

Universidad de las Ciencias Informáticas

Facultad 6



TÍTULO: “alasClínicas: Desarrollo de funcionalidades para el monitoreo de los datos a partir del Sistema OpenClinica.”

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Autores:

Yurien Santos Valdés
Denlis Laffita Hernández

Tutores:

Ing. Ana Lupe Delgado Montero
Ing. Richard Díaz Pompa
Ing. Liván Rodríguez Miranda

Ciudad de La Habana, Junio 2009



“Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio conscientes de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte.”

Ernesto Che Guevara.

DECLARACIÓN DE AUTORÍA

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

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

Denlis Laffita Hernández

Firma del autor

Yurien Santos Valdés

Firma del autor

Ing. Ana Lupe Delgado Montero

Firma de la tutora

Ing. Richard Díaz Pompa

Firma del tutor

Ing. Liván Rodríguez Miranda

Firma del tutor

AGRADECIMIENTOS

Ya han pasado cinco cursos de sueños, alegrías y ha llegado el momento de enfrentarnos al mundo como profesionales. Cada uno de nuestros logros trae consigo el amor, la comprensión y ayuda incondicional de todas aquellas personas que de una forma u otra fueron partícipes de que este momento existiese.

En primer lugar agradecer a toda la familia, a nuestras Madres, Padres y Hermanos por el gran esfuerzo y la paciencia para enfrentar numerosos obstáculos, por apoyarnos en cada momento de nuestras vidas por darnos las fuerzas para seguir adelante, por amarnos tanto y darnos lo mejor de sí, gracias por todo, gracias por no exigir, por no presionar, por saber guiarnos y sobre todo gracias por confiar. Los quiero mucho.

Agradecer a nuestros amigos que nos han apoyado durante toda la carrera universitaria. A nuestra líder de proyecto Lucía Rodríguez García, nuestra antigua líder de proyecto y ahora líder de Polo: Aidacelys López Díaz. Y especialmente a nuestros tutores: Liván Rodríguez Miranda, Richard Díaz Pompa, y Ana Lupe Delgado Montero; a nuestros profesores que de una manera u otra nos ayudaron Y a todos que incondicionalmente nos apoyaron durante todo este tiempo.

A todos... muchas gracias.

DEDICATORIA

Dedicamos este esfuerzo a toda nuestra familia en especial nuestros padres y hermanos que son la razón por la que vivimos y que han logrado soportarnos durante tanto tiempo y más aún en estos últimos 5 años. ¡Sí!, a ellos, que han luchado por vernos crecer en todos nuestros aspectos, que han sabido ayudarnos y apoyarnos en todos los momentos. A nuestras amistades que han convivido con nosotros en diferentes años de la carrera, a nuestros compañeros de aula, a nuestros profesores y a todas las personas que conocemos.

RESUMEN

El presente trabajo forma parte del proyecto Ensayos Clínicos de la facultad 6 de la Universidad de las Ciencias Informáticas, titulado: Desarrollo de funcionalidades para el monitoreo de los datos a partir del Sistema OpenClinica. Seccionado en cuatros capítulos, aborda el análisis, diseño e implementación de funcionalidades para el monitoreo proporcionándole al sistema la usabilidad requerida por el cliente encargado de utilizar el mismo. Se justifica la utilización de herramientas y Metodologías, además proporciona una breve descripción de un Ensayo Clínico, importancia para nuestro país y necesidad de aplicarlos en el mismo. Por ser nuestro país tan humano y tener tanto personal de una gran calidad humana y profesional, merecen el esfuerzo de muchos para lograr un trabajo eficiente por la garantía de vida de nuestro pueblo y la humanidad en general.

Palabras Claves: Ensayos Clínicos, OpenClinica.

