



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 2

Trabajo de Diploma para Optar por el Título de
Ingeniero en Ciencias Informáticas

Desarrollo de funcionalidades para la gestión de sujetos y cronogramas específicos en centros sin acceso al sistema Clínicas 1.0

Autores: Ardelio Madrigal Morera

Karina Barbara Martinez Casas

Tutores: Ing. Lenia Palau Yú

Ing. Gelsys González Souto

Co-tutor: Ing. Daniellys La O Cervante

**La Habana, junio de 2014
“Año 56 de la Revolución”**

Declaración de autoría

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

Para que así conste firmamos la presente a los 17 días del mes de junio del año 2014.

Karina Barbara Martinez Casas
Firma del Autor

Ardelio Madrigal Morera
Firma del Autor

Ing. Lenia Palau Yú
Firma del Tutor

Ing. Gelsys González Souto
Firma del Tutor

Ing. Daniellys La O Cervante
Firma del Tutor

Datos de Contacto

Ing. Lenia Palau Yú.

Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2010. En la vinculación con la producción pertenece al Departamento de Sistemas Especializados en Salud (SES) del Centro de Informática Médica (CESIM) y específicamente trabaja en el desarrollo del proyecto Prótesis 2.0 donde se desempeña como Líder de Proyecto y Planificadora.

Correo electrónico: lpalau@uci.cu

Ing. Gelsys González Souto.

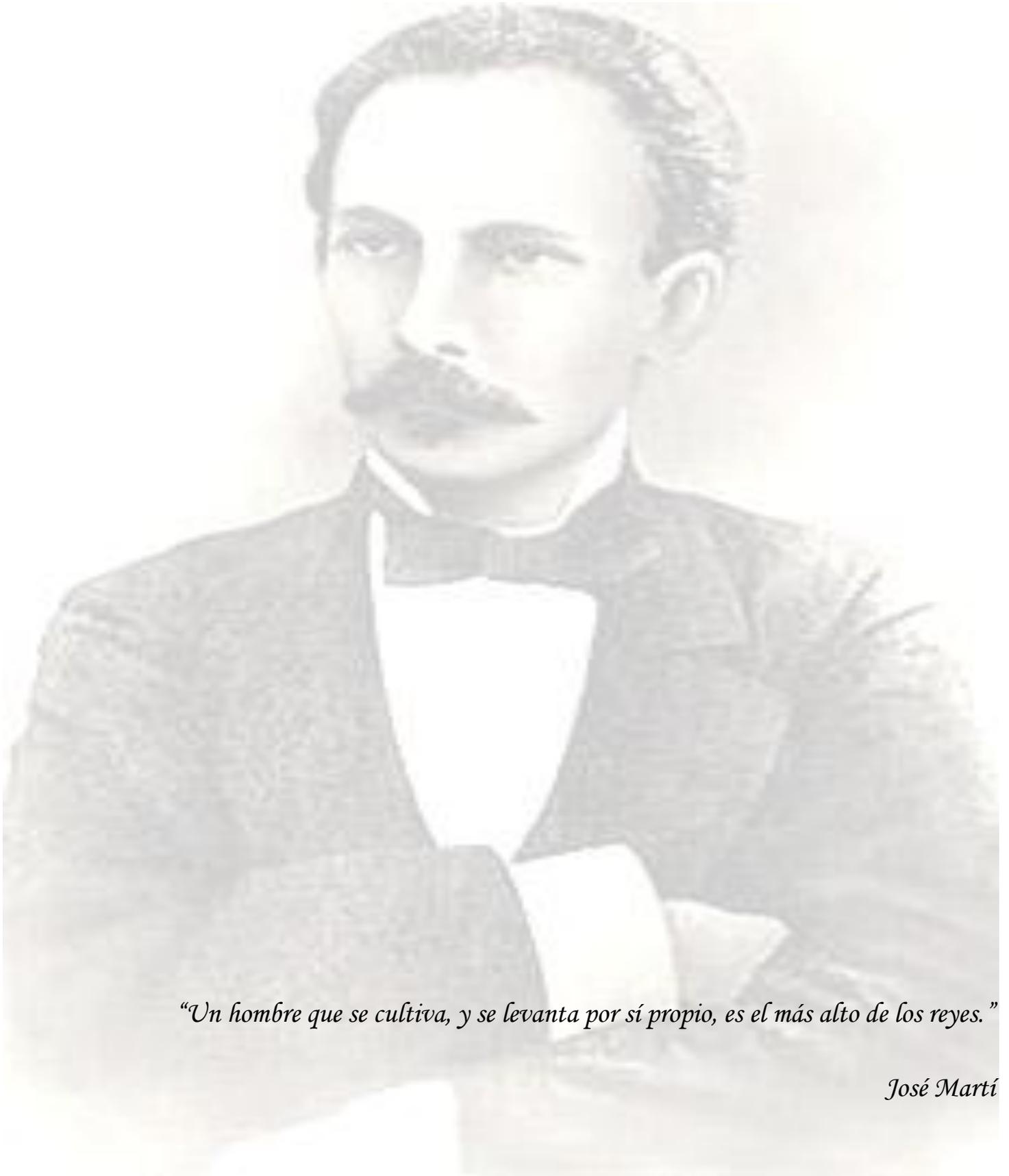
Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2012. En la vinculación con la producción pertenece al Departamento de Sistemas Especializados en Salud (SES) del Centro de Informática Médica (CESIM) y específicamente trabaja en el desarrollo del proyecto SEND donde se desempeña como Analista.

Correo electrónico: gsouto@uci.cu

Ing. Daniellys La O Cervante.

Graduado de Ingeniería en Ciencias Informáticas en el año 2013 en la Universidad de las Ciencias Informáticas y actualmente desempeña el rol de Administrador de la Calidad en el Departamento Sistemas Especializados en Salud (SES).

Correo electrónico: dlao@uci.cu



“Un hombre que se cultiva, y se levanta por sí propio, es el más alto de los reyes.”

José Martí

Resumen

En el Centro de Inmunología Molecular (CIM), actualmente para la gestión de sujetos y cronogramas específicos en la conducción de ensayos clínicos, se utiliza el sistema informático Clínicas 1.0. Sin embargo algunos de los centros que colaboran con el CIM en estos procesos no cuentan con acceso a dicho sistema. Es por ello que el presente trabajo propone desarrollar funcionalidades que permitan la gestión de sujetos y cronogramas específicos en dichos centros.

El desarrollo de las funcionalidades se basa en tecnologías de escritorio, multiplataforma y utiliza el patrón arquitectónico Modelo Vista Controlador. Emplea Java como lenguaje de programación, NetBeans como entorno de desarrollo y PostgreSQL como sistema gestor de bases de datos. El proceso de desarrollo estuvo guiado por el Proceso Unificado de Desarrollo (RUP).

Las funcionalidades facilitarán el manejo de toda la información referente a la gestión de sujetos y cronogramas específicos a los especialistas de los centros sin acceso al sistema Clínicas 1.0. Permitirán además realizar la búsqueda de las acciones realizadas por los especialistas de forma rápida y se evitará la duplicidad, pérdidas y deterioro de la información.

Palabras clave: Centro de Inmunología Molecular, Centros sin acceso al sistema Clínicas 1.0, Clínicas 1.0, Conducción, Ensayos Clínicos, Sujetos.

Índice

Introducción 1

Capítulo 1: Fundamentación teórica 5

 1.1 Términos relacionados con la investigación 5

 1.2 Tendencias actuales de los sistemas de gestión de ensayos clínicos 6

 1.3 Funcionalidades del módulo Enviar Datos del sistema Clínicas 1.0 9

 1.4 Metodología de Desarrollo de Software..... 9

 1.5 Tecnologías y Herramientas a utilizar..... 10

Capítulo 2: Características de las funcionalidades 18

 2.1 Propuesta de solución 18

 2.2 Flujo actual de los procesos involucrados en el campo de acción 18

 2.3 Objeto de automatización..... 19

 2.4 Modelo de Dominio 19

Descripción de clases 20

 2.3 Especificación de los Requisitos 21

Requisitos Funcionales 21

Requisitos no Funcionales 23

 2.4 Diagrama de Casos de Uso del Sistema 25

 2.5 Actores del Sistema 26

 2.6 Descripción de Casos de Uso del Sistema 26

Capítulo 3: Análisis y Diseño de las funcionalidades..... 32

 3.1 Modelo de análisis..... 32

Diagramas de Clases del Análisis 32

Diagramas de Colaboración 35

 3.2 Modelo de diseño 38

Diagramas de Clases de Diseño 38

Descripción de las Clases del Diseño 41

Descripción de la Arquitectura..... 42

Patrón Arquitectónico: Modelo Vista Controlador (MVC) 42

Patrones de diseño 44

Capítulo 4: Implementación de las funcionalidades 49

 4.1 Modelo de implementación..... 49

TABLA DE CONTENIDOS

4.2 Modelo de datos.....	51
4.3 Tratamiento de excepciones	55
4.4 Seguridad e Integridad de la información	55
4.5 Estrategias de codificación. Estándares y estilos a utilizar	56
Conclusiones	59
Recomendaciones	60
Referencias Bibliográficas.....	61
Bibliografía.....	65
Anexos.....	69
Anexo 1: Entrevista	69
Anexo 2: Diagramas de clases de diseño	70
Anexo 3: Interfaces de usuario.....	76

Índice de Figuras

Figura.1 Modelo Conceptual20

Figura.2 Diagrama de Casos de Uso del Sistema.....25

Figura.3 Diagrama de clase de análisis CU Añadir sujeto.....33

Figura.4 Diagrama de clase de análisis CU Interrumpir sujeto33

Figura.5 Diagrama de clase de análisis CU Visualizar listado de sujetos34

Figura.6 Diagrama de clase de análisis CU Visualizar datos de sujeto34

Figura.7 Diagrama de clase de análisis CU Visualizar momentos de seguimiento por sujeto34

Figura.8 Diagrama de colaboración CU Añadir sujeto.....35

Figura.9 Diagrama de colaboración CU Interrumpir sujeto.....36

Figura.10 Diagrama de colaboración CU Visualizar listado de sujetos.....36

Figura.11 Diagrama de colaboración CU Visualizar datos de sujeto37

Figura.12 Diagrama de colaboración CU Visualizar momentos de seguimiento por sujeto37

Figura.13 Diagrama de clase de diseño CU Añadir sujeto38

Figura.14 Diagrama de clase de diseño CU Interrumpir sujeto39

Figura.15 Diagrama de clase de diseño CU Visualizar listado de sujetos39

Figura.16 Diagrama de clase de diseño CU Visualizar datos de sujeto.....40

Figura.17 Diagrama de clase de diseño CU Visualizar momentos de seguimientos por sujeto41

Figura.18 Patrón MVC42

Figura.19 Uso del patrón DAO45

Figura.20 Estructura del DAO45

Figura.21 Uso del patrón Bajo acoplamiento46

Figura.22 Uso del patrón Alta cohesión46

Figura.23 Comportamiento del patrón Experto.....47

Figura.24 Comportamiento del patrón Creador47

Figura.25 Comportamiento del patrón Controlador47

Figura.26 Comportamiento del patrón Iterator.....48

Figura.27 Diagrama de componentes “Gestionar sujeto”	50
Figura.28 Diagrama de despliegue del sistema	51
Figura.29 Modelo de datos de las funcionalidades	52
Figura.30 Diagrama de clase de diseño CU Autenticar usuario	70
Figura.31 Diagrama de clase de diseño CU Actualizar perfil.....	70
Figura.32 Diagrama de clase de diseño CU Cambiar centro/estudio	71
Figura.33 Diagrama de clase de diseño CU Generar cronograma específico	71
Figura.34 Diagrama de clase de diseño CU Buscar sujeto	72
Figura.35 Diagrama de clase de diseño CU Visualizar momentos de seguimiento	72
Figura.36 Diagrama de clase de diseño CU Añadir Momento de seguimiento no programado	73
Figura.37 Diagrama de clase de diseño CU Visualizar listado de hojas CRD	73
Figura.38 Diagrama de clase de diseño CU Visualizar cronograma específico	74
Figura.39 Diagrama de clase de diseño CU Imprimir cronograma específico	74
Figura.40 Diagrama de clase de diseño CU Imprimir momentos de seguimiento por sujeto	75
Figura.41 Diagrama de clase de diseño CU Firmar hoja CRD	75
Figura.42 Interfaz de usuario CU Autenticar usuario.....	76
Figura.43 Interfaz de usuario CU Actualizar perfil	76
Figura.44 Interfaz de usuario CU Cambiar centro/estudio.....	77
Figura.45 Interfaz de usuario CU Generar cronograma específico.....	77
Figura.46 Interfaz de usuario CU Buscar sujeto.....	77
Figura.47 Interfaz de usuario CU Visualizar momentos de seguimiento.....	78
Figura.48 Interfaz de usuario CU Añadir Momento de seguimiento no programado.....	78
Figura.49 Interfaz de usuario CU Visualizar listado de hojas CRD.....	79
Figura.50 Interfaz de usuario CU Visualizar cronograma específico	79
Figura.51 Interfaz de usuario CU Imprimir cronograma específico.....	79
Figura.52 Interfaz de usuario CU Imprimir momentos de seguimiento por sujeto.....	80
Figura.53 Interfaz de usuario CU Firmar hoja CRD.....	80

Índice de Tablas

Tabla 1. Sistemas de gestión de EC6

Tabla 2. Actores del sistema26

Tabla 3. Descripción CU Añadir sujeto.....27

Tabla 4. Descripción CU Interrumpir sujeto.....28

Tabla 5. Descripción CU Visualizar listado de sujetos.....29

Tabla 6. Descripción CU Visualizar datos de sujeto30

Tabla 7. Descripción CU Visualizar momentos de seguimiento por sujeto30

Tabla 8. Descripción de estereotipos32

Tabla 9. Descripción de las clases del diseño.....42

Tabla 10. Descripción de las tablas del modelo de datos.....55

Introducción

En la década de los setenta la revolución electrónica marcó un punto de partida a la llamada "Era Digital" y diez años más tarde convergían la electrónica, la informática y las telecomunicaciones, dando surgimiento a las Tecnologías de la Información y las Comunicaciones (TIC). Actualmente el desarrollo vertiginoso de las TIC ha traído consigo cambios para diferentes campos de la sociedad, aportándole grandes adelantos a los mismos.

La aplicación de las TIC en el área de la salud mediante el uso de equipamiento y soluciones informáticas, se conoce como Informática Médica. Ello garantiza la disminución del margen de error médico, siendo su principal objetivo el mejoramiento de la calidad y eficiencia de la atención sanitaria a través del uso de las nuevas tecnologías. Una de las aplicaciones de la Informática Médica se evidencia al utilizar tecnologías de punta en la creación de medicamentos, los cuales se aplican en el tratamiento y prevención de enfermedades que provocan altos niveles de muertes en el mundo.

En Cuba, el sector de la salud se encuentra en proceso de la informatización de sus servicios a pesar de que el país no cuenta con la infraestructura tecnológica suficiente. El Ministerio de Salud Pública (MINSAP) conjuntamente con la Universidad de las Ciencias Informáticas (UCI) y otras empresas, trabajan en este proceso para contribuir a la mejora de los servicios de atención sanitaria a todos los niveles de atención médica. Una de las instituciones beneficiadas con este proceso es el Centro de Inmunología Molecular (CIM) de La Habana, inaugurado el 5 de diciembre de 1994.

El CIM es una institución biotecnológica cubana dedicada a la investigación básica, desarrollo y fabricación de productos a partir del cultivo de células de mamíferos, de acuerdo con las regulaciones de las actuales Buenas Prácticas de Manufactura. Los proyectos de investigación básica están concentrados en la inmunoterapia del cáncer, especialmente en el desarrollo de "vacunas moleculares", ingeniería de anticuerpos, ingeniería celular, bioinformática y regulación de la respuesta inmune. La búsqueda de nuevos fármacos para el diagnóstico, tratamiento del cáncer y enfermedades relacionadas con el sistema inmune, constituye el principal objetivo de esta institución, por lo que realiza Ensayos Clínicos (EC) en centros altamente especializados. [1]

Los EC son estudios en los que se compara, mediante medición de varios parámetros, los efectos de dos o más tratamientos [2]. Son una parte fundamental en el proceso de desarrollo, aprobación e introducción en el mercado de nuevos fármacos y tratamientos contra las enfermedades. El principal objetivo de la mayor parte de ellos es descubrir si los nuevos tratamientos son seguros y eficaces. Sin embargo, también pueden estar destinados a aprender cómo prevenir la enfermedad, diagnosticar precozmente la patología o bien mejorar la calidad de vida de los enfermos. [3]

En un ensayo clínico se genera y maneja un gran volumen de información, que pueden ser: datos del ensayo, imágenes, aprobaciones, modificaciones, además de los datos del tratamiento de cada paciente. Toda esta documentación es necesaria para cumplir con las buenas prácticas clínicas exigidas por todas las agencias reguladoras a nivel mundial, y debe ser almacenada por no menos de 15 años posterior al cierre del estudio de modo que pueda ser inspeccionada en cualquier momento por cualquiera de las agencias reguladoras. [4]

Para la gestión de los EC los especialistas del CIM utilizan el sistema informático Clínicas 1.0, desarrollado por informáticos del proyecto Ensayos Clínicos, del Departamento Sistemas Especializados en Salud (SES) en el Centro de Informática Médica (CESIM) de la UCI. Clínicas 1.0 consta de cuatro módulos que permiten el diseño y la conducción de los ensayos: Administrar Sistema, Gestionar Estudio, Extraer Datos y Enviar Datos, en este último se realiza la conducción de EC. La conducción está formada por varios procesos, entre los que se encuentran: gestión de los sujetos y gestión de los datos asociados al estudio de los sujetos. Estos procesos se encargan de gestionar los datos personales de los pacientes que participan en el EC y la planificación de sus actividades durante el estudio, las cuales se recogen en un cronograma específico.

Algunos centros especializados que colaboran con el CIM en la gestión de sujetos y cronogramas específicos, no tienen acceso al sistema Clínicas 1.0; por tanto la recogida de la información referente a estos procesos se realiza de forma manual y se registra en formato duro, lo que puede ocasionar pérdida, duplicidad y retraso en la entrega de la misma. En un EC se maneja un alto volumen de datos, por lo que la gestión de sujetos y cronogramas específicos resulta tediosa y difícil de realizar; exigiendo mayor tiempo por parte de los especialistas transcribiendo información de los EC realizados, para luego exponerla al deterioro provocado por su continua manipulación.

Por lo antes planteado se propone como **problema a resolver**: ¿Cómo facilitar la gestión de sujetos y cronogramas específicos en centros sin acceso al sistema Clínicas 1.0?

Para darle respuesta al problema propuesto se define como **objeto de estudio**: el proceso de gestión de ensayos clínicos, enmarcado en el **campo de acción**: la gestión de sujetos y cronogramas específicos en ensayos clínicos.

Se define como **objetivo general**: desarrollar funcionalidades para la gestión de sujetos y cronogramas específicos en centros sin acceso al sistema Clínicas 1.0.

Para darle cumplimiento al objetivo se proponen como **tareas de la investigación**:

- ✓ Caracterización del estado del arte relacionado con la informatización del proceso de gestión de ensayos clínicos que puedan ser reutilizados.

- ✓ Descripción de las funcionalidades del módulo Enviar Datos del sistema Clínicas 1.0 vinculadas con el campo de acción.
- ✓ Descripción de las herramientas y tecnologías a utilizar para el desarrollo de las funcionalidades.
- ✓ Elaboración de los artefactos correspondientes a las disciplinas: Modelado de Negocio, Requisitos, Análisis y Diseño e Implementación según la metodología Rational Unified Process.
- ✓ Desarrollo de funcionalidades para la gestión de sujetos y cronogramas específicos.

En el transcurso y desarrollo de esta investigación científica se utilizaron los siguientes métodos teóricos y empíricos:

Teóricos:

- ✓ Histórico – Lógico: con la ayuda de este método se estudió cómo ha evolucionado la informatización de los procesos de gestión de EC a través de los años.
- ✓ Inductivo – Deductivo: se utilizó para analizar los sistemas existentes que gestionan información referente a los EC a nivel internacional y nacional, permitiendo llegar a conclusiones lógicas a partir de los conocimientos adquiridos en la investigación y así poder plantear una solución.
- ✓ Analítico – Sintético: se utilizó para el análisis de la documentación relacionada con la gestión de EC, donde se determinarán los conceptos relacionados con el objeto de estudio para el entendimiento del problema planteado y luego poder sintetizarlos para la elaboración de la solución.
- ✓ Modelación: se empleó mediante la realización de los artefactos correspondientes a las fases: Modelación del Negocio, Requisitos, Análisis y Diseño e Implementación de la solución propuesta.

Empíricos:

- ✓ Entrevista: se utilizó la entrevista no estructurada o no estandarizada para obtener información sobre la gestión de sujetos y cronogramas específicos en el sistema Clínicas 1.0.
- ✓ Observación: se empleó para la obtención de información acerca del objeto de investigación en el sistema Clínicas 1.0.

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

- ✓ Facilitará el manejo de toda la información referente a la gestión de sujetos y cronogramas específicos, eliminando redundancias y duplicidades.
- ✓ Proveerá la integridad de la información, referente a la gestión de sujetos y cronogramas específicos.
- ✓ Permitirá optimizar los recursos empleados para la gestión de sujetos y cronogramas específicos.

El siguiente trabajo está estructurado en 4 capítulos:

Capítulo 1. Fundamentación teórica: se realiza un estudio del estado del arte de los sistemas que gestionan EC; se describen las metodologías, tecnologías y las herramientas utilizadas para la solución propuesta.

Capítulo 2. Características de las funcionalidades: se describen los conceptos del dominio, se realiza la propuesta de solución, se muestran los requisitos funcionales y no funcionales, se presentan los diagramas de casos de uso del sistema y su descripción.

Capítulo 3. Análisis y Diseño de las funcionalidades: se describen los patrones de diseño, la arquitectura a utilizar, se presentan los diagramas de clases del análisis, colaboración y clases del diseño. Se describen las principales clases en el diseño de las funcionalidades.

Capítulo 4. Implementación de las funcionalidades: se realizan los diagramas de componentes, el diagrama de despliegue del sistema, el modelo de datos y la descripción de las entidades. Se describen el tratamiento de excepciones, la seguridad del sistema, las estrategias de codificación, estándares y estilos que se utilizan.

Capítulo 1: Fundamentación teórica

El presente capítulo ofrece definiciones y conceptos relacionados con la gestión de sujetos y cronogramas específicos en los EC para conocer cómo se realiza actualmente este proceso. Se caracterizan sistemas que gestionan EC y se describen la metodología, tecnologías y herramientas a utilizar para el desarrollo de las funcionalidades.

1.1 Términos relacionados con la investigación

Ensayos clínicos

Un ensayo clínico es un estudio experimental que se lleva a cabo en seres humanos, permite a los médicos determinar si un nuevo tratamiento, medicamento o dispositivo contribuirá a prevenir, detectar o tratar una enfermedad. Surgen como la respuesta científica a la necesidad ética de garantizar la eficacia y la seguridad de los tratamientos que reciben los pacientes. Son siempre prospectivos, se planifican y a continuación se realizan, siguiendo la evolución de los sujetos de investigación a lo largo del tiempo. Los EC proporcionan un método objetivo, reproducible y controlado para medir los efectos de un tratamiento sobre la salud. [2]

Conducción de Ensayos Clínicos

La conducción de los EC está formada por tres procesos:

- ✓ La gestión de los sujetos: en este proceso se incluyen los sujetos a los cuales se les realizarán los estudios, además se gestionan sus datos personales.
- ✓ La gestión de los datos asociados al estudio de los sujetos: se gestionan los datos asociados al estudio realizado a cada sujeto, si se culmina o se interrumpe su tratamiento, entre otras actividades asociadas a la planificación de la gestión de los datos.
- ✓ Monitorización de los datos asociados al estudio de los sujetos: en la monitorización, el monitor puede hacerle auditorías a los datos asociados al estudio correspondiente a cada sujeto. [5]

Cronograma General

Planificación detallada de los momentos de seguimiento de todos los sujetos que participan en el estudio.

Cronograma Específico

Planificación detallada de los momentos de seguimiento definidos a un sujeto específico.

Sujeto

Individuo incluido en un estudio o EC.

Cuaderno de Recogida de Datos (CRD)

Documento impreso, óptico o electrónico, diseñado para registrar toda la información requerida en el protocolo de investigación para ser reportada al promotor sobre cada individuo del estudio [6]. Está formado por una o varias Hojas CRD.

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.

1.2 Tendencias actuales de los sistemas de gestión de ensayos clínicos

En la siguiente tabla se muestran los sistemas estudiados.

Sistemas de Gestión de EC			Tipo de Licencia		Web
Nombre	País	Multiplataforma	Libre	Propietario	
Oracle Health Sciences Mobile Clinical Research Associate	EE.UU	✗	✗	✓	✗
Entrypoint Plus	EE.UU	✗	✗	✓	✗
BCX Clinical	España	✗	✗	✓	✗
Clínicas 1.0	Cuba	✓	✓	✗	✓

Tabla 1. Sistemas de gestión de EC

A continuación se presentan las características de los sistemas mostrados en la tabla anterior, considerados significativos para el objeto de estudio de la investigación.

Sistemas internacionales

Oracle Mobile Clinical Research Associate

Aplicación para dispositivos móviles desarrollada sobre software propietario, presentada por Oracle Corporation diseñada específicamente trabajos de campo, con el fin de ayudar a las organizaciones de investigación médica a acortar los tiempos de los estudios clínicos y reducir costos. Se trata de una aplicación nativa para el sistema operativo iOS, disponible ya para iPhone y iPad en Apple iTunes Store.

Está integrada al sistema de gestión de ensayos clínicos Oracle Health Sciences y brinda a los equipos de investigación, la información esencial de los ensayos clínicos en tiempo real; tanto si el usuario está o no conectado al sistema. Cuenta con una función que captura los detalles de ubicación gracias a las capacidades de geo-localización, e incorpora esta información en el informe, para que sea lo más completo posible y conforme a los requisitos legales.

Permite acceder a todos los indicadores de rendimiento del estudio y ver cómo progresan los resultados, lo que posibilita planificar visitas más productivas y eficientes a los sitios de estudio. La aplicación incorpora un sistema de alertas para notificar hechos destacados o problemas potenciales presentados en los sitios que se están monitorizando. [7]

Entrypoint Plus

Sistema propietario de captura de datos diseñado por Phoenix para crear, desarrollar y administrar ensayos clínicos. Es una aplicación de escritorio que incluye un conjunto de planillas de CRD con campos para datos demográficos, historia médica, reacciones adversas y exámenes de laboratorios; soporta la entrada remota de datos y utiliza ODBC¹ para interactuar con bases de datos SQL². Dentro de sus avanzadas opciones de administración incluye variedad de funciones que generan un alto nivel de seguridad.

Uno de sus módulos se denomina Administrar Entrypoint, el cual permite la gestión de usuarios del sistema para el control de acceso de estos a la aplicación, define grupos de usuarios para el acceso a módulos individuales. Permite exportar la información a varios formatos como Microsoft Excel (CSV), Access y XML³. [8]

BCX Clinical

Sistema propietario utilizado 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 digital del CRD y BCX clinical core, que brinda la interfaz de recogida de los datos de individuos del estudio, facilitando las herramientas necesarias para la validación y exportación de los datos almacenados.

Este software permite controlar la etapa de recogida de datos de un ensayo clínico, en cualquiera de las fases clínicas, a través de las múltiples funcionalidades que incorpora. Gracias a esta aplicación no sólo se agiliza el proceso de monitorización de un ensayo clínico, sino que la entidad investigadora está respaldada durante todo el proceso por medidas de seguridad, que ofrecen gran fiabilidad a los datos recopilados acordes con la legislación vigente.

Este tipo de software está diseñado para ser usado por parte de grandes entidades, como hospitales, laboratorios farmacéuticos o centros de investigación, que precisan de una herramienta para diseñar y cumplimentar el cuaderno de recogida de datos de sus EC. [9]

¹ Open Database Connectivity

² Structured Query Language

³ Extensible Markup Language

Luego de realizar un estudio de los sistemas antes mencionados, se concluye que los mismos no presentan una solución al problema planteado; debido a que están desarrollados sobre software propietario limitando al país su adquisición y posibles modificaciones en sus bases de datos para lograr un diseño compatible con el sistema Clínicas 1.0 utilizado por el CIM. Además el sistema Oracle Mobile Clinical Research Associate, es una aplicación para dispositivos móviles, y los centros especializados sin acceso al sistema Clínicas 1.0 no cuentan con recursos para el despliegue de un sistema de este tipo.

Sistemas nacionales

Clínicas 1.0

Sistema utilizado por el CIM para la gestión de EC, desarrollado por la Universidad de las Ciencias Informáticas a partir del sistema OpenClínica v2.5. Permite la realización de un cronograma de ejecución general para cada ensayo y la generación de un cronograma específico para cada sujeto, así como la validación de las variables de los ensayos, entre otras funcionalidades.

Ofrece entre sus ventajas la disminución y detección de los errores en los datos del EC y restringe las direcciones IP desde las cuales se puede acceder al mismo. El sistema Clínicas 1.0 es accesible desde cualquier sistema operativo, solo es necesario un navegador. Al igual que el OpenClínica v2.5 cuenta con cuatro módulos: Administrar Sistema, Gestionar Estudio, Enviar Datos y Extraer Datos, los cuales están estrechamente vinculados, de modo que no permite el funcionamiento de estos por separados.

En el módulo Enviar Datos se realiza la conducción de ensayos clínicos. Permite generar el cronograma específico y se realiza el llenado del Cuaderno de Recogida de Datos (CRD); este cuaderno no es más que un formulario para registrar los datos de cada uno de los sujetos incluidos en el estudio. Por otra parte, el módulo permite realizar el monitoreo de los datos de los sujetos y de las notas asociadas a cada uno de estos.

La seguridad del sistema es controlada en dos etapas. Una vez que el usuario se autentica, se verifica que su nombre de usuario y contraseña sean correctos, y que su acceso sea desde una computadora cuya dirección IP se encuentre registrada en el sistema para ese usuario. El usuario accede a la zona de trabajo que le corresponde teniendo en cuenta el rol, el centro al que pertenece y el ensayo clínico al que tiene acceso.

El estudio del sistema Clínicas 1.0, permitió conocer que el módulo Enviar Datos posee las funcionalidades necesarias para la gestión de sujetos y cronogramas específicos; sin embargo no constituye una solución debido a que no permite el funcionamiento independiente de este módulo. Los centros sin acceso a este sistema, no cuentan con la infraestructura necesaria ni con el presupuesto

requerido para el despliegue del sistema Clínicas 1.0 en su totalidad, ni otro sistema basado en tecnologías web. Por lo que se decide estudiar las funcionalidades de gestión de sujetos y cronogramas específicos en el módulo Enviar Datos, con el fin de desarrollar una solución basada en tecnologías de escritorio que cuente con un diseño compatible con el utilizado por el sistema Clínicas 1.0.

1.3 Funcionalidades del módulo Enviar Datos del sistema Clínicas 1.0

A continuación se describen las funcionalidades vinculadas con el campo de acción.

Gestión de sujetos: Permite la inserción de sujetos en un estudio, gestionar sus momentos de seguimientos programados y no programados, obtener las hojas CRD de un momento de seguimiento seleccionado, además ofrece la opción de interrumpir el sujeto deseado y visualizar los datos asociados a cada uno de ellos.

Creación del cronograma específico: Permite generar el cronograma específico para cada sujeto que participa en el estudio, a partir del cronograma general. El cronograma general contiene todos los sujetos que participan en el estudio y sus momentos de seguimientos mientras que el específico solo contiene los momentos de un solo sujeto.

1.4 Metodología de Desarrollo de Software

Conjunto de procedimientos, técnicas y ayudas a la documentación para asistir al proceso de desarrollo de software. Una metodología puede seguir uno o varios modelos del ciclo de vida, indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. [10]

A continuación se presenta la metodología que será empleada en el desarrollo de las funcionalidades.

RUP

RUP es una metodología iterativa e incremental centrada en la arquitectura y guiada por los casos de uso. Utiliza como lenguaje de presentación visual UML⁴. Se caracteriza por asignar tareas y responsabilidades. Divide el proceso en cuatro fases Inicio, Elaboración, Construcción y Transición, realizando en cada una de ellas una serie de artefactos. [11]

Establece además nueve disciplinas, seis de ingeniería y tres de apoyo a realizar en cada fase del proyecto.

Disciplinas de Ingeniería:

⁴ Unified Modelling Language

- ✓ Modelado del Negocio: describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- ✓ Requisitos: define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- ✓ Análisis y Diseño: describe cómo el sistema será elaborado a partir de las funcionalidades previstas y los requisitos impuestos.
- ✓ Implementación: define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- ✓ Prueba: busca los defectos a lo largo del ciclo de vida del software.
- ✓ Despliegue: realiza las actividades de empaque, instalación y asistencia a usuarios para entregar el software a los usuarios finales.

Disciplinas de Apoyo:

- ✓ Gestión del proyecto: involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- ✓ Gestión de cambio y configuraciones: describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a utilización y actualización concurrente de elementos, control de versiones.
- ✓ Gestión de Entorno: contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto, así como el procedimiento para implementar el proceso en una organización. [12]

A continuación se describen las 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_SISEC_010216_5 ArqSWVtaEntorno_v2.0”.

1.5 Tecnologías y Herramientas a utilizar

UML Versión 2.0

Lenguaje de modelado gráfico utilizado para visualizar, especificar, construir y documentar los artefactos que se generan en el proceso de modelado y construcción de un software. UML es un lenguaje para especificar y no para describir métodos o procesos, está compuesto por diversos elementos gráficos que

se combinan para formar diagramas, ofrece un estándar para describir de la manera más sencilla y entendible para cualquier desarrollador los aspectos conceptuales tales como procesos de negocios y funciones del sistema. [13]

Las ventajas que proporciona:

- ✓ Permite modelar sistemas.
- ✓ Permite especificar todas las decisiones de análisis, diseño e implementación, construyendo así modelos precisos, no ambiguos y completos.
- ✓ Permite documentar todos los artefactos de un proceso de desarrollo.
- ✓ La trazabilidad y documentación del proyecto se realiza de una forma ordenada y guiada por los casos de uso. [5]

Funciones:

- ✓ Visualizar: expresa de forma gráfica un sistema.
- ✓ Especificar: especifica cuáles son las características de un sistema antes de su construcción.
- ✓ Construir: a partir de los modelos especificados se puede construir los sistemas diseñados.
- ✓ Documentar: los elementos gráficos sirven como documentación del sistema desarrollado, sirviendo para su futura revisión. [14]

Visual Paradigm 8.0

Es la herramienta CASE⁵ profesional que soportará el ciclo de vida completo del desarrollo de software. Esta herramienta multiplataforma proporciona abundantes tutoriales de UML, garantiza una rápida construcción de aplicaciones de calidad a menor costo y permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Visual Paradigm 8.0 permite una mayor comunicación en equipos de trabajo, ayuda a determinar la complejidad del proyecto y esfuerzos necesarios. [15]

Características de Visual Paradigm:

⁵ Computer Aided Software Engineering

- ✓ Brinda un diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- ✓ Presenta un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Disponibilidad en múltiples plataformas y soporta el análisis y diseño orientado a objetos.[16]

PostgreSQL 9.2

Es un sistema gestor de base de datos relacional y multiplataforma destacado por su escalabilidad, robustez y cumplimiento de los estándares SQL. PostgreSQL se considera uno de los gestores de bases de datos de código abierto más potente del mercado, minimiza los errores utilizando multiprocesos, garantizando que si un proceso falla el sistema no se vea afectado. Es altamente extensible, soporta operadores, métodos de acceso y tipos de datos definidos por el usuario.

Soporta integridad referencial para garantizar la validez de los datos en la base de dato. Aproxima los datos a un modelo objeto-relacional, siendo capaz de manejar reglas y rutinas complejas, como son las consultas SQL declarativas, herencia, arrays y optimización de consultas. PostgreSQL posee una instalación ilimitada, estabilidad y confidencialidad legendaria, está diseñado para ambientes de alto volumen, soportando grandes cantidades de datos y una alta concurrencia. [17]

PgAdmin III 1.16.1

PgAdmin es una herramienta de código abierto multiplataforma que no requiere controles adicionales para la comunicación con el servidor de base de datos. Facilita la administración gracias a la compatibilidad de la interfaz gráfica con las características de PostgreSQL y está diseñado para responder a las necesidades de los usuarios, desde la escritura simple en consultas SQL hasta desarrollar bases de datos complejas. [18]

Incluye:

- ✓ Editor SQL con resaltado de sintaxis.
- ✓ Un editor de código de la parte del servidor. [19]

Java

Lenguaje de programación orientado a objetos desarrollado por la compañía Sun Microsystems. Es un lenguaje de plataforma independiente que permite el desarrollo de aplicaciones bajo el esquema Cliente Servidor. Aun cuando cuenta con una gran similitud con C y C++ elimina la complejidad de estos.

Java es multiplataforma, no solo a nivel de código fuente, sino a nivel de código compilado también, puede ejecutarse en cualquier sistema operativo que cuente con una máquina virtual compatible [20]. Se caracteriza por ser:

- ✓ Orientado a objetos: los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos.
- ✓ Indiferente a la arquitectura: está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos.
- ✓ Portable: la indiferencia a la arquitectura representa solo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas. Estas dos últimas características se conocen como la Máquina Virtual de Java.
- ✓ Interpretado y compilado a la vez: Java es compilado en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador. Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).
- ✓ Robusto: fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros), ya que se ha prescindido por completo de los punteros.
- ✓ Multihilo: soporta la sincronización de múltiples hilos de ejecución a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.
- ✓ Dinámico: el lenguaje y el sistema “run-time” son dinámicos en sus etapas de conexión. Las clases son conectadas solamente si son necesarias. Nuevos módulos de código pueden ser conectados en una demanda desde una variedad de fuentes, incluso de fuentes a lo largo de la red. [21]

Plataforma Java 2 Enterprise Edition (J2EE)

Java 2 Enterprise Edition es un estándar que define una plataforma de desarrollo empresarial basada en Java. Ofrece un modelo de aplicaciones distribuidas multicapa con componentes reusables, un modelo de seguridad unificado, un control flexible de transacciones y soporte para servicios web. Proporciona soluciones al diseño, desarrollo, montaje y despliegue de aplicaciones empresariales independientes del sistema operativo, en un entorno distribuido. [21]

Ventajas de J2EE sobre otras plataformas:

- ✓ Independencia de plataforma: las aplicaciones J2EE son portables. Si se cambia el sistema operativo de los servidores y/o el programa servidor de aplicaciones, se puede seguir utilizando sin modificaciones las aplicaciones J2EE desarrolladas.
- ✓ Objetos gestionados por los contenedores: el hecho de que los contenedores gestionen los objetos permite al desarrollador abstraerse de un conjunto de aspectos que requieren tiempo y dedicación. Como consecuencia, los desarrollos son más rápidos, con menos coste y se obtiene un código mucho menos propenso a fallos y de más fácil mantenimiento. Además las aplicaciones J2EE son declarativas, lo que quiere decir que se puede modificar su comportamiento sin modificar el código, lo cual contribuye también a un mantenimiento mucho más rápido, sencillo y de menor coste.
- ✓ Reusabilidad: Las aplicaciones J2EE están constituidas por la unión de componentes, como si se tratase de las piezas de un rompecabezas. Así se consiguen desarrollos más rápidos y de menor coste, y se proporciona mayor robustez a las aplicaciones.
- ✓ Modularidad: La modularidad facilita enormemente el mantenimiento de las aplicaciones, pues si se desean realizar modificaciones en algún módulo, éstas no deberían afectar al resto. Vistas estas ventajas, se puede entender mejor lo que se quiere decir con eso de que la idea principal tras J2EE es la de proveer un estándar simple y unificado para aplicaciones distribuidas a través de un modelo de aplicaciones basado en componentes. [22]

Swing

Swing es un conjunto de librerías que forman parte de la JFC⁶, las cuales implementan una serie de componentes utilizados en la construcción de interfaces gráficas.

Entre sus principales características se encuentran:

⁶ Java Foundation Classes

- ✓ Utiliza la arquitectura Modelo Vista Controlador, lo que proporciona un alto grado de extensibilidad y de personalización de los componentes visuales de la librería.
- ✓ Permite establecer estilos de interfaz llamados “look and feels” que modifican la forma en que se muestra y se comporta toda la interfaz de usuario.
- ✓ 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. [23]

Máquina Virtual de Java open-jdk 8.0

Una máquina virtual es un intérprete capaz de ejecutar instrucciones expresadas en un código binario especial (bytecode). La Máquina Virtual Java (JVM) es el entorno en el que se ejecutan los programas Java, su misión principal es la de garantizar la portabilidad de las aplicaciones Java.

Las tareas principales de la JVM son las siguientes:

- ✓ Reservar espacio en memoria para los objetos creados.
- ✓ Liberar la memoria no usada.
- ✓ Asignar variables a registros y pilas.
- ✓ Vigilar el cumplimiento de las normas de seguridad de las aplicaciones Java. [24]

Ejecuta el código resultante de la compilación del código fuente y se encarga de traducir el bytecode en instrucciones nativas de la plataforma destino. Esto permite que una misma aplicación Java, pueda ser ejecutada en una gran variedad de sistemas con arquitecturas distintas, siempre que posea una implementación adecuada de la máquina virtual de Java. [25]

Hibernate

Hibernate es una herramienta de ORM⁷ 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 de las entidades que permiten establecer estas relaciones.

Diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible. [26]

⁷ Mapeo Objeto-Relacional

Las características de esta tecnología son:

- ✓ Software libre.
- ✓ Soluciona el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el de orientación a objetos y el modelo relacional.
- ✓ Flexible en cuanto al esquema de tablas utilizado.
- ✓ Ofrece un lenguaje de consulta de datos llamado HQL⁸. [19]

NetBeans 7.4

NetBeans es un IDE⁹ para programadores, pensado para escribir, compilar, depurar y ejecutar programas. Está escrito en Java y contiene además un número importante de módulos para extenderlo; entre sus principales características se encuentran:

- ✓ Generación automática de código.
- ✓ Utilidades de búsqueda y reemplazo muy intuitivos y amigables.
- ✓ Coloración de código.
- ✓ Refactorización inteligente.
- ✓ Soporte para múltiples máquinas virtuales.
- ✓ Soporte para librerías de terceros. [27]

Subversion 1.4.5

Subversion es un software libre diseñado para el control de versiones. Es un repositorio en forma de árbol con una jerarquía de directorios y archivos, su objetivo es permitir editar de forma colaborativa y compartir información. Recuerda cada cambio que se haya realizado en el repositorio, así como los realizados a cada archivo y en el árbol de directorios. Gestiona cambios a través del tiempo, provee la habilidad de leer estados anteriores del sistema de archivos. Permite conocer los cambios realizados, cuándo se realizaron y quién los realizó. Se caracteriza por su transparencia al eliminar y cambiar nombres de archivos, independientemente del número de ramificaciones creadas mantiene un árbol diferencial de cambios, minimizando así el espacio consumido en el depósito. [28]

⁸ Hibernate Query Language

⁹ Integrated Development Environment

TortoiseSVN 1.7

Herramienta de código abierto desarrollado bajo la licencia GLP¹⁰. Cuenta con fácil acceso a los comandos de Subversion, debido a que están disponibles desde el menú contextual del explorador, donde añade su propio submenú. Indica el estado de cada carpeta y archivo versionado por pequeños íconos superpuestos, permitiendo ver fácilmente el estado en el que se encuentra su copia de trabajo.

Su interfaz gráfica de usuario permite que una vez listados los cambios realizados a un archivo o carpeta, pueda acceder a una revisión para ver los comentarios de esa confirmación. Puede también ver una lista de archivos modificados y el diálogo de confirmación lista todos los ítems que han de ser incluidos en una confirmación, donde cada ítem permite escoger qué cambios desea incluir. En caso que haya olvidado incluir algún archivo, los archivos que no están versionados pueden ser también listados. [29]

En el presente capítulo se describieron las definiciones y conceptos relacionados con la gestión de sujetos y cronogramas específicos en los EC, lo cual permitió un mejor entendimiento de los mismos. Se caracterizaron sistemas que gestionan EC, arribándose a la conclusión de que estos no presentan una solución al problema planteado. Se describieron las tecnologías y herramientas a utilizar para el desarrollo de las funcionalidades, permitiendo profundizar en las ventajas y características que estas ofrecen.

¹⁰ General Public License

Capítulo 2: Características de las funcionalidades

En el presente capítulo se realiza la propuesta de solución para darle respuesta a la problemática planteada. Se describen los procesos involucrados en el campo de acción, para un mejor entendimiento de los mismo. Se hace uso del modelo de dominio, con el objetivo de comprender y describir los conceptos más importantes referentes a la gestión de sujetos y cronogramas específicos en EC. Se presentan además los requisitos funcionales, los requisitos no funcionales y el diagrama de casos de uso del sistema junto con su descripción, persiguiendo un mejor entendimiento del sistema por los usuarios.

2.1 Propuesta de solución

Para darle solución a la problemática planteada, se propone desarrollar funcionalidades basadas en tecnologías de escritorio para el sistema SIGECConductor¹¹, que permitan la gestión de sujetos y cronogramas específicos en centros donde no se cuente con acceso al sistema Clínicas 1.0. Se utilizarán las tablas de la base de datos del sistema Clínicas 1.0 relacionadas con la gestión de sujetos y cronogramas específicos, en aras de lograr un diseño compatible con este; garantizando de esta forma la correcta sincronización entre las bases de datos de los sistemas Clínicas 1.0 y SIGECConductor a través del sistema SIGECSincronizador¹².

Se permitirá el acceso de los usuarios según los niveles establecidos de acuerdo a la función que realizan. Los usuarios que interactuarán con las funcionalidades son: el coordinador de la investigación clínica, responsable de generar el cronograma específico para cada sujeto y el investigador principal, encargado de firmar las hojas CRD. Para el encryptamiento de las contraseñas de los usuarios se empleará el algoritmo de reducción criptográfico MD5¹³ y se guardarán las trazas de todas las acciones llevadas a cabo en el sistema.

2.2 Flujo actual de los procesos involucrados en el campo de acción

Gestión de sujetos y cronograma específicos en centros sin acceso al sistema Clínicas 1.0

El sujeto se presenta en un centro especializado del país donde se realicen EC, para participar por decisión propia o por remisión de otro centro especializado. El coordinador de la investigación clínica procede a llenar un modelo con los datos personales del sujeto y desde ese momento comienza la etapa evaluación inicial del mismo. En la evaluación inicial se realizan un conjunto de pruebas al sujeto para ver si puede o no participar en el EC; este momento termina cuando son llenadas las hojas CRD con los datos obtenido.

¹¹ Sistema basado en tecnologías de escritorio para la conducción de ensayos clínicos

¹² Sistema basado en tecnologías de escritorio para la sincronización de los sistemas SIGECConductor y Clínicas 1.0

¹³ Message Digest Algorithm 5

CAPITULO 2. CARACTERISTICAS DE LAS FUNCIONALIDADES

Una vez completada la evaluación inicial, si el sujeto cumple con los parámetros para participar en el estudio, se le genera el cronograma específico a partir del cronograma general del estudio en el que se encuentre y pasa a la etapa de tratamiento. En la etapa de tratamiento se le efectúan exámenes programados en su cronograma y se recogen los datos de interés para el estudio. El tratamiento puede ser interrumpido por razones personales o por fallecimiento del sujeto, en términos clínicos se manejan como momentos de seguimientos no programados. Concluida la etapa de tratamiento satisfactoriamente, el sujeto entra a la etapa de seguimiento, en la que a través de exámenes se comprueba la efectividad del tratamiento.

2.3 Objeto de automatización

Para darle solución a la problemática, se propone desarrollar funcionalidades que permitan la gestión de sujetos y cronogramas específicos en los centros donde no se cuente con acceso al sistema Clínicas1.0. Por tanto se desea automatizar las actividades y documentos que se derivan de estos procesos.

2.4 Modelo de Dominio

El Modelo de Dominio es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Este representa los conceptos del dominio del problema y cómo se relacionan y puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema. [30]

A continuación se presenta el modelo de dominio del sistema en el que el estudio representa los estudios son llevados a cabo sobre sujetos bajo investigación clínica, en él participan el coordinador de investigación clínica y el investigador principal. El coordinador de la investigación clínica es el encargado de gestionar los sujetos en el estudio. Los sujetos cuentan con un cronograma específico y un estado de tratamiento que puede ser: Evaluación, Tratamiento, Seguimiento e Interrumpido. El cronograma específico es generado por el coordinador y cuenta con momentos de seguimientos programados y no programados, estos últimos son añadidos tanto por el investigador principal como por el coordinador de la investigación clínica. Los momentos de seguimientos están formados por hojas CRD, las cuales son firmadas por el investigador principal. Las hojas CRD y los momentos de seguimientos cuentan con un estado que puede ser: No Iniciada, Iniciada, Completada, Monitoreo Iniciado, Monitoreo Completado, Atrasado y Firmado.

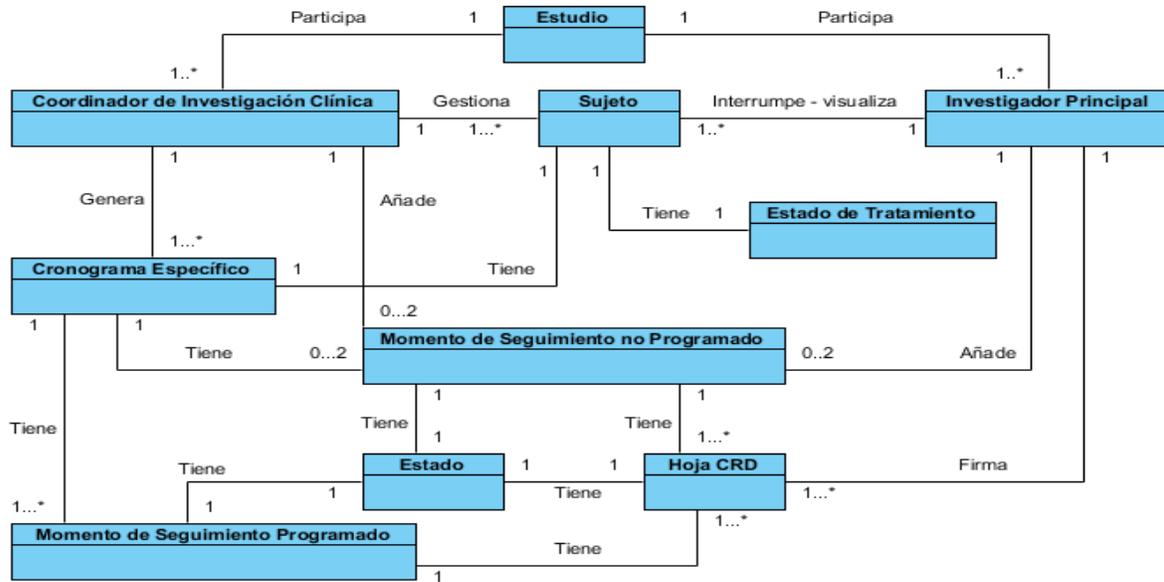


Figura.1 Modelo Conceptual

Descripción de clases

Sujeto: pacientes incluidos en los estudios.

Estado de Tratamiento: representa la etapa en que se encuentra el tratamiento del sujeto (Evaluación, Tratamiento, Seguimiento e Interrumpido).

Hoja CRD: recoge el conjunto de datos pertenecientes a un examen, definido por los conductores del estudio.

Momento de Seguimiento Programado: asociación de una o varias hojas CRD a un sujeto en una fecha, con un plazo de tiempo para su llenado.

Momento de Seguimiento no Programado: representa un suceso adverso en un sujeto, el cual tiene asociada una o varias hojas CRD.

Estado: representa el estado en que se encuentran los momentos de seguimiento tanto programados como no programados y las hojas CRD (No Iniciada, Iniciada, Completada, Monitoreo Iniciado, Monitoreo Completado, Atrasado y Firmado).

Cronograma Específico: representa la asociación de una planificación general de momentos de seguimiento a un sujeto específico. Contiene la fecha para la cual se planificó cada momento de seguimiento, la etapa a la que pertenece cada uno y el estado.

CAPITULO 2. CARACTERISTICAS DE LAS FUNCIONALIDADES

Coordinador de Investigación Clínica: representación del rol de coordinador de investigación clínica, el cual es el responsable de gestionar los sujetos y además de generar el cronograma específico de cada sujeto.

Investigador Principal: representación del rol Investigador principal, puede interrumpir y visualizar sujetos, añadir momentos de seguimiento no programados y además firmar las hojas CRD.

Estudio: representa los estudios que son llevados a cabo sobre sujetos bajo investigación clínica, en el cual participan uno o más coordinadores de investigación clínica y uno o más investigadores principales.

2.3 Especificación de los Requisitos

Requisitos Funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, estos describen lo que el sistema debe ser capaz de realizar para satisfacer las necesidades del usuario [31]. A continuación se muestra el listado de los requisitos funcionales identificados.

RF 1 Adicionar Momentos de Seguimiento No Programados

A partir de la fecha de inclusión de un sujeto y antes del último día definido en el Cronograma Específico se puede asignar un Momento de Seguimiento No Programado a cualquier sujeto, indicando como fecha de inicio la fecha actual.

RF 2 Adicionar Momento de Seguimiento de la etapa de evaluación al insertar un sujeto

Al insertar un sujeto se le agregan los Momentos de Seguimiento definidos para la Etapa de “Evaluación” en el Cronograma General a partir de la fecha de inclusión del sujeto.

RF 3 Autenticar usuario

Permitir la entrada de usuario y contraseña; verificando la validez de los mismos.

RF 4 Buscar momentos de seguimiento por sujeto

Permite realizar una búsqueda de momentos de seguimiento en una lista de todos los momentos de seguimiento de los sujetos que participan en el estudio.

RF 5 Buscar sujeto

Permite realizar una búsqueda de sujetos en una lista de todos los sujetos que participan en el estudio.

RF 6 Cambiar de centro

Cambiar de centro activo para otro donde el usuario autenticado tenga algún rol.

RF 7 Cambiar de estudio

CAPITULO 2. CARACTERISTICAS DE LAS FUNCIONALIDADES

Cambiar de estudio activo para otro donde el usuario autenticado tenga algún rol.

RF 8 Editar perfil

Modificar los datos del perfil de un usuario, incluyendo el cambio de contraseña.

RF 9 Firmar hoja CRD

Se le asigna el estado de “Firmado” a una hoja CRD.

RF 10 Generar fecha para cada Momento de Seguimiento programado

A partir de una fecha indicada por el coordinador, se le asigna una fecha a cada Momento de Seguimiento Programado de un sujeto, pertenecientes a las etapas de “Tratamiento” y “Seguimiento”, según el orden de la planificación de los mismos en el Cronograma General.

RF 11 Insertar un sujeto

Se añaden un sujeto al estudio activo.

RF 12 Imprimir datos de sujeto

Se imprimen los datos del sujeto seleccionado.

RF 13 Imprimir cronograma específico

Se imprimen el cronograma específico del sujeto seleccionado.

RF 14 Interrumpir sujeto

Se le establece el estado de tratamiento “Interrumpido” al sujeto y el sistema no obligará el llenado de los momentos posteriores al último Momento de Seguimiento Llenado.

RF 15 Mostrar datos del sujeto

Se muestra de un sujeto determinado todos sus datos, incluido el nombre del estudio al que pertenece.

RF 16 Visualizar listado de sujetos

Se muestra de todos los sujetos que están asociados a un estudio determinado, el nombre del sujeto, estado de inclusión, estado de tratamiento, próxima vista, última visita y las acciones asociadas a los mismos.

RF 17 Visualizar Momentos de Seguimientos programados

Se muestra de todos los Momentos de Seguimiento Programado definidos para un sujeto con fecha de inicio menor o igual a la fecha actual, el nombre, el estado, fecha de inicio, plazo de tiempo para el llenado, cantidad de Notas de Monitoreo con estado “Nueva” o “Actualizada” y las acciones asociadas a los mismos.

CAPITULO 2. CARACTERISTICAS DE LAS FUNCIONALIDADES

RF 18 Visualizar Momentos de Seguimiento no programados

Se muestra todos los Momentos de Seguimiento No Programado definidos para un sujeto con fecha de inicio menor o igual a la fecha actual, el nombre, el estado, fecha de inicio, cantidad de Notas de Monitoreo con estado “Nueva” o “Actualizada” y las acciones asociadas a los mismos.

RF 19 Visualizar cantidad de Momentos de Seguimiento por sujeto

Se muestra la cantidad de momentos de seguimiento programados asignados a cada sujeto del estudio activo.

RF 20 Visualizar cronograma específico de un sujeto

Se muestra de todos los Momentos de Seguimiento definidos para un sujeto, el nombre, el estado, fecha y etapa asociados a los mismos.

RF 21 Visualizar listado de hojas CRD

Se muestra de todas las hojas CRD asociadas a un Momento de Seguimiento, el nombre, el estado, la fecha de llenado, la cantidad de Notas de Monitoreo con estado “Nueva” o “Actualizada” y las acciones asociadas a las mismas.

Requisitos no Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable [31].

Existen múltiples categorías para clasificar a los requisitos no funcionales, a continuación se muestran los presentes en el sistema, descritos en el documento “SES_SICEC_010113_ERS_v1.0”.

RNF de Interfaz de usuario

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

RNF de Software

Estaciones de trabajo

CAPITULO 2. CARACTERISTICAS DE LAS FUNCIONALIDADES

- ✓ Para la instalación de la aplicación se debe disponer del sistema operativo Windows o GNU Linux.
- ✓ Debe estar instalada una máquina virtual de Java en su versión 8.0 o superior.

Servidor de Base de Datos

- ✓ Sistema operativo GNU Linux.
- ✓ Servidor de Base de Datos PostgreSQL 9.0 o superior.

RNF de Hardware

Estaciones de trabajo

- ✓ Procesador Celeron o PIII a 1.72 GHz o superior.
- ✓ Memoria RAM de 128 MB o superior.

Servidor de Base de Datos

- ✓ Procesador Intel(R) Xeon(R) 5140 Dual-Core o superior.
- ✓ Memoria RAM de 4 GB o superior.
- ✓ 1TB de espacio disponible como mínimo, debido a que el volumen de información es bastante grande y perdura en el tiempo hasta 15 años.

RNF de Seguridad

- ✓ Se garantizará el acceso de los usuarios al sistema según los niveles establecidos de acuerdo a la función que realizan. Cada usuario tendrá asignado uno o varios roles en el sistema.
- ✓ Se emplea el algoritmo de reducción criptográfico MD5 para el encryptamiento de las contraseñas de los usuarios.
- ✓ Las acciones llevadas a cabo en el sistema serán guardadas en el registro de trazas en la base de datos.
- ✓ Las contraseñas deberán tener más de 7 caracteres de longitud y tener una fortaleza media.
- ✓ Los usuarios estarán obligados a cambiar la contraseña cada 60 días como máximo.

RNF de Extensibilidad

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

2.4 Diagrama de Casos de Uso del Sistema

El diagrama de casos de uso del sistema describe las relaciones y dependencias entre un grupo de casos de uso y los actores participantes en el proceso, estos proporcionan una mejor comunicación con los futuros usuarios del sistema. [32]

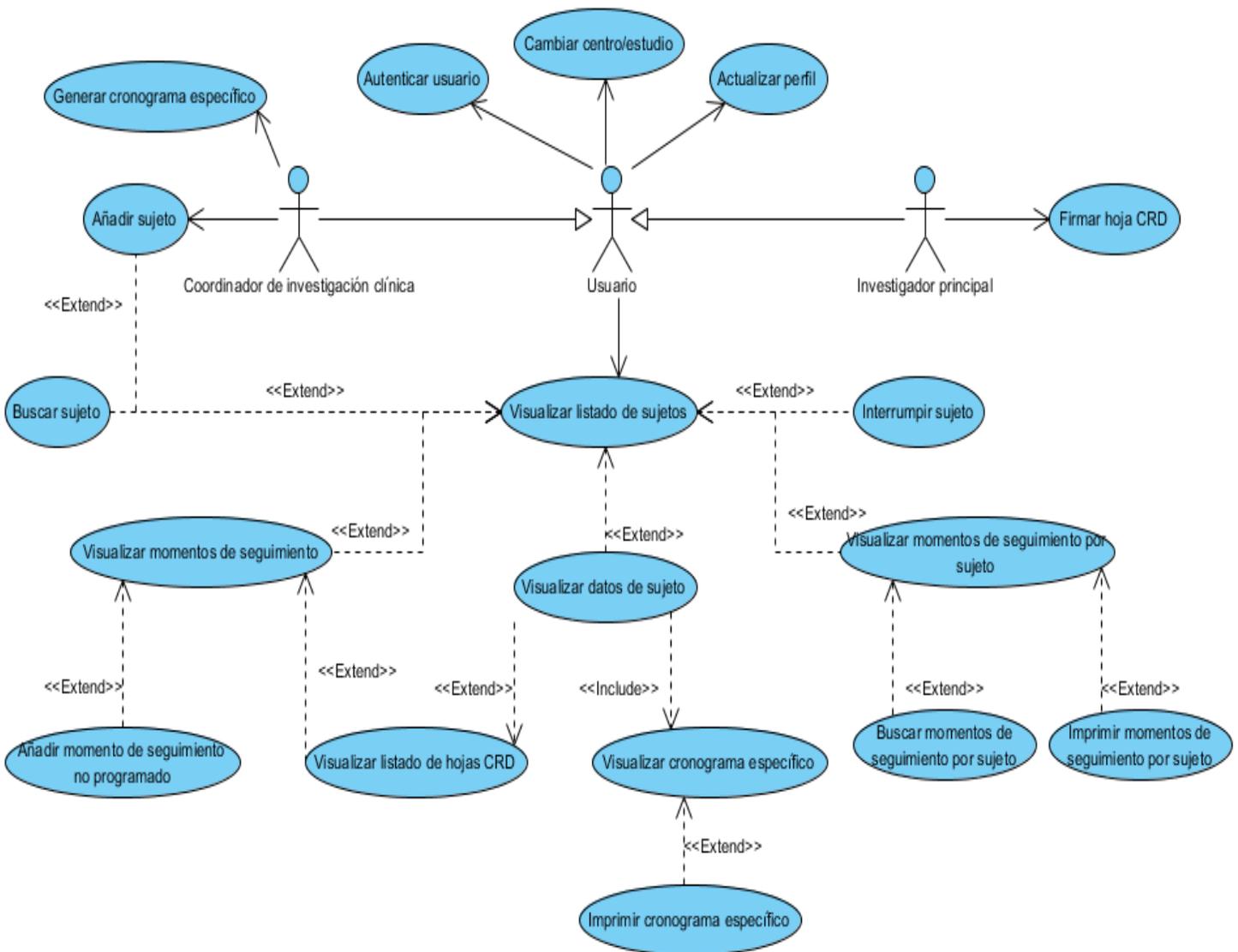


Figura.2 Diagrama de Casos de Uso del Sistema

2.5 Actores del Sistema

Actor	Descripción
Usuario	Interrumpe, busca y visualiza los sujetos del estudio, insertar los momentos de seguimiento no programados asociados a estos y visualiza los programados.
Coordinador de investigación clínica	Realiza las mismas acciones que el Usuario común del sistema y además añade nuevos sujetos y genera el cronograma específico.
Investigador Principal	Realiza las mismas acciones que el Usuario común del sistema y además firma las hojas CRD.

Tabla 2. Actores del sistema

2.6 Descripción de Casos de Uso del Sistema

A continuación se describen algunos casos de uso del sistema, detallando la secuencia de eventos que el actor lleva a cabo para completar un proceso determinado a través de la aplicación.

Caso de Uso Añadir sujeto

Objetivo	Añadir un sujeto.	
Actores	Coordinador de investigación clínica.	
Resumen	El caso de uso inicia cuando el Coordinador de investigación clínica accede a la opción Añadir un sujeto. El sistema muestra los campos a llenar y cuando estos han sido introducidos, el sistema añade un sujeto, el caso de uso termina.	
Complejidad	Media.	
Prioridad	Crítico.	
Precondiciones	El Coordinador de investigación clínica debe estar autenticado en el sistema.	
Postcondiciones	Se adiciona un nuevo sujeto en el estudio.	
Flujo de eventos		
Flujo básico Añadir un sujeto		
	Actor	Sistema
1.	Selecciona la opción añadir un sujeto en el estudio.	
2.		Muestra una interfaz con los datos que deben ser introducidos: <ul style="list-style-type: none"> • Identificador del sujeto en el estudio. • Identificador de la persona. • Fecha de inclusión.

CAPITULO 2. CARACTERISTICAS DE LAS FUNCIONALIDADES

		<ul style="list-style-type: none"> • Sexo. • Fecha de nacimiento.
3.	Introduce los datos requeridos.	
4.	Selecciona la opción salvar y finalizar.	
5.		Valida los datos. Si hay datos incorrectos, ver Evento N° 1 . “Los datos son incorrectos”.
6.		Añade el sujeto en el sistema y le asigna como estado de tratamiento “Evaluación”.
7.		Termina el caso de uso.
Flujos alternos		
N° 1 Los datos son incorrectos.		
	Actor	Sistema
1.		Muestra un mensaje de notificación “Datos incorrectos”. Ir al paso 2 del flujo normal de eventos.
Relaciones	CU incluidos	-
	CU extendidos	-
Requisitos no funcionales	Interfaz de usuario, Seguridad y Extensibilidad.	
Asuntos pendientes	No existen.	

Tabla 3. Descripción CU Añadir sujeto

Caso de Uso Interrumpir Sujeto

Objetivo	Interrumpir sujeto.	
Actores	Usuario.	
Resumen	El Usuario selecciona la opción interrumpir sujeto. El sistema muestra una interfaz con todos los datos del sujeto que se desea interrumpir y un campo obligatorio para seleccionar la fecha de interrupción, el usuario selecciona la fecha de interrupción, el sistema interrumpe el sujeto, termina el caso de uso.	
Complejidad	Media.	
Prioridad	Secundario.	
Precondiciones	El Usuario debe estar autenticado en el sistema y debe de existir al menos un sujeto en el estudio.	
Postcondiciones	Se interrumpe un sujeto en el estudio.	
Flujo de eventos		
Flujo básico Interrumpir sujeto		
	Actor	Sistema
1.	Selecciona la opción del sistema Interrumpir sujeto.	
2.		Muestra en una interfaz los datos del sujeto que se desea interrumpir y un campo para introducir la fecha de interrupción, esta es obligatoria.

CAPITULO 2. CARACTERISTICAS DE LAS FUNCIONALIDADES

3.	Inserta la fecha de interrupción.	
4.		Muestra un mensaje para confirmar la interrupción.
5.	Confirma que desea interrumpir el sujeto.	
6.		Valida los datos. Si hay datos incorrectos, ver Evento N° 1 . "Fecha incorrecta".
7.		Interrumpe el sujeto, asignándole como estado de tratamiento "Interrumpido".
8.		Termina el caso de uso.
Flujos alternos		
N° 1 Fecha incorrecta		
	Actor	Sistema
1.		Muestra un mensaje de notificación "Fecha incorrecta".
Relaciones	CU incluidos	-
	CU extendidos	-
Requisitos no funcionales	Interfaz de usuario, Seguridad y Extensibilidad.	
Asuntos pendientes	No existen.	

Tabla 4. Descripción CU Interrumpir sujeto

Caso de Uso Visualizar listado de sujetos

Objetivo	Visualizar todos los sujetos presentes en el estudio.	
Actores	Usuario.	
Resumen	El caso de uso inicia cuando el Usuario indica ver el listado de sujetos de un estudio. El sistema muestra el listado de sujetos con un grupo de datos de interés asociados a cada uno, termina el caso de uso.	
Complejidad	Media.	
Prioridad	Crítico.	
Precondiciones	El Usuario debe estar autenticado en el sistema y debe estar creado al menos un sujeto en el estudio activo.	
Postcondiciones	-	
Flujo de eventos		
Flujo básico Visualizar listado de sujetos		
	Actor	Sistema
1.	Indica gestionar sujetos en el menú Principal.	
2.		Busca todos los sujetos del estudio que está activo.
3.		Muestra el listado de sujetos especificando: <ul style="list-style-type: none"> • Identificador del sujeto en el estudio. • Estado de inclusión.

CAPITULO 2. CARACTERISTICAS DE LAS FUNCIONALIDADES

		<ul style="list-style-type: none"> Estado del tratamiento. Fecha del momento de seguimiento anterior. Fecha de próximo momento de seguimiento. <p>Además muestra las acciones que puede hacer el Usuario con cada uno de los pacientes.</p>
4.		Termina el caso de uso.
Relaciones	CU incluidos	-
	CU extendidos	<ul style="list-style-type: none"> Interrumpir sujeto en el caso de uso: Interrumpir sujeto. Añadir sujeto en el caso de uso: Añadir sujeto. Buscar sujeto en el caso de uso: Buscar sujeto. Visualizar momentos de seguimiento en el caso de uso: Visualizar momentos de seguimiento. Visualizar datos del sujeto en el caso de uso: Visualizar datos del sujeto. Visualizar momentos de seguimiento por sujetos en el caso de uso: Visualizar momentos de seguimiento por sujetos.
Requisitos no funcionales	Interfaz de usuario, Seguridad y Extensibilidad.	
Asuntos pendientes	No existen.	

Tabla 5. Descripción CU Visualizar listado de sujetos

Caso de Uso Visualizar datos de sujeto

Objetivo	Visualizar los datos de un sujeto.	
Actores	Usuario	
Resumen	El caso de uso inicia cuando el Usuario selecciona la opción "Ver sujeto". El sistema muestra una interfaz con todos los datos referentes al sujeto, termina el caso de uso.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	El Usuario debe estar autenticado en el sistema.	
Postcondiciones	-	
Flujo de eventos		
Flujo básico Visualizar datos del sujeto		
	Actor	Sistema
1.	Selecciona la opción "Ver sujeto".	
2.		Muestra una interfaz con todos los datos referentes al sujeto.
3.		Termina el caso de uso.
Relaciones	CU incluidos	Visualizar cronograma específico. <u>Ver CU Visualizar</u>

CAPITULO 2. CARACTERISTICAS DE LAS FUNCIONALIDADES

	<u>cronograma específico.</u>
	CU extendidos -
Requisitos no funcionales	Interfaz de usuario, Seguridad y Extensibilidad.
Asuntos pendientes	No existen.

Tabla 6. Descripción CU Visualizar datos de sujeto

Caso de Uso Visualizar momentos de seguimiento por sujeto

Objetivo	Visualizar momentos de seguimiento por sujeto.	
Actores	Usuario.	
Resumen	El caso de uso inicia cuando el Usuario selecciona la opción “Visualizar momentos de seguimiento por sujeto”. El sistema muestra una interfaz con todos los sujetos y todos los momentos de seguimiento asociados a ellos, termina el caso de uso.	
Complejidad	Alta.	
Prioridad	Secundario.	
Precondiciones	El Usuario debe estar autenticado en el sistema.	
Postcondiciones	-	
Flujo de eventos		
Flujo básico Visualizar momentos de seguimiento por sujeto		
	Actor	Sistema
1.	Selecciona la opción “Visualizar momentos de seguimiento por sujeto”.	
2.		Muestra una interfaz con todos los sujetos y todos los momentos de seguimiento asociados a ellos.
3.		Termina el caso de uso.
Relaciones	CU incluidos	-
	CU extendidos	- Buscar momentos de seguimiento por sujeto en el CU: Buscar momentos de seguimiento por sujeto. - Imprimir Buscar momentos de seguimiento por sujeto en el CU: Imprimir Buscar momentos de seguimiento por sujeto.
Requisitos no funcionales	Interfaz de usuario, Seguridad y Extensibilidad.	
Asuntos pendientes	No existen.	

Tabla 7. Descripción CU Visualizar momentos de seguimiento por sujeto

En el presente capítulo se realizó el modelado del dominio permitiendo un entendimiento de los conceptos relacionados con la gestión de sujetos y cronogramas específicos en los EC. Se modeló además el

CAPITULO 2. CARACTERISTICAS DE LAS FUNCIONALIDADES

Diagrama de Casos de Uso del Sistema describiendo los Actor y Casos de Uso que forman parte de este, posibilitando un mayor entendimiento de las características y restricciones que debe poseer el sistema para cumplir con los requisitos establecidos.

Capítulo 3: Análisis y Diseño de las funcionalidades

El presente capítulo ofrece los diagramas de clases del análisis, los diagramas de colaboración y los diagramas de clases del diseño generados en el flujo de trabajo análisis y diseño, a través de los cuales se traducen los requisitos a una especificación que describe cómo implementarlos. Se describen además los patrones de diseño y la arquitectura que será utilizada en el desarrollo de las funcionalidades, permitiendo establecer la estructura, funcionamiento e interacción de las partes del sistema.

3.1 Modelo de análisis

El modelo de análisis es una primera aproximación al diseño que ofrece una especificación más precisa de los requerimientos utilizando el lenguaje de los desarrolladores, ofrece una visión conceptual precisa y unificada de alternativas para la implementación de los sistemas. La estructura impuesta por el modelo de análisis se define mediante una jerarquía y su estructura se representa mediante un sistema de análisis que denota el paquete de más alto nivel del modelo. Este modelo crece incrementalmente conforme se analizan más casos de uso, de manera que el sistema se construye como una estructura de clases de análisis y relaciones entre dichas clases. [33]

Diagramas de Clases del Análisis

Los Diagramas de Clases del Análisis representa la abstracción de una o varias clases y subsistemas del diseño del sistema, en estos los casos de uso se describen mediante clases de análisis.

Existen tres estereotipos estandarizados en UML que ayudan a los desarrolladores a distinguir el ámbito de las diferentes clases [33]:

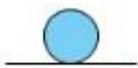
Representación	Nombre	Descripción
	Entidad (Entity)	Modela la información y comportamiento asociado de algún fenómeno o concepto. Se utiliza para modelar información que posee una vida larga y que a menudo es persistente.
	Control (Control)	Representa coordinación secuencial, transacciones y control de otros objetos, encapsula el control de un caso de uso en concreto. También se utiliza para representar derivaciones y cálculos complejos.
	Interfaz (Boundary)	Representa la abstracción de ventanas, formularios, paneles, interfaz de comunicación, interfaz de impresión, etc. Se utiliza para modificar la interacción entre el sistema y sus actores, lo que implica recibir y representar informaciones y peticiones de los usuarios y sistemas externos.

Tabla 8. Descripción de estereotipos

CAPITULO 3. ANALISIS Y DISEÑO DE LAS FUNCIONALIDADES

A continuación se muestran los diagramas de clases de análisis de los casos de uso Añadir sujeto, Interrumpir sujeto, Visualizar listado de sujetos, Visualizar datos de sujeto y Visualizar momentos de seguimiento por sujeto, en los que se muestran las clases que intervienen en estos y la relación existente entre ellas.

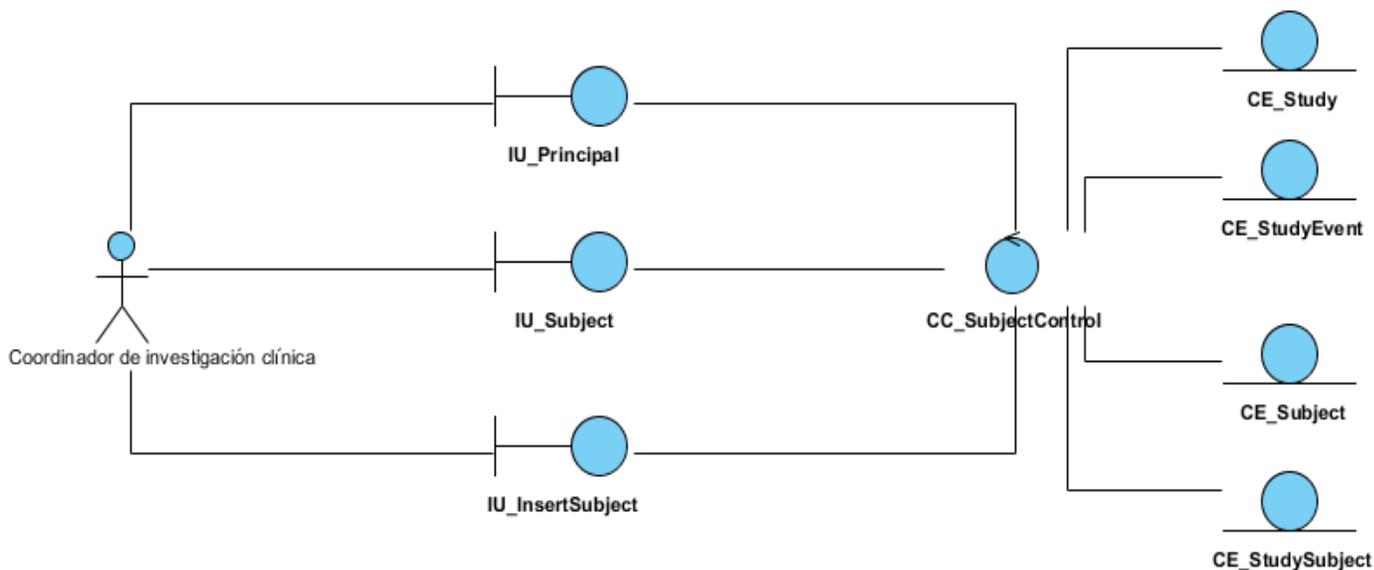


Figura.3 Diagrama de clase de análisis CU Añadir sujeto

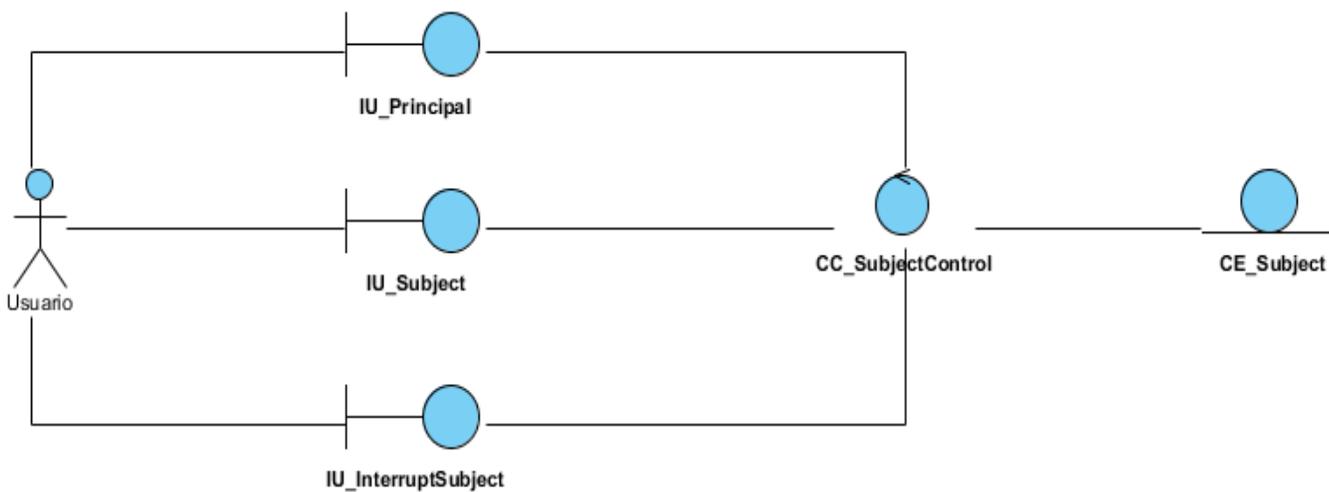


Figura.4 Diagrama de clase de análisis CU Interrumpir sujeto

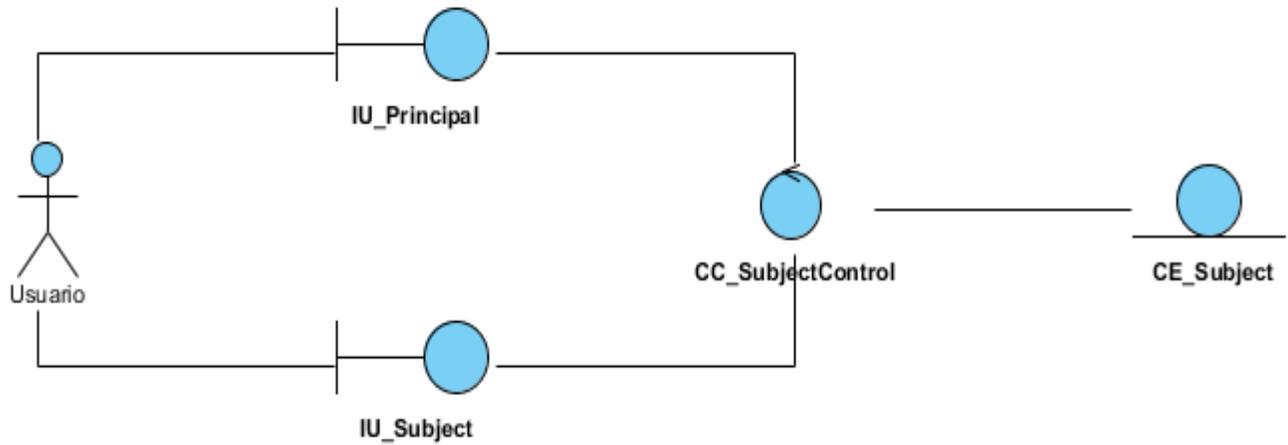


Figura.5 Diagrama de clase de análisis CU Visualizar listado de sujetos

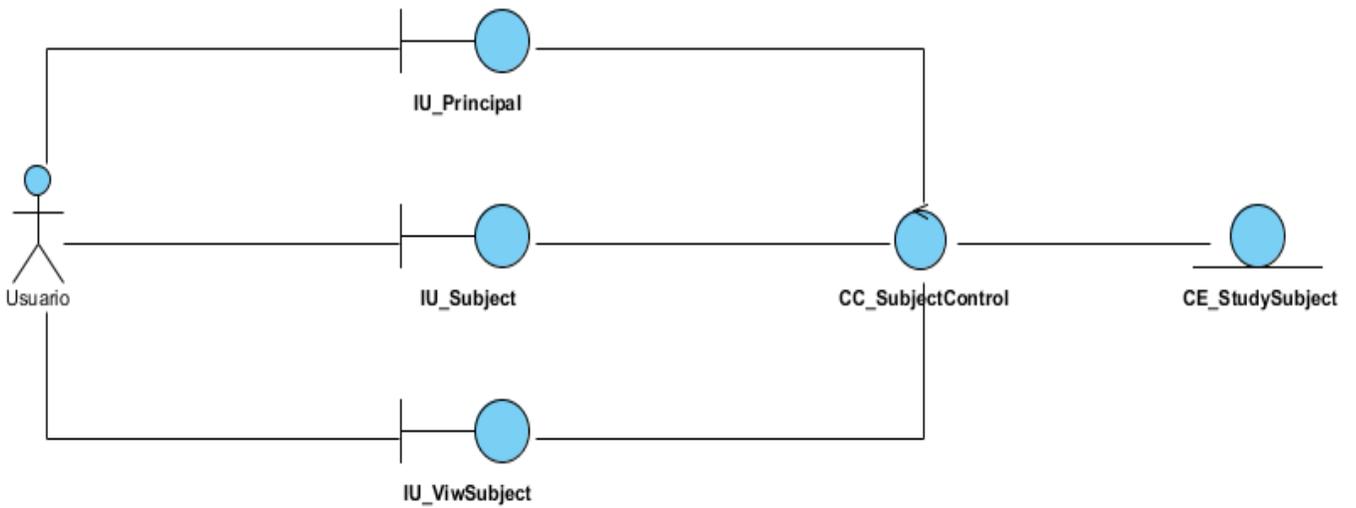


Figura.6 Diagrama de clase de análisis CU Visualizar datos de sujeto

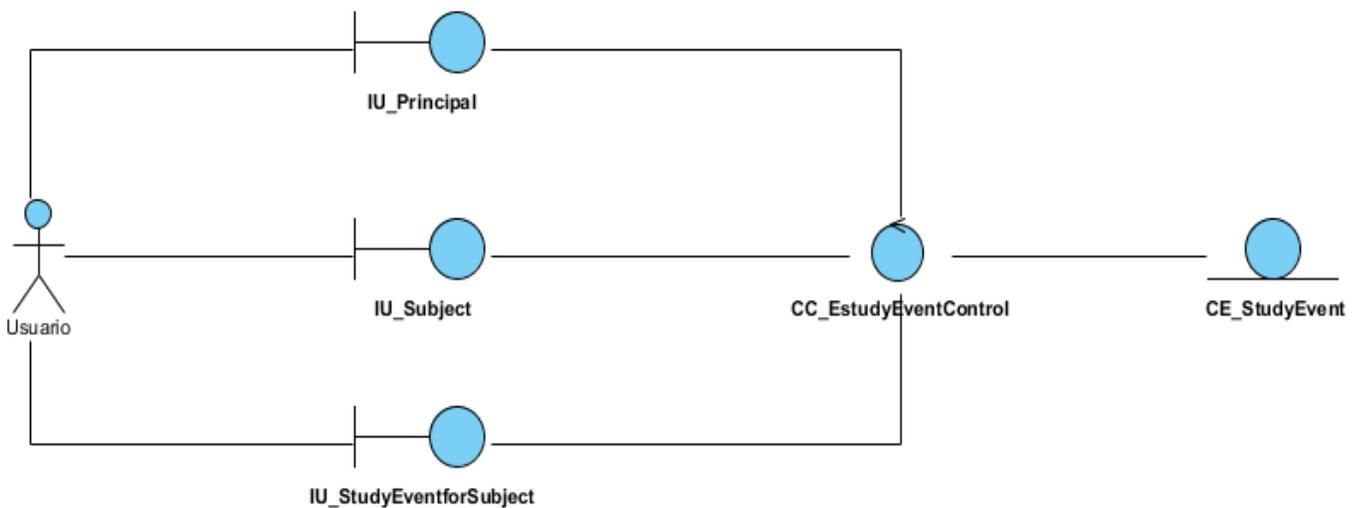


Figura.7 Diagrama de clase de análisis CU Visualizar momentos de seguimiento por sujeto

Diagramas de Colaboración

Los Diagramas de Colaboración se utilizan para modelar interacciones entre objetos, son similares a un diagrama de clases, pero muestran instancias y ligas en lugar de clases y asociaciones. Muestran cómo interactúan los objetos secuencialmente o en paralelo, numerando los mensajes intercambiados. Ilustran la vista dinámica del sistema. [34]

A continuación se muestran los diagramas de colaboración de los casos de uso Añadir sujeto, Interrumpir sujeto, Visualizar listado de sujetos, Visualizar datos de sujeto y Visualizar momentos de seguimiento por sujeto, en los que se muestra la interacción a través de mensajes entre las clases que intervienen en estos.

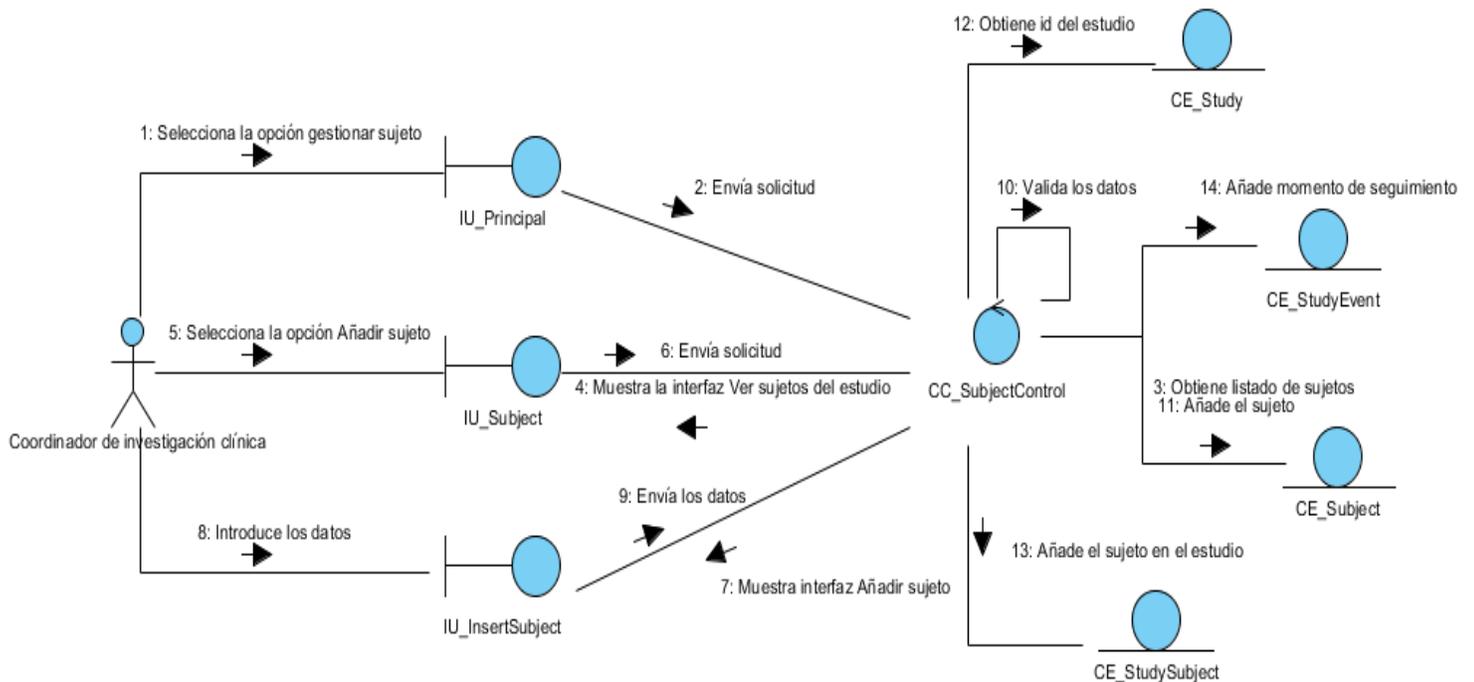


Figura.8 Diagrama de colaboración CU Añadir sujeto

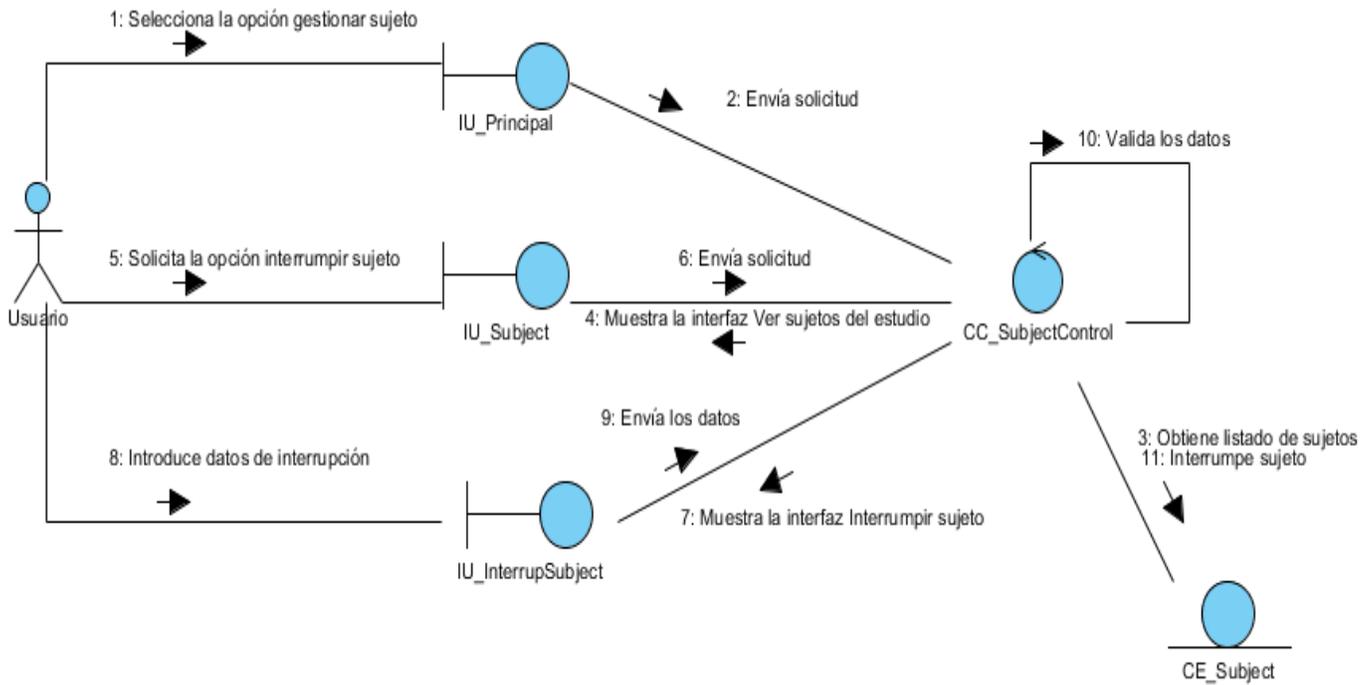


Figura.9 Diagrama de colaboración CU Interrumpir sujeto

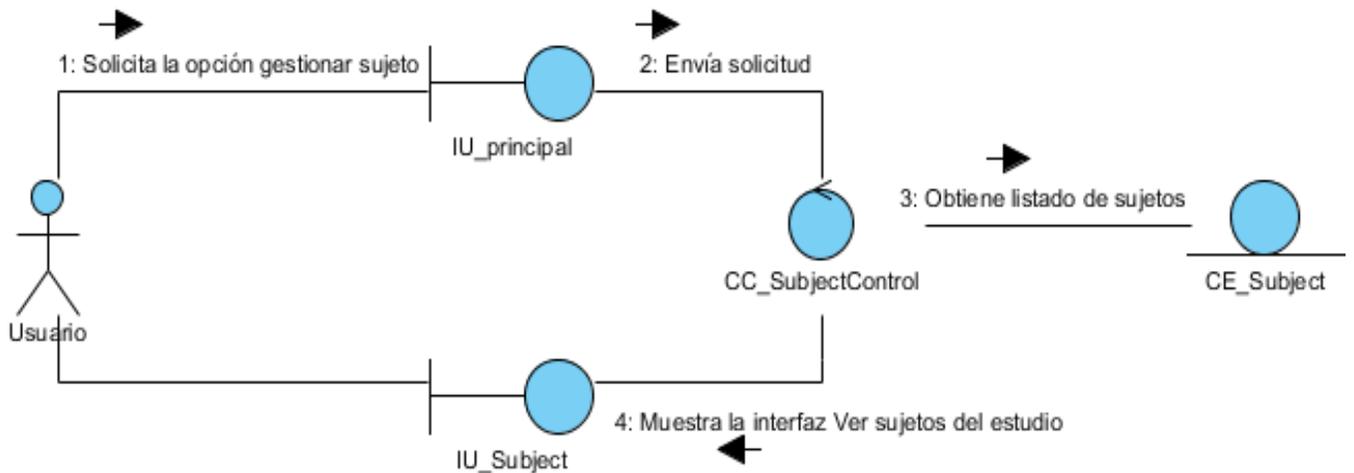


Figura.10 Diagrama de colaboración CU Visualizar listado de sujetos

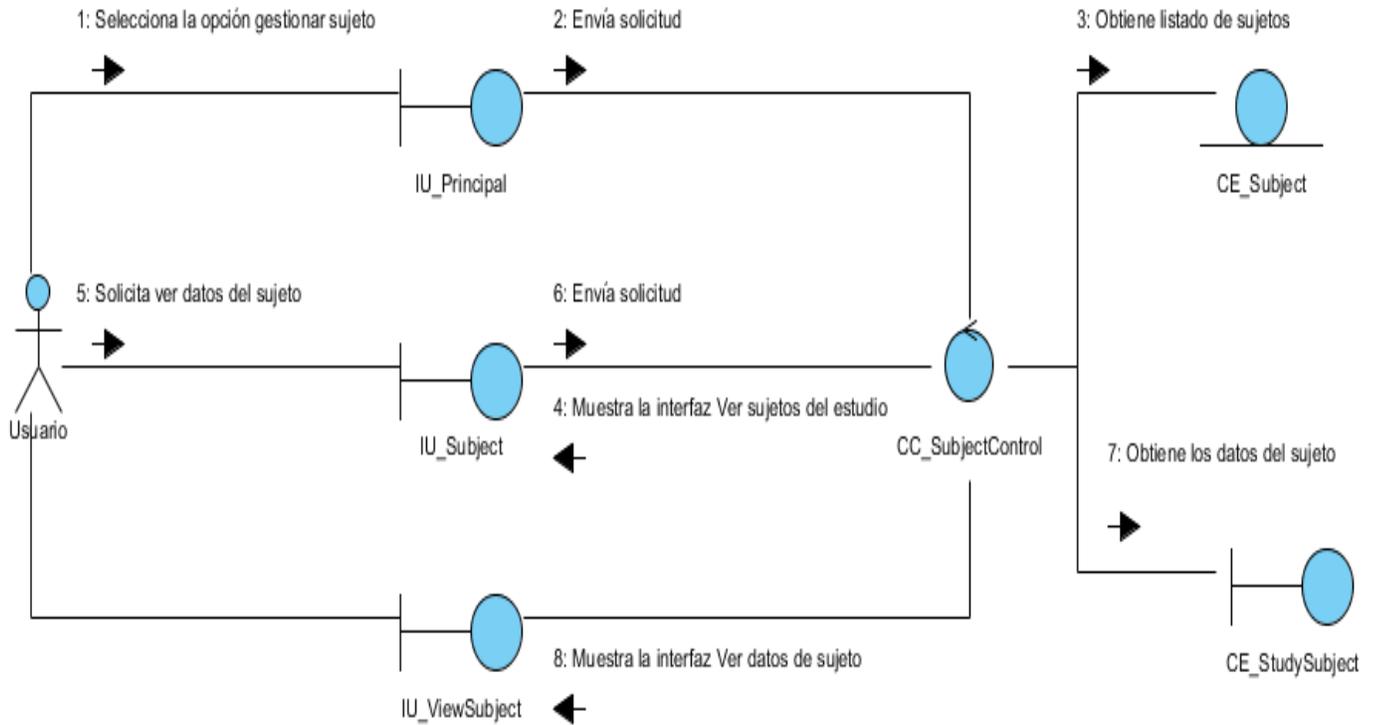


Figura.11 Diagrama de colaboración CU Visualizar datos de sujeto

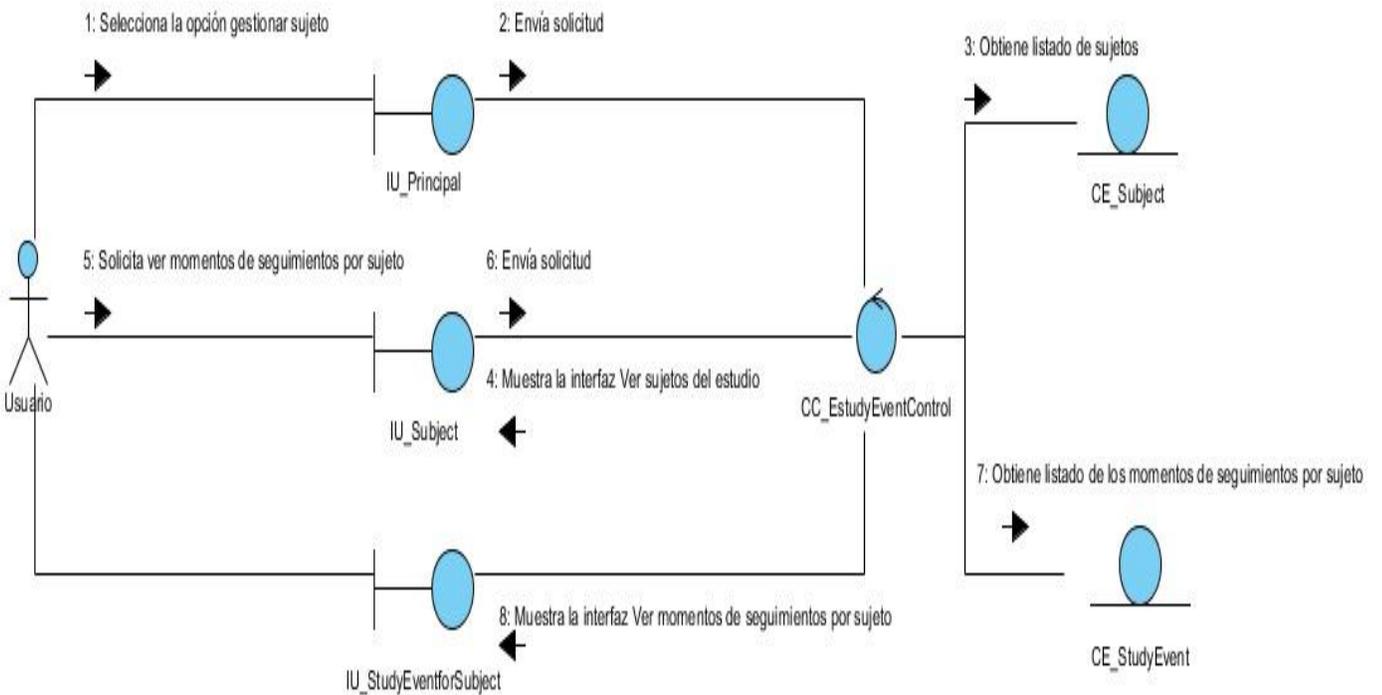


Figura.12 Diagrama de colaboración CU Visualizar momentos de seguimiento por sujeto

3.2 Modelo de diseño

Es una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas a implementación. Representa a los casos de uso en el dominio de la solución. [35]

Diagramas de Clases de Diseño

El Diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces (las de Java, por ejemplo) en una aplicación. A diferencia del modelo conceptual, un diagrama de este tipo contiene las definiciones de las entidades del software en vez de conceptos del mundo real. [34]

A continuación se presenta el diagrama de clases de diseño del caso de uso Añadir sujeto, en el que se agrupan las clases de acuerdo al patrón arquitectónico utilizado (Modelo Vista Controlador). En el paquete Vista se encuentra la interfaz de usuario InsertSubject, en el paquete Control se encuentran las clases SubjectControl, StudyControl, StudySubjectControl y StudyEventControl. Las clases controladoras se comunican con las clases DAO contenidas en el paquete Modelo las cuales permiten el acceso a los datos del mapeo de la base de datos, en este caso las clases Subject, Study, StudySubject y StudyEvent contenidas en el mismo paquete.

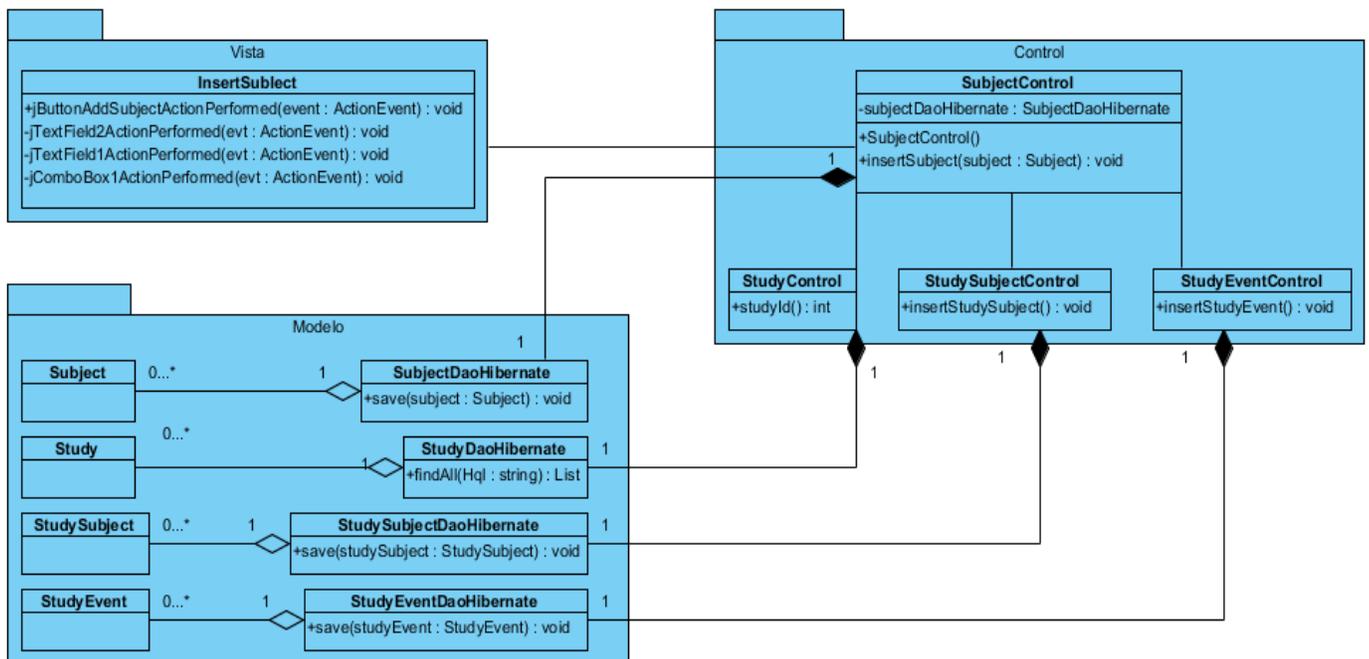


Figura.13 Diagrama de clase de diseño CU Añadir sujeto

A continuación se presenta el diagrama de clases de diseño del caso de uso Interrumpir sujeto, en el que se agrupan las clases de acuerdo al patrón arquitectónico utilizado. En el paquete Vista se encuentra la

CAPITULO 3. ANALISIS Y DISEÑO DE LAS FUNCIONALIDADES

interfaz de usuario InterrupSubject; en el paquete Control se encuentran las clases SubjectControl y StudyEventControl. Esta última se comunica con la clase StudyEventDaoHibernate contenida en el paquete Modelo la cual permite el acceso a los datos de la clase StudyEvent contenida en el mismo paquete.

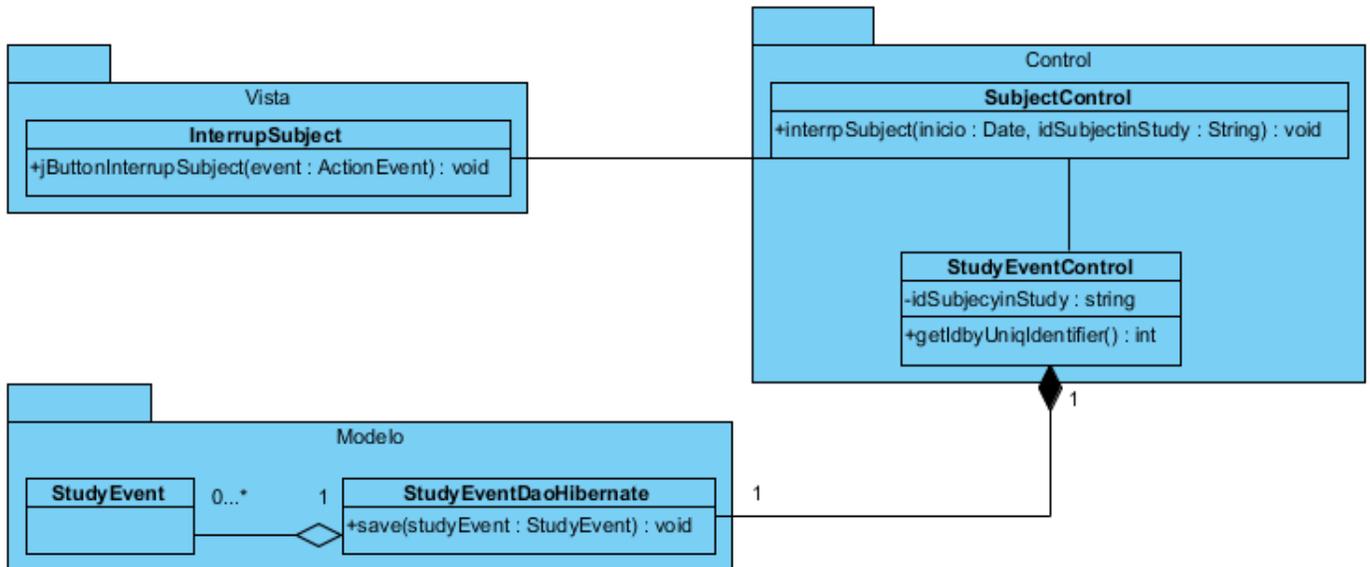


Figura.14 Diagrama de clase de diseño CU Interrumpir sujeto

A continuación se presenta el diagrama de clases de diseño del caso de uso Visualizar listado de sujetos, en el que se agrupan las clases de acuerdo al patrón arquitectónico utilizado. En el paquete Vista se encuentra la interfaz de usuario Subject; en el paquete Control se encuentra la clase SubjectControl, la cual se comunica con la clase SubjectDaoHibernate contenida en el paquete Modelo y permite el acceso a los datos de la clase Subject contenida en el mismo paquete.

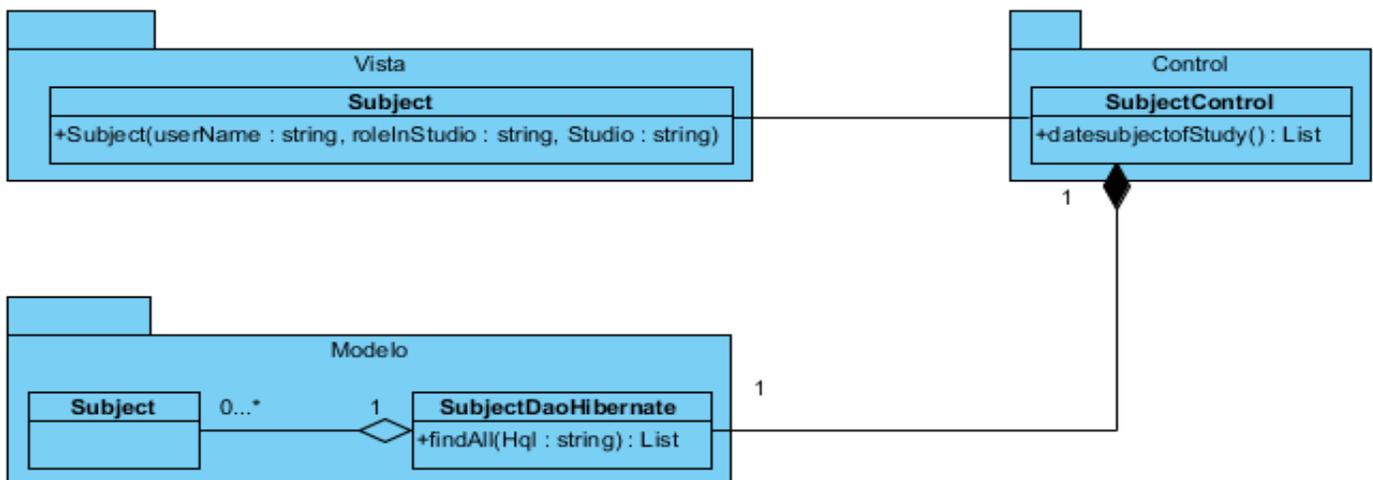


Figura.15 Diagrama de clase de diseño CU Visualizar listado de sujetos

CAPITULO 3. ANALISIS Y DISEÑO DE LAS FUNCIONALIDADES

A continuación se presenta el diagrama de clases de diseño del caso de uso Visualizar datos de sujeto, en el que se agrupan las clases de acuerdo al patrón arquitectónico utilizado. En el paquete Vista se encuentra la interfaz de usuario ViewSubject; en el paquete Control se encuentra la clase SubjectControl, la cual se comunica con la clase StudySubjectDaoHibernate contenida en el paquete Modelo y permite el acceso a los datos de la clase StudySubject contenida en el mismo paquete.



Figura.16 Diagrama de clase de diseño CU Visualizar datos de sujeto

A continuación se presenta el diagrama de clases de diseño del caso de uso Visualizar momentos de seguimientos por sujeto, en el que se agrupan las clases de acuerdo al patrón arquitectónico utilizado. En el paquete Vista se encuentra la interfaz de usuario StudyEventforSubject; en el paquete Control se encuentra la clase StudyEventControl, la cual se comunica con la clase StudyEventDaoHibernate contenida en el paquete Modelo y permite el acceso a los datos de la clase StudyEvent.

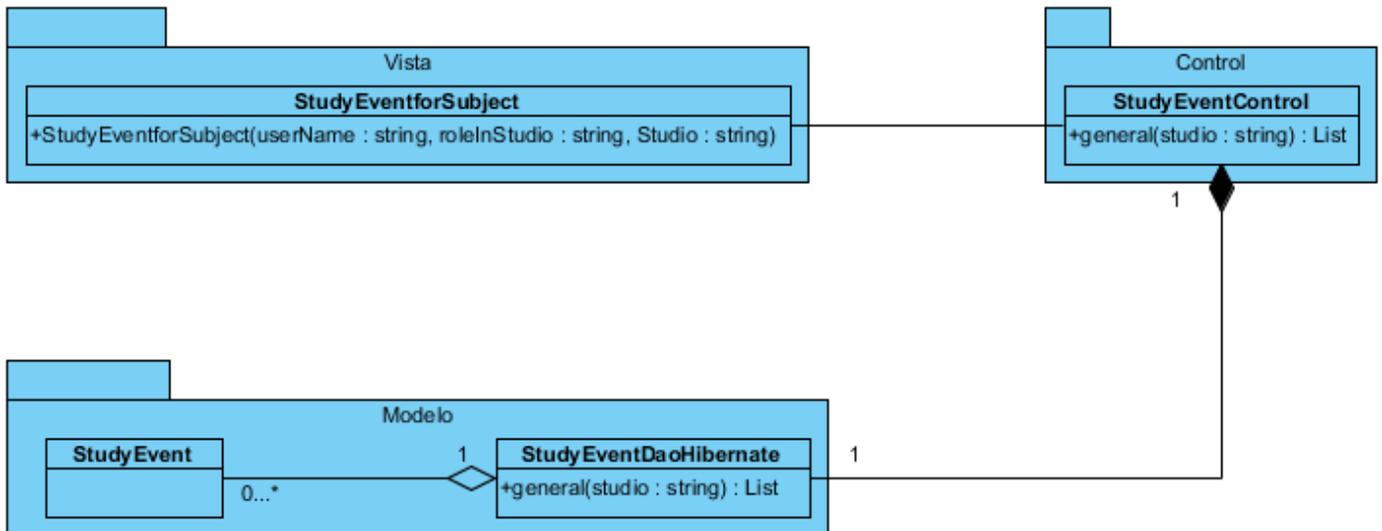


Figura.17 Diagrama de clase de diseño CU Visualizar momentos de seguimientos por sujeto

Descripción de las Clases del Diseño

Nombre	Tipo	Descripción
SubjectControl	Controladora	Contiene los métodos necesarios para la gestión de los sujetos.
StudyControl	Controladora	Contiene los métodos necesarios para obtener toda la información referente al estudio.
StudySubjectControl	Controladora	Contiene los métodos necesarios para obtener toda la información referente a los sujetos en el estudio.
StudyEventControl	Controladora	Contiene los métodos necesarios para la gestión de momentos de seguimiento.
SubjectDaoHibernate	Entidad	Contiene los métodos necesarios para gestionar la información de los sujetos en la base de datos.
StudyDaoHibernate	Entidad	Contiene los métodos necesarios para gestionar la información del estudio en la base de datos.
StudySubjectDaoHibernate	Entidad	Contiene los métodos necesarios para gestionar la información de los sujetos en un estudio en la base de datos.
StudyEventDaoHibernate	Entidad	Contiene los métodos necesarios para gestionar los momentos de seguimiento en la base de datos.
Subject	Entidad	Representación Orientada a Objetos (OO) de los datos de un sujeto, contiene variables y métodos

CAPITULO 3. ANALISIS Y DISEÑO DE LAS FUNCIONALIDADES

		para acceder o actualizar los mismos.
Study	Entidad	Representación OO de los datos de un estudio, contiene variables y métodos para acceder o actualizar los mismos.
StudySubject	Entidad	Representación OO de los datos de un sujeto en el estudio, contiene variables y métodos para acceder o actualizar los mismos.
StudyEvent	Entidad	Representación OO de los datos de los momentos de seguimiento, contiene variables y métodos para acceder o actualizar los mismos.

Tabla 9. Descripción de las clases del diseño

Descripción de la Arquitectura

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software.[36]

Patrón Arquitectónico: Modelo Vista Controlador (MVC)

Este patrón arquitectónico separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos: el Modelo, las Vistas y los Controladores.

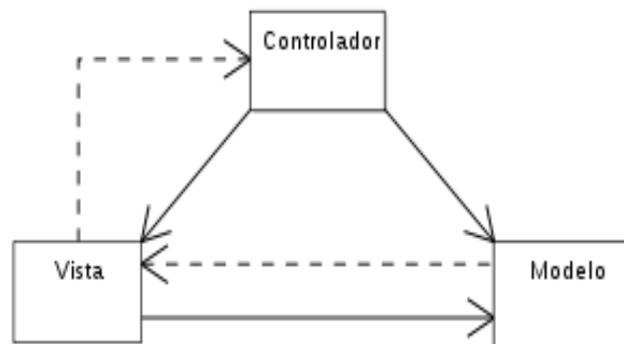


Figura.18 Patrón MVC

Descripción de los componentes:

- ✓ **Modelo:** es la representación específica de la información con la cual el sistema opera. Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- ✓ **Vista:** presenta el modelo en un formato adecuado para interactuar con el usuario, es responsable de la lógica de presentación y capturar de datos del sistema. Pueden existir múltiples vistas del modelo, cada una tiene asociado un componente controlador.
- ✓ **Controlador:** reciben las entradas como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista. Carga objetos y opera con ellos.

Ventajas del patrón MVC:

- ✓ Localización ágil de errores.
- ✓ Diseño modular.
- ✓ Facilidad para la realización de pruebas unitarias de los componentes.
- ✓ Reutilización de los componentes.
- ✓ Conexión entre el modelo y sus vistas dinámicas (se produce en tiempo de ejecución).
- ✓ Simplicidad en el mantenimiento de los sistemas. [37]

El uso de este patrón queda expuesto en los diagramas de clases del diseño contenidos en el acápite 3.2 y en los diagramas de componentes que se muestran más adelante en el presente documento. En cada diagrama se agrupan las clases o componentes de acuerdo a este patrón, lo cual permitió separar los datos de la aplicación, la interfaz de usuario y la lógica de control. A continuación se muestra las cuatro clases más significativas contenidas en cada capa.

Modelo

- ✓ **Subject:** Representación orientada a objeto de los datos de un sujeto, contiene variables y métodos para acceder o actualizar los mismos.
- ✓ **Study:** Representación orientada a objeto de los datos de un estudio, contiene variables y métodos para acceder o actualizar los mismos.

CAPITULO 3. ANALISIS Y DISEÑO DE LAS FUNCIONALIDADES

- ✓ StudySubject: Representación orientada a objeto de los datos de un sujeto en el estudio, contiene variables y métodos para acceder o actualizar los mismos.
- ✓ StudyEvent: Representación orientada a objeto de los datos de los momentos de seguimiento, contiene variables y métodos para acceder o actualizar los mismos.

Vista

- ✓ Subject: Muestra todos los sujetos que participan en el estudio.
- ✓ InsertSubject: Muestra los campos que deben ser llenados para insertar un sujeto en el estudio.
- ✓ InterrupSubject: Muestra los campos que deben ser llenados para interrumpir un sujeto en el estudio.
- ✓ StudyEventforSubject: Muestra todos los momentos de seguimiento de todos los sujetos que participan en el estudio.

Control

- ✓ SubjectControl: Contiene los métodos necesarios para la gestión de los sujetos.
- ✓ StudyControl: Contiene los métodos necesarios para obtener toda la información referente al estudio.
- ✓ StudySubjectControl: Contiene los métodos necesarios para obtener toda la información referente a los sujetos en el estudio.
- ✓ StudyEventControl: Contiene los métodos necesarios para la gestión de momentos de seguimiento.

Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. Brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares [38], estos pretenden:

- ✓ Proporcionar catálogos de elementos reusables en el diseño de software.
- ✓ Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- ✓ Formalizar un vocabulario común entre diseñadores.
- ✓ Estandarizar el modo en que se realiza el diseño.

CAPITULO 3. ANALISIS Y DISEÑO DE LAS FUNCIONALIDADES

- ✓ Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente. [39]

Para el diseño de las funcionalidades se utilizaron varios patrones de diseño como son:

DAO¹⁴

Este patrón utiliza un objeto de acceso a datos para abstraer y encapsular todos los accesos a la fuente de datos. Los Objetos de Acceso a Datos pueden usarse en Java para aislar a una aplicación de la tecnología de JPA¹⁵, la cual podría ser JDBC¹⁶, JDO¹⁷, EJB¹⁸, CMP¹⁹, TopLink²⁰, Hibernate²¹, iBATIS²², o cualquier otra tecnología de persistencia. Usar Objetos de Acceso de Datos permite que al cambiar o actualizar la tecnología subyacente, no sea necesario cambiar otras partes de la aplicación. [40]

El uso de este patrón queda expuesto en los diagramas de clases del diseño contenidos en el acápite 3.2 y en los diagramas de componentes que se muestran más adelante en el presente documento. En cada diagrama se incluye en el paquete Modelo un grupo de clases DAO, encargadas de gestionar la información de las tablas en la bases de datos; el siguiente ejemplo pertenece al Caso de Uso Interrumpir sujeto.



Figura.19 Uso del patrón DAO

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

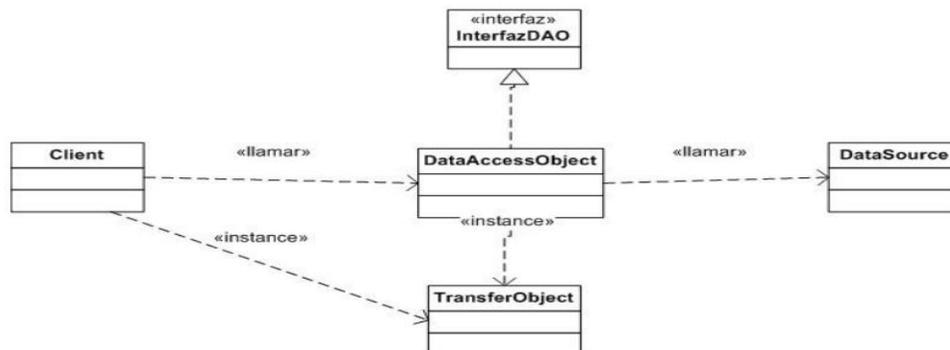


Figura.20 Estructura del DAO

¹⁴ Data Access Object

¹⁵ Java Persistence API (Interfaz de Programación de Aplicaciones)

¹⁶ Java Database Connectivity

¹⁷ Java Data Object

¹⁸ Enterprise Java Beans

¹⁹ Persistencia controlada por el Contenedor

²⁰ Paquete Java para mapeo objeto-relacional

²¹ Herramienta para mapeo objeto-relacional

²² Framework de persistencia para Java

Patrones GRASP²³

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. [41]

✓ Bajo acoplamiento

El acoplamiento mide qué tan fuerte se encuentra una clase conectada con otras. Una clase con bajo o débil acoplamiento no depende de muchas otras clases, por lo que en caso de producirse una modificación en alguna de estas, se tendrá la mínima repercusión posible en el resto de clases, potenciando la reutilización. Mantener bajo el acoplamiento entre clases, más que un patrón que se pueda implementar, es un principio que servirá para elegir entre alternativas de diseño [42]. Su uso se evidencia en todas las clases del sistema, un ejemplo es la clase StudyEventControl, la cual puede sufrir cambios sin que se vean afectadas el resto de las clases del sistema.

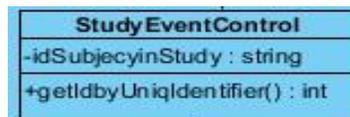


Figura.21 Uso del patrón Bajo acoplamiento

✓ Alta cohesión

En la perspectiva del diseño orientado a objetos, la cohesión funcional es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme [42]. Un ejemplo de su uso se evidencia en la clase SubjectControl contenida en el paquete Control, la cual tiene responsabilidades estrechamente relacionadas con otras clases como se muestra en la siguiente figura.

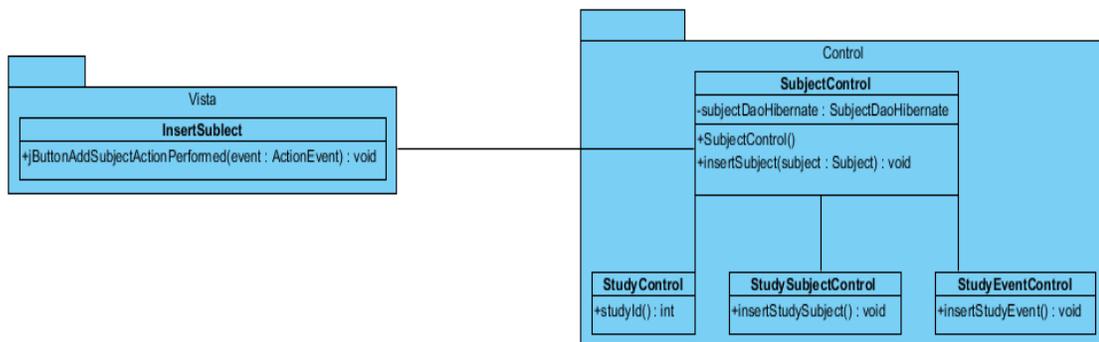


Figura.22 Uso del patrón Alta cohesión

²³ General Responsibility Assignment Software Patterns

✓ Experto

Asigna la responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad [42]. Este patrón se tiene en cuenta en todo el sistema, fundamentalmente en las clases contenidas en el paquete Modelo. Un ejemplo de su uso se evidencia en la clase Subject, la cual contiene la información de los sujetos en el sistema.

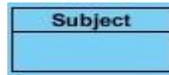


Figura.23 Comportamiento del patrón Experto

✓ Creador

Ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases, 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 [42]. Su uso se aprecia en la clase SubjectDaoHibernate contenida en el paquete Modelo, la cual es la responsable de la creación de objetos de la clase Subject contenida en el mismo paquete.

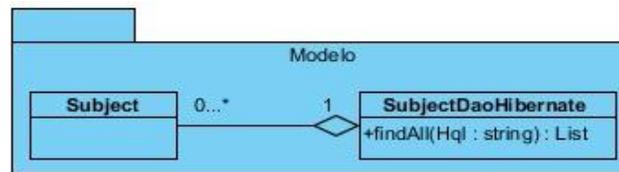


Figura.24 Comportamiento del patrón Creador

✓ Controlador

Asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión [42]. Un ejemplo de su uso se evidencia en la clase SubjectControl contenida en el paquete Control, la cual controla el flujo de procesos correspondiente a los sujetos en el sistema.

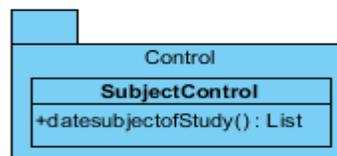


Figura.25 Comportamiento del patrón Controlador

Patrones GoF²⁴

Los patrones de diseño el grupo de GoF clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. Los patrones de comportamiento ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos. [43]

✓ Iterator

Patrón GoF de Comportamiento que se encarga de proveer una forma de acceso secuencial a los elementos de un objeto agregado sin tener que exponer su representación interna. El mismo define una interfaz mediante la cual se ofrece este acceso secuencial a un agregado, soportando múltiples recorridos de objetos agregados. En la implementación se usa mucho este patrón para recorrer colecciones de datos que contienen las tuplas que se obtienen de las consultas a la base de datos [44]. Un ejemplo de su uso se evidencia en la clase StudyEventDaoHibernate contenida en el paquete Modelo, encargada de buscar la información referente a los momentos de seguimiento almacenados en la base de datos. En ocasiones una petición devuelve como resultado varias tuplas y es a través del patrón Iterator que se asumen todas las filas en un arreglo.

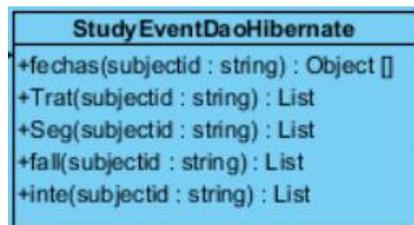


Figura.26 Comportamiento del patrón Iterator

En el presente capítulo se caracterizó la arquitectura a utilizar para el desarrollo de las funcionalidades, permitiendo conocer su estructura y funcionamiento. Se identificaron las principales clases del diseño y se elaboró el modelo de diseño; lo cual permitió a los desarrolladores entender mejor las funcionalidades requeridas por el cliente, garantizando la correcta elaboración de la fase de Implementación.

²⁴ Gang of Four

Capítulo 4: Implementación de las funcionalidades

El presente capítulo ofrece los diagramas de componentes, de despliegue del sistema y el modelo de datos que permitió una mejor estructuración de la información que se manejará en las funcionalidades. Además se aborda sobre el tratamiento de excepciones, la seguridad del sistema, así como las estrategias de codificación, estándares y estilos que se utilizan.

4.1 Modelo de implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. [45]

El Modelo de Implementación de las funcionalidades para la gestión de sujetos y cronogramas específicos, expone una organización en capas, jerarquías de paquetes y subsistemas de implementación. Los mismos contienen componentes y sus relaciones y divide el sistema en partes más manejables. Esto posibilita la reutilización, que se pueda implementar por separado y disminuye el impacto que pueda traer consigo un cambio.

Diagrama de componentes

Los diagramas de componentes muestran los elementos de diseño de un sistema de software. Un diagrama de componentes permite 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. [46]

A continuación se muestra el diagrama de componente para gestionar sujetos, realizado para dar paso a la implementación del sistema. Los componentes se agrupan en tres paquetes principales, siendo estos Vista, Control y Modelo, en correspondencia con el patrón de arquitectura utilizado. En la Vista se encuentran las interfaces de usuario las cuales utilizan las librerías swing 1.0.4 y jendelar 1.4 para los componentes visuales. En el paquete Control se encuentran las clases encargadas de la lógica del negocio en la gestión de sujetos y en el paquete Modelo se encuentran las clases DAO junto con sus propiedades y el mapeo de las tablas para la gestión de sujetos de la base de datos. En el paquete Base de datos se encuentra la base de datos.

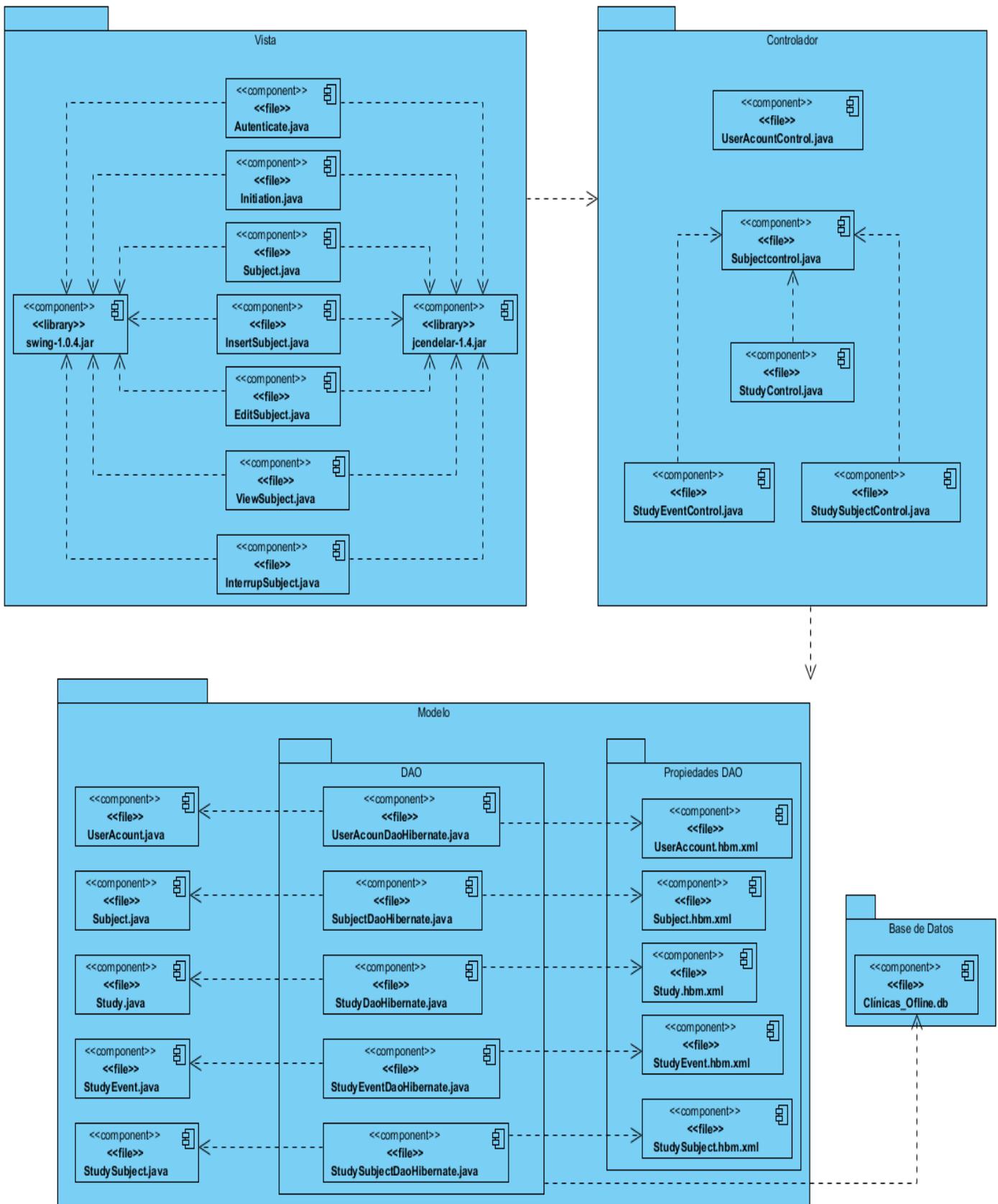


Figura.27 Diagrama de componentes "Gestionar sujeto"

Diagrama de despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un Server computador, un dispositivo o memoria. [47]

Para la estructura de la distribución física de las funcionalidades, 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 serán desplegadas las funcionalidades. La estación de trabajo se comunicará a través del protocolo TCP/IP²⁵ con el servidor de bases de datos y contará además con una impresora conectada por del puerto USB²⁶ o por el puerto LTP²⁷. A continuación se presenta el diagrama de despliegue propuesto para el sistema.

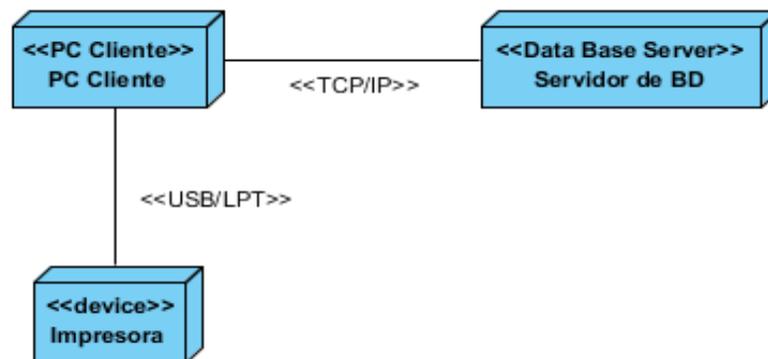


Figura.28 Diagrama de despliegue del sistema

4.2 Modelo de datos

Los modelos de datos son un conjunto de conceptos, reglas y convenciones que permiten describir y manipular los datos de la parcela de un cierto mundo real que se pueden almacenar en la base de datos. Comprenden aspectos relacionados con: estructuras y tipos de datos, operaciones y restricciones. [48]

A continuación se presenta el modelo de datos de las funcionalidades, el cual está compuesto por las 24 tablas utilizadas para la gestión de sujetos y cronogramas específicos del modelo de datos del sistema Clínicas 1.0. Entre las tablas más utilizadas se encuentran study y subject, las cuales contienen los datos referentes a los estudios y a los sujetos que participan en ellos respectivamente; de la relación entre estas dos tablas surge study_subject, la cual está relacionada con study_event que almacena los datos de los momentos de seguimientos en los estudios.

²⁵ Protocolo de Control de Transmisión/ Protocolo de Internet

²⁶ Bus Universal en Series

²⁷ Line Print Terminal

CAPITULO 4. IMPLEMENTACION DE LAS FUNCIONALIDADES

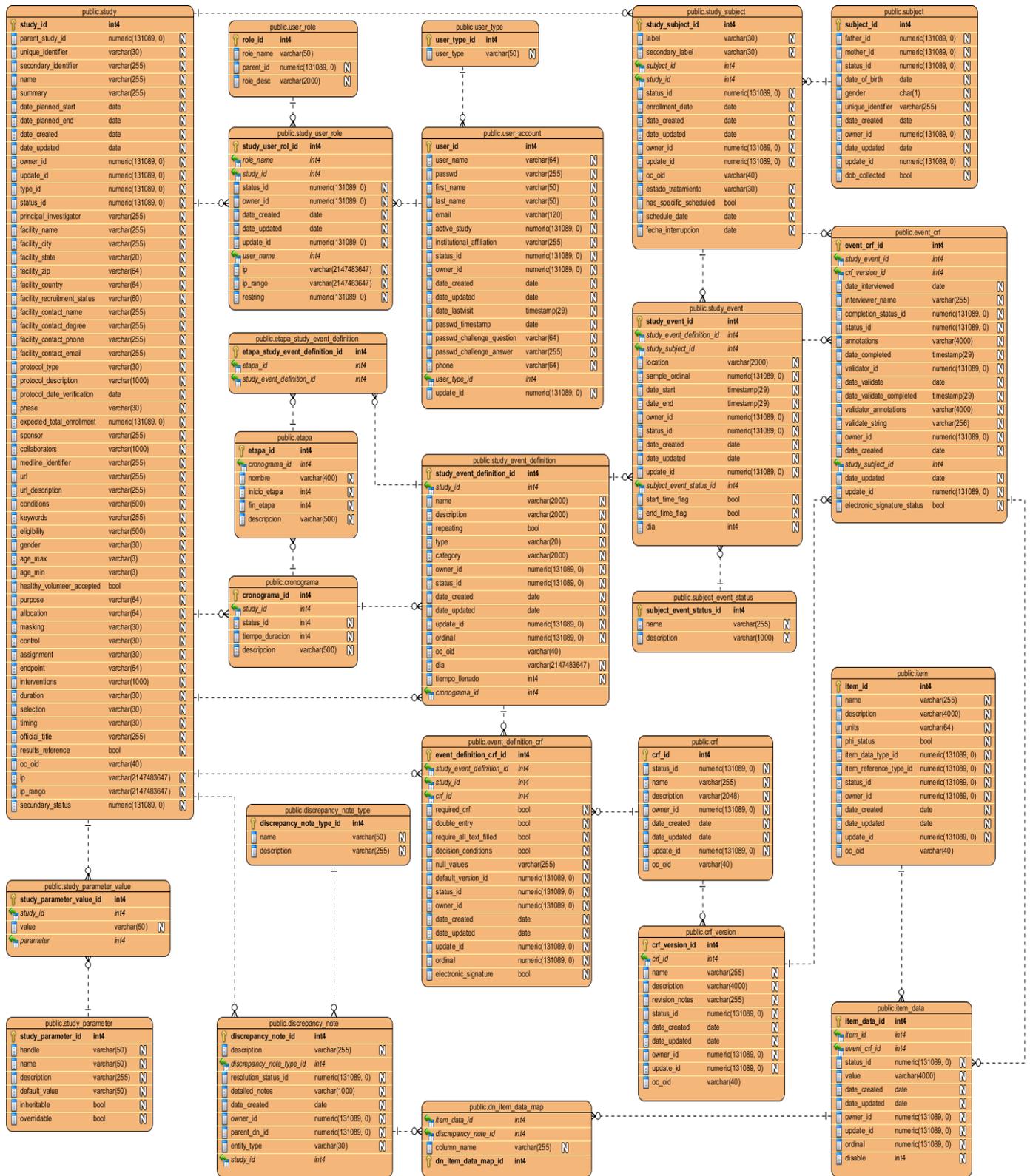


Figura.29 Modelo de datos de las funcionalidades

CAPITULO 4. IMPLEMENTACION DE LAS FUNCIONALIDADES

Descripción de las tablas del Modelo de Datos

A continuación se muestra para cada una de las entidades: su nombre, llave primaria, número de campos que contiene dicha entidad y una descripción de la misma.

Nº	Entidad	Llave Primaria	Nº de campos	Descripción
1.	crf	crf_id	9	Contiene atributos de las hojas CRD.
2.	crf_version	crf_version_id	11	Contiene la versión de cada hoja CRD.
3.	cronograma	cronograma_id	5	Contiene los atributos del cronograma general.
4.	discrepancy_note	discrepancy_note_id	10	Contiene los atributos de las notas de discrepancia.
5.	discrepancy_note_type	discrepancy_note_type_id	3	Contiene el tipo de cada nota de discrepancia.
6.	dn_item_data_map	dn_item_data_map_id	4	Nomenclador entre los atributos de las notas de discrepancia y los datos de dichas notas.
7.	etapa	etapa_id	6	Contiene los atributos de las etapas en que se encuentran los sujetos en el estudio.
8.	etapa_study_event_definition	etapa_study_event_definition_id	3	Contiene la etapa en que se encuentra el momento de seguimiento en el estudio.
9.	event_crf	event_crf_id	20	Contiene los atributos relacionados con la entrada de datos de una hoja CRD.
10.	event_definition_crf	event_definition_crf_id	17	Contiene los atributos asociados a las acciones que se realizan sobre una hoja CRD.
11.	item	item_id	13	Contiene los atributos para

CAPITULO 4. IMPLEMENTACION DE LAS FUNCIONALIDADES

				el guardado de los datos.
12.	item_data	item_data_id	11	Contiene los datos guardados.
13.	study	study_id	58	Contiene los atributos de los estudios.
14.	study_event	study_event_id	15	Contiene los momentos de seguimiento de los sujetos en los estudios.
15.	study_event_definition	study_event_definition_id	17	Contiene los atributos para la creación del cronograma específico de un sujeto a partir de un cronograma general.
16.	study_parameter	study_parameter_id	7	Contiene una especificación de cómo se mostrarán los componentes visuales en el sistema.
17.	study_parameter_value	study_parameter_value_id	4	Contiene el valor de los componentes visuales en el sistema.
18.	study_subject	study_subject_id	16	Contiene los atributos de los sujetos en los estudios.
19.	study_user_role	study_user_role_id	12	Contiene los atributos de los usuarios y sus respectivos roles en los estudios.
20.	subject	subject_id	12	Contiene los atributos de los sujetos.
21.	subject_event_status	subject_event_status_id	3	Contiene los estados que pueden adquirir los sujetos en el sistema.
22.	user_account	user_id	19	Contiene los atributos de los usuarios.
23.	user_role	role_id	4	Contiene los atributos de los roles.

24.	user_type	user_type_id	2	Contiene atributos relacionados con los tipos de usuarios
-----	-----------	--------------	---	---

Tabla 10. Descripción de las tablas del modelo de datos

4.3 Tratamiento de excepciones

Una excepción es un evento que ocurre durante la ejecución de un programa interrumpiendo el flujo normal de las sentencias. Dicho evento puede ser desde serios problemas de hardware, como la avería de un disco duro, hasta los simples errores de programación y pueden ser tratados mediante una estructura de control que poseen los lenguajes de programación de alto nivel, diseñada para manejar condiciones anormales que pueden ser tratadas por el mismo programa que se desarrolla. A esta estructura de control se le conoce como tratamiento de excepciones. [49]

Un adecuado tratamiento de las excepciones brinda robustez, fiabilidad y calidad al sistema. Este se llevó a cabo en todo el código del sistema, principalmente en aquellas regiones donde puedan surgir algunas situaciones inesperadas como por ejemplo donde se ejecutan sentencias que manipulan los valores de la base de datos. Se controlaron además los eventos adversos que pueden surgir en la validación de datos provenientes de la interfaz de usuario. Se utilizó el bloque try para detectar cuando ocurra algún fallo y el catch para manejar las excepciones, mediante mensajes que se muestran en la interfaz de usuario.

4.4 Seguridad e Integridad de la información

La seguridad informática es la disciplina que se ocupa de diseñar las normas, procedimientos, métodos y técnicas, orientados a proveer condiciones seguras y confiables, para el procedimiento de datos en sistemas informáticos. Para lograr sus objetivos, la seguridad informática se fundamenta en tres principios, que debe cumplir todo sistema informático:

- ✓ Confidencialidad: se refiere a la privacidad de los elementos de información almacenados y procesados en un sistema informático.
- ✓ Integridad: se refiere a la validez y consistencia de los elementos de información almacenados y procesados en un sistema informático.
- ✓ Disponibilidad: se refiere a la continuidad de acceso a los elementos de información almacenados y procesados en un sistema informático. [23]

La unión de estas características trae consigo que la información sea modificada y vista por el personal autorizado de forma controlable y que esté disponible cuando se necesite. A continuación se describen detalladamente las funcionalidades que garantizan la seguridad que ofrece el sistema.

Iniciar/Cerrar sesión de trabajo

Cuando el usuario desea acceder al sistema, deberá introducir su nombre de usuario y contraseña. El sistema verifica que los datos introducidos sean válidos, si es así, el usuario accede al mismo. El sistema muestra las funcionalidades a las que tiene permiso de acceder el usuario, lo que garantiza el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. El sistema brinda la opción de cerrar sesión.

Encryptación de datos

Se utilizó para el encryptamiento de las contraseñas de los usuarios MD5, algoritmo de reducción criptográfico diseñado en 1991 por el profesor Ronald Riverst del Instituto Tecnológico de Massachusetts.

Registrar trazas

Cuando el usuario realiza una acción sobre el sistema, que puede ser: inicio de sesión, cierre de sesión, modificación a un atributo de una entidad o cualquier otra operación sobre el sistema, este registra una traza en la base de datos en la tabla correspondiente a la entidad modificada.

4.5 Estrategias de codificación. Estándares y estilos a utilizar

Los estándares de codificación son un conjunto de reglas que se siguen para la escritura del código fuente, las cuales permiten obtener un código de alta calidad y cumplir con las buenas prácticas establecidas en la Ingeniería de Software. Posibilita además que el software que se obtiene sea fácil de comprender y de mantener en el tiempo . A continuación se describen los estándares de codificación que se utilizaron para el desarrollo del sistema.

Identación

El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador, ya que este puede variar según la computadora o la configuración de dicha tecla. Los inicios ({) y cierre (}) de ámbito deben de tener su propia línea y estar alineados debajo de la declaración a la que pertenecen, deben evitarse si hay sólo una instrucción.

Variables, parámetros, métodos, clases y objetos

El nombre empleado, debe permitir que con solamente leerlo se conozca su propósito, a continuación se describen las pautas que deben seguir:

CAPITULO 4. IMPLEMENTACION DE LAS FUNCIONALIDADES

- ✓ El nombre de las variables, métodos, objetos y parámetros debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto, se empleará notación CamellCasing.

Ejemplo: camellCasing.

- ✓ El nombre de las clases debe comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing.

Ejemplo: PascalCasing ().

Comentarios, separadores, líneas, espacios en blanco y márgenes

Para la utilización de comentarios, separadores, márgenes, líneas y espacios en blanco se deben seguir las siguientes pautas:

- ✓ Comentar al inicio de la clase o función su objetivo y el de los parámetros que usa.
- ✓ Antes y después de la declaración de una clase, una estructura o una función, dejar una línea en blanco.
- ✓ Se deben usar espacios en blanco entre los operadores para lograr una mayor legibilidad en el código.
Ejemplo: producto = nombreSujeto.
- ✓ Evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción.
- ✓ 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.

Bases de datos, tablas, esquemas y campos

El nombre de las bases de datos, las vistas, las tablas, los campos y los procedimientos almacenados, deben permitir que con sólo leerlos se conozca el propósito de los mismos. A continuación se describen las pautas que deben seguir:

CAPITULO 4. IMPLEMENTACION DE LAS FUNCIONALIDADES

- ✓ El nombre de las bases de datos deben comenzar con mayúscula, en caso de ser un nombre compuesto cada palabra debe comenzar con mayúscula separadas con underscore. Ejemplo: "Clinica_Offline".
- ✓ El nombre de las tablas y de los campos contenidos en ellas deben escribirse con todas las letras en minúscula, en caso de que sea un nombre compuesto se utilizará underscore para separarlas. Ejemplo: "subject_event_status".
- ✓ Todos los campos identificadores van a comenzar con el nombre del campo seguido de underscore y posteriormente el identificador id. Ejemplo: "subject_event_status_id".

En el presente capítulo se abordaron cuestiones pertenecientes a la etapa de implementación como el tratamiento de errores, el cual garantiza que las excepciones sean tratadas correctamente, brindándole robustez, fiabilidad y calidad al sistema. Los estándares y estilos de codificación, permitieron lograr la organización del código de las funcionalidades. Se obtuvieron los diagramas de componentes, los cuales garantizaron un mejor entendimiento y organización de los componentes en la implementación. Se obtuvo además la distribución física que se debe seguir para el despliegue del sistema y el modelo de datos el cual permitió describir la estructura de la base de datos.

Conclusiones

Con el desarrollo de la presente investigación se puede arribar a las siguientes conclusiones:

- ✓ Los sistemas analizados no ofrecieron una solución viable al problema planteado en la investigación, por lo que se propuso desarrollar funcionalidades que permitan la gestión de sujetos y cronogramas específicos en centros sin acceso al sistema Clínicas 1.0.
- ✓ La descripción de la metodología, las herramientas y las tecnologías definidas por el arquitecto principal del proyecto Ensayos Clínicos, reafirmó que las mismas brindan los beneficios necesarios para desarrollar las funcionalidades que permitan la gestión de sujetos y cronogramas específicos en centros sin acceso al sistema Clínicas 1.0.
- ✓ La elaboración de los artefactos correspondientes a las disciplinas: Modelado de Negocio, Requisitos, Análisis y Diseño e Implementación según la metodología RUP, permitió establecer la documentación necesaria de la investigación, sirviendo como guía al desarrollador para implementar la solución obtenida.
- ✓ El desarrollo de las funcionalidades para la gestión de sujetos y cronogramas específicos en centros sin acceso al sistema Clínicas 1.0, permitió darle solución al problema planteado.

Recomendaciones

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

1. Realizar una nueva versión de las funcionalidades para la gestión de sujetos y cronogramas específicos, que pueda ser usada en dispositivos móviles.
2. Incorporar las funcionalidades necesarias para que el sistema se capaz de:
 - ✓ Modificar el cronograma específico de los sujetos.
 - ✓ Modificar los datos de los sujetos que participan en el ensayo clínico.

Referencias Bibliográficas

1. CIMAB S.A.; Disponible en: <http://www.cimab-sa.com/index.php?action=cim>.
2. *Sociedad Española Farmacología Clínica*. Disponible en: <http://se-fc.org/gestor/ensayos-clinicos/informacion-general.html>.
3. *Institut Ferran de Reumatología*. 2013-12-10]; Disponible en: http://www.institutferran.org/ensayos_clinicos.htm.
4. Martínez Jera, Erislán y González Enríquez, Landy, *alasClínicas: Desarrollo de la Gestión de Datos de Ensayos Clínicos a partir del sistema OpenClinica.*, 2009. p. 131.
5. Cabalé, Dianne y Hernández, Alexey, *Desarrollo del módulo Enviar Datos del sistema alas Clínicas 2.0*, 2012, UCI. p. 143.
6. *BCX research |Software cuaderno de recogida de datos (CRD) | Médicos, investigadores , estudiantes y doctorandos |*. 2014-01-24]; Disponible en: http://www.bcxresearch.com/web/Glosario_de_terminos_es.asp.
7. *Medicina 21 - Notas de Prensa - Oracle lanza una aplicación móvil para investigación clínica*. 2014-02-21]; Disponible en: http://www.medicina21.com/Notas_De_Prensa/V5148-Oracle_lanza_una_aplicacion_movil_para_investigacion_clinica.html.
8. International, P.S. *EntrypointPlus*. 2.
9. *Biocapax Technologies S.L. | Software para ensayos clinicos en CRO´s, hospitales, laboratorios farmaceuticos y centros de investigación*. 2014-02-21]; Disponible en: http://www.biocapax.com/web/BCX_Clinical_es.asp.
10. *Apuntes. Ingenieria del software. Sistemas Informaticos. Nivel de madurez software. Informatica Aplicada a la Gestión Pública. 2005/06-2. Universidad de Murcia. Rafael Barzanallana*. 2014-02-20]; Disponible en: <http://www.um.es/docencia/barzana/IAGP/lagp2.html>.
11. Chesini, U., *Selección e implementación de prácticas de Diseño Centrado en el Usuario en un Proceso RUP* 2008. p. 95.
12. *Cualxpfddrup.pdf*. 2014-02-20]; Disponible en: <http://www.willydev.net/descargas/articulos/general/cualxpfddrup.pdf>.

13. *Base de conocimiento de Epidata : Introducción a UML 2.0.* 2014-02-20]; Disponible en: http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15.
14. *Tecnologías de la Informacion BD: CARACTERISTICAS Y FUNCIONES DEL UML.* 2014-02-16]; Disponible en: <http://luismanuelsarandingua.blogspot.com/2009/07/introduccion-es-un-lenguaje-grafico.html>.
15. *Visual Paradigm for UML Standard 8.0 - Free Download.* 2014-02-20]; Disponible en: <http://visual-paradigm-for-uml-standard.soft112.com/>.
16. Reyes Lamoth, Katileydis y López Sosa, Raúl, *Desarrollo de los módulos Administrar Sistema y Extraer Datos del sistema alas Clínicas V2.0*, 2012, UCI. p. 142.
17. Kasián, Fernando y Reyes, Nora, *Búsquedas por similitud en PostgreSQL*, 2012. p. 10.
18. *pgAdmin 1.12 Visual Tour.* 2014-02-20]; Disponible en: <http://www.pgadmin.org/visualtour12.php>.
19. Pérez Marrero, Dunia y Sarmiento Rosa, Henry, *Desarrollo de la Hoja de consulta de prótesis bucomaxilofacial y el módulo Laboratorio del sistema alas BMF*, 2012. p. 113.
20. *Principales Características de Java.* 2014-02-20]; Disponible en: <http://personales.upv.es/rmartin/cursoJava/Java/Introduccion/PrincipalesCaracteristicas.htm>.
21. Diaz, D.B., *Diseño del módulo Visualizar Información del sistema alas Clínicas v2.0*, 2012, UCI. p. 120.
22. Sosa Pérez, Mairim y Galiano Gamboa, Ernesto Rafael, *Desarrollo del módulo Gestionar Estudios del sistema alas Clínicas 2.0.*, 2012, UCI. p. 146.
23. Rodríguez Gómez, Zuleidy y Mirabal Cantillo, Hanssel, *Sistema para el levantamiento de inventario de equipos médicos por los especialistas del Centro de Ingeniería Clínica y Electromedicina.*, 2013, UCI. p. 73.
24. García, A.F. *La Máquina Virtual Java.* 2000 2014-03-15]; Disponible en: <http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/introduccion/virtual.htm>.
25. *JDK 7 Características.* 2014-02-20]; Disponible en: <http://openjdk.java.net/projects/jdk7/features/&usg=ALkJrhhrz0HgKRnA2bLy7VSKe2zsn1mQyA>.
26. *tutorialspoint.* 2014-02-05]; Disponible en: <http://www.tutorialspoint.com/hibernate/>.

27. *NetBeans IDE 7.2.1 Release Information*. 2014-02-20]; Disponible en: <https://netbeans.org/community/releases/72/>.
28. *subversion: Discussion topic*. 2014-02-20]; Disponible en: <http://subversion.tigris.org/ds/viewMessage.do?dsMessageId=917537&dsForumId=445>.
29. *Características de TortoiseSVN*. 20-2-2014]; Disponible en: http://tortoisesvn.net/docs/release/TortoiseSVN_es/tsvn-preface-features.html.
30. Larman, C., *Modelo Del Dominio*, in *UML y Patrones* 2003. p. 23.
31. *Tipos de requisitos: Funcional vs. No Funcional | Tecnología y Synergix*. 2014-03-08]; Disponible en: <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>.
32. Tello, J.C. *Diagramas de Casos de Uso*. 4.
33. *Guía para realizar el Modelo de Clases de Análisis*, 2008, Universidad Politécnica de Catalunya.
34. Larman, C., *UML y Patrones*. p. 507.
35. Venezuela, C.N.d.T.d.I. *MeRinde - Modelo de Diseño **. 2014-03-17]; Disponible en: http://merinde.net/index.php?option=com_content&task=view&id=92&Itemid=296.
36. *Arquitectura de software - EcuRed*. 2014-04-10]; Disponible en: http://www.ecured.cu/index.php/Arquitectura_de_software.
37. Catalani, E. *ARQUITECTURA Modelo/Vista/Controlador | Exequiel Catalani*. 2014-03-10]; Disponible en: <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>.
38. Tedeschi, N. *¿Qué es un Patrón de Diseño?*
39. Colombia, U.C.d., *INGENIERIA DE SOFTWARE: PATRONES DE DISEÑO Y FRAMEWORKS.*, in *Ingengería de Software II: Colombia*.
40. Mesa, Jose Luis, Ricardo Torres, Anrdes , Patricia Oviedo, Claudia y Andrea Tenorio, Jennifer. *Patron DAO*.
41. Gutierrez, J.A.S. *Patrones GRASP*. 2009 2014-04-07]; Disponible en: <http://jorgesaavedra.wordpress.com/about/>.
42. Visconti, Marcello y Astudillo, Hernán. *Patrones Grasp*.

43. Fire, A. *Patrones GoF*. 2014-04-07]; Disponible en: <http://geektheplanet.net/5462/patrones-gof.xhtml>.
44. Madrid, U.C.I.d., *Patrones de Diseño*, 2008: Madrid.
45. Hernandez, L. *MODELO DE IMPLEMENTACION*. 2013 2014-04-02]; Disponible en: <http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html>.
46. Network, M.D. *Diagrama de componentes de UML*. 2014-04-11]; Disponible en: <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
47. Marca Hualpara, Hugo Michael y Quisbert Limachi, Nancy Susana. *Diagrama de Despliegue*. 2014-04-10]; Disponible en: <http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc>.
48. Pantaleón, M.E.Z., *Modelos de datos*, 2010, Cantabria. p. 100.
49. Davesa, Arianna y Muñoz, Reinel, *Módulo Hospitalización del Sistema de Información Hospitalaria alas HIS*, 2009, UCI. p. 117.

Bibliografía

¿Qué es un ensayo clínico? - Instituto del Corazón de Texas (Texas Heart Institute). Disponible en: http://www.texasheart.org/hic/topics_esp/faq/clinical_trials_span.cfm.

Apuntes. Ingeniería del software. Sistemas Informaticos. Nivel de madurez software. Informatica Aplicada a la Gestión Pública. 2005/06-2. Universidad de Murcia. Rafael Barzanallana. 2014-02-20]; Disponible en: <http://www.um.es/docencia/barzana/IAGP/lagp2.html>.

Arquitectura de software - EcuRed. 2014-04-10]; Disponible en: http://www.ecured.cu/index.php/Arquitectura_de_software.

Base de conocimiento de Epidata : Introducción a UML 2.0. 2014-02-20]; Disponible en: http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15.

BCX research |Software cuaderno de recogida de datos (CRD) | Médicos, investigadores , estudiantes y doctorandos |. 2014-01-24]; Disponible en: http://www.bcxresearch.com/web/Glosario_de_terminos_es.asp.

Biocapax Technologies S.L. | Software para ensayos clinicos en CRO's, hospitales, laboratorios farmaceuticos y centros de investigación. 2014-02-21]; Disponible en: http://www.biocapax.com/web/BCX_Clinical_es.asp.

Cabalé, Dianne y Hernández, Alexey, *Desarrollo del módulo Enviar Datos del sistema alas Clínicas 2.0*, 2012, UCI. p. 143.

Características de TortoiseSVN. 20-2-2014]; Disponible en: http://tortoisesvn.net/docs/release/TortoiseSVN_es/tsvn-preface-features.html.

Catalani, E. *ARQUITECTURA Modelo/Vista/Controlador | Exequiel Catalani. 2014-03-10*]; Disponible en: <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>.

Chesini, U., *Selección e implementación de prácticas de Diseño Centrado en el Usuario en un Proceso RUP 2008*. p. 95.

CIMAB S.A.; Disponible en: <http://www.cimab-sa.com/index.php?action=cim>.

Colombia, U.C.d., *INGENIERIA DE SOFTWARE: PATRONES DE DISEÑO Y FRAMEWORKS.*, in *Ingeniería de Software II: Colombia*.

- Cualxpfdgrup.pdf*. 2014-02-20]; Disponible en: <http://www.willydev.net/descargas/articulos/general/cualxpfdgrup.pdf>.
- Davesa, Arianna y Muñoz, Reinel, *Módulo Hospitalización del Sistema de Información Hospitalaria alas HIS*, 2009, UCI. p. 117.
- Diaz, D.B., *Diseño del módulo Visualizar Información del sistema alas Clínicas v2.0*, 2012, UCI. p. 120.
- Fire, A. Patrones GoF. 2014-04-07]; Available from: <http://geektheplanet.net/5462/patrones-gof.xhtml>.
- García, A.F. La Máquina Virtual Java. 2000 2014-03-15]; Available from: <http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/introduccion/virtual.htm>.
- Guía para realizar el Modelo de Clases de Análisis*, 2008, Universidad Politécnica de Catalunya.
- Gutierrez, J.A.S. *Patrones GRASP*. 2009 2014-04-07]; Disponible en: <http://jorgesaavedra.wordpress.com/about/>.
- Hernandez, L. MODELO DE IMPLEMENTACION. 2013 2014-04-02]; Available from: <http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html>.
- Institut Ferran de Reumatología*. 2013-12-10]; Disponible en: http://www.institutferran.org/ensayos_clinicos.htm.
- International, P.S. *EntrypointPlus*. 2.
- JDK 7 Características*. 2014-02-20]; Disponible en: <http://openjdk.java.net/projects/jdk7/features/&usg=ALkJrhhrz0HgKRnA2bLy7VSKe2zsn1mQyA>.
- Kasián, Fernando y Reyes, Nora, *Búsquedas por similitud en PostgreSQL*, 2012. p. 10.
- Larman, C., *Modelo Del Dominio*, in *UML y Patrones* 2003. p. 23.
- Larman, C., *UML y Patrones*. p. 507.
- Madrid, U.C.I.d., *Patrones de Diseño*, 2008: Madrid.
- Marca Huallpara, Hugo Michael y Quisbert Limachi, Nancy Susana. *Diagrama de Despliegue*. 2014-04-10]; Disponible en: <http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc>.

Martínez Jera, Erilán y González Enríquez, Landy, *alasClínicas: Desarrollo de la Gestión de Datos de Ensayos Clínicos a partir del sistema OpenClinica.*, 2009. p. 131.

Medicina 21 - Notas de Prensa - Oracle lanza una aplicación móvil para investigación clínica. 2014-02-21]; Disponible en: http://www.medicina21.com/Notas_De_Prensa/V5148-Oracle_lanza_una_aplicacion_movil_para_investigacion_clinica.html.

medidata. 2014-02-10]; Disponible en: http://www.mdsol.com/products/rave_overview.htm.

Mesa, Jose Luis, Ricardo Torres, Anrdes , Patricia Oviedo, Claudia y Andrea Tenorio,Jennifer, *Patron DAO.*

NetBeans IDE 7.2.1 Release Information. 2014-02-20]; Disponible en: <https://netbeans.org/community/releases/72/>.

Network, M.D. *Diagrama de componentes de UML.* 2014-04-11]; Disponible en: <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.

Pantaleón, M.E.Z., *Modelos de datos*, 2010, Cantabria. p. 100.

Paumier, S.C.V.S.S., *SACCEM: Módulo de Gestión para el Departamento de Radiofísica del CCEEM (versión 2.0)*, 2010, UCI. p. 67.

Pérez Marrero, D.y.S.R., Henry Desarrollo de la Hoja de consulta de prótesis bucomaxilofacial y el módulo Laboratorio del sistema alas BMF, 2012. p. 113.

pgAdmin 1.12 Visual Tour. 2014-02-20]; Disponible en: <http://www.pgadmin.org/visualltour12.php>.

Principales Características de Java. 2014-02-20]; Disponible en: <http://personales.upv.es/rmartin/cursoJava/Java/Introduccion/PrincipalesCaracteristicas.htm>.

Reyes Lamoth, Katileydis y López Sosa, Raúl, *Desarrollo de los módulos Administrar Sistema y Extraer Datos del sistema alas Clínicas V2.0*, 2012, UCI. p. 142.

Rodríguez García, Lucía, Hernández Ramírez, Martha Denia, Díaz Pompa, Richard, Martínez Jera, Erilán, Hernández Cuello, Leandro y González Enríquez, Landy. *“ALASCLÍNICAS”: SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS “ALASCLÍNICAS”: CLINICAL TRIAL MANAGEMENT SYSTEM.* 2011. 18.

Rodríguez Gómez, Zuleidy y Mirabal Cantillo, Hanssel, *Sistema para el levantamiento de inventario de equipos médicos por los especialistas del Centro de Ingeniería Clínica y Electromedicina.*, 2013, UCI. p. 73.

Santos Valdés, Yurien y Laffita Hernández, Denlis, *alasClínicas: Desarrollo de funcionalidades para el monitoreo de los datos a partir del Sistema OpenClinica*, 2009, UCI. p. 128.

Sociedad Española Farmacología Clínica. Disponible en: <http://se-fc.org/gestor/ensayos-clinicos/informacion-general.html>.

Sosa Pérez, M.y.G.G., Ernesto Rafael, *Desarrollo del módulo Gestionar Estudios del sistema alas Clínicas 2.0.*, 2012, UCI. p. 146.

Strategies, Q.-Q.E. *CMMI: Modelo Integrado de Capacidad y Madurez*. 2009-2014 2014-02-15]; Disponible en: <http://www.qescalidad.com/cmml.htm>.

subversion: Discussion topic. 2014-02-20]; Disponible en: <http://subversion.tigris.org/ds/viewMessage.do?dsMessageId=917537&dsForumId=445>.

Tecnologías de la Información BD: CARACTERISTICAS Y FUNCIONES DEL UML. 2014-02-16]; Disponible en: <http://luismanuelarandina.blogspot.com/2009/07/introduccion-es-un-lenguaje-grafico.html>.

Tedeschi, N. *¿Qué es un Patrón de Diseño?*

Tello, J.C. *Diagramas de Casos de Uso*. 4.

Tipos de requisitos: Funcional vs. No Funcional | Tecnología y Synergix. 2014-03-08]; Disponible en: <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>.

tutorialspoint. 2014-02-05]; Available from: <http://www.tutorialspoint.com/hibernate/>.

Venezuela, C.N.d.T.d.I. *MeRinde - Modelo de Diseño **. 2014-03-17]; Disponible en: http://merinde.net/index.php?option=com_content&task=view&id=92&Itemid=296.

Visconti, Marcello y Astudillo, Hernán. *Patrones Grasp*.

Visual Paradigm for UML Standard 8.0 - Free Download. 2014-02-20]; Disponible en: <http://visual-paradigm-for-uml-standard.soft112.com/>.

Anexos

Anexo 1: Entrevista

Objetivo: obtener información sobre la gestión de sujetos y cronogramas específicos en el sistema Clínicas 1.0.

Entrevistado: Ing. Erislán Martínez Jera, jefe del proyecto Ensayos Clínicos en la UCI.

Preguntas:

1. ¿Qué es un sujeto?
2. ¿Cómo se gestionan los sujetos en el estudio?
3. ¿Cuáles son las etapas por la que pueden pasar los sujetos en el estudio?
4. ¿Qué es un cronograma específico?
5. ¿Qué incluye un cronograma específico?
6. ¿Quién puede generar el cronograma específico?
7. ¿Cuándo se le puede generar el cronograma específico a un sujeto que participa en el estudio?
8. ¿Qué son los momentos de seguimiento programados y no programados?
9. ¿Qué son las hojas CRD?
10. ¿Cuáles son los estados que pueden tener los momentos de seguimiento y las hojas CRD?
11. ¿Cuándo se puede firmar una hoja CRD y quién puede hacerlo?

Anexo 2: Diagramas de clases de diseño

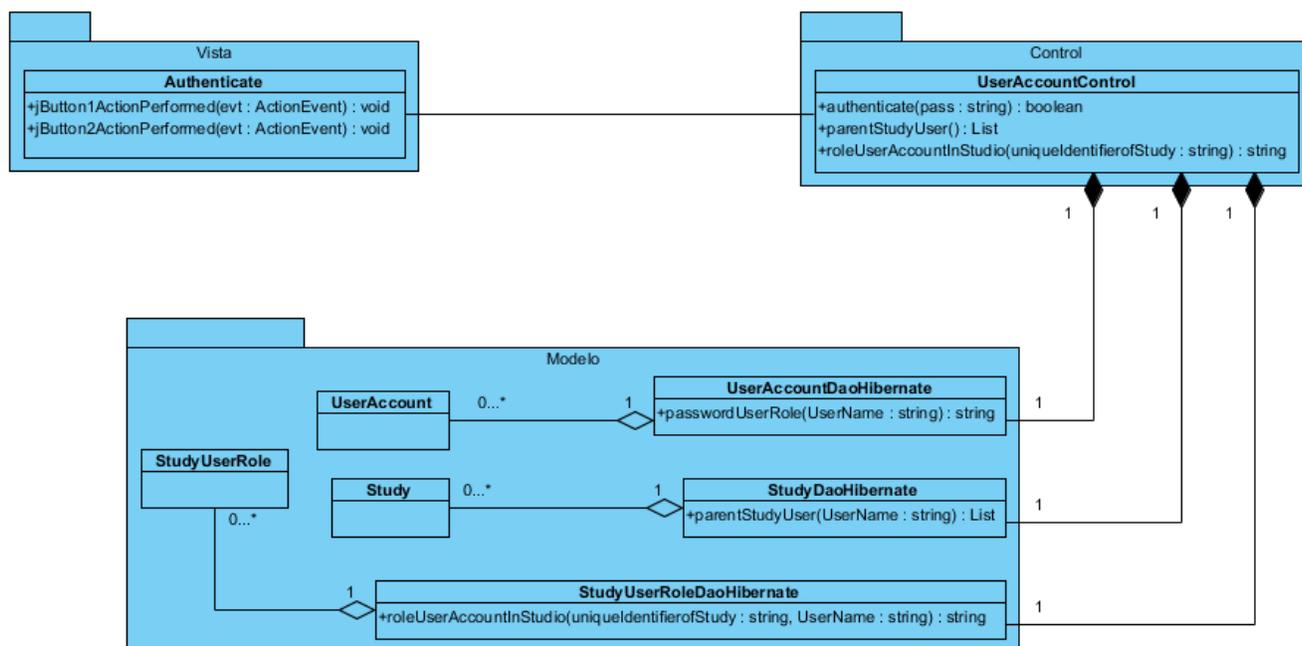


Figura.30 Diagrama de clase de diseño CU Autenticar usuario

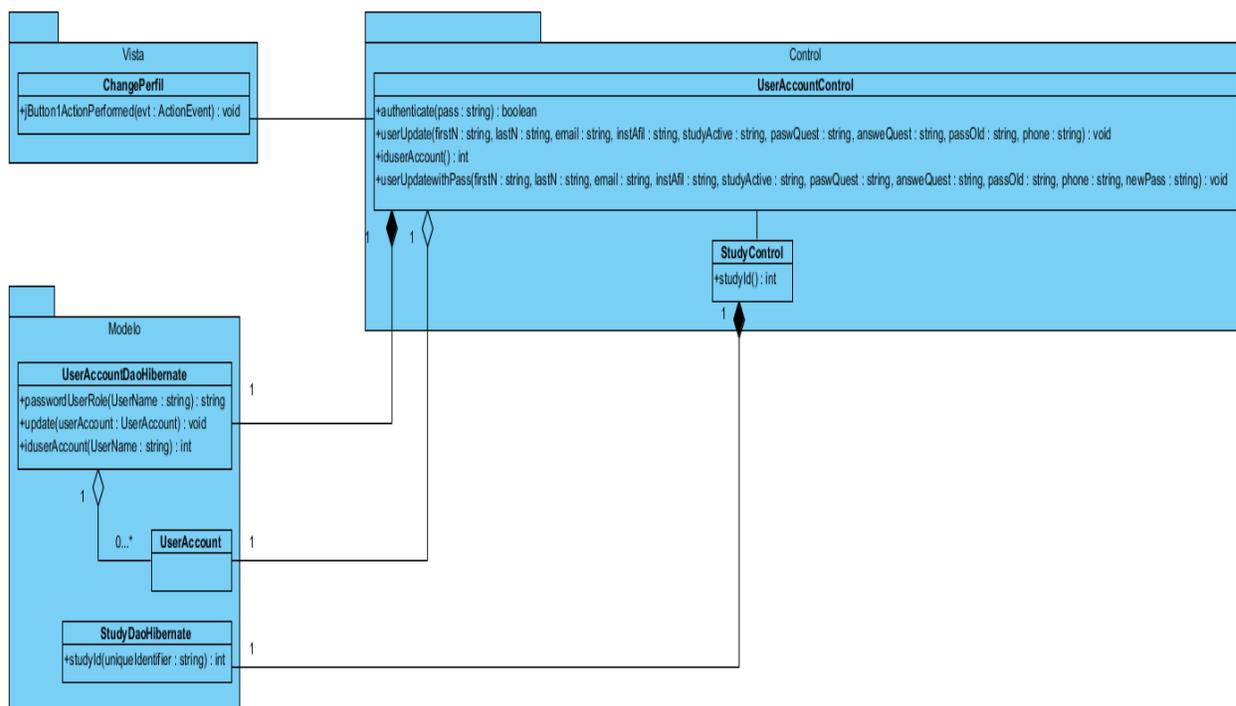


Figura.31 Diagrama de clase de diseño CU Actualizar perfil

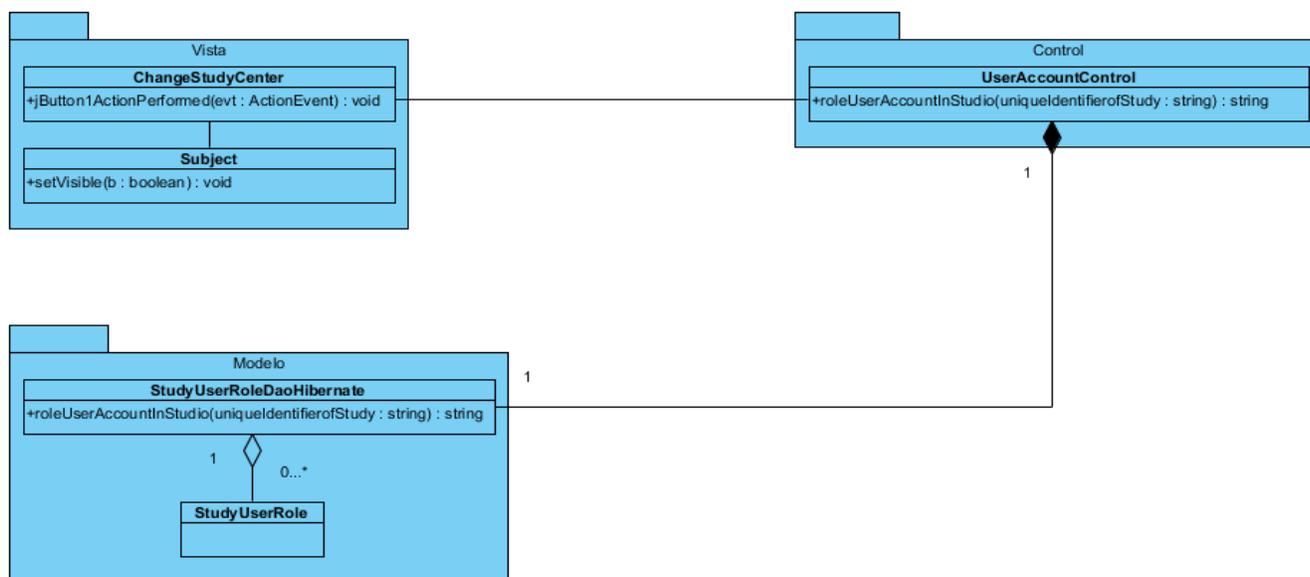


Figura.32 Diagrama de clase de diseño CU Cambiar centro/estudio

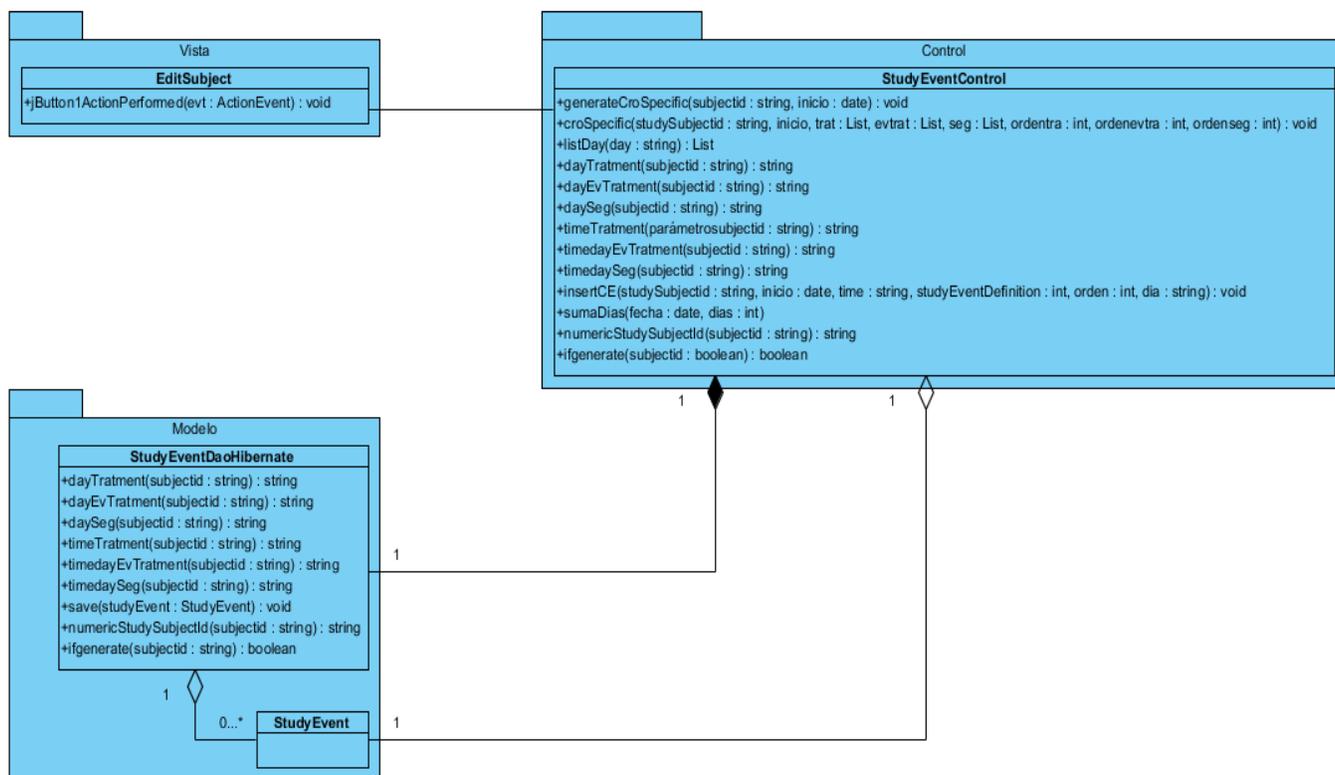


Figura.33 Diagrama de clase de diseño CU Generar cronograma específico

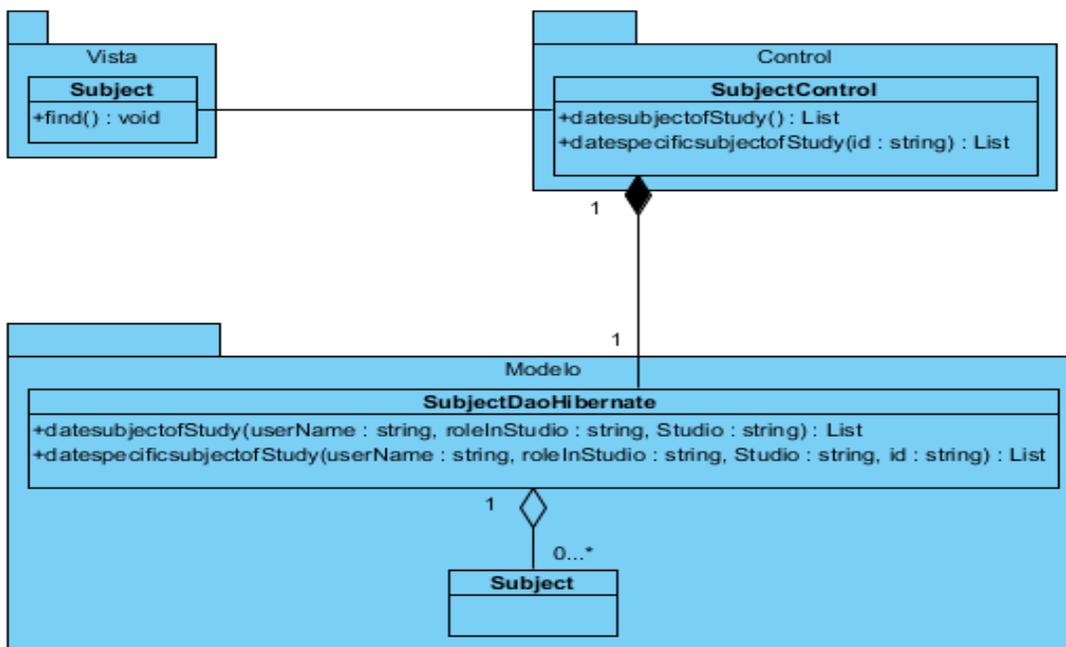


Figura.34 Diagrama de clase de diseño CU Buscar sujeto

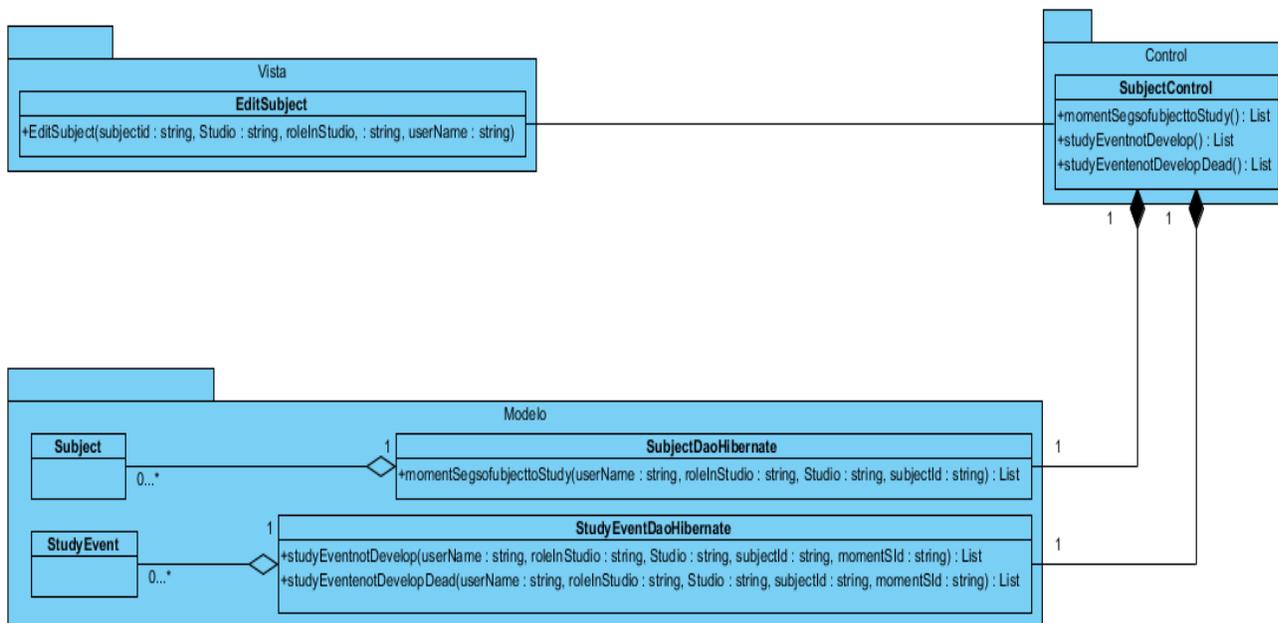


Figura.35 Diagrama de clase de diseño CU Visualizar momentos de seguimiento

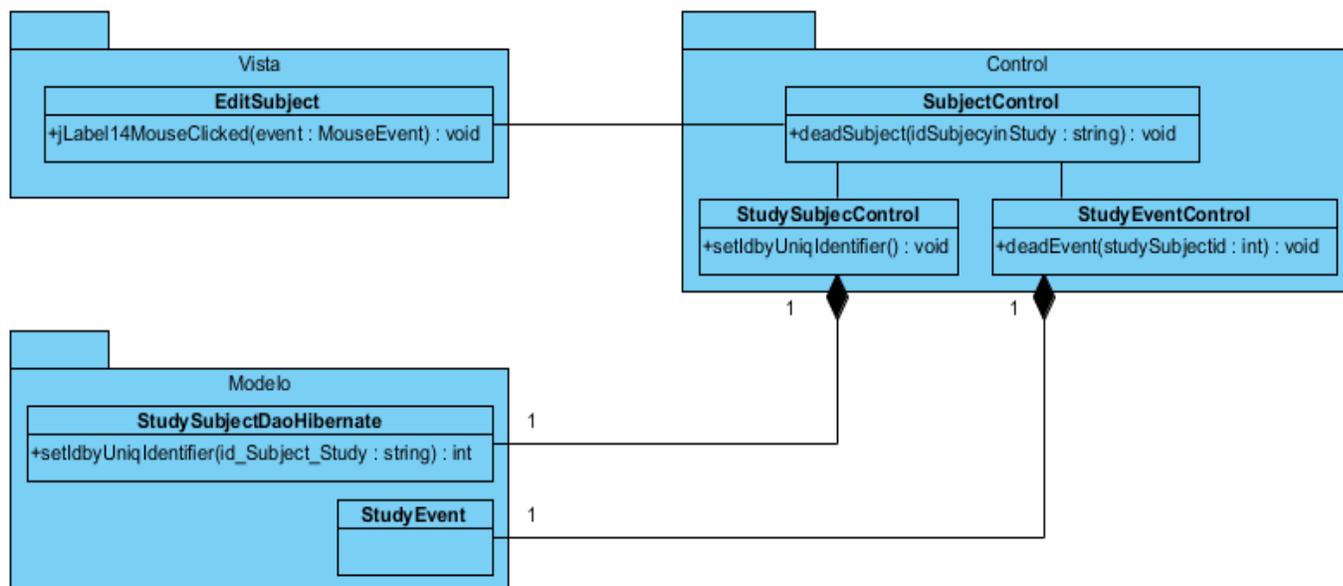


Figura.36 Diagrama de clase de diseño CU Añadir Momento de seguimiento no programado

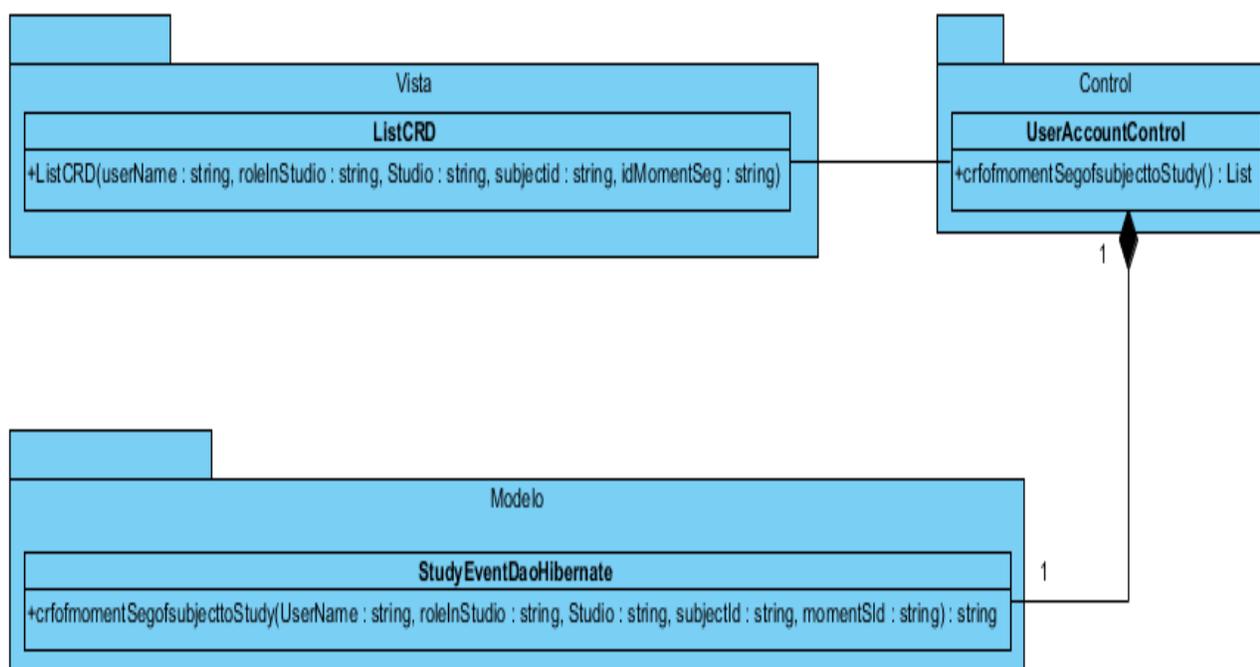


Figura.37 Diagrama de clase de diseño CU Visualizar listado de hojas CRD

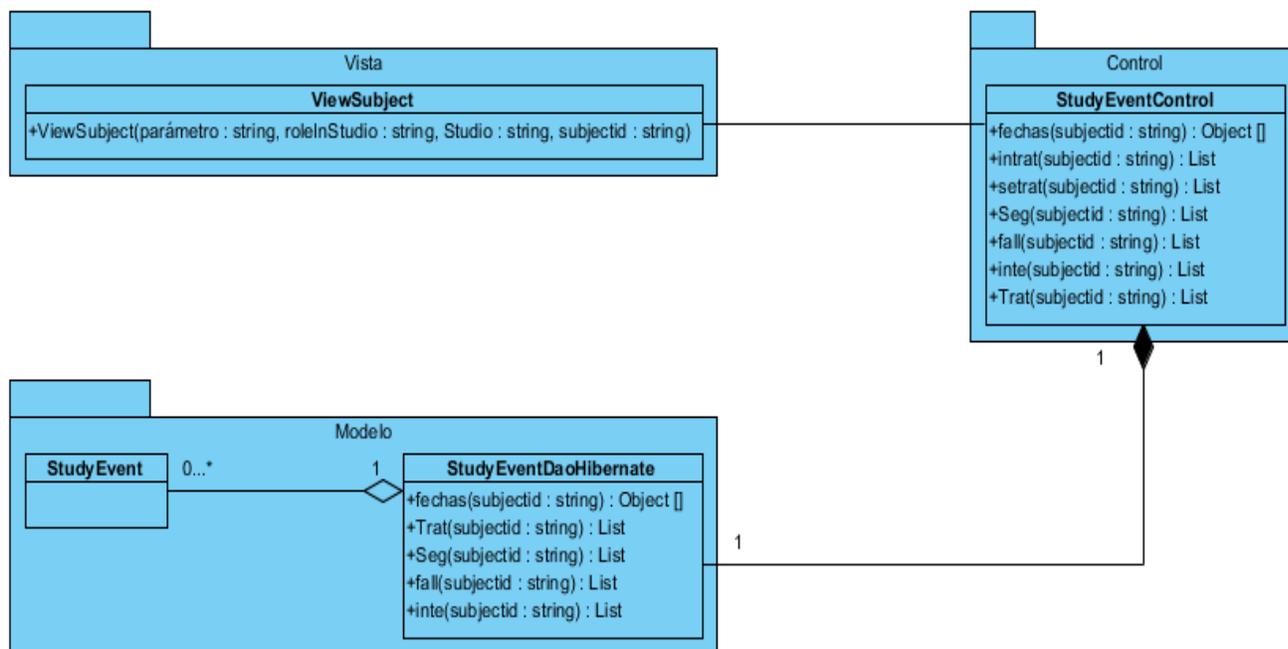


Figura.38 Diagrama de clase de diseño CU Visualizar cronograma específico

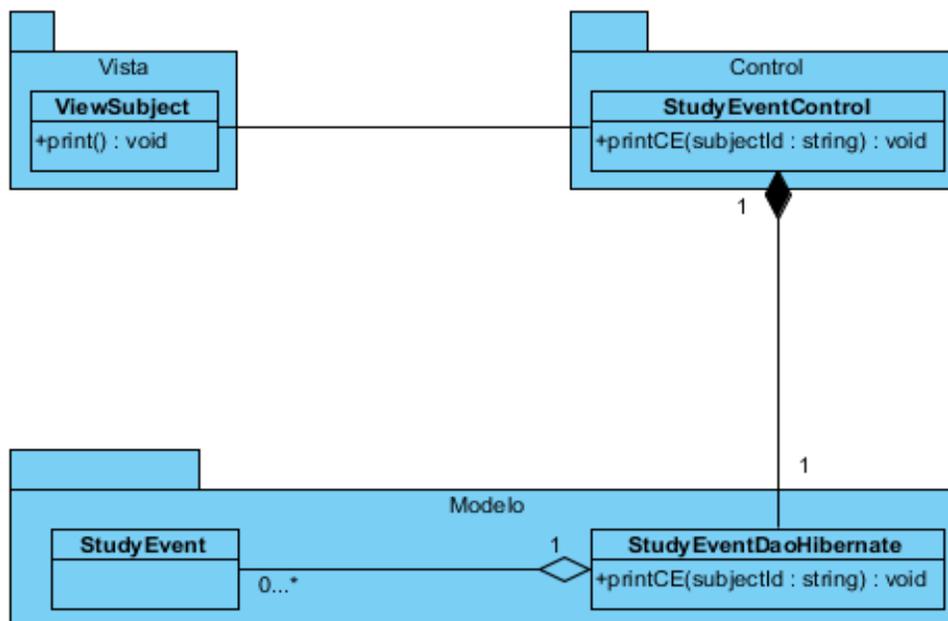


Figura.39 Diagrama de clase de diseño CU Imprimir cronograma específico

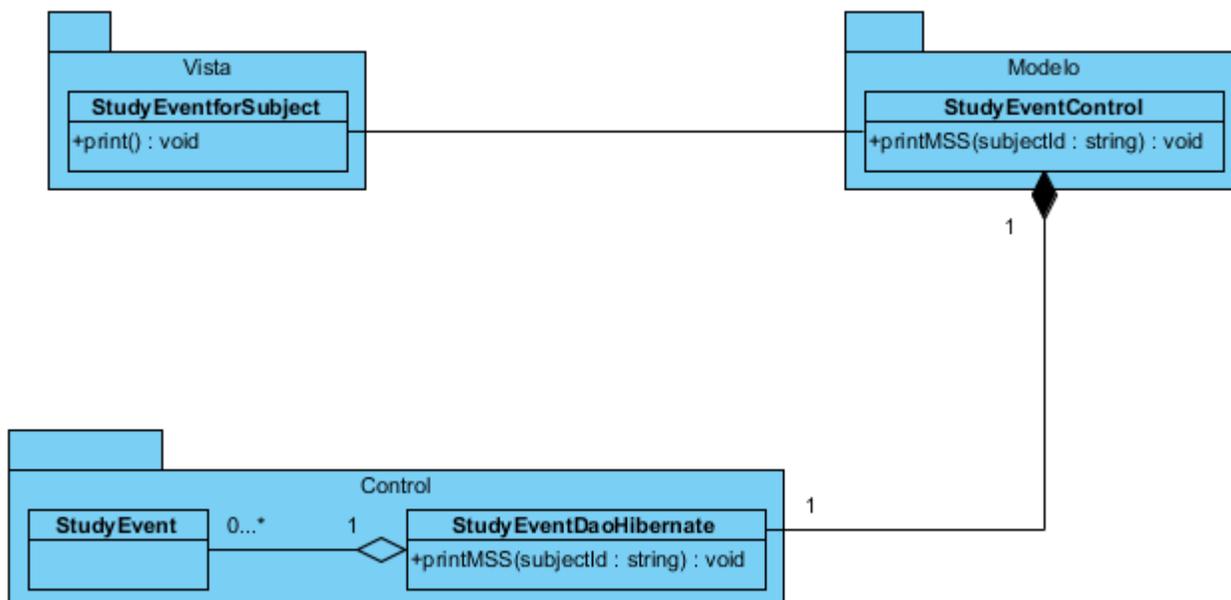


Figura.40 Diagrama de clase de diseño CU Imprimir momentos de seguimiento por sujeto

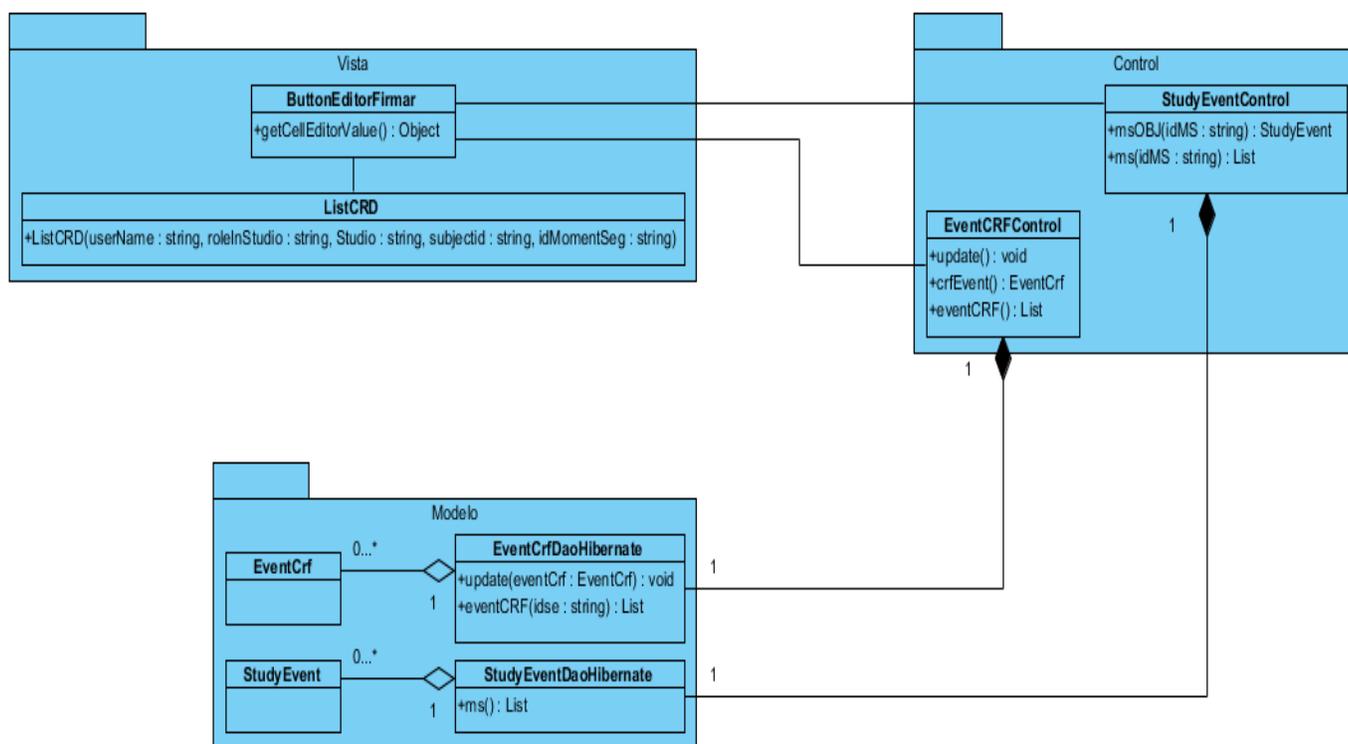


Figura.41 Diagrama de clase de diseño CU Firmar hoja CRD

Anexo 3: Interfaces de usuario

Autenticarse

Usuario:

Contraseña:

Figura.42 Interfaz de usuario CU Autenticar usuario

SIGECConductor
SISTEMA PARA LA CONDUCCIÓN DE ENSAYOS CLÍNICOS

Usuario: coordinador
Estudio: Estudio 1
Id: Estudio 1
Rol: Coordinador de Investigación Clínica
▶ Cambiar Centro/Estudio
▶ Cerrar Aplicación

Instrucciones
Alertas y Mensajes
Alertas y Mensajes
Tiene 1 notas de discrepancias

Cambiar perfil de usuario

Nombre: *

Apellido: *

Correo electrónico: *

Afilación institucional: *

Estudio activo por defecto: *

Pregunta recordatorio para la contraseña: *

Respuesta clave para la contraseña:

Contraseña antigua:

Contraseña nueva:

Deje esto en blanco si no quiere cambiar la contraseña

Confirmar nueva contraseña:

Teléfono: *

* indica que el campo es obligatorio.

Figura.43 Interfaz de usuario CU Actualizar perfil

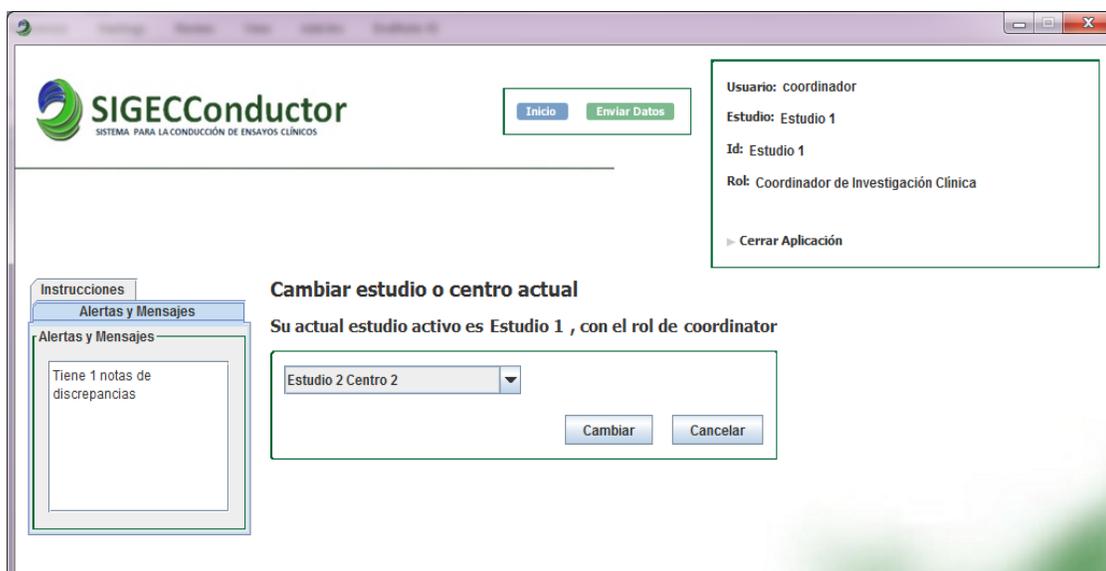


Figura.44 Interfaz de usuario CU Cambiar centro/estudio

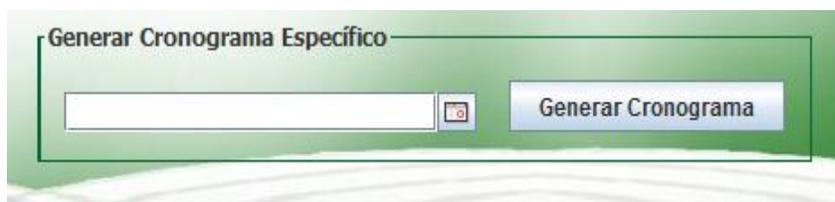


Figura.45 Interfaz de usuario CU Generar cronograma específico



Figura.46 Interfaz de usuario CU Buscar sujeto

SIGECConductor
SISTEMA PARA LA CONDUCCIÓN DE ENSAYOS CLÍNICOS

Inicio Enviar Datos
Sujetos Notas y Discrepancias

Usuario: coordinador
Estudio: Estudio 1
Id: Estudio 1
Rol: Coordinador de Investigación Clínica
Cambiar Centro/Estudio
Cerrar Aplicación

Instrucciones
Alertas y Mensajes
Alertas y Mensajes
Tiene 1 notas de discrepancias

Ver todos los momentos de seguimiento para Sujeto7

Ver todos los momentos de seguimiento programados

Momento de seguimiento	Plazo de tiempo	Fecha de inicio	Estado	Notas de monitoreo	Gesti
Seguimiento	-4.0	2014-05-25	Atrasado	0	
Evaluacion del tratamiento	-15.0	2014-05-21	Atrasado	0	
Tratamiento	-16.0	2014-05-20	Atrasado	0	
Evaluacion del tratamiento	-20.0	2014-05-16	Atrasado	0	
Tratamiento	-21.0	2014-05-15	Atrasado	0	
Evaluacion del tratamiento	-25.0	2014-05-11	Atrasado	0	
Tratamiento	-26.0	2014-05-10	Atrasado	0	
Evaluacion del tratamiento	-30.0	2014-05-06	Atrasado	0	
Tratamiento	-31.0	2014-05-05	Atrasado	0	
Evaluacion del tratamiento	-35.0	2014-05-01	Atrasado	0	
Tratamiento	-36.0	2014-04-30	Atrasado	0	
Evaluacion del tratamiento	-40.0	2014-04-26	Atrasado	0	

Todos los momentos de seguimiento no programados Fallecimiento

Ver todos los momentos de seguimiento no programados

Momento de seguimiento	Fecha de inicio	Estado	Notas de monitoreo	Gestionar Datos
Interrupción	2014-04-26	Atrasado	0	
Fallecimiento	2014-04-30	Atrasado	0	

Figura.47 Interfaz de usuario CU Visualizar momentos de seguimiento

Todos los momentos de seguimiento no programados Fallecimiento

► Adicionar momento

Momento de seguimiento	Fecha de inicio	Estado	Notas de monitoreo	Gestionar Datos

Figura.48 Interfaz de usuario CU Añadir Momento de seguimiento no programado



Figura.49 Interfaz de usuario CU Visualizar listado de hojas CRD

Cronograma Especifico

Momentos/Fechas	2014-04-10	2014-04-11	2014-04-15	2014-04-16	2014-04-20	2014-04-21	2014-04-25	2014-04-26
Tratamiento	☒		☒		☒		☒	
Evaluación de Tratamiento		☒		☒		☒		
Seguimiento								
Interrupción								
Fallecimiento								

Figura.50 Interfaz de usuario CU Visualizar cronograma específico

Imprimir

Cronograma Especifico

Momentos/Fechas	2014-04-10	2014-04-11	2014-04-15	2014-04-16	2014-04-20	2014-04-21	2014-04-25	2014-04-26
Tratamiento	☒		☒		☒		☒	
Evaluación de Tratamiento		☒		☒		☒		
Seguimiento								
Interrupción								
Fallecimiento								

Figura.51 Interfaz de usuario CU Imprimir cronograma específico

The screenshot shows the SIGECConductor web application interface. At the top left is the logo and name 'SIGECConductor SISTEMA PARA LA CONDUCCIÓN DE ENSAYOS CLÍNICOS'. In the top center, there are buttons for 'Inicio', 'Enviar Datos', 'Sujetos', and 'Notas y Discrepancias'. On the top right, a user information box displays: 'Usuario: coordinador', 'Estudio: Estudio 1', 'Id: Estudio 1', and 'Rol: Coordinador de Investigación Clínica', with links for 'Cambiar Centro/Estudio' and 'Cerrar Aplicación'.

On the left side, there are tabs for 'Instrucciones' and 'Alertas y Mensajes'. The 'Alertas y Mensajes' tab is active, showing a message: 'Tiene 1 notas de discrepancias'.

In the center, there is a search bar labeled 'Buscar' with a 'Buscar' button and an 'Imprimir' link to the right.

The main content area is titled 'Ver momentos por sujeto' and contains a table with the following data:

Sujeto	Fallecimiento	Interrupción	Evaluación inicial	Tratamiento	Evaluación del tratamiento	Seguimiento
Sujeto3	0	0	0	0	0	0
Sujeto4	0	0	0	0	0	0
kuko	0	0	0	0	0	0
Sujeto5	0	0	0	0	0	0
Sujeto6	0	0	0	0	0	0
Sujeto2	0	0	0	0	0	0
juju	0	0	1	1	1	0
Sujeto8	0	0	0	0	0	0
Sujeto9	0	0	0	0	0	0
Sujeto11	0	0	0	0	0	0
Sujeto7	0	0	0	0	0	0
Sujeto15	0	0	0	0	0	0
Sujeto16	0	0	0	0	0	0
Sujeto10	0	0	0	0	0	0
Sujeto12	0	0	0	0	0	0
Sujeto1	0	0	1	4	4	2
juan	0	0	0	0	0	0
Sujeto13	0	0	0	0	0	0
Sujeto14	0	0	0	0	0	0
tutu	0	0	0	0	0	0
rere	0	0	0	0	0	0

Figura.52 Interfaz de usuario CU Imprimir momentos de seguimiento por sujeto

The screenshot shows a table with the following data:

Nombre de la hoja CRD	Versión	Estado	Fecha de inicio	Notas de monitoreo	Gestionar Datos	Ver	Imprimir	Firmar
Hoja1	1	Atrasado	2014-05-21	0				

Below the table, there is a large empty rectangular area for additional information or actions.

Figura.53 Interfaz de usuario CU Firmar hoja CRD