidos)

Es un sistema completo desarrollado por Phoenix para la creación, implementación y la administración personalizada de datos electrónicos. Es una aplicación de escritorio que tiene incluido un conjunto de plantillas de CRDs electrónicos con campos para datos demográficos, historia médica, reacciones adversas y exámenes de laboratorios. (8)

Entrypoint tiene muchos atributos avanzados incorporados que hacen la adición de características tales como controles de rango, búsquedas de la tabla, la duplicación y comprobar dígitos, criterios de validación de una sencilla operación de punto de clic. (9)

Este sistema incluye servicios como: línea de ayuda contextual, documentación clara y concisa, incluyendo un tutorial que te ha de diseñar y utilizar las aplicaciones en tan sólo unas horas, el soporte técnico por correo electrónico, fax o teléfono, actualizaciones regulares del producto, las clases y los programas de capacitación, servicios de diseño de aplicaciones. (9)

Oracle Mobile Clinical Research Associate (de la compañía Oracle, situada en los Estados Unidos)

Es una aplicación móvil que ayuda a las organizaciones de investigación médica a acortar los tiempos de los estudios clínicos, gracias a la mejora de la eficiencia y la productividad. La aplicación diseñada específicamente para ser utilizada en trabajos de campo, está integrada con el sistema de gestión de ensayos clínicos y las analíticas de desarrollo clínico de Oracle, para suministrar a los equipos de

Capítulo 1: Fundamentación Teórica de la Investigación

investigación un acceso en tiempo real a la información esencial de los ensayos clínicos, los datos utilizados para la toma de decisiones y completos informes de viaje, tanto si el usuario está conectado al sistema o no. Es posible acceder a todos los indicadores de rendimiento del estudio y ver cómo progresan los resultados, lo que permite planificar visitas más productivas y eficientes a los sitios de estudio. (10)

BCX Clinical (de la compañía Biocapax, situada en España)

Es un software de captura de datos electrónicos (EDC por sus siglas en inglés), que se utiliza para gestionar la etapa de recogida de datos de los pacientes incluidos en un ensayo clínico. Está compuesta por dos aplicaciones de escritorio: BCX Clinical studio, que proporciona las herramientas necesarias para el diseño en versión digital, que tendrá el cuaderno de recogida de datos y BCX Clinical core, proporciona la interfaz de recogida de los datos de los individuos del estudio, facilitando además las herramientas necesarias para la validación y exportación de los datos almacenados. (11)

Este potente software permite controlar exhaustivamente la etapa de recogida de datos de un ensayo clínico, en cualquiera de las fases clínicas a través de las múltiples funcionalidades que incorpora. Su uso no sólo agiliza el proceso de monitorización de un ensayo clínico, también posibilita que la entidad investigadora esté respaldada durante todo el proceso de medidas de seguridad acordes con la legislación vigente que ofrecen gran fiabilidad a los datos recopilados. Este tipo de software está orientado para su uso por parte de grandes entidades, como hospitales, laboratorios farmacéuticos o centros de investigación, que precisan de una herramienta para diseñar y perfeccionar el cuaderno de recogida de datos de sus ensayos clínicos. (11)

El estudio de los sistemas referenciados ha aportado una visión más clara de la línea a seguir durante el desarrollo del presente trabajo. Se han observado características comunes en estos sistemas que se pueden tener en cuenta durante el desarrollo de la solución, entre las que se destacan: la captura de datos electrónicos, la gestión de usuarios y roles como: Monitor, Investigador, Data Manager y Data entry, el almacenamiento de datos de los pacientes del estudio, así como la creación, modificación y eliminación de estudios.

A pesar de las diversas funcionalidades que presentan, ninguno ofrece una solución para la problemática que se plantea en la presente investigación. Esto se debe a que son sistemas realizados por empresas

bajo licencias no libres, lo cual no cumple con el principio de independencia tecnológica que promueve el país, limitando al país su adquisición y futuras mejoras a las funcionalidades que brindan. Por otra parte son sistemas desarrollados para problemas específicos de una institución por lo que solamente se enfocan en la captura de datos electrónicos, además, no realizan la gestión de notas durante el proceso de monitoreo.

1.3.2 Sistemas nacionales

Clínicas 1.0

En Cuba el CIM conduce más de 60 ensayos clínicos para varias enfermedades, entre las que destaca el cáncer. Dicho centro para la gestión electrónica de ensayos clínicos utiliza el sistema Clínicas 1.0, el cual tiene como propósito acelerar y automatizar la conducción de ensayos clínicos en el Centro. (12)

Clínicas 1.0 es un sistema desarrollado en la Universidad de las Ciencias Informáticas en Cuba, abarca las funcionalidades de Open Clínica y además brinda otras como: la realización de un Cronograma de ejecución general para los ensayos y la generación del cronograma específico para cada paciente, la validación de las variables de los EC, permitiendo la disminución y detección de errores en los datos, el reporte de las trazas de las acciones realizadas en el sistema; así como la restricciones de las direcciones IP desde las cuales se puede acceder al mismo, entre otras. (13)

El sistema Clínicas 1.0 está formado por cuatro módulos:

- **El módulo Administrar Empresa:** permite gestionar los permisos y roles de los usuarios que accederán al sistema, así como la administración de direcciones IP desde las cuales se accederá al sistema.
- **El módulo Gestionar Estudio:** permite la creación de EC, así como la confección de CRD y la creación del Cronograma general de ejecución de cada ensayo clínico. Además permite realizar la validación de las variables de un CRD a través del establecimiento de reglas de validación y derivación; permitiendo la disminución y detección de los errores que pueden cometerse al entrar los datos recogidos de los pacientes en el ensayo.
- **El módulo Enviar Datos:** permite generar el cronograma de ejecución específico para cada paciente dentro de un ensayo, así como el llenado de las hojas CRD, además permite realizar el

monitoreo de los datos de los pacientes de un estudio y la creación de notas o discrepancias asociadas a cada uno de los datos monitoreados.

- **El módulo Extraer Datos:** permite realizar reportes a partir de los datos almacenados por pacientes en el ensayo y realizar un reporte de las trazas de auditoría asociado fundamentalmente a las variables de los modelos y a las acciones en general que realiza determinado usuario sobre los elementos del sistema. Garantiza la salida de información en diversos formatos de uso extendido: SPSS, EXCEL y HTML.

Específicamente el módulo Enviar Datos, que se encarga de la gestión de sujetos y notas en el sistema, constituyó el punto de partida para el entendimiento de la gestión y el monitoreo de datos de ensayos clínicos, favoreciendo la obtención de las funcionalidades correspondientes para la solución propuesta.

El estudio del sistema anteriormente descrito sirvió de base durante la investigación para la familiarización con el proceso de gestión de ensayos clínicos. Por otra parte este módulo cuenta con potentes funcionalidades para la conducción de ensayos clínicos en el país, pero no constituye una posible forma de cumplimiento del objetivo propuesto en la presente investigación. Esto se debe a que es una aplicación web donde el acceso a la misma solo puede ser desde un centro que contenga conexión disponible prácticamente todo el tiempo a la red nacional y al servidor central que radica en el CIM.

1.4 Aplicación Desktop (también llamada de Escritorio)

La realidad de cada empresa es única y por eso muchas veces es necesaria la implementación de un software desarrollado para conseguir respuestas adecuadas a sus necesidades.

Una **aplicación Desktop** es aquella que está instalada en el ordenador del usuario, que es ejecutada directamente por el sistema operativo y cuyo rendimiento depende de diversas configuraciones de hardware como memoria RAM, disco duro, memoria de video, etc. (14)

Ventajas de los sistemas desktop:

- Habitualmente su ejecución no requieren comunicación con el exterior, sino que se realiza de forma local. Esto repercute en mayor velocidad de procesamiento, y por tanto en mayores capacidades a la hora de programar herramientas más complicadas o funcionales.
- Suelen ser más robustas y estables que las aplicaciones Web.

- Rendimiento: el tiempo de respuesta es muy rápido.
- Seguridad: pueden ser muy seguras.

Desventajas de los sistemas desktop:

- Su acceso se limita al ordenador donde están instaladas.
- Son dependientes del sistema operativo que utilice el ordenador y sus capacidades (video, memoria, etc.).
- Requieren instalación personalizada.
- Requieren actualización personalizada.
- Suelen tener requerimientos especiales de software y librerías.

1.5 Metodología, tecnologías y herramientas

A continuación se describen la metodología, tecnologías y herramientas utilizadas en la solución al problema planteado. Las mismas, fueron definidas por el arquitecto del proyecto Ensayos Clínicos y se encuentran en el documento “SES_SICEC_0120_6 ArqSWVtaEntorno_v1.0”.

1.5.1 Metodología de desarrollo

En la actualidad existen diversas metodologías que han surgido producto de necesidades específicas en distintas regiones del mundo, lo que impide visualizar un estándar único que se pueda aplicar a todos los procesos de desarrollo de software. (15)

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de actitud y diferentes tamaños de proyectos. (16)

El **Proceso Unificado de Desarrollo** (RUP por sus siglas en inglés) es un proceso formal, provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por

Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. Es guiado por casos de uso y centrado en la arquitectura, y utiliza lenguaje de modelado unificado (UML) como lenguaje de notación. (17)

Las cuatro fases del ciclo de vida son: inicio o conceptualización, elaboración, construcción y transición; los flujos de trabajo están comprendidos en dos grupos, seis de ingeniería (modelación del negocio, requerimientos, análisis y diseño, implementación, prueba e instalación) y tres de soporte (administración del proyecto, administración de configuración y cambios y ambiente). (17)

Ventajas de usar esta metodología:

- ✓ Evaluación en cada fase que permite cambios de objetivos.
- ✓ Funciona bien en proyectos de innovación.
- ✓ Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- ✓ Seguimiento detallado en cada una de las fases.

En RUP se utilizan los casos de uso no sólo como una herramienta para especificar los requisitos del sistema, sino también como una guía de su diseño, implementación y prueba; es decir los casos de uso constituyen un elemento integrador y una guía del trabajo. Por otra parte como metodología no ágil, es adaptable al contexto y las necesidades de cada organización y asegura la producción de software de calidad, dentro de plazos y presupuestos predecibles.

1.5.2 Lenguaje de programación

Java es un lenguaje de programación de propósito general orientado a objetos desarrollado por Sun Microsystems. También se puede decir que Java es una tecnología que no sólo se reduce al lenguaje sino que además provee de una máquina virtual Java que permite ejecutar código compilado Java, sea cual sea la plataforma que exista por debajo; plataforma tanto hardware, como software (el sistema operativo que soporte ese hardware). El apoyo a esta tecnología viene dado por la gran cantidad de fabricantes que apoyan esta especificación de máquina virtual. (18)

Se caracteriza por ser un lenguaje dinámico, de arquitectura neutral y multiplataforma lo bastante potente para desarrollar aplicaciones en cualquier ámbito. Todo esto unido a la portabilidad, potencia y seguridad

que presenta lo ha hecho convertirse en un lenguaje muy utilizado por programadores de todo el mundo. (19)

1.5.3 Entorno de Desarrollo Integrado (IDE): Netbeans versión 7.4

Netbeans es un IDE² gratuito de código abierto para Oracle Solaris, Oracle Linux, otras distribuciones de Linux, Mac y Windows que permite a los desarrolladores crear rápidamente aplicaciones Web, empresariales, de escritorio y móviles utilizando principalmente Java, así como PHP³ y C/C++⁴. Ofrece mejoras considerables para crear interfaces gráficas de usuario Swing y admite herramientas para la depuración visual de interfaces de usuario (IU). (20)

Este IDE es un producto sin restricciones de uso, completo y modular. Facilita la construcción de aplicaciones completas para el cliente, además de permitir la creación de ventanas, menús, barras de herramientas y acciones de una forma más amena.

1.5.4 Librería Swing

Swing es un conjunto de librerías que forman parte de la JFC⁵ en la plataforma java, las cuales incluye componentes para interfaz gráfica de usuarios tales como cajas de texto, botones, desplegados y tablas. La interfaz de usuario es la parte del programa que permite a éste interactuar con el usuario, dichas interfaces pueden adoptar muchas formas, que van desde la simple línea de comandos hasta las interfaces gráficas que proporcionan las aplicaciones más modernas. Es más dependiente de la plataforma Java y no del sistema operativo, facilitándole con ello a los programadores mayor variedad de componentes gráficos a la hora de diseñar las interfaces. Sus componentes se identifican porque pertenecen al paquete *javax.swing* (21)

² Entorno de Desarrollo Integrado

³ Hypertext Preprocessor

⁴ Lenguaje de Programación.

⁵ Java Foundation Classes

1.5.5 Máquina Virtual de Java: Open-jdk-8.0

Una **máquina virtual Java** (en inglés Java Virtual Machine, JVM) es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el bytecode Java), el cual es generado por el compilador del lenguaje Java. Su misión principal es la de garantizar la portabilidad de las aplicaciones Java. Define esencialmente un ordenador abstracto y especifica las instrucciones (bytecode) que este ordenador puede ejecutar. (22)

Las tareas principales de la JVM son las siguientes:

- Reservar espacio en memoria para los objetos creados
- Liberar la memoria no usada (garbage collection).
- Asignar variables a registros y pilas.
- Llamar al sistema huésped para ciertas funciones, como los accesos a los dispositivos.
- Vigilar el cumplimiento de las normas de seguridad de las aplicaciones Java.

1.5.6 Herramienta de Mapeo objeto-relacional (ORM): Hibernate versión 3.6.10

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML)⁶. (23)

Proporciona un lenguaje para el manejo de consultas a la base de datos y cuenta con el Hibernate Query Language (HQL, por sus siglas en inglés) que es un lenguaje de consulta similar al SQL. Además brinda, la posibilidad de generar bases de datos en cualquiera de los entornos soportados: PostgreSQL, Oracle, MySQL, entre otros. El API⁷ de Hibernate es una arquitectura de dos capas (Capa de persistencia y Capa de Negocio). (23)

⁶ Lenguaje de Marcado Extensible.

⁷ Application Programming Interface

1.5.7 Herramienta CASE: Visual Paradigm para UML versión 8.0

Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos (24). Visual Paradigm también ofrece:

- ✓ Navegación intuitiva entre la escritura del código y su visualización.
- ✓ Potente generador de informes en formato PDF⁸/HTML⁹.
- ✓ Documentación automática.
- ✓ Ambiente visualmente superior de modelado.
- ✓ Sincronización de código fuente en tiempo real.

Visual Paradigm como herramienta CASE profesional soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Logra integrarse con varias herramientas entre las que se encuentran: Eclipse, Netbeans, Microsoft Visual Studio y LDeveloper. Es una herramienta multiplataforma que permite su uso en cualquier sistema operativo. Además, permite dibujar todos los diagramas de clases, código inverso y generar códigos desde diagramas. (24)

1.5.8 Lenguaje Unificado de Modelado: UML versión 2.0

El Lenguaje Unificado de Modelado (Unified Modeling Language, UML) se define como un lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. (25)

Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. UML incluye conceptos semánticos, notación, y principios generales. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo, pero no especifica en sí mismo qué metodología o proceso usar. (25)

⁸ Formato de Documento Portátil

⁹ Lenguaje de Marcado de Hipertexto.

1.5.9 Herramienta de Administración de la Base de Datos: PgAdmin3 versión 1.16.1

PgAdmin es una herramienta gráfica de código abierto para administrar y desarrollar bases de datos en PostgreSQL. Está diseñado para cumplir las necesidades de todos los usuarios, desde escribir consultas SQL¹⁰ simples hasta el desarrollo de bases de datos complejas. (26)

La interfaz gráfica soporta todas las características de PostgreSQL facilitando su administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor y un agente para lanzar scripts programados. Este administrador de base de datos se encuentra disponible en más de una docena de lenguajes y para varios sistemas operativos como: Windows, Linux, entre otros. Además de estar desarrollado por una comunidad de expertos de PostgreSQL en todo el mundo está disponible en más de una docena de idiomas. (26)

1.5.10 Sistema Gestor de Base de Datos: PostgreSQL versión 9.2.4

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD¹¹ y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. (27)

Esta nueva versión incorpora avanzadas características en materia de seguridad, soporte de aplicaciones, seguimiento y control, rendimiento y almacenamiento de data especializada. Los usuarios se sienten especialmente interesados por las nuevas funcionalidades de replicación y hot standby, que acelerarán su adopción para alojamiento en la nube y aplicaciones escalables. (28)

Las definiciones y conceptos abordados en este capítulo permiten tener un mayor entendimiento del proceso de conducción de ensayos clínicos. Los sistemas investigados a nivel internacional no ofrecen una solución a la problemática planteada en la presente investigación, los mismos son sistemas

¹⁰ Lenguaje de Consulta Estructurado.

¹¹ Berkeley Software Distribution.

Capítulo 1: Fundamentación Teórica de la Investigación

propietarios que fueron diseñados para resolver problemas específicos de una institución; sin embargo la investigación sobre estos sistemas aportó un acercamiento a las tendencias mundiales en cuanto a lo que de gestión de ensayos clínicos se conoce y realiza. Por otra parte el estudio del módulo Enviar Datos del sistema Clínicas 1.0 sentó las bases para el desarrollo de la solución de la presente investigación. El estudio y caracterización de las tecnologías y herramientas definidas permitió adquirir los conocimientos necesarios para el desarrollo de la solución.

Capítulo 2: Características del Sistema

El presente capítulo tiene como objetivo describir las características del sistema a construir. Contiene además la identificación de los requisitos funcionales y no funcionales de la solución a desarrollar, se describen los actores y el diagrama de caso de uso del sistema, así como sus respectivas descripciones de los casos de uso considerados más significativos.

2.1 Objeto de automatización

Actualmente ante la inexistencia de conexión de red, algunos de los centros que colaboran con el CIM en la gestión y monitoreo de datos de ensayos clínicos desarrollan el proceso de la manera habitual, donde toda la información se encuentra en papel y la transmisión de información de los hospitales a los centros promotores y viceversa, se hace generalmente vía correo electrónico o vía telefónica.

Un complicado proceso de monitoreo de los datos clínicos se realiza visitando periódicamente cada uno de los sitios participantes, incluyendo las provincias orientales, por un profesional capacitado (Monitor) con la necesaria competencia clínica, elegido para controlar la calidad y uniformidad de los datos recogidos. Durante las visitas se recoge toda la información proveniente de cada ensayo, ya sean la fecha de inclusión y de inicio de tratamiento para cada sujeto, hospitales participantes, las causas de no inclusión en el estudio y todos las características propias de cada ensayo.

Finalmente se preparan bases de datos, en versión electrónica, a partir de los Cuadernos de Recogida de Datos llenados en los hospitales; con el objetivo de facilitar el análisis final de los resultados de cada ensayo. La generación de estas bases de datos resulta compleja para el CIM, dado la gran variabilidad de la información recogida en los hospitales y por la dificultad intrínseca del proceso de transcribir de manera fiable, información de un formato impreso a un formato electrónico.

Para un mejor desarrollo de las actividades antes expuestas se hace necesaria la automatización del proceso de gestión y monitoreo facilitando el trabajo a los especialistas cubanos y de esta forma contribuir a la creación de nuevos fármacos.

2.2 Modelado del dominio

El modelo del dominio muestra las clases conceptuales significativas en un dominio del problema; es el artefacto más importante que se crea durante el análisis orientado a objetos (29). Es un diccionario visual

de términos importantes del dominio y se representa en el Lenguaje de Modelado Unificado con un diagrama de clases.

2.2.1 Descripción del flujo del diagrama del modelo de dominio

En el Centro de Inmunología Molecular se realizan diversos estudios o ensayos clínicos para la creación de fármacos. En estos estudios participan diferentes especialistas que pueden ocupar el cargo de Monitor, Coordinador de Investigación Clínica o Investigador Principal. Durante la gestión y el monitoreo de los datos asociados a los estudios se generan tres tipos de notas (notas de sitio, generales o de monitoreo). El especialista que ocupa el cargo de monitor es el responsable de crear las notas generales y de monitoreo, además de monitorear las hojas CRD que han sido creadas en el sistema durante un estudio. El especialista que ocupa el cargo de investigador principal es el responsable de firmar las hojas CRD una vez que hayan sido monitoreadas. El especialista que ocupa el cargo de coordinador es el responsable de gestionar los pacientes o sujetos que son incluidos en un estudio, así como las hojas CRD y las notas de sitios. La siguiente figura muestra de forma gráfica la descripción del modelo de dominio.

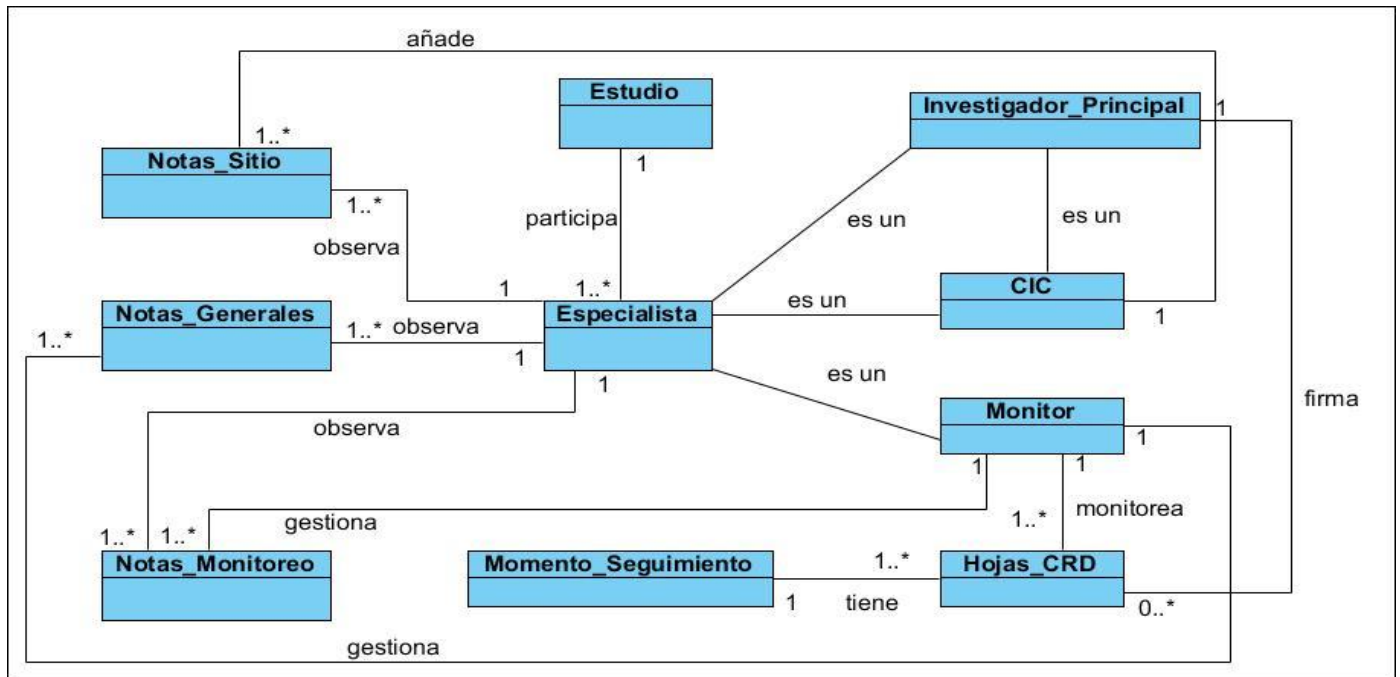


Ilustración 1. Modelo de dominio.

2.2.2 Definición de los conceptos del modelo del dominio

CIC: es una representación del rol Coordinador de Investigación Clínica. Es el responsable de gestionar pacientes, los datos de las hojas CRD asociadas a los momentos de seguimiento definidos para los pacientes bajo su responsabilidad y las notas del sitio.

Especialista: es una generalización de los actores Investigador Principal, Coordinador y Monitor. Es el encargado de responder notas generales y visualizar todas las notas del sistema.

Estudio: representa los estudios que son llevados a cabo sobre sujetos bajo investigación clínica, en el cual participan los especialistas.

Hojas CRD: representa el conjunto de datos pertenecientes a un examen de cada paciente que debe ser llenado en algún momento durante la conducción del estudio.

Investigador Principal: es una representación del rol Investigador principal. Es el responsable de toda la información clínica recogida en su sitio de investigación y además es el encargado en el sistema de insertar y retirar su firma de las hojas de CRD completadas.

Momento de seguimiento: representa la asociación de una o varias hojas CRD a un paciente con un plazo de tiempo para su llenado.

Monitor: es una representación del rol Monitor. Es el responsable de monitorear los datos de los sujetos, las Hojas CRD, además de gestionar las notas generales y las notas de monitoreo.

Notas de monitoreo: representa una aclaración que se le agrega a un dato específico durante el proceso de monitoreo de los datos por el monitor.

Notas de sitio: representa una aclaración que se le agrega a un dato específico en el llenado de una hoja CRD por el CIC.

Notas generales: representa una aclaración que no está asociada directamente a una variable y se puede levantar en cualquier momento en la ejecución de un ensayo.

2.3 Propuesta del Sistema

Dado que se pretende informatizar el proceso de conducción de los ensayos clínicos en el CIM y los hospitales del país, es necesario desarrollar un sistema offline con las funcionalidades del sistema Clínica

1.0, donde sea más rápido el tiempo de respuesta a los usuarios y la disponibilidad solo dependa de la PC donde esté instalada la aplicación y del servidor de la Base de Datos.

Se propone desarrollar funcionalidades sobre tecnologías de escritorio, que permitan la gestión y monitoreo de datos en los diferentes hospitales del país que no tienen acceso al sistema Clínicas 1.0. Se utilizarán las tablas de la base de datos del sistema Clínicas 1.0 relacionadas con la gestión y el monitoreo, en aras de lograr un diseño compatible con este. Los datos generados se sincronizarán con el sistema web a través de la herramienta SIGEC Sincronizador evitando la duplicidad de la información gestionada por los especialistas.

Para acceder al sistema el usuario deberá introducir su nombre de usuario y contraseña de acuerdo a su rol, los roles de los usuarios a interactuar con el sistema son: Investigador Principal, Coordinador de Investigaciones Clínicas y Monitor. Las funcionalidades encargadas de la gestión y el monitoreo de datos, son las mismas funcionalidades implementadas en el módulo Enviar Datos del sistema Clínicas 1.0. El especialista que interactúe con el sistema será capaz de gestionar los datos de los sujetos asociados a los ensayos a través de la recogida de estos en hojas CRD, además, de monitorear los datos correspondientes a dichas hojas. Por otra parte podrá realizar acciones como crear, visualizar listados, ver detalles y responder las diferentes notas de discrepancias, ya sean de monitoreo, de sitio o generales.

2.3.1 Requisitos funcionales

El propósito de la definición de requisitos es especificar las condiciones o capacidades que el sistema debe cumplir y las restricciones bajo las cuales debe operar, logrando un mayor entendimiento entre el equipo de desarrollo y el cliente. Los requerimientos del software se dividen en requerimientos funcionales (RF) y requerimientos no funcionales (RNF).

A continuación se muestran los requisitos funcionales y una breve descripción de cada uno.

Requisitos	Requisitos
RF1. Visualizar hojas CRD	Se muestra la hoja CRD asociada a un Momento de Seguimiento con sus respectivos datos, el nombre, el estado, la fecha de llenado, la cantidad de Notas de Monitoreo con estado "Nueva" o "Actualizada" y las acciones

Capítulo 2: Características del Sistema

	asociadas a las mismas.
RF2. Monitorear hoja de CRD	El monitor revisa los datos introducidos en una hoja CRD asociada a un momento de seguimiento definido para un sujeto y revisa el estado de las notas.
RF3. Modificar estado de cada momento de seguimiento	Se modifica el estado a cada momento de seguimiento, según el estado de las hojas CRD asociadas a él.
RF4. Modificar datos a cada hoja de CRD	El coordinador introduce los datos a modificar de una hoja CRD asociada a un Momento de Seguimiento definido para un sujeto y posteriormente el sistema actualiza el valor de los datos que ya tenía almacenado de esa hoja CRD.
RF5. Crear nota de monitoreo	El monitor crea una nota en el monitoreo de los datos de una hoja CRD en caso de ser necesaria alguna aclaración o algún comentario sobre el dato.
RF6. Crear notas de sitio	El coordinador crea una nota de sitio en el llenado de un dato en caso de ser necesarios alguna aclaración o algún comentario sobre el dato.
RF7. Crear nota general	El monitor crea una nota del estudio en caso de ser necesarios alguna aclaración o algún comentario sobre algo del estudio activo.
RF8. Mostrar notas generales	Se muestran en la interfaz de gestionar notas las notas generales creadas en el sistema.
RF9. Mostrar notas de sitio	Se muestran en la interfaz de gestionar notas las notas de sitio creadas en el sistema.
RF10. Mostrar notas de monitoreo	Se muestran en la interfaz de gestionar notas la notas de monitoreo creadas en el sistema.

Capítulo 2: Características del Sistema

RF11. Visualizar listado de notas de sitio por estudio	Se muestran todas las notas de sitio asociadas a un estudio determinado que han sido creadas en el sistema.
RF12. Visualizar listado de notas de monitoreo por estudio	Se muestran todas las notas de monitoreo asociadas a un estudio determinado que han sido creadas en el sistema.
RF13. Visualizar listado de notas generales por estudio	Se muestran todas las notas generales asociadas a un estudio determinado que han sido creadas en el sistema.
RF14. Responder nota general	Se responde una nota general y se actualiza el estado de la nota.
RF15. Responder nota de monitoreo	Se responde una nota de monitoreo y se actualiza el estado de la nota.
RF16. Firmar estado hoja CRD	Permite que el Investigador principal tenga la opción de firmar el estado de monitoreo de una hoja CRD si y solo si la hoja CRD tiene el estado de monitoreo completado.
RF17. Mostrar detalles de la nota de monitoreo	Permite mostrar los detalles específicos de las Notas de Monitoreo que han sido creadas en el sistema.
RF18. Mostrar detalles de la nota de sitio	Permite mostrar los detalles específicos de las Notas de Sitio que han sido creadas en el sistema.
RF19. Mostrar detalles de la nota general	Permite mostrar los detalles específicos de las Notas Generales que han sido creadas en el sistema.
RF20. Visualizar listado de notas de monitoreo por	Se muestran todas las notas de monitoreo asociadas a un sujeto

sujeto	determinado que han sido creadas en un estudio.
RF21. Visualizar listado de notas de sitio por sujeto	Se muestran todas las notas de sitio asociadas a un sujeto determinado que han sido creadas en un estudio.

Tabla 1. Requisitos funcionales del sistema

2.3.2 Requisitos no funcionales

Los requisitos no funcionales (RNF) son propiedades o cualidades que el sistema debe presentar como: el rendimiento, mantenimiento, seguridad, disponibilidad del equipo, entre otros. Estos forman una parte significativa de la especificación de requisitos siendo importantes para que clientes y usuarios puedan valorar las características no funcionales del producto. A continuación se muestran los requisitos no funcionales identificados en este sistema, los cuales están descritos en el documento “SES_SIGEC_010113_ERS_v1.0”.

1. RNF de Usabilidad

- La aplicación tendrá un ambiente sencillo de manera que pueda ser usado por cualquier persona que posea conocimientos básicos en el manejo de una computadora o sistema informático.

2. RNF de Seguridad

- El acceso a cualquier manipulación del sistema, tanto entrada como análisis de datos estará sometido a un proceso de autenticación del usuario donde será especificado el usuario y contraseña.
- Cada usuario tendrá asignado uno o varios roles en el sistema. Cada rol definido tendrá niveles de acceso al Software.

3. RNF de Disponibilidad

- Se garantizará el funcionamiento de la aplicación durante las 24 horas del día y los siete días de la semana.

4. RNF de Interfaz

- La entrada de datos incorrecta será detectada claramente e informada al usuario. Todos los textos y mensajes en pantalla aparecerán en idioma español.
- Las interfaces no estarán cargadas de imágenes y contendrán bien estructurados los datos, permitiendo la interpretación correcta de la información a los usuarios.
- Se hará uso de simbología mediante íconos que contendrán funcionalidades específicas.

5. RNF de Soporte

- Las necesidades de los usuarios deben quedar satisfechas después de la puesta en explotación del sistema. Una vez aprobada la aplicación y lista para comenzar su ejecución, se instalará en los hospitales donde se realice la conducción de ensayos clínicos.

6. RNF de Software

- El sistema debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma Java (Java Virtual Machine y PostgreSQL).

7. RNF de Hardware

- Las estaciones de trabajo deben tener como mínimo 256 Mb de memoria RAM o superior y un microprocesador a 2.0 GHz o superior.

8. RNF de Restricciones de Diseño

- La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en la implementación del motor de persistencia Hibernate.

9. RNF de Portabilidad

- El sistema debe permitir ejecutarse en cualquiera de los sistemas operativos más usados en la actualidad. Destacándose las distribuciones de GNU Linux: Ubuntu, Nova, etc.

2.3.3 Actores del Sistema

ACTORES DEL SISTEMA

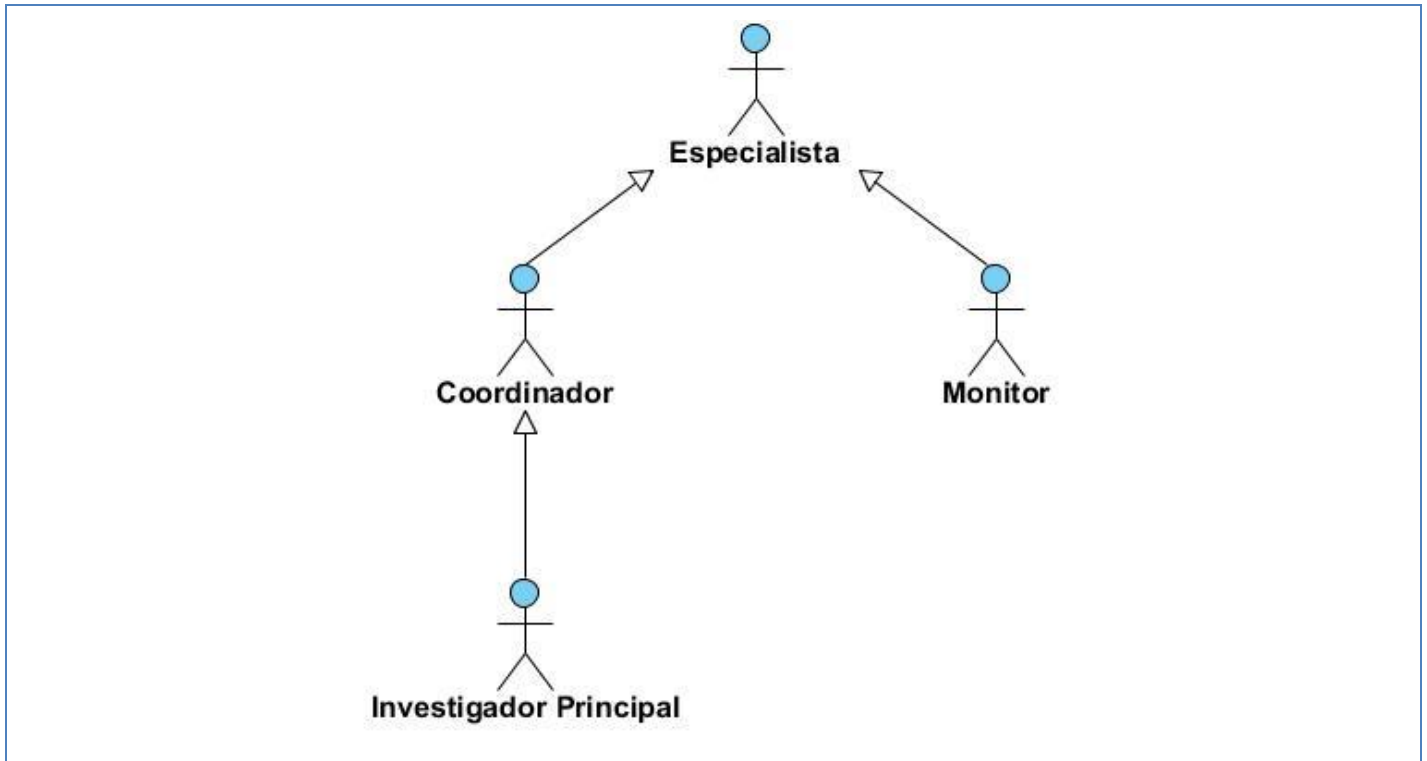


Ilustración 2. Actores del Sistema

A continuación se describen los actores del sistema con sus respectivas descripciones para un mejor entendimiento de las responsabilidades que les son autorizadas.

Actor	Descripción
Monitor	Es el responsable de Monitorear toda la información referente a los estados del sujeto. Además gestiona las notas generales y de monitoreo y observa las notas de sitio.
Investigador Principal	Es el máximo responsable de toda la información clínica recogida en su sitio de investigación y además es el encargado en el sistema de insertar y retirar su firma de las hojas de CRD completadas. Puede modificar las hojas, gestionar notas de sitio y responder notas generales y de monitoreo.
Coordinador de la	Es el responsable de gestionar sujetos por estudios, insertar los datos de las

Investigación Clínica (CIC)	hojas de CRD asociadas a los Momentos de Seguimiento definidos para los sujetos bajo su responsabilidad, gestionar de Sitio y responder notas generales y de monitoreo.
Especialista	Es el encargado de responder notas generales y visualizar todas las notas del sistema. Un especialista es una generalización de los actores Investigador Principal, Coordinador y Monitor.

Tabla 2. Descripción de los Actores del Sistema

2.3.4 Diagrama de Casos de Uso del Sistema

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente. (30)

En la Ilustración 3 se muestra el Diagrama de Caso de Uso del Sistema, donde se muestran los casos de usos en los que se agruparon los requisitos funcionales.

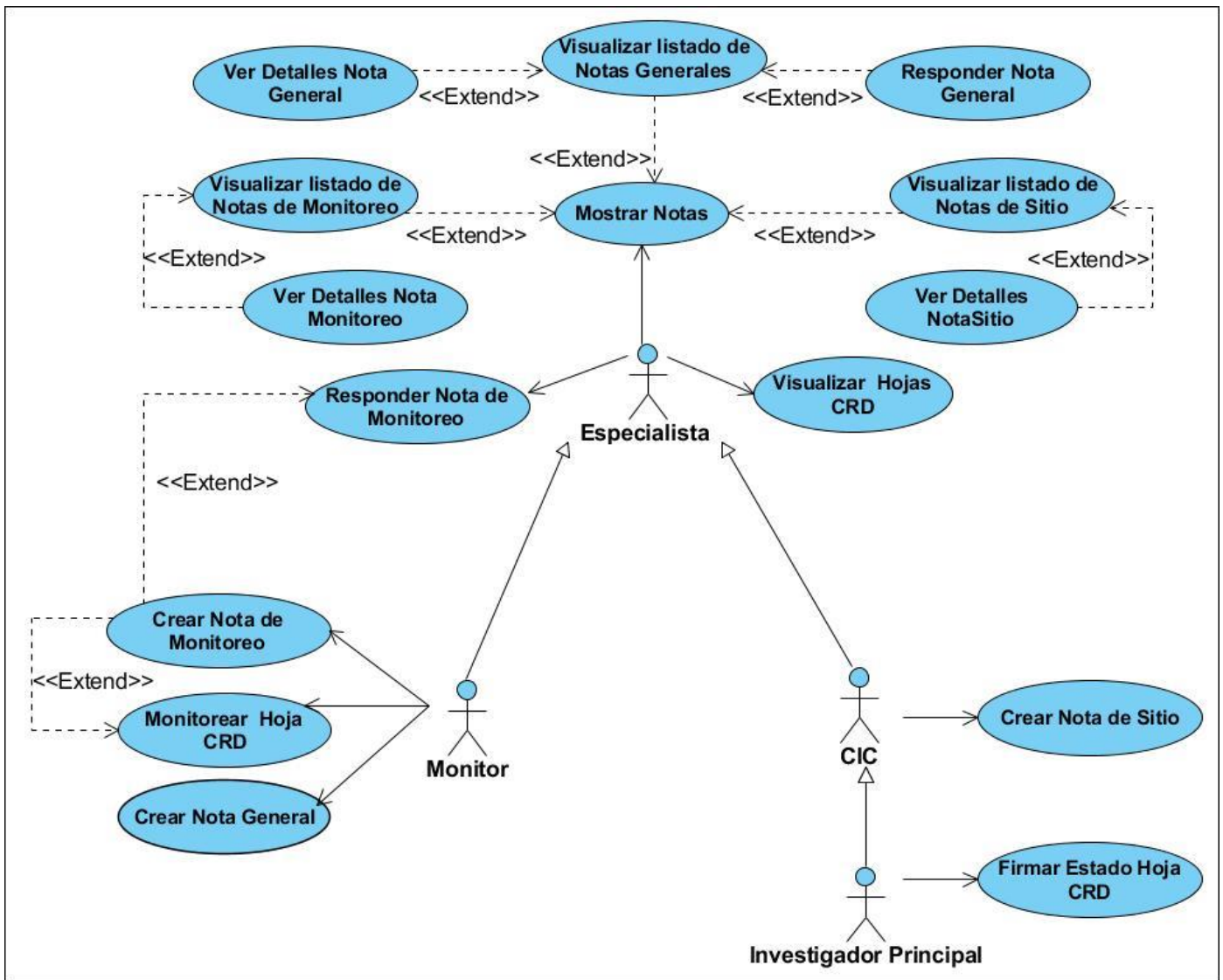


Ilustración 3. Diagrama de Casos de Usos del Sistema

2.3.5 Descripción de los Casos de Uso del Sistema

A continuación se describen los casos de usos Mostrar Notas, Crear Nota General y Responder Nota General, detallando la secuencia de eventos que el actor lleva a cabo para completar un proceso determinado a través de la aplicación.

2.3.5.1 CU: Mostrar Notas

Caso de uso	
CU-7	Mostrar Notas
Propósito:	Observar las notas y discrepancias que existen en el sistema.
Actores:	Especialista (inicia)
Resumen:	El caso de uso inicia cuando el actor decide ver las notas y discrepancias en un estudio. El sistema muestra un listado de notas dividido en 3 secciones: Notas de Monitoreo, Notas de Sitio y Notas Generales. El caso de uso termina cuando el especialista indica otra opción en el sistema.
Referencias	RF9, RF10, RF11, RF15, RF16, RF17
Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Ver Notas y Discrepancias” en un estudio.	<p>1.1 Muestra un listado de notas dividido en 3 secciones: Notas de Monitoreo, Notas de Sitio y Notas Generales.</p> <p>Para las Notas de Monitoreo muestra:</p> <ul style="list-style-type: none"> • Id del paciente • Momentos de seguimientos asociados. • Cantidad de notas de monitoreo • Fecha • Descripción <p>Para las Notas de Sitio muestra:</p> <ul style="list-style-type: none"> • Id del paciente

	<ul style="list-style-type: none"> • Momentos de seguimientos asociados. • Cantidad de notas de monitoreo • Fecha • Descripción <p>Para las Notas Generales muestra:</p> <ul style="list-style-type: none"> • ID de la Nota • Fecha de Creación • Fecha de Actualización • Descripción • Detalles de la Nota • Número de Notas • Estado de Resolución • Tipo de Nota • Autor <p>Permite realizar la acción:</p> <ul style="list-style-type: none"> • Ver listado de las notas por sujeto para las notas de monitoreo y las de sitio. Ver Sección 1: “Listar notas”. <p>Termina el caso de uso.</p>
Flujos alternativos	
Sección1: Listar Notas	
Acción del actor	Respuesta del sistema
1.	<p>1.1 Muestra una ventana con los datos de las notas asociadas a un sujeto:</p> <ul style="list-style-type: none"> • Id del sujeto • Fecha de creación • Fecha de última actualización

	<ul style="list-style-type: none"> • Número de notas • Fecha del momento de seguimiento • Descripción • Detalles de la nota • Momento de seguimiento
2. Selecciona "Cancelar".	2.1 Vuelve a la interfaz anterior.
Punto de extensión	
<ul style="list-style-type: none"> • Muestra el listado de notas de monitoreo de un estudio. Ver caso de uso: Visualizar listado Notas Monitoreo. • Muestra el listado de notas de sitio de un estudio. Ver caso de uso: Visualizar listado Notas de Sitio. • Muestra el listado de notas generales de un estudio. Ver caso de uso: Visualizar listado Notas Generales. 	

Tabla 3. Caso de uso Mostrar Notas

2.3.5.2 CU: Crear Nota General

Caso de uso	
CU-6	Crear Nota General
Propósito:	Crear notas generales en el sistema.
Actores:	Monitor (inicia)
Resumen:	El caso de uso inicia cuando el Monitor selecciona Gestionar Nota y escoge la opción Crear Nota General. El sistema muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando el Monitor decide guardar los cambios.

Referencias	RF-7
Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Crear Nota General”.	1.1 Muestra una interfaz con los campos siguientes: <ul style="list-style-type: none"> • Tipo <ul style="list-style-type: none"> ○ Anotación ○ Consulta • Detalles de la Nota • Descripción • Estado de Resolución
2. Edita los datos: <ul style="list-style-type: none"> • Tipo • Detalles de la Nota • Descripción • Estado de Resolución 	
3. Indica enviar la nota.	
4. Selecciona enviar.	4.1 Salva la nota en un registro de notas.
5. Selecciona cerrar.	5.1 Vuelve a la interfaz anterior. Termina el caso de uso.
Punto de extensión	
No existe.	

Tabla 4. Caso de uso Crear Nota General

2.3.5.3 CU: Responder Nota General

Caso de uso	
CU-11	Responder Nota General
Propósito:	Responder las notas generales que han sido creadas en un estudio.
Actores:	Especialista (inicia)
Resumen:	El caso de uso inicia cuando el Especialista, en la interfaz Principal para gestionar notas y discrepancias, en el listado de notas generales, selecciona la opción Responder nota general. El sistema muestra la interfaz correspondiente y ejecuta las acciones necesarias. El caso de uso finaliza cuando el Especialista guarda los cambios.
Referencias	RF14
Acción del actor	Respuesta del sistema
3 El caso de uso inicia cuando el actor, en la interfaz de gestionar notas, escoge la opción Ver detalles.	1.1 Muestra una interfaz con los datos de la nota: <ul style="list-style-type: none">• ID de la nota• Tipo• Estado de Resolución• Nº de Notas• Fecha de Creación• Autor• Estatus• Detalles de la Nota

	Descripción
2. Si la nota en cuestión tiene como Estado de Resolución: Nueva o Actualiza.”.	2.1 Muestra la opción Responder Nota General.
3. Escoge la opción Resolver Nota General como respuesta a la nota padre.	3.1 Muestra una interfaz que permite llenar los siguientes campos: <ul style="list-style-type: none"> • Destalles de la Nota • Descripción • Estado de Resolución <ul style="list-style-type: none"> ○ Actualizado ○ Resuelto ○ Cerrado
4. Introduce los datos de los campos: <ul style="list-style-type: none"> • Detalles de la Nota • Descripción 	
5. Selecciona la opción Enviar.	5.1 Salva la nota en un registro de notas.
6. Selecciona la opción Cerrar.	6.1 Vuelve a la interfaz anterior. El caso de uso termina.
Punto de extensión	
<ul style="list-style-type: none"> • Responde las notas generales. Ver caso de uso: Ver Detalles Nota General. 	

Tabla 5. Caso Uso Responder Nota General

Capítulo 2: Características del Sistema

Con el desarrollo de este capítulo se describe el funcionamiento del sistema y las características necesarias para su correcto funcionamiento a través del modelo de dominio, de la descripción de los requisitos funcionales, los no funcionales, del diagrama de casos de uso y de la descripción de cada caso de uso del sistema. Se identificaron 21 requisitos funcionales que se agrupan en 15 casos de uso que guiarán el proceso de desarrollo de la solución.

Capítulo 3: Diseño del Sistema

Con el desarrollo del presente capítulo se efectúa el análisis y diseño de la propuesta de solución para el componente. Se describe la arquitectura y los patrones a utilizar para el diseño e implementación. Además, se realizan los diagramas de clases, los diagramas de colaboración y el diagrama de despliegue según los casos de usos más relevantes definidos en el capítulo anterior.

3.1 Descripción de la Arquitectura

El diseño arquitectónico representa la estructura de datos y los componentes del programa necesarios para construir un sistema computacional. Asume el estilo arquitectónico que tomará el sistema, la estructura y las propiedades de los componentes que constituyen el sistema y las interrelaciones entre todos los componentes arquitectónicos de un sistema. (31)

3.1.1 Arquitectura en 3 capas

Esta arquitectura soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite la partición de un problema complejo en una secuencia de pasos incrementales. Proporciona amplia reutilización. Este tipo de arquitectura separa los datos de una aplicación, la interfaz de usuario y la lógica en tres componentes o capas distintas. La arquitectura está basada en un modelo de tres capas: Capa de datos, Capa de lógica de negocio y Capa de presentación, que soportan todas las funcionalidades necesarias. (32)

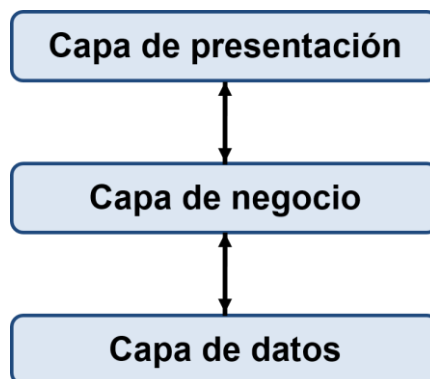


Ilustración 4. Arquitectura en 3 capas

Descripción de las capas:

- **Capa de presentación:** Está formada por formularios tales como: **CreateGeneralNote.form**, **DetailGeneralNote.form** y **ListGeneralNote.form** y sus controles visuales (textBox, comboBox, etc.) desarrollados con el uso de la librería Swing y permiten la interacción del sistema con el usuario.
- **Capa de negocio:** Las clases controladoras **DiscrepancyNotegeneralControl**, **DiscrepancyNoteMonitoryControl** y **SiscrepancyNoteSitioControl** procesan la información necesaria referente a las notas, contienen la validación de los datos e interactúan con los datos considerando si se realiza algún cambio en la capa de datos. Esta capa se comunica con la de presentación, para recibir las solicitudes realizadas por los usuarios y generar los resultados; interactúa además con la capa de acceso a datos, para realizar solicitudes que le permitan controlar la información almacenada, tanto para obtenerla como para registrarla o modificarla.
- **Capa de datos:** Este es la representación de la información con la cual el sistema opera. Contiene el dominio de la aplicación con sus clases, los métodos get y set y los objetos de acceso a datos (DAO). Estos objetos de acceso a datos serán manejados a través de interfaces que exponen las operaciones de persistencia definidas para cada uno de los **DAOs** e implementadas utilizando **Hibernate**, que es el framework de mapeo a objeto-relacional utilizado.

Se escogió la Arquitectura 3 capas para el desarrollo del componente, ya que con la puesta en práctica se reducen las dependencias existentes entre las clases, permitiendo la modificación de una capa sin afectar a las otras. La arquitectura permitirá una mejor organización durante el desarrollo así como brindar un mejor soporte al sistema. Esta arquitectura se diferencia de Modelo Vista Controlador (MVC) en que la relación entre los componentes MVC es triangular, donde existe relación entre la vista y el controlador, entre el controlador y el modelo y entre la vista y el modelo; mientras que la relación en 3 capas es lineal es decir que no hay una comunicación directa entre las diferentes capas, pues todas las comunicaciones deben pasar por una capa intermedia.

3.1.2 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular Dichos patrones pretenden formalizar un vocabulario común entre diseñadores así como también estandarizar el modo en que se realiza el diseño

(33). Entre los más utilizados se encuentran los General Responsibility Assignment Software Patterns (GRAPS por sus siglas en inglés) y el Data Access Object (DAO por sus siglas en inglés).

Patrón DAO

El patrón Objeto de Acceso a Datos es un patrón de diseño considerado una buena práctica. Básicamente este patrón consiste en centralizar los procesos de acceso a la base de datos evitando inconsistencias y posibles problemáticas, con este patrón se independiza la lógica de negocio de la lógica de acceso a datos obteniendo mayor organización y flexibilidad.

DAO¹² puede usarse en Java para aislar a una aplicación de la tecnología de persistencia Java subyacente (API de Persistencia Java), la cual podría ser JDO¹³, JDBC¹⁴, Hibernate o cualquier otra tecnología de persistencia. El uso de este patrón queda expuesto en los diagramas de clases del diseño, cada diagrama incluye en el paquete Modelo un grupo de clases DAO, con el objetivo de independizar la lógica del negocio de la lógica de acceso a datos obteniendo mayor organización y flexibilidad en el sistema, un ejemplo constituye el siguiente donde la clase `DiscrepancyNoteDaoHibernate` maneja la información de las notas siendo el puente de interacción entre la base de datos y la aplicación.



Ilustración 5. Uso del Patrón DAO

Patrones GRASP

Otros de los patrones usados son los *GRASP*¹⁵ (por sus siglas en inglés), que no son más que patrones generales de software para asignación de responsabilidades. Aunque se considera que más que patrones

¹² Data Access Object

¹³ Java Data Object

¹⁴ Java Database Connectivity

¹⁵ *General Responsibility Assignment Software Patterns*

propriadamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software. (34)

Alta Cohesión: caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Las clases con baja cohesión a menudo han asumido responsabilidades que deberían haber delegado a otros objetos o hacen un trabajo excesivo. La solución es asignar una responsabilidad de modo que la cohesión siga siendo alta y se incremente la claridad y facilidad de la comprensión del diseño. El uso del patrón se evidencia en la clase DiscrepancyNote la cual tiene responsabilidades estrechamente relacionadas con las clases DiscrepancyNoteType y ResolutionStatus.

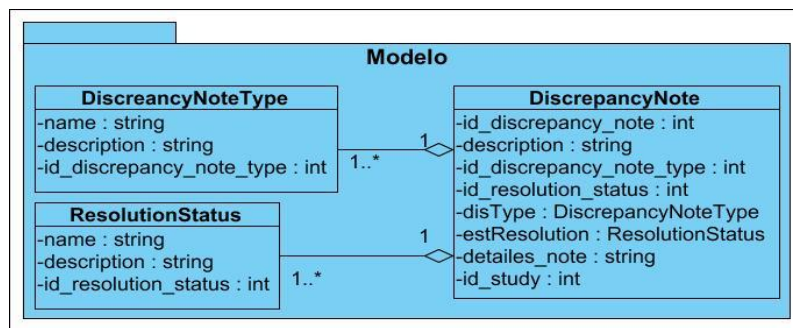


Ilustración 6. Uso del patrón Alta cohesión

Bajo acoplamiento: soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. Es la idea de tener las clases lo menos ligadas entre sí que se pueda, de tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. El uso del patrón se evidencia en la clase MonitoryNoteControl, la cual puede sufrir cambios sin que se vean afectadas el resto de las clases del sistema.

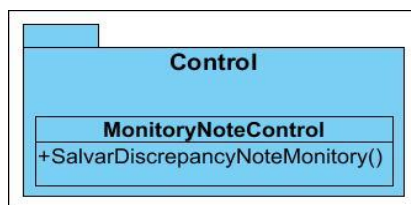


Ilustración 7. Uso del patrón Bajo acoplamiento

Patrón Experto: es el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. El uso del patrón se evidencia en la clase DiscrepancyNote, la cual contiene toda la información referente a las notas del sistema.

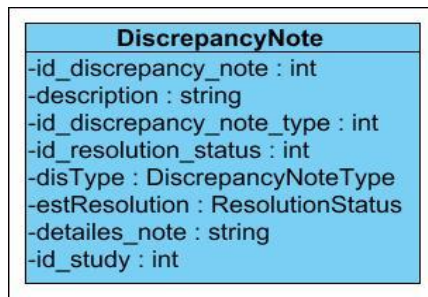


Ilustración 8. Uso del patrón Experto

Patrón Creador: es un patrón que guía la asignación de responsabilidades relacionadas con la creación de objetos. La intención básica del patrón es encontrar un creador que necesite conectarse al objeto creado en alguna situación, logrando así mayor reutilización. El uso del patrón se evidencia entre las clases DiscrepancyNoteDAOHibernate y DiscrepancyNote contenidas en el paquete modelo, donde las clases DAO son las encargadas de la creación de objetos.

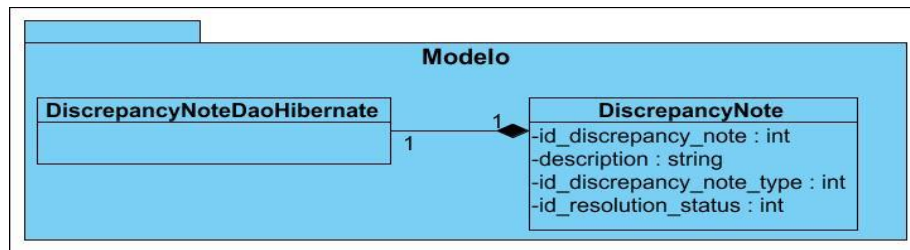


Ilustración 9. Uso del patrón Creador

Patrón Controlador: es un patrón que sirve como intermediario entre una determina interfaz y el algoritmo que la implementa. Se encarga de asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Las clases Control son clases controladoras que se encargan de gestionar todos los eventos del sistema. El uso del patrón se evidencia en todas las clases Control que conforman los diagramas de clases del diseño.

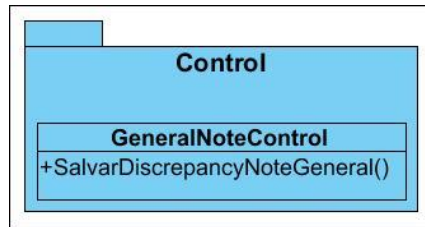


Ilustración 10. Uso del patrón Controlador

3.2 Modelo de Diseño

3.2.1 Diagramas de clases del diseño

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos (34). En el presente documento se realizaron los diagramas de clases del diseño por cada caso de uso, estos permiten que se describa detalladamente y de forma gráfica las especificaciones de las clases del software. A continuación se muestran los diagramas de clases del diseño confeccionados haciendo referencia al patrón arquitectónico expuesto anteriormente: Modelo Vista Controlador (MVC). Se muestran entonces los elementos que componen cada uno de estos paquetes y posteriormente se ilustran con los diagramas de clases del diseño realizados.

Paquetes:

➤ Modelo

<<Nombre Clase DAO Hibernate>> Clases de acceso a datos, que encapsulan la fuente de datos y ocultan la forma de acceder a ellas.

<<Nombre Clase>> Clases entidades portadoras de los datos del sistema, cada objeto de ellas constituye una tupla de una tabla en la base de datos.

➤ Vista

<<View Nombre Clase>> Es la interfaz a través de la cual interactúan los usuarios con el sistema. A través de ella se envían datos del cliente al servidor.

➤ Controlador

<<Nombre Clase Control>> Clases controladoras que se encargan de la gestión de los datos del modelo y de enviárselos a la vista, de la misma forma, recibe los datos de la vista y los envía a el modelo para almacenarlos.

3.2.1.1 Diagrama de clases del diseño CU “Mostrar Notas”

A continuación se muestra el diagrama el diagrama de clases del diseño de la funcionalidad Mostrar Notas donde la clase ViewNotas muestra los datos necesarios referentes a las notas como respuesta, sus atributos son los elementos de entrada del formulario (input boxes, text áreas, check boxes). Las clases controladoras almacenan todos los métodos referentes a las notas, siendo las encargadas gestionar la visualización de respuesta del sistema. Por otra parte la clase DiscrepancyNoteDAOHibernate permite el acceso a los datos almacenados en la base de datos y las clases restantes tienen como propósito proveer el mapeo con la base de datos, las mismas forman parte de las tablas del modelo relacional.

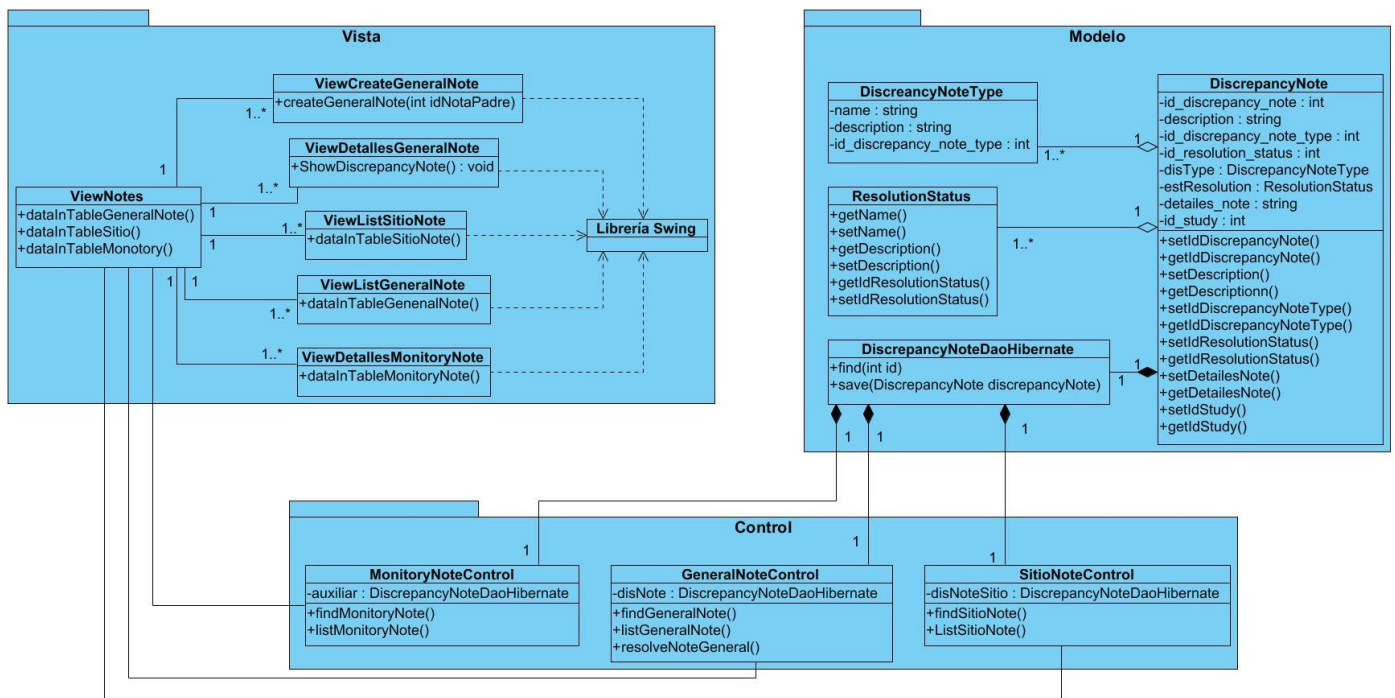


Ilustración 11. Diagrama de clases del diseño Mostrar Notas

Descripción de las clases

Caso Uso: Mostrar Notas	
Nombre: MonitoryNoteControl	
Tipo: Controladora (Se encarga del manejo de los datos para posteriormente mostrar los detalles de la nota de monitoreo en cuestión).	
Atributo	Tipo
auxiliar	DiscrepancyNoteDAOHibernate
Responsabilidades	
Nombre	Descripción
listMonitoryNote ()	Este método se encarga de gestionar los datos para la conformación del listado de las notas de monitoreo.
findMonitoryNote()	Este método se encarga de gestionar los datos para mostrar los detalles de las notas de monitoreo.
Nombre: GeneralNoteControl	
Tipo: Controladora (Se encarga del manejo de los datos para posteriormente mostrar los detalles de la nota general en cuestión).	
Atributo	Tipo
disNote	DiscrepancyNoteDAOHibernate
Responsabilidades	
Nombre	Descripción

listGeneralNote ()	Este método se encarga de gestionar los datos para la conformación del listado de las notas generales.
resolveGeneralNote()	Método que se encarga de gestionar los datos para responder las notas generales.
findGeneralNote()	Este método se encarga de gestionar los datos para mostrar los detalles de las notas generales.
Nombre: SitioNoteControl	
Tipo: Controladora (Se encarga del manejo de los datos para posteriormente mostrar los detalles de la nota de sitio en cuestión).	
Atributo	Tipo
disNoteSitio	DiscrepancyNoteDAOHibernate
Responsabilidades	
Nombre	Descripción
ListSitioNote ()	Este método se encarga de gestionar los datos para la conformación del listado de las notas de sitio.
FindSitioNote()	Este método se encarga de gestionar los datos para mostrar los detalles de las notas de sitio.

Tabla 6. Descripción de las clases: Mostrar Notas

3.2.1.2 Diagrama de clases del diseño CU “Crear Nota General”

En el diagrama de clases del diseño de la funcionalidad Crear Nota General la clase GeneralNoteControl almacena todos los métodos referentes a las notas generales, siendo la encargada de gestionar la

visualización de respuesta del sistema. La clase ViewCreateGeneralNote contiene los elementos de entrada para la inserción de todos los datos referentes a las notas generales. Por otra parte la clase DiscrepancyNoteDAOHibernate permite el acceso a los datos almacenados en la base de datos y las clases restantes tienen como propósito proveer el mapeo con la base de datos, las mismas forman parte de las tablas del modelo relacional.

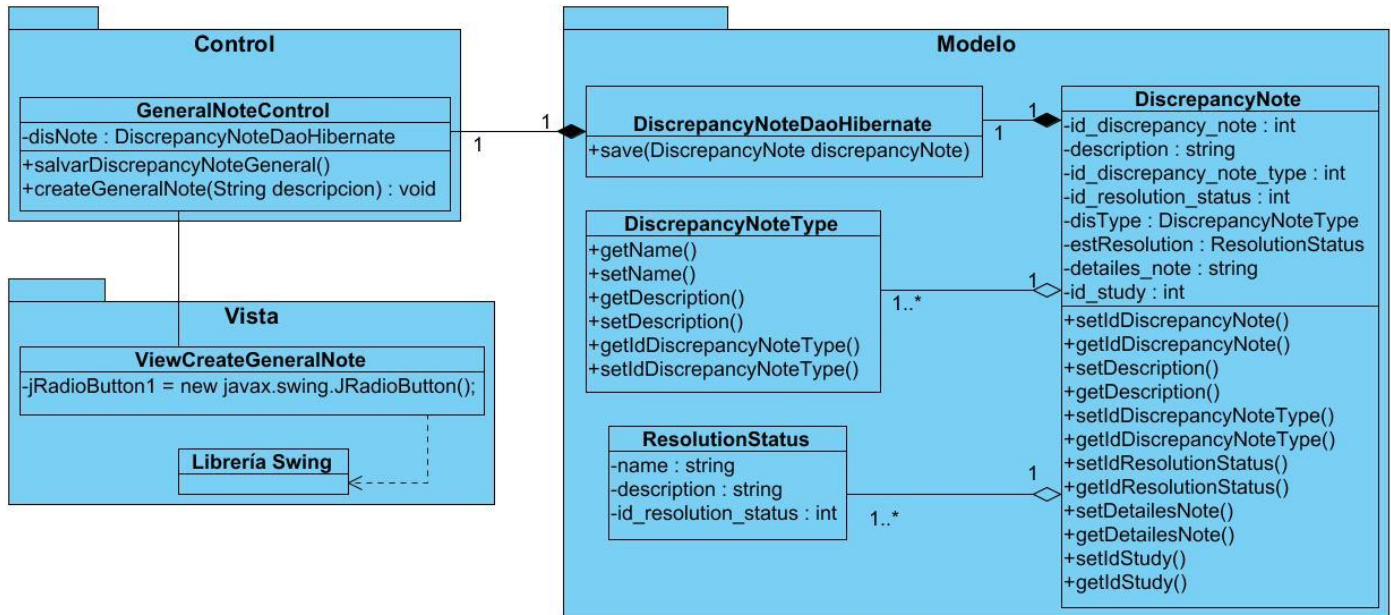


Ilustración 12. Diagrama de clases del diseño Crear Nota General

Descripción de las clases

Caso Uso: Crear Nota General	
Nombre: GeneralNoteControl	
Tipo: Controladora (Se encarga del manejo de los datos para posteriormente mostrar los detalles de la nota general en cuestión, para que pueda ser creada).	
Atributo	Tipo
disNote	DiscrepancyNoteDAOHibernate

Responsabilidades	
Nombre	Descripción
salvarDiscrepancyNoteGeneral()	Este método se encarga de guardar las notas creadas en un registro de notas.
createGeneralNote(String descripción)	Este método se encarga de crear las notas generales.

Tabla 7. Descripción de las clases: Crear Nota General

3.2.1.3 Diagrama de clases del diseño del CU “Responder Nota General”

En el diagrama de clases del diseño de la funcionalidad Responder Nota General la clase ViewResolveGeneralNote contiene los elementos de entrada para la inserción de todos los datos referentes a las notas generales así como la clase ViewDetallesGeneralNote muestra todos los datos necesarios referentes a las notas generales. La clase GeneralNoteControl almacena todos los métodos referentes a las notas generales, siendo la encargada de gestionar la visualización de respuesta del sistema. Por otra parte la clase DiscrepancyNoteDAOHibernate permite el acceso a los datos almacenados en la base de datos y las clases restantes tienen como propósito proveer el mapeo con la base de datos, las mismas forman parte de las tablas del modelo relacional.

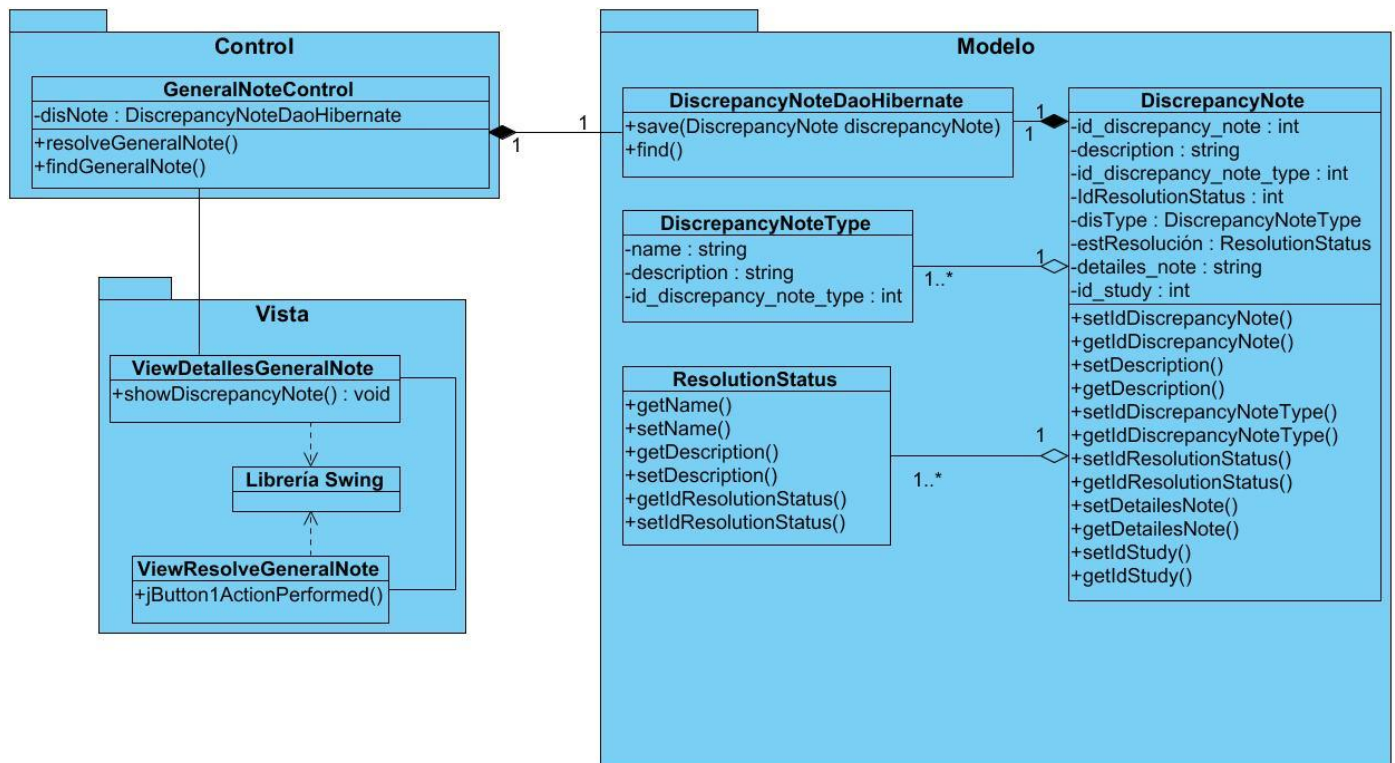


Ilustración 13. Diagrama de clases del diseño Responder Nota general

Descripción de las clases

Caso Uso: Responder Nota General	
Nombre: GeneralNoteControl	
Tipo: Controladora (Se encarga del manejo de los datos para posteriormente mostrar los detalles de la nota general en cuestión, para que pueda ser resuelta).	
Atributo	Tipo
disNote	DiscrepancyNoteDAOHibernate
Responsabilidad	

Nombre	Descripción
resolveGeneralNote()	Método que se encarga de gestionar los datos para responder las notas generales.
findGeneralNote()	Este método se encarga de gestionar los datos para mostrar los detalles de las notas generales.

Tabla 8. Descripción de las clases: Responder Nota General

3.2.2 Diagramas de colaboración

Un diagrama de colaboración es una forma de representar la interacción entre objetos, enlaza las asociaciones del diagrama de clases mediante el paso de mensajes de un objeto a otro (35). A continuación se muestran ejemplos de los diagramas de colaboración del sistema.




Representación	Nombre	Descripción
	Entidad	Modela la información que posee una vida larga y que a menudo es persistente, modelando la información y comportamiento asociado de algún fenómeno o concepto. Se derivan de una clase de entidad del negocio o del dominio.
	Control	Representa una coordinación secuencial, transacciones y control de otros objetos. Se usa para encapsular el control de un caso de uso en concreto y para representar derivaciones y cálculos complejos.
	Interfaz	Modifica la interacción entre el sistema y sus actores, lo que implica recibir y representar informaciones y peticiones de los usuarios y sistemas externos. Representa la abstracción de ventanas, formularios, paneles, interfaz de comunicación, interfaz de impresión, sensores, terminales, aplicaciones, entre otros.

Tabla 9. Descripción de estereotipos

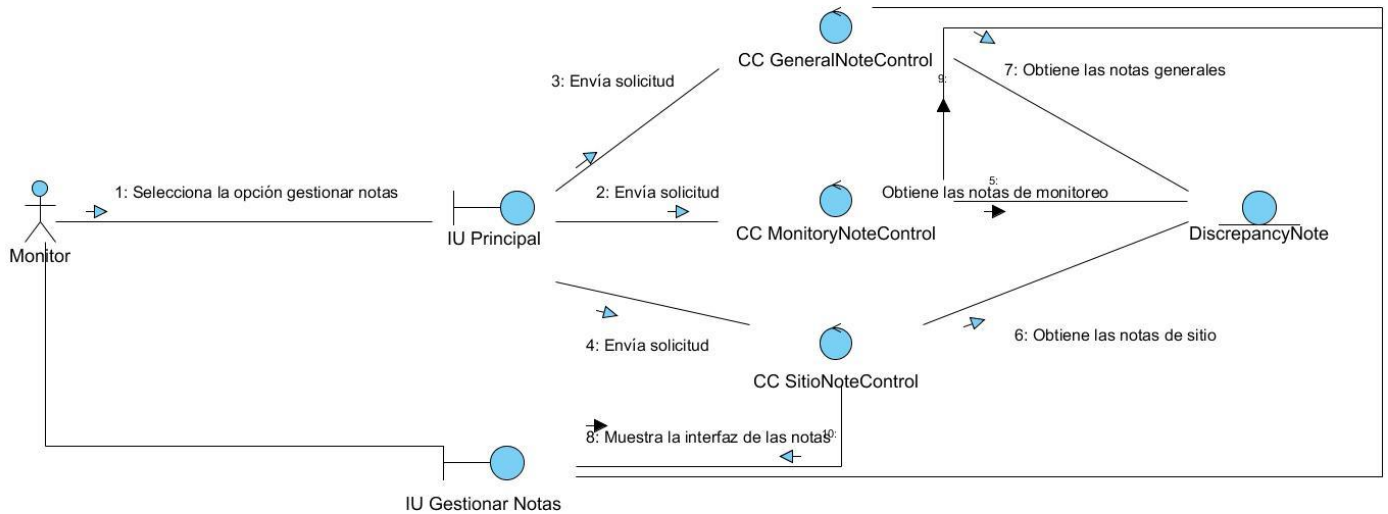


Ilustración 14. Diagrama de colaboración Mostrar Notas

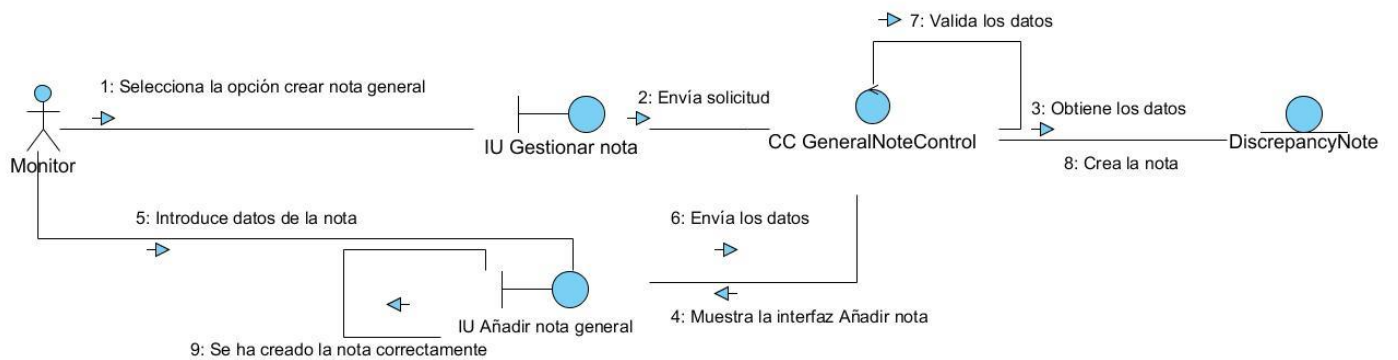


Ilustración 15. Diagrama de colaboración Crear Nota General

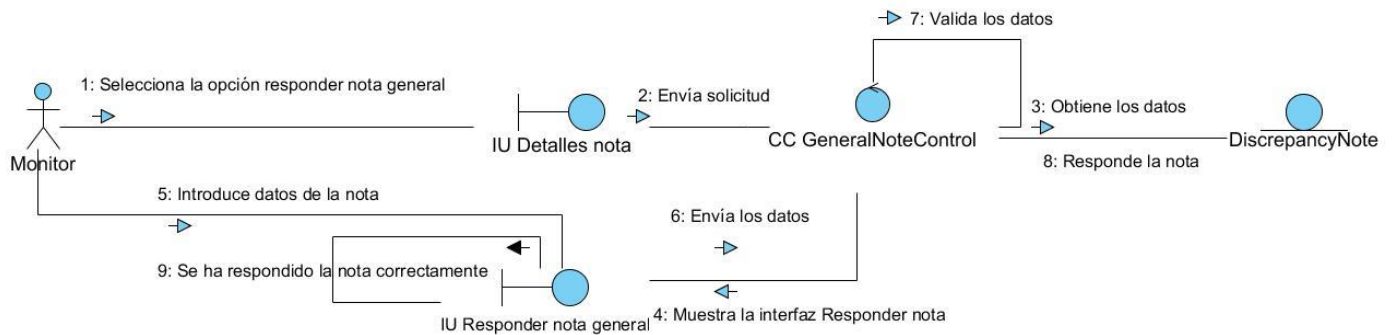


Ilustración 16. Diagrama de colaboración Responder Nota General

En este capítulo se realizó el diseño de la propuesta de solución para lograr el desarrollo de la aplicación. La caracterización de la arquitectura a utilizar durante el desarrollo de la solución permitió establecer la estructura y funcionamiento de las partes del sistema. Se identificaron las clases necesarias para el correcto funcionamiento del sistema, se modelaron además los diagramas de colaboración que facilitan a los desarrolladores actuales o futuros el entendimiento de las funcionalidades requeridas por el cliente. La utilización de los patrones GRASP y DAO incorporó buenas prácticas al resultado de la investigación.

Capítulo 4: Implementación del Sistema

El presente capítulo tiene como objetivo describir todo lo relacionado con la implementación del sistema de acuerdo a la arquitectura propuesta por el diseño. Se muestran las tablas correspondientes al modelo de datos y la representación física del modelo de despliegue del sistema. Por otro lado se expone el diagrama de componentes estableciendo las dependencias existentes y la representación física de la estructura interna del sistema.

4.1 Modelo de datos

Al diseñar una base de datos y gestionar su información, se plasma una parte del mundo real en las tablas, registros y campos ubicados en el ordenador; creando un modelo semejante a la realidad. El principal paso para crear las tablas de la base de datos es diseñar un modelo de datos.

Un Modelo de Datos no es más que un conjunto de conceptos que definen la estructura de una base de datos, ya sean los datos, las relaciones entre estos y las restricciones que rigen dichos datos, sirviendo de representación física de las tablas en una base de datos determinada. (36)

A continuación se muestra el modelo de datos para las funcionalidades de la gestión y el monitoreo de datos, contando con 16 tablas, cada una de estas tablas presentan su nombre, llave primaria (PK de sus siglas en inglés primary key), tipo de dato de la llave primaria, los atributos y el tipo de dato por cada atributo, quedando evidenciado las relaciones entre las tablas. El modelo de datos generado es una parte de la representación de la base de datos del sistema web Clínicas 1.0. Las tablas utilizadas son las que interactúan en la gestión y el monitoreo de datos con el objetivo de obtener un componente compatible con la versión web.

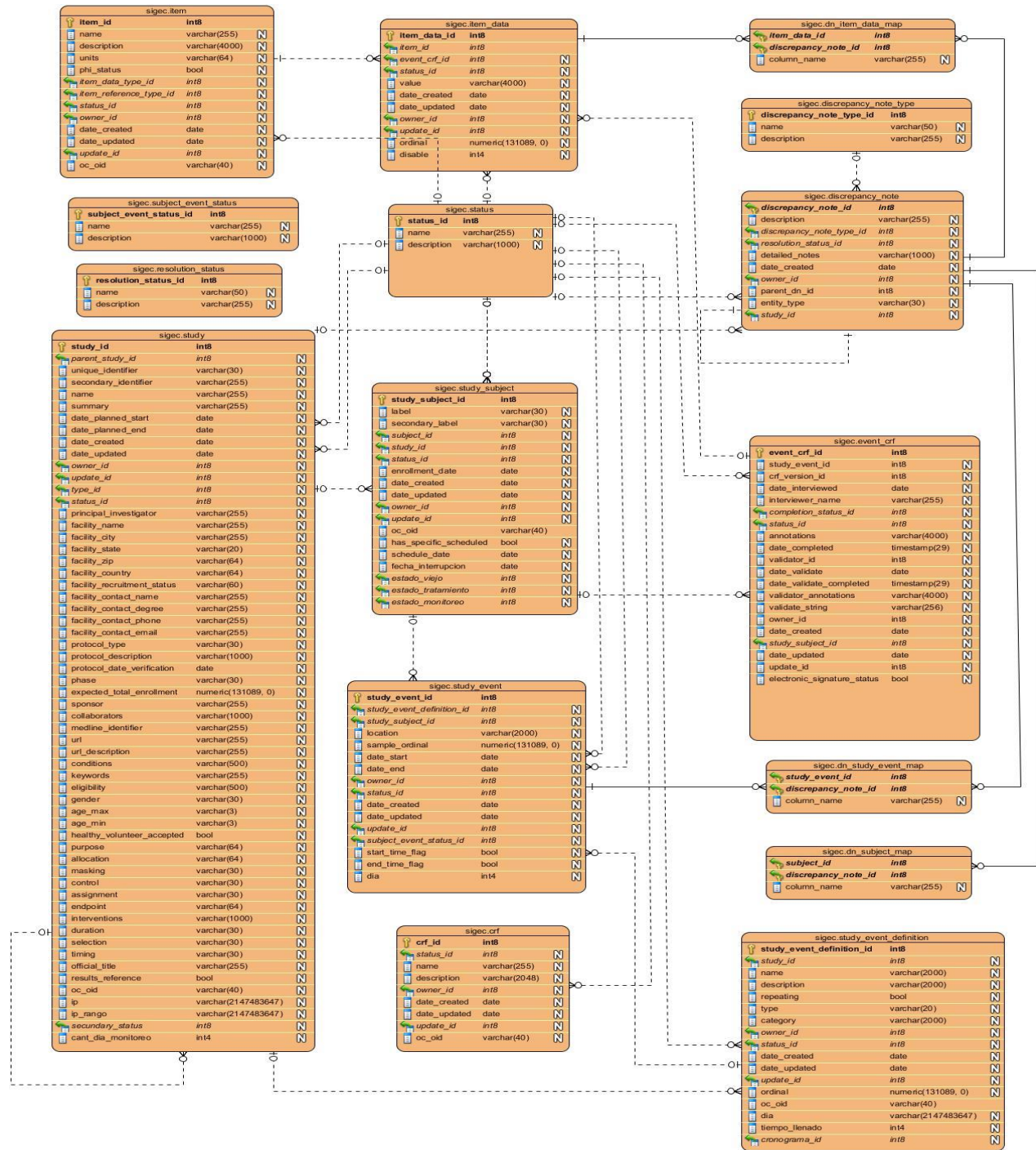


Ilustración 17. Modelo de datos

Descripción de las tablas del Modelo de Datos

A continuación se muestra la descripción de tres de las entidades que conforman el modelo de datos del sistema a desarrollar con sus respectivos atributos, tipo de datos y la descripción de cada atributo.

Nombre: discrepancy_note		
Descripción: Entidad que contiene las “notas de monitoreo” existentes.		
Atributo	Tipo	Descripción
discrepancy_note_id	integer	Representa el identificador de las notas.
description	varchar	Representa una descripción de una nota realizada por estudio.
discrepancy_note_type_id	numeric	Representa el tipo de nota.
resolution_status_id	numeric	Representa el estado de resolución de una nota.
owner_id	numeric	Representa el identificador del paciente asociado a la hoja CRD.
detailed_notes	varchar	Representa los detalles de una nota.
study_id	integer	Representa un estudio realizado.
date_created	date	Representa la fecha de creación de una nota.
parent_dn_id	numeric	Representa el padre de un sujeto.
entity_type	varchar	Representa el tipo de entidad.

Tabla 10. Descripción de la entidad Nota de discrepancias

Nombre: discrepancy_note_type

Descripción: Entidad que con tienes los tipos de notas ya sea “monitoreo”, “sitio” o “generales”.

Atributo	Tipo	Descripción
discrepancy_note_type_id	integer	Representa el identificador de los tipos de notas.
name	varchar	Representa el nombre de la nota.
descripción	varchar	Representa la descripción de la nota.

Tabla 11. Descripción de la entidad Tipo de notas de discrepancia

Nombre: subject_event_status

Descripción: Entidad que contiene los diferentes estados que pueden asumir los momentos de seguimiento.

Atributo	Tipo	Descripción
subject_event_status_id	integer	Representa el identificador del estado del sujeto.
name	varchar	Representa el nombre del estado de los sujetos.
descripción	varchar	Representa la descripción de los estados de un sujeto.

Tabla 12. Descripción de la entidad Estado de momentos de seguimiento

Nombre: resolution_status

Descripción: Entidad que contiene el estado de resolución de las Notas.

Atributo	Tipo	Descripción
resolution_status_id	integer	Representa el identificador del estado de resolución de una nota.

name	varchar	Representa el nombre del estado resolución.
descripción	varchar	Representa la descripción del estado.

Tabla 13. Descripción de la entidad Estado de resolución

4.2 Implementación

4.2.1 Diagrama de Despliegue

El despliegue es la etapa del desarrollo que describe la configuración del sistema para su ejecución en un ambiente del mundo real. El Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar detalladamente la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Además, muestra la configuración de nodos que participan en la ejecución y de los componentes que residen en ellos. (37)

Para la estructura de la distribución física se empleó un nodo PC_Cliente que representa la estación de trabajo de los usuarios en los centros sin acceso al sistema Clínicas 1.0 donde será desplegado el sistema. La estación de trabajo se comunicará mediante el Protocolo de Control de Transmisión/Protocolo de Internet (Transmission Control Protocol/Internet Protocol – TCP/IP por sus siglas en inglés) con el servidor de bases de datos. A continuación se presenta el diagrama de despliegue de la aplicación.

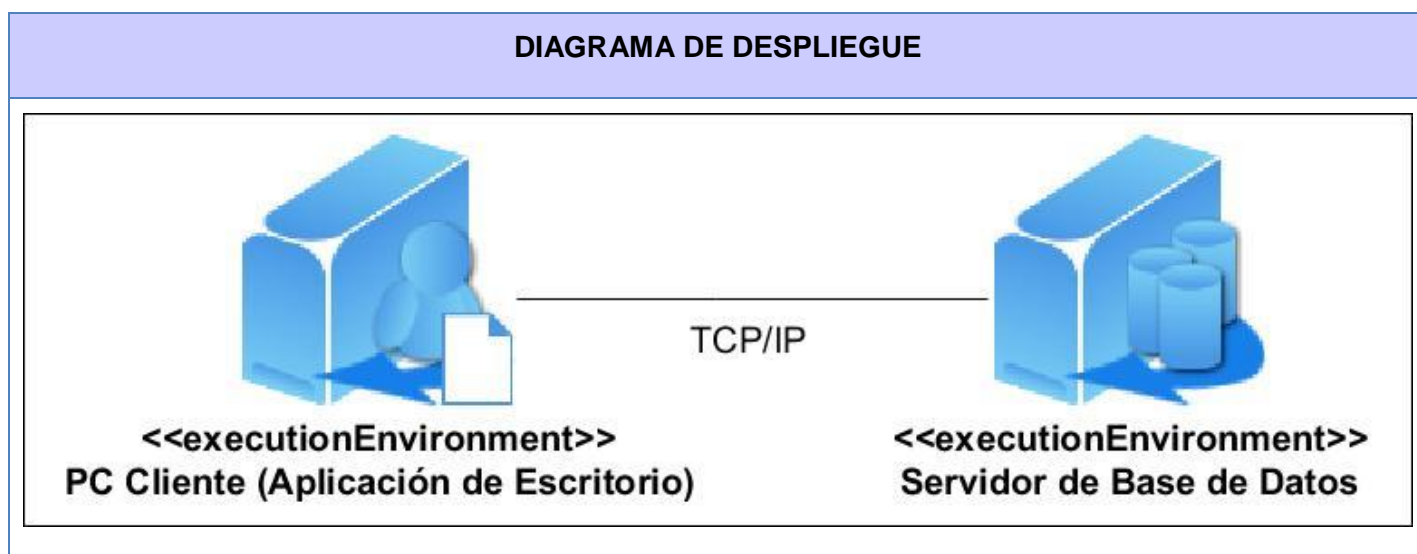


Ilustración 18. Diagrama de despliegue

4.2.2 Diagrama de Componentes

Un componente es una parte física de un sistema (módulo, base de datos, programa ejecutable, etc.). Los diagramas de componentes permiten visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. (38)

A continuación se muestra el diagrama de componente realizado para dar paso a la implementación del sistema. Los componentes se pueden agrupar en 3 paquetes principales, siendo estos Vista, Control y Modelo, en correspondencia con el patrón de arquitectura utilizado Modelo-Vista-Controlador.

En el paquete **Vista**: pueden encontrarse paquetes de componentes formados por archivos .form.

En el paquete **Controlador**: pueden encontrarse paquetes de componentes formados por archivos .java que no son más que las controladoras dentro del sistema.

En el paquete **Modelo**: se encuentran los componentes formados por archivos .java que se encargan de la abstracción del sistema con la BD llamados "X DAO Hibernate" y componentes formados por archivos .java que no son más que las entidades dentro del sistema.

En el paquete **Acceso DAO**: el componente DAO.java contiene las clases AbstractDAO DataAccessLayerException e HibernateFactory, clases necesarias para la búsqueda y edición de las entidades de la base de datos.

En el paquete **Hibernate**: se encuentran los componentes formados por archivos .java y .xml, que no son más que las clases mapeadas de la base de datos.

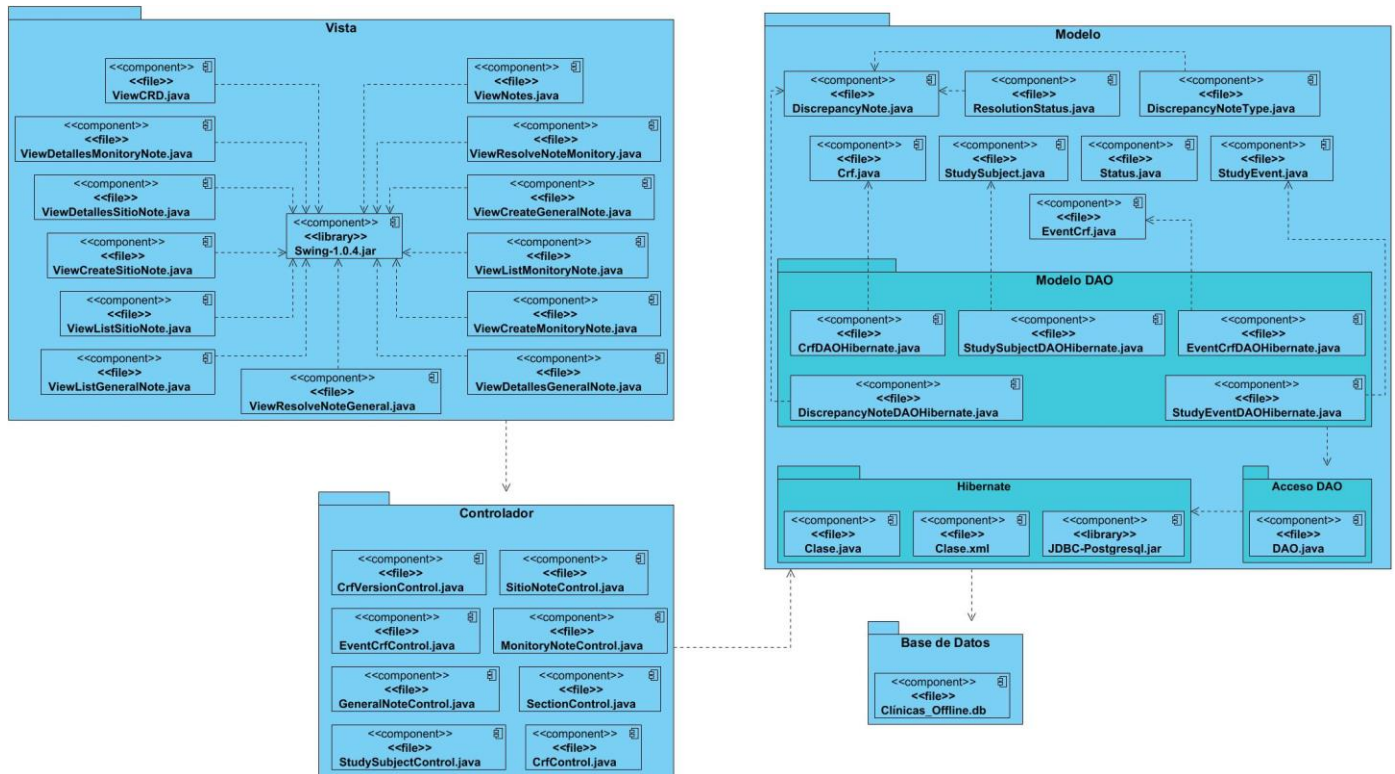


Ilustración 19. Diagrama de Componentes

4.2.3 Estilos y estándares de codificación

Los estándares de codificación son reglas que se deben seguir en el desarrollo de software con el objetivo de posibilitar el entendimiento del código fuente generada, a otros programadores del mismo grupo de trabajo (39). Los estándares de codificación facilitan en gran medida el mantenimiento posterior de los sistemas que hayan sido creados basándose en ellos.

A continuación se muestran los estándares de codificación más relevantes utilizados en la implementación de la propuesta de solución.

1. Los comentarios multilíneas se escriben comenzando con los caracteres “/*” y terminando con “*/”, los comentarios de una sola línea comienzan con los caracteres “//”.

➤ Ejemplo de un comentario multilíneas:

/* Esto es un comentario multilíneas

```
Esto es un comentario multilíneas
```

```
Esto es un comentario multilíneas */
```

➤ **Ejemplo de un comentario de una sola línea:**

```
// Esto es un comentario de una sola línea
```

➤ **No se debe usar espacio en blanco:**

- ✓ Antes y después de un corchete
- ✓ Antes y después de un paréntesis
- ✓ Antes de un punto y coma

2. Las líneas de código que debido a su tamaño ocupan más del espacio visible que ofrece el NetBeans para escribir código (se refiere al ancho de la ventana maximizada), deben ser divididas en varias, preferiblemente después de una coma o un operador.

➤ **Ejemplo:**

```
DiscrepancyNoteTypeDaoHibernate auxiliar = new
```

```
DiscrepancyNoteTypeDaoHibernate ();
```

3. Los nombres de las clases utilizan el estándar UpperCamelCase y los métodos, variables, objetos y parámetros utilizan en sus nombres el estándar CamelCase.

➤ **Ejemplo de declaración de una clase con la notación UpperCamelCase en su nombre:**

```
“public class GeneralNoteControl () {...}”.
```

➤ **Ejemplo de variables recibidas por parámetro con la notación CamelCase:**

```
(int idSubject, String momento) {...}”.
```

4. El nombre de las tablas y de los campos contenidos en ellas deben escribirse en singular y con letra minúscula, comenzando por el prefijo “sigec_”. En caso de ser un nombre compuesto se utilizará para separarlas underscore.

5. Todos los atributos de las clases del modelo están encapsulados. Son privados y el acceso a ellos se hace por medio de los métodos `getAtributo` y/o `setAtributo`.
6. Las variables deben ser explícitas, aunque se pueden usar abreviaturas siempre y cuando no violen este principio.

➤ **Ejemplo de una declaración de variable:**

```
int auxiliar= 0;  
  
string nota=" ";
```

El uso de estándares facilitará en gran medida una mejor comprensión del código por los desarrolladores, durante la implementación y en futuros mantenimientos. El código mostrará una mayor limpieza, claridad y organización, elementos estos que aportan un valor agregado a la solución.

4.2.4 Tratamiento de Excepciones

Una excepción es un evento que ocurre durante la ejecución del programa que interrumpe el flujo normal de las sentencias. Generalmente estos eventos ocurren cuando existe algún dato o instrucción que no se apega al funcionamiento del programa por lo que se produce un error. El manejo de excepciones permite al usuario crear aplicaciones tolerantes a fallas y robustos (resistentes a errores) para controlar estas excepciones y que pueda seguir ejecutando el programa sin verse afectado por el problema. (40)

El tratamiento de excepciones ayuda a lidiar con los errores de una aplicación por medio de la manipulación del código para hacer programas más robustos. En el código del sistema se usaron las herramientas que ayudan a manejarlas, tal es el caso de los bloques `try` (intentar) que encierran el código que puede lanzar una excepción y los bloques `catch` (atrapar) que lidian con las excepciones que surjan.

En este capítulo se desarrollaron los diagramas de componentes pertenecientes a los casos de uso y el modelo de datos, los cuales sirvieron de patrón para la implementación del sistema. Se diseñó la distribución física para representar el futuro despliegue del sistema. Además, se abordan los estilos y estándares de codificación utilizados en la implementación de la propuesta de solución logrando la organización del código, así como el tratamiento de excepciones.

Conclusiones

Luego de finalizada la investigación referida a la implementación del componente para la Gestión y el Monitoreo de Datos de Ensayos Clínicos se concluye:

- Los sistemas de gestión de ensayos clínicos investigados, no ofrecieron una solución al problema planteado en la presente investigación, por lo que se propuso el desarrollo del componente para la gestión y el monitoreo de datos en centros sin acceso al sistema Clínica 1.0.
- Las funcionalidades del módulo Enviar Datos del sistema Clínica 1.0 sirvieron como base para el desarrollo de la aplicación de escritorio, logrando un diseño compatible con el sistema Clínica.
- La utilización de la metodología RUP como guía, permitió detallar a través de los artefactos generados, cada una de las fases transitadas en el desarrollo del sistema, lo cual fue de gran ayuda en la obtención del producto final.
- La aplicación de la Arquitectura en 3 capas, permitió la organización de las funcionalidades en los tres componentes de diseño según sus responsabilidades, propiciando que los cambios que se realicen en algún componente no repercuta en los otros.
- El componente obtenido permitirá la gestión y el monitoreo de datos de forma segura y rápida, facilitando este trabajo a las diferentes instituciones hospitalarias del país que no tienen acceso al sistema Clínicas 1.0.

Recomendaciones

Luego de haber concluido el presente trabajo de diploma se recomienda:

- Realizar una nueva versión de funcionalidades para la gestión y el monitoreo de datos, que pueda ser usada en dispositivos móviles.

Referencias Bibliográficas

1. Medline Plus. *Medline Plus*. [En línea] Biblioteca Nacional de Medicina de EE.UU, 2013. [Citado el: 17 de 1 de 2014.] <http://www.nlm.nih.gov/medlineplus/spanish/clinicaltrials.html>. 2.
2. CIMAB SA. *CIMAB SA*. [En línea] Centro de Inmunología Molecular, 2009. [Citado el: 19 de 1 de 2014.] <http://www.cimab-sa.com/index.php?action=cim>.
3. Ensayos clínicos. Legislación Vigente de Medicamentos Huérfanos. [En línea] Instituto de Salud Carlos.Universidad de Barcelona. [Citado el: 25 de febrero de 2014.] <http://www.ub.edu/legmh/ereensay.htm>.
4. Erislán Martínez J., Landy Gonzáles E. *Clínicas:Desarrollo de la Gestión de Datos de Ensayos Clínicos a partir del sistema Open Clínica*. 2009.
5. Metoder.nu. [En línea] [Citado el: 25 de febrero de 2014.] <http://www.metoder.nu/cgi-bin/met.cgi?d=s&w=2052&l=es&s=mt>.
6. III Curso para Investigadores sobre Normas de Buena Práctica Clínica (BPC). [En línea] [Citado el: 18 de marzo de 2014.] http://www.farmacovigilanciacanarias.org/docs/CUADERNO_DE_RECOGIDA_DE_DATOS_DISENO_Y_CUMPLIMENTACION_P_Rdquez_Fortunez.pdf.
7. Mairin Sosa P., Ernesto R. Galiano G. *Desarrollo del módulo Gestionar Estudios*. 2012. pág. 8.
8. Phoenix Software International. *Phoenix Software International*. [En línea] Entrypoint Plus, Electronic data , 2012. [Citado el: 10 de 2 de 2014.] http://www.sciencesoftware.com.cn/search/search_soft_detail12.asp?id=966.
9. Phoenix Software International. *Phoenix Software International*. [En línea] Entrypoint Plus, 2012. [Citado el: 10 de 2 de 2014.] http://www.phoenixsoftware.com/entrypoint/entrypoint_90.htm.
10. Medicina21. *Medicina21*. [En línea] [Citado el: 10 de 2 de 2014.] http://www.medicina21.com/Notas_De_Prensa/V5148_Oracle_lanza_una_aplicacion_movil_pa.
11. Biocapax. *Biocapax*. [En línea] 2010. [Citado el: 5 de 3 de 2014.] http://www.biocapax.com/web/BCX_Clinical_Caracteristicas_es.asp.

12. AIN. *AIN*. [En línea] Agencia Información Nacional, 22 de 10 de 2013. [Citado el: 14 de 1 de 2014.] http://www.ain.cu/2013/octubre/22aem_ensayos_clinicos.htm.
13. Scielo "CLÍNICAS 1.0": Sistema de gestión de ensayos clínicos. *Scielo "CLÍNICAS 1.0": Sistema de gestión de ensayos clínicos*. [En línea] Revista Cubana de Informática Médica, 2012. [Citado el: 14 de 1 de 2014.] Disponible en: <http://scielo.sld.cu/scielo.php?pid=S1684->.
14. Buyto. *Buyto*. [En línea] 2009. [Citado el: 21 de 2 de 2014.] <http://www.buyto.es/general-diseno-web/diferencias-entre-aplicaciones-web-y-aplicaci>.
15. Toro Lazo, Alonso; Cardona Benjumea, Lorena. *Estado del arte de la ingeniería del software en el ámbito nacional e internacional de acuerdo a organizaciones que tratan la disciplina*. Universidad Católica Popular del Risaralda. Pereira, 2010.
16. Jacobson Grady, Ivar; James Rumbaugh, Booch. Parte 1 *El Proceso Unificado de Desarrollo de Software*. En: Núñez de Balboa (editor). *El Proceso Unificado de Desarrollo de Software*. Madrid: Pearson Education, S.A, 2000.464p.
17. G. Figueroa, Robert; J. Solís, Camilo; A. Cabrera, Armando. *Metodologías tradicionales vs. Metodologías ágiles*. Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.
18. Principales Características de JAVA. Principales Características de JAVA. [En línea] [Citado el: 20 de abril de 2014.] <http://personales.upv.es/rmartin/cursoJava/Java/Introduccion/PrincipalesCaracteristicas.htm>.
19. JAVA. JAVA. [En línea] [Citado el: 18 de abril de 2014.] <http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html>.
20. Gonzales, Carlos Daniel. Curso Java Interfaz Gráfica. Desarrollo de Software. *Curso Java Interfaz Gráfica. Desarrollo de Software*. [En línea] 2008. [Citado el: 20 de 4 de 2014.] <http://www.usabilidadweb.com.ar/JavaInterfacesGraficas.php>.
21. Marcela Calderón, Emilio Davis. Swing, la solución actual de Java para crear GUIs. Swing, la solución actual de Java para crear GUIs. [En línea] [Citado el: 20 de 4 de 2014.] <http://users.dcc.uchile.cl/~lmateu/CC60H/Trabajos/edavis/swing.html>.

22. Curso de Lenguaje Java. [En línea]. [Consultado el: 11 de febrero de 2014]. Disponible en: <http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/introduccion/virtual.htm>
23. Curso Hibernate. *Curso Hibernate*. [En línea] Introducción a los ORM, Curso Hibernate, 2013. [Citado el: 10 de 2 de 2014.] http://www.cursohibernate.es/doku.php?id=unidades:01_introduccion_orm:06_intro_orm.
24. Visual Paradigm International. *Visual Paradigm International*. [En línea] Visual Paradigm International. [Citado el: 17 de 2 de 2014.] <http://www.visual-paradigm.com..>
25. Lenguaje Unificado de Modelado (UML). *Lenguaje Unificado de Modelado (UML)*. [En línea] [Citado el: 17 de 2 de 2014.] <http://www.uml.org..>
26. TEAM, D. PgAdmin. *PgAdmin*. [En línea] PgAdmin PostgreSQL Tools, 2012. [Citado el: 19 de 2 de 2012.] <http://www.pgadmin.org..>
27. PostgreSQL-es. *PostgreSQL-es*. [En línea] [Citado el: 20 de 2 de 2014.] www.postgresql.org.es/sobre_postgresql.
28. PostgreSQL. *PostgreSQL*. [En línea] [Citado el: 20 de 2 de 2014.] <http://www.postgresql.org/about/news/1235/>.
29. Pearson Education S.A. Universidad de Granada. *Pearson Education S.A. Universidad de Granada*. [En línea] Pearson Education S.A. . [Citado el: 8 de 3 de 2014.] <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>.
30. Tello, Jesús Cáceres. *www2. www2*. [En línea] [Citado el: 8 de 3 de 2014.] <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>.
31. Pressman, S Roger. *Diseño Arquitectónico. Diseño Arquitectónico*. [En línea] [Citado el: 17 de 3 de 2014.] <http://eva2.uci.cu.pdf>.
32. Newcomlab. *Newcomlab*. [En línea] [Citado el: 16 de junio de 2014.] http://www.newcomlab.com/default.aspx?id_seccion=936.
33. Developer Network. *Diagramas de clases de UML*. [En línea] Microsoft, 2014. [Citado el: 18 de marzo de 2014.] <http://msdn.microsoft.com/es-es/library/dd409437.aspx>.

34. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Desing Patterns*. s.l. : Addison Wesley, 1995.
35. Conceptos básicos en un Diagrama de Colaboración. *Conceptos básicos en un Diagrama de Colaboración*. [En línea] [Citado el: 17 de mayo de 2014.] <http://webdocs.cs.ualberta.ca/~pfiguero/soo/uml/colaboracion01.html>.
36. Tutorial de fundamentos de base de datos. *Tutorial de fundamentos de base de datos*. [En línea] Instituto tecnologico de colina. [Citado el: 3 de mayo de 2014.] http://labredes.itcolima.edu.mx/fundamentosbd/sd_u2_1.htm.
37. Sparx Systems. *Sparx Systems*. [En línea] [Citado el: 18 de 4 de 2014.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.
38. Network, M.D. Diagrama de componentes de UML. [En línea] [Citado el: 25 de abril de 2014.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
39. Recursos web. [En línea] Universidad Nacional Autónoma de México, 2009. [Citado el: 7 de mayo de 2014.] <http://recursosweb.unam.mx/recursos-web/creacion-de-paginas-web/estandares-de-codificacion/>.
40. MOROS VALLE, BEGOÑA. *PROGRAMACIÓN ORIENTADA A OBJETOS*. [En línea] [Citado el: 25 de abril de 2014.] <http://dis.um.es/~bmoros/Tutorial/parte9/cap9-3.html>.

Bibliografía

EVA. Entorno Virtual de Aprendizaje.[En línea] [Citado el: 17 de marzo de 2014.] http://eva.uci.cu/file.php/180/2._Clases/Tema_1/Materiales_basicos/4.Patrones_de_diseno_de_BD.pdf..

BCX Biocapax. *BCX Clinical Glosario de términos.* [En línea] [Citado el: 5 de marzo de 2014.] http://www.biocapax.com/web/BCX_Clinical_Glosario_de_términos_es.asp.

Becerril C., Francisco. *Java a su alcance.* México : Litográfica Ingramex. 1998. 736.

Biocapax Technologies S.L. *Software para ensayos clinicos en CRO's, hospitales, laboratorios farmaceuticos y centros de investigación.* [En línea] [Citado el: 27 de febrero de 2014.] http://www.biocapax.com/web/BCX_Clinical_es.asp.

2014. BuscoCirujano.Com. *Oracle Health Sciences Mobile Clinical Research Associate.* [En línea] 2014. [Citado el: 27 de mayo de 2014.] <http://www.buscocirujano.com/noticias-buscocirujano-cirujanos-articulos/noticias-empresas-buscocirujano/1218-oracle-lanza-una-nueva-aplicacion-medica-oracle-health-sciences-mobile-clinical-research-associate.html>.

Camacho, Erika, Cardeso, Fabio y Nuñez, Gabriel. 2004. Entorno Virtual de Aprendizaje. [En línea] abril de 2004. http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_1/Arquitectura_de_Software/Arquitecturas_de_software._Guias_de_estudio.pdf.

2009. CIMAB SA. *Centro de Inmunología Molecular.* [En línea] 2009. [Citado el: 20 de enero de 2014.] <http://www.cimab-sa.com/index.php?action=cim>.

Clinical trial. *La gestión de ensayos clínicos.* [En línea] [Citado el: 23 de marzo de 2014.] <http://es.scribd.com/doc/14476388/Software-para-ensayos-clinicos>.

2007. ClinicalTrials, Understanding Clinical Trials. [En línea] National Institutes of Health, 2007. [Citado el: 27 de enero de 2014.] <http://clinicaltrials.gov/ct2/info/understand>.

Conceptos básicos en un Diagrama de Colaboración. *Conceptos básicos en un Diagrama de Colaboración.* [En línea] [Citado el: 17 de mayo de 2014.] <http://webdocs.cs.ualberta.ca/~pfiguero/soo/uml/colaboracion01.html>.

Correa, Niurka Fiel. 2012. Enciclopedia Virtual . *Contribuciones a las Ciencias Sociales*. [En línea] mayo de 2012. [Citado el: 9 de marzo de 2014.] <http://www.eumed.net/rev/cccss/20/nfc.html>.

2014. Cubadebate.Contra el Terrorismo Mediático. *Noticias*. [En línea] 2014. [Citado el: 20 de enero de 2014.] <http://www.cubadebate.cu/noticias/2012/12/28/crecen-en-mercado-externo-ventas-de-centro-de-inmunologia-molecular/>.

desarrolloweb.com. [En línea] [Citado el: 30 de marzo de 2014.] <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.

E. Gamma, R. Helm, R. Johnson, and J. Vlissides. 1995. *Desing Patterns*. s.l: Addison Wesley, 1995.

Ensayos clínicos. *Legislación Vigente de Medicamentos Huérfanos*. [En línea] Instituto de Salud Carlos.Universidad de Barcelona. [Citado el: 25 de febrero de 2014.] <http://www.ub.edu/legmh/ereensay.htm>.

2010. GCP-Service. *Clinical Trial Management System*. [En línea] GCP-Service International Ltd., 2010. [Citado el: 22 de marzo de 2014.] <http://www.gcp-service.eu/CTMS>.

Hernández León, Rolando Alfredo y Coello González, Sayda. 2011. *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA*. La Habana: Editorial Universitaria del Ministerio de Educación Superior, 2011. 978-959-16-1307-3.

Hernández Sampieri, Roberto, Fernández Collado, Carlos y Baptista Lucio, Pilar. *Metodología de la investigación*. México : Interamericana editores S.A. 2006. 970-10-5753-8.

Institut Ferran de Reumatología, S.L. [En línea] [Citado el: 23 de febrero de 2014.] <http://www.institutferran.org/>.

Instituto tecnológico de colima. Tutorial de fundamentos de base de datos. [En línea] [Citado el: 7 de mayo de 2014.] http://labredes.itcolima.edu.mx/fundamentosbd/sd_u1_6.htm.

JAVA. JAVA. [En línea] [Citado el: 18 de abril de 2014.] <http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html>.

Larman, Craig. 1999. UML Y PATRONES . *Introducción al análisis y diseño orientado a objetos*. [En línea] 1999. http://eva.uci.cu/file.php/158/Documentos/Bibliografia_general/Textos_Basicos/UML_y_Patrones/00_Presentacion_y_contenido.pdf. 970-17-0261-1.

- López, Angel.** *JAVA, la programación del futuro.* Buenos Aires MP Ediciones S.A. 1997. 987-9131-38-X.
- Mairin Sosa P., Ernesto R. Galiano G. 2012.** *Desarrollo del módulo Gestionar Estudios.* 2012. pág. 8.
- Marcela Calderón, Emilio Davis.** Swing, la solución actual de Java para crear GUIs. *Swing, la solución actual de Java para crear GUIs.* [En línea] [Citado el: 20 de 4 de 2014.] <http://users.dcc.uchile.cl/~lmateu/CC60H/Trabajos/edavis/swing.html>.
- MATO GARCIA, ROSA MARIA. 1999.** Entorno Virtual de Aprendizaje. [En línea] , octubre de 1999.
- MOROS VALLE, BEGOÑA.** *PROGRAMACIÓN ORIENTADA A OBJETOS.* [En línea] [Citado el: 25 de abril de 2014.] <http://dis.um.es/~bmosos/Tutorial/parte9/cap9-3.html>.
- Network, M.D.** *Diagrama de componentes de UML.* [En línea] [Citado el: 25 de abril de 2014.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
- OpenClinica®** - Gestión de estudios clínicos. [En línea] [Citado el: 18 de marzo de 2014.] <http://www.caduceus.es/productos-y-servicios/15-productos/55-openclinica-gestion-estudios-clinicos.html>.
- Oracle.** *Oracle En la Industria Clínica.* [En línea] [Citado el: 16 de febrero de 2014.] <http://www.oracle.com/us/industries/life-sciences/clinical/overview/index.html>.
- Phoenix Software.** *Entrypoint.* [En línea] [Citado el: 9 de febrero de 2014.] <http://www.phoenixsoftware.com/entrypoint.htm>.
- Phoenix Software International.** *Electronic Data Capture.* [En línea] [Citado el: 7 de febrero de 2014.] http://www.phoenixsoftware.com/entrypoint/entrypoint_electronicdatacapture.htm.
- Pressman, Roger S. 2010.** *Ingeniería de Software 7ma edición.* 2010. 978-0-07-337597-7.
- Pressman, S Roger.** *Diseño Arquitectónico.* [En línea] [Citado el: 17 de marzo de 2014.] <http://eva2.uci.cu.pdf>.
- Principales Características de JAVA.** *Principales Características de JAVA.* [En línea] [Citado el: 20 de abril de 2014.] <http://personales.upv.es/rmartin/cursoJava/Java/Introduccion/PrincipalesCaracteristicas.htm>.
- Rumbaugh, James, Jacobson, Ivar y Booch, Grady California.** *El Lenguaje Unificado de Modelado. Manual de Referencia.* California: s.n. 1998.

Schmuller, Joseph. 2000.. *Entorno Virtual de Aprendizaje*. [En línea] 2000. http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_1/Diseno_de_software/Aprendiendo_UML_en_24_Horas/00_Presentacion_y_contenido.pdf. 968-444-463-X.

SOMERVILLE, Ian. 2005. *Ingeniería de Software 7ma edición*. [En línea] 2005. [Citado el: 15 de febrero de 2014.] http://books.google.com.cu/books?id=gQWd49zSut4C&pg=PA80&dq=Herramientas+Case&hl=es&sa=X&ei=uOG3T-_cE8jggged2PSiCg&ved=0CEQQ6AEwAw#v=onepage&q=Herram.

Suárez, Liliana Cortés. 2011. Enciclopedia Virtual. *Contribuciones a las Ciencias Sociales*. [En línea] enero de 2011. [Citado el: 10 de marzo de 2014.] <http://www.eumed.net/rev/cccss/11/lcs.htm>.

Glosario de Términos

A.

API: Conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Representa una interfaz de comunicación entre componentes software.

B.

Biofármacos: Producto medicinal, terapéutico o de

C.

CASE: Ingeniería de Software Asistida por Ordenador (Por sus siglas en inglés Computer Aided Software Engineering).

CIC: Coordinador de investigación clínica. Persona implicada en la Gestión de los Datos de un Ensayo Clínico.

CIM: Centro de Inmunología Molecular.

CRD: Cuaderno de Recogida de Datos. Formulario diseñado para anotar las variables recogidas durante un ensayo clínico.

CU: Caso de Uso.

D.

E.

EC: Ensayo Clínico.

Especialista: Persona que agrupa un conjunto de roles dentro del sistema tales como: Investigador Principal, Coordinador y Monitor.

Estudio: Ensayo Clínico

F.

G.

GDCE: Gestión de Datos de Ensayos Clínicos. Conjunto de procesos de un ensayo clínico dirigidos por el CIC para gestionar información asociada a Pacientes, Hojas CRD y Momentos de Seguimiento de un estudio.

H.

Hoja CRD: Formulario que agrupa un conjunto de variables sobre datos de interés del paciente para el estudio.

I.

IDE: Entorno de Desarrollo Integrado por sus siglas en inglés (Integrated Development Environment).

Investigador Principal: Persona encargada de toda la investigación de un ensayo.

J.

K.

L.

M.

Momento de Seguimiento: Momento en el cual se le realizan exámenes a los sujetos o se le aplica algún fármaco y se recogen datos de interés para un estudio.

Monitor: Persona implicada en el proceso de monitoreo de un Ensayo Clínico.

Monitoreo: Proceso que se encarga de revisar los datos asociados a una Hoja CRD con el objetivo de identificar los posibles errores desde el punto de vista clínico y solicitar explicaciones de estos errores mediante las notas de monitoreo generadas en dicho proceso.

N.

Notas de Discrepancias: Anotaciones dentro del sistema realizadas por un especialista.

O.

P.

Q.

R.

S.

Sujeto: persona implicada en un estudio determinado.

T.

U.

UML: Lenguaje de Modelado Unificado.

V.

W.

X.

Y.

Z.

Anexo

Anexo1

Entrevista

Entrevista realizada a Erilán Martínez Jera.

Objetivo de la entrevista

Se realizó con el objetivo de profundizar en el tema de la investigación, obteniendo diferentes puntos de vistas respecto al problema a resolver así como la identificación de las funcionalidades a implementar.

1. ¿Cuáles son los problemas que en la actualidad impiden la utilización del sistema Clínicas 1.0 a lo largo del territorio nacional?
2. ¿Cómo se realiza la gestión de ensayos clínicos en el CIM?
3. ¿Cómo se realiza el monitoreo de los datos en las instituciones hospitalarias?
4. ¿Qué tipo de notas se necesitarán gestionar en el sistema?
5. ¿Cuáles son los roles que intervienen durante la conducción de ensayos clínicos en el CIM?
6. ¿Cuáles son los estados de resolución que puede llegar a tener una nota?
7. ¿Cómo se logrará la homogeneidad de la información almacenada en el sistema Clínica 1.0 con la gestionada en la aplicación creada?