ÍNDICE

DECLARACIÓN DE AUTORÍA I

AGRADECIMIENTOS II

DEDICATORIA III

RESUMEN IV

ÍNDICE V

TABLA DE IMÁGENES IX

INTRODUCCIÓN 1

CAPÍTULO 1: Fundamentación Teórica 6

 Introducción 6

 1 Ensayos Clínicos 6

 2 Aplicaciones Web para Sistemas de Manejo de Datos de ensayos clínicos. 8

 2.1 MACRO 8

 2.2 Administrador de Estudio (Study Manager) 9

 2.3 OpenClinica 9

 3 Metodología y Herramientas a Utilizar 10

 3.1 Proceso de desarrollo de software 10

 3.2 RUP - Metodología de desarrollo de software OO (Orientado a Objetos) 10

 3.3 Lenguaje de Modelado (UML Versión 2.0) 12

 3.4 Herramienta CASE para la modelación del sistema. 14

 3.5 Herramienta para el desarrollo 15

 3.5.1 Eclipse (versión 3.4.2) 16

 3.5.2 Apache Tomcat (versión 5.5.27) 16

 3.5.3 Sistema gestor de base de datos 17

 3.6 Herramienta de control de versiones 18

CAPÍTULO 2: Características del Sistema 19

 Introducción 19

 1 Modelo de Dominio 19

 2 Requerimientos 20

 2.1 Requisitos funcionales 20

 2.2 Requisitos no Funcionales 21

 2.2.1 RNF de Apariencia o Interfaz Externa 21

 2.2.2 RNF de Usabilidad 21

 2.2.3 RNF de Soporte 21

| | | |
|-------|--|----|
| 2.2.4 | RNF de Seguridad..... | 21 |
| 2.2.5 | RNF de Software..... | 21 |
| 2.2.6 | RNF de Hardware..... | 22 |
| 2.2.7 | RNF de Disponibilidad..... | 22 |
| 2.2.8 | RNF de Licencia..... | 22 |
| 2.2.9 | RNF de Estándares Aplicables..... | 22 |
| 3 | Patrones de Casos de Uso..... | 23 |
| 3.1 | Múltiples actores (roles comunes):..... | 23 |
| 4 | Descripción de los Casos de Usos del Sistema..... | 23 |
| 4.1 | Actores del Sistema..... | 24 |
| 4.2 | CU: Monitorear Hoja de CRD..... | 25 |
| 4.3 | CU: Visualizar Notas..... | 27 |
| 4.4 | CU: Responder Nota de Monitoreo..... | 33 |
| 4.5 | CU: Responder Nota General..... | 36 |
| 4.6 | CU: Crear Nota de Monitoreo..... | 38 |
| 4.7 | CU: Crear Nota general..... | 41 |
| | Conclusiones..... | 43 |
| | CAPÍTULO3: Análisis y Diseño del Sistema..... | 44 |
| | Introducción..... | 44 |
| 1 | Patrones..... | 44 |
| 1.1 | Patrones de Diseño..... | 44 |
| 1.1.1 | DAO..... | 44 |
| 1.1.2 | Iterador (Iterator)..... | 45 |
| 1.1.3 | Fachada de Sesión (Session Façade)..... | 46 |
| 1.1.4 | Vista compuesta (Composite View)..... | 46 |
| 1.1.5 | Experto..... | 47 |
| 1.1.6 | Creador:..... | 47 |
| 1.1.7 | Bajo acoplamiento:..... | 47 |
| 1.1.8 | Alta cohesión:..... | 47 |
| 2 | Patrones arquitectura..... | 48 |
| 2.1 | El patrón de arquitectura de software Modelo – Vista – Controlador (MVC)...... | 48 |
| 3 | Uso del Digester..... | 50 |
| 4 | Diagramas de clases del diseño..... | 50 |
| 4.1 | Diagrama de Clases de Diseño del CU Crear Nota General..... | 51 |

| | | |
|-----|--|----|
| 4.2 | Diagrama de Clases del Diseño del CU Responder Nota General | 53 |
| 4.3 | Diagrama de Clases del Diseño del CU Monitorear Hoja de CRD | 55 |
| 4.4 | Diagrama de Clases del Diseño del CU Visualizar Notas | 57 |
| 4.5 | Diagrama de Clases del Diseño del CU Crear Nota de Monitoreo | 60 |
| 4.6 | Diagrama de Clases del Diseño del CU Responder Nota de Monitoreo. | 61 |
| 5 | Diagramas de interacción..... | 63 |
| 5.1 | Diagrama de Secuencia del Caso de Uso Crear Nota de General..... | 64 |
| 5.2 | Diagrama de Secuencia del Caso de Uso Responder Nota de General. | 66 |
| 5.3 | Diagrama de Secuencia del Caso de Uso Crear Nota Monitoreo. | 67 |
| 5.4 | Diagrama de Secuencia del Caso de Uso Responder Nota Monitoreo. | 69 |
| 5.5 | Diagrama de Secuencia del Caso de Uso Monitorear Hoja de CRD..... | 70 |
| 5.6 | Diagrama de Secuencia del Caso de Uso Visualizar Notas..... | 71 |
| 6 | Modelo de Datos..... | 72 |
| 7 | Diagrama de Despliegue..... | 75 |
| | Conclusiones | 77 |
| | CAPÍTULO 4: Implementación del Sistema..... | 78 |
| | Introducción | 78 |
| 1 | Componentes..... | 78 |
| 1.1 | Diagrama de componentes..... | 78 |
| 1.2 | Diagrama de componente del caso de uso Crear Nota de Monitoreo..... | 80 |
| 1.3 | Diagrama de componente del caso de uso Responder Nota de Monitoreo..... | 81 |
| 1.4 | Diagrama de componente del caso de uso Crear Nota de General. | 82 |
| 1.5 | Diagrama de componente del caso de uso Responder Nota de Monitoreo..... | 83 |
| 1.6 | Diagrama de componente del caso de uso Monitorear Hoja de CRD. | 84 |
| 1.7 | Diagrama de componente del caso de uso Visualizar Notas. | 85 |
| 2 | Ejemplos de Códigos Relevantes..... | 86 |
| 2.1 | CU Visualizar Notas | 86 |
| 2.2 | CU Crear Nota General | 87 |
| 2.3 | CU Crear Nota de Monitoreo | 88 |
| 2.4 | Monitorear una Hoja de CRD..... | 89 |
| 2.5 | Función JavaScript que ejecuta funciones de AJAX. CU Visualizar Notas..... | 90 |
| 3 | Principales Interfaces | 91 |
| 3.1 | CU Crear Nota de Monitoreo..... | 91 |
| 3.2 | Responder Nota de Monitoreo..... | 92 |

| | | |
|-----|--|-----|
| 3.3 | Crear Nota General | 94 |
| 3.4 | Responder Nota General | 95 |
| 3.5 | Monitorear Hoja de CRD | 97 |
| 3.6 | Visualizar Notas..... | 98 |
| | Conclusiones | 99 |
| | Conclusiones Generales | 100 |
| | Recomendaciones: | 101 |
| | Referencias Bibliográficas..... | 102 |
| | Bibliografía | 103 |
| | Anexos..... | 104 |
| 1 | CU: Mostrar Informe de las Notas Generales | 104 |
| 2 | CU: Mostrar Informe de las Notas de Sitio | 105 |
| 3 | CU: Mostrar Informe de las Notas de Monitoreo..... | 106 |
| 4 | Diagramas de secuencias | 107 |
| 5 | Diagramas de Componentes..... | 110 |
| 6 | Diagramas de Clase de Diseño | 113 |
| | Glosario de Términos..... | 115 |

TABLA DE IMÁGENES

| | |
|--|----|
| Ilustración 1 RUP..... | 11 |
| Ilustración 2 UML..... | 13 |
| Ilustración 3 Visual Paradigm..... | 14 |
| Ilustración 4 Eclipse..... | 16 |
| Ilustración 5 Apache Tomcat (versión 5.5.27)..... | 16 |
| Ilustración 6 PostgreSQL (Gestor de BD) (versión 8.2.9.1)..... | 17 |
| Ilustración 7 Modelo de Dominio..... | 19 |
| Ilustración 8 Licencia..... | 22 |
| Ilustración 9 Diagrama de Casos de Uso del Sistema..... | 23 |
| Ilustración 10 Diagrama de Actores del Sistema..... | 24 |
| Ilustración 11 Patrón DAO..... | 45 |
| Ilustración 12 Patrón Iterador..... | 45 |
| Ilustración 13 Patrón Fachada de Sesión..... | 46 |
| Ilustración 14 Patrón Vista Compuesta..... | 46 |
| Ilustración 15 Crear Nota General..... | 51 |
| Ilustración 16 Responder Nota General..... | 53 |
| Ilustración 17 Responder Nota General..... | 55 |
| Ilustración 18 Visualizar Notas..... | 57 |
| Ilustración 19 Crear Nota de Monitoreo..... | 60 |
| Ilustración 20 Responder Nota de Monitoreo..... | 61 |
| Ilustración 21 Crear Nota General #2..... | 65 |
| Ilustración 22 Responder Nota General..... | 66 |
| Ilustración 23 Crear Nota Monitoreo # 1..... | 67 |
| Ilustración 24 Crear Nota Monitoreo # 2..... | 68 |
| Ilustración 25 Responder Nota Monitoreo..... | 69 |
| Ilustración 26 Monitorear Hoja de CRD..... | 70 |
| Ilustración 27 Visualizar Notas Sección#1..... | 71 |
| Ilustración 28 Modelo de Datos..... | 72 |
| Ilustración 29 Diagrama de Despliegue..... | 76 |
| Ilustración 30 Diagrama de Componentes..... | 79 |
| Ilustración 31 DC Crear Nota de monitoreo..... | 80 |
| Ilustración 32 DC Responder Nota de Monitoreo..... | 81 |

| | |
|--|-----|
| Ilustración 33 DC Crear Nota General..... | 82 |
| Ilustración 34 DC Responder Nota de Monitoreo | 83 |
| Ilustración 35 DC Monitorear Hoja de CRD..... | 84 |
| Ilustración 36 DC Vizualizar Notas | 85 |
| Ilustración 37 Ejemplo de código del CU Vizualizar Notas | 86 |
| Ilustración 38Ejemplo de código del CU Crear Nota General..... | 87 |
| Ilustración 39 Ejemplo de código de CU Crear Nota de Monitoreo | 88 |
| Ilustración 40 Ejemplo de código del CU Monitorear Hoja de CRD | 89 |
| Ilustración 41 Función AJAX | 90 |
| Ilustración 42 Diagrama de secuencia del CU Visualizar Notas: ver Notas de Monitoreo y Sitio | 107 |
| Ilustración 43 Diagrama de secuencia del CU Visualizar Notas: listar Notas Generales | 108 |
| Ilustración 44 Diagrama de secuencia del CU Visualizar Notas: listar Notas de Monitoreo | 108 |
| Ilustración 45 Diagrama de secuencia del CU Visualizar Notas: listar Notas de Sitio | 109 |
| Ilustración 46 Mostrar informe de Notas Generales | 110 |
| Ilustración 47 Mostrar informe de Notas de Monitoreo | 111 |
| Ilustración 48 Mostrar informe de Notas de Sitio | 112 |
| Ilustración 49 Diagrama de clase del diseño de las Notas de Sitio | 113 |
| Ilustración 50 Diagrama de clase del diseño de las Notas Generales | 113 |
| Ilustración 51 Diagrama de clase del diseño de las Notas de Monitoreo | 114 |

INTRODUCCIÓN

El cáncer no es una enfermedad, sino más bien muchas enfermedades. De hecho, hay 100 tipos diferentes de cáncer. Todos los cánceres empiezan en las células. Las células son las unidades básicas que forman los tejidos del cuerpo. Para entender mejor qué es el cáncer, es necesario saber cómo las células normales se vuelven cancerosas.

El cuerpo está compuesto de muchos tipos de células. Estas células crecen y se dividen para producir nuevas células conforme el cuerpo las necesita. Cuando las células envejecen, mueren y éstas son reemplazadas por células nuevas.

Pero a veces, este proceso ordenado de división de células se descontrola. Células nuevas se siguen formando cuando el cuerpo no las necesita. Cuando esto pasa, las células viejas no mueren cuando deberían morir. Estas células que no son necesarias pueden formar una masa de tejido. Esta masa de tejido es lo que se llama tumor.(ELEC 2008)

Según datos estadísticos de la Organización Mundial de la Salud (OMS), de los 58 millones de muertes que se registraron en el mundo en el año 2007, 7.6 millones (13%) se debieron al cáncer. Los que más contribuyen a la mortalidad son los cánceres de:

- pulmón (1,4 millones de defunciones)
- estómago (866 000 defunciones)
- hígado (653 000 defunciones)
- colon (677 000 defunciones)
- mama (548 999 defunciones)(OMS 2009)

En el país hay más de 100 mil personas con algún tipo de enfermedad maligna y aparecen cada año unos 25 mil nuevos casos. Y en cuanto a la mortalidad, se sitúa en la segunda causa en la población en general, pero es la primera causa de muerte antes de los 74 años.

El 5 de diciembre de 1994 Fidel Castro Ruz inaugura el Centro de Inmunología Molecular (CIM). Con la principal misión obtener y producir nuevos biofármacos destinados al tratamiento del cáncer y otras enfermedades crónicas no transmisibles e introducirlos en la salud pública cubana. Para la aprobación e introducción de estos productos en el mercado se realizan Ensayos Clínicos (EC). Estos constituyen

un tipo de estudio clínico que conforma una de las fases del proceso de desarrollo de fármacos que a través de su aplicación a seres humanos pretende valorar su eficacia y seguridad.

En la medida que se avanza en el desarrollo de los productos, se avanza en las fases de los EC. Esto trae aparejado el incremento del número de pacientes a estudiar y por consiguiente el número de hospitales que se enrolan en el reclutamiento de pacientes. Consecuentemente una extensa documentación: Cuadernos de Recogida de Datos (CRD), imágenes, y datos primarios de los pacientes. Como un estimado conservador se puede decir que por cada paciente de un Ensayo Clínico se genera por lo menos 1000 datos diferentes, lo cual permite fácilmente estimar el gran volumen de información que se genera y maneja en esta actividad.

Por ejemplo, para 2500 pacientes que previó manejar el CIM en un determinado momento se estimó generar no menos de 2 millones 500 mil datos, los cuales son de muy diverso tipo, desde información referida a mediciones cuantitativas del estado de un paciente (presión sanguínea, hemoglobina y mucho otros), hasta imágenes que caracterizan el tamaño del tumor, o hasta información referida al proceso de intercambio de mensajes entre los investigadores que conducen los estudios.

En la actualidad, los ensayos clínicos en el CIM y en Cuba se desarrollan de la manera clásica, toda la información se encuentra en papel, y la transmisión de información de los hospitales a los centros promotores y viceversa se hace generalmente vía e-mail o vía telefónica. Un complicado proceso de monitoreo de la información clínica se realiza visitando periódicamente cada uno de los sitios participantes (incluyendo las provincias orientales), con el objetivo de controlar la calidad y uniformidad de los datos recogidos. Finalmente bases de datos en versión electrónica se preparan a partir de los CRD llenados en los hospitales, con el objetivo de facilitar el análisis final de los resultados de cada ensayo. Tales bases de datos se confeccionan hoy en el CIM en los software, Microsoft Access y EPINFO, teniendo que hacerse una entrada doble de datos desde los CRD, por parte de dos operadores de máquina no relacionados con la recogida primaria de los datos del estudio.

Este proceso tiene como principales desventajas:

- Retraso y errores en el llenado de los modelos.
- Falta de estandarización.
- Interpretaciones erróneas.
- Lo cual trae como consecuencia el incumplimiento del protocolo establecido.

El CIM tiene la necesidad de informatizar el proceso de conducción de los ensayos clínicos de forma que permita la transmisión remota de datos entre la Red Nacional de Hospitales y el Centro, además

de la actualización constante de las bases de datos y el procesamiento de la información clínica que de estas se deriven.

La Universidad de Ciencias Informáticas en coordinación con el Centro de Inmunología Molecular decide el estudio de un potente sistema para el manejo de datos de ensayos clínicos.

OpenClinica: Es un software de captura electrónica de datos y de gestión de datos clínicos para ensayos clínicos y estudios de investigación. El software está basado en web y está diseñado para dar cobertura a todo tipo de estudios clínicos. Está desarrollado a partir de los estándares de más prestigio para alcanzar altos niveles de interoperabilidad con otros servicios y plataformas. Su arquitectura modular, transparencia y su modelo de desarrollo colaborativo ofrecen una gran flexibilidad a la vez que permiten desplegar soluciones de alto rendimiento y escalabilidad. Permite cumplir con los más altos estándares de calidad, seguridad y buenas prácticas clínicas según GCP (good clinical practices), HIPAA (Health Insurance Portability and Accountability Act) y apoya las 21 pautas de la parte 11 de CFR. Permitiendo además por ser de código abierto basado en java, la posibilidad de ser adaptado a la ejecución de Ensayos Clínicos Cubanos. Presenta una interfaz fácil de usar, permite la extracción de datos y filtrado de archivo, posibilita el intercambio de recursos de forma segura y transparente, admite exportación de herramientas para la migración de datos clínicos en las hojas de cálculo Excel y base de datos locales. OpenClinica presenta una infraestructura escalable, utiliza Java J2EE, con base de datos relacionales PostgreSQL incluidas y Oracle 10G, además usa el patrón de arquitectura Modelo Vista Controlador y Subversión como controlador de versiones.

Este estudio se ha realizado junto con especialistas del CIM vinculados al manejo de ensayos clínicos en un período de un mes y medio. Concluido el estudio, se determinó que algunas de las funcionalidades del sistema son muy similares a los procesos realizados en la conducción de Ensayos Clínicos Cubanos. Sin embargo, existen un conjunto de requisitos que no están contemplados en el sistema y otros que no están a fin con el manejo de ensayos clínicos en Cuba, funcionalidades que minimizarían el trabajo durante la ejecución del ensayo mejorando el monitoreo del sistema.

Ejemplos:

- El monitoreo de datos por hojas de CRD.
- Notas de Monitoreo.
- Notas Generales.
- Notas de Sitio.
- Mostrar informe de las notas.

De la situación antes expuesta se define el siguiente **problema científico**:

- ¿Cómo adaptar el monitoreo de los datos del sistema OpenClinica al proceso de conducción de los Ensayos Clínicos Cubanos?

A partir del problema planteado se define como **objeto de estudio**:

- Proceso de conducción de los Ensayos Clínicos Cubanos en el sistema OpenClinica.

Del objeto de estudio se define como **campo de acción**:

- Monitoreo de los datos de Ensayos Clínicos Cubanos del sistema OpenClinica.

Identificándose como **objetivo general**:

- Desarrollar funcionalidades para el monitoreo de datos de Ensayos Clínicos Cubanos del sistema OpenClinica.

Donde se definen los siguientes **objetivos específicos**:

- Modelar el entorno del negocio.
- Analizar las funcionalidades para el monitoreo de datos de los Ensayos Clínicos Cubanos en el sistema OpenClinica.
- Diseñar las funcionalidades para el monitoreo de los datos de los Ensayos Clínicos Cubanos en el sistema OpenClinica
- Implementar las funcionalidades para el monitoreo de datos de los Ensayos Clínicos Cubanos en el sistema OpenClinica.

Para dar cumplimiento a los objetivos específicos se trazaron las siguientes **tareas**:

- Estudio del sistema OpenClinica.
- Estudio de los procesos del negocio de los Ensayos Clínicos Cubanos.
- Modelación del dominio.
- Levantamientos de requisitos a adaptar al sistema OpenClinica.
- Definición y descripción de los casos de uso del sistema.

- Realización de los diagramas de clase de diseño.
- Realización de los diagramas de interacción.
- Realización de los diagramas de clase persistentes.
- Realización de los diagramas de componentes.
- Implementación de las funcionalidades para el sistema.

Estructura del trabajo:

- **Capítulo 1 Fundamentación Teórica:** En este capítulo se brinda el estado del arte del objeto de estudio; se fundamentan las metodologías, tecnologías y herramientas a utilizar para el desarrollo de la aplicación web.
- **Capítulo 2 Características del Sistema:** Este capítulo describe la modelación del negocio (Modelo de dominio) .Contiene además la captura de los requisitos funcionales y no funcionales, el diagrama de casos de uso del sistema, así como sus respectivas descripciones de los casos de uso considerados más significativos.
- **Capítulo 3 Análisis y Diseño del Sistema:** Describe los patrones de diseño y arquitectura a utilizar y define los artefactos correspondientes al rol de diseñador, dígame diagrama de clases del diseño, los diagramas de interacción y diagrama de despliegue.
- **Capítulo 4 Implementación del Sistema:** Implementa elementos del diseño en componentes de implementación. Define la organización del código en términos de subsistemas de implementación organizados por capas.

CAPÍTULO 1: Fundamentación Teórica.

Introducción

En este capítulo se brinda el estado del arte del objeto de estudio; se fundamentan las metodologías, tecnologías y herramientas a utilizar para el desarrollo de la aplicación web.

1 Ensayos Clínicos

¿Qué es un Ensayo Clínico?

Un Ensayo Clínico constituye un tipo de estudio clínico que conforma una de las fases del proceso de desarrollo de fármacos que a través de su aplicación a seres humanos pretende prevenir, detectar o tratar una enfermedad y ayudan a los médicos a descubrir si estos nuevos tratamientos son inocuos y si son mejores que los actuales.

¿Por qué aplicarlos en Cuba?

Desde hace muchos años la especie humana ha sido amenazada por diversas enfermedades, algunas de ellas mortales, como es el caso del cáncer; con el objetivo de combatir las mismas, científicos de Cuba y el resto del mundo se han dado a la tarea de crear productos químicos y convertirlos en medicamentos para tratamientos a seres humanos. Uno de los centros dedicados a esta labor es el CIM. Pero, para su aplicación y exportación deben ser aprobados por las Agencias Regulatorias.

Después de una profunda investigación a nivel mundial sobre la informatización de la gestión de ensayos clínicos se obtienen una serie de ventajas prácticas, como son:

- ✓ Elimina la doble entrada de los datos. Los datos son entrados directamente por quienes los recogen (lo que reduce la posibilidad de errores en los datos).
- ✓ Permite una regularización mayor e inmediata de los datos recogidos.
- ✓ Permite un acceso inmediato y actualizado del estado de desarrollo del ensayo clínico.
- ✓ Permite una optimización del proceso de monitoreo. Se plantea que es posible reducir el número de visitas requeridas a los sitios de inclusión, así como reducir el tiempo requerido en las mismas.
- ✓ Reduce el tiempo real de conducción del ensayo clínico, acelerando de ese modo el desarrollo del producto estudiado.

- ✓ Permite mayor meta-análisis de la información recogida en el ensayo clínico.

Lo antes expuesto revela la importancia de los ensayos clínicos en Cuba, es la vía segura para que los medicamentos destinados a combatir el cáncer, sean lanzados al mercado sin correr el riesgo de que no cumplan con las normas requeridas para ser usados por una persona que padezca de esta enfermedad.

¿Cuáles son los tipos de ensayos existentes?

De acuerdo con el Instituto Nacional del Cáncer, hay diferentes tipos de ensayos clínicos para el cáncer, los cuales incluyen:

- Los ensayos de prevención están diseñados para prevenir el desarrollo del cáncer en las personas que no han tenido cáncer anteriormente.
- Los ensayos de prevención están diseñados para prevenir el desarrollo de un tipo nuevo de cáncer en las personas que ya han tenido cáncer.
- Los ensayos para la detección temprana están diseñados para encontrar el cáncer especialmente en sus etapas más tempranas.
- Los ensayos de tratamiento están diseñados para probar terapias nuevas en las personas que tienen cáncer.
- Los estudios de la calidad de vida están diseñados para mejorar la comodidad y la calidad de la vida de las personas que tienen cáncer. Los estudios conductuales para evaluar las formas de modificar las conductas que causan el cáncer, como el uso del tabaco.
- Los estudios genéticos para abordar el modo en el que la composición genética afecta la detección, diagnóstico.

¿Dónde se realizan?

Los ensayos clínicos son patrocinados por organizaciones a cargo de médicos y otros prestadores de atención de la salud, estos determinarán la localización de los ensayos. En general se llevan a cabo en grandes centros para tratamiento del cáncer, hospitales o centros médicos locales. En el caso del CIM los ensayos clínicos se realizan de forma multicéntrica en su mayoría. Solo los estudios de farmacología clínica se conducen de forma unicéntrica, o sea, en un solo centro hospitalario.

Esta área posee laboratorios equipados para inmunoquímica, radioquímica, biología molecular, cultivo celular e instalaciones para la experimentación con modelos animales y una Planta Piloto cGMP que suministra los productos para ensayos clínicos.

2 Aplicaciones Web para Sistemas de Manejo de Datos de ensayos clínicos.

A nivel mundial se trabaja en base al desarrollo de las tecnologías y a su vez se lleva a cabo la transformación de la información a formato digital y electrónico. El CIM en su investigación de los ensayos clínicos se integra en el avance de las nuevas tecnologías, teniendo en cuenta esto se traza como tarea el estudio y selección de las herramientas y software que existen para realizar las tareas relacionadas con este campo. Entre ellas se encuentran el MACRO, Study Manager y OpenClinica.

2.1 MACRO

MACRO brinda soporte para todos los ensayos, desde la primera fase hasta la cuarta, escalando de un solo sitio, hasta grandes ensayos internacionales. InferMed está trabajando en conjunto con las principales compañías farmacéuticas, para implementar MACRO como una solución global para la recolección electrónica de datos y con los principales centros académicos de investigación, para brindar soporte a ensayos no comerciales y registros de pacientes. Posee una interfaz amigable para el usuario y es fácil de instalar, actualizar y validar.

MACRO cumple con las normas de Buenas Prácticas Clínicas (BPC), y con las directivas de la agencia reguladora europea. La utilización de algunos de estos sistemas en el CIM, quien tributa al CECMED, que es la agencia reguladora cubana, conllevaría a una reestructuración de la metodología de trabajo adoptada por el CIM y el resto del Polo Científico de manera más general, lo que se traduciría en un gasto importante de recursos y tiempo. Cualquier contrato subsiguiente con InferMed, de ser concertado, sería realizado bajo los términos de los acuerdos de la licencia del software InferMed Ltd., de los acuerdos de mantenimiento o cualquier otro que se detalle; creando una dependencia del Polo Científico de este producto.(GEORGIA 2008)

De adquirir MACRO, se incurriría en un gasto muy grande. Los costos de viaje y estancia, por parte de los miembros de InferMed que viajen a nuestro país, en cada ocasión que lo amerite, serían adicionados a la cuenta. Por otro lado, esta aplicación emplea como principal gestor de base de datos

a SQL Server, además de otros componentes y herramientas, cuyas licencias deben ser pagadas para su uso de manera legal.

2.2 Administrador de Estudio (Study Manager)

Es un sistema de gerencia de ensayo clínico windows-based, es una aplicación web creada para los investigadores profesionales, coordinadores, y los médicos que necesitan supervisar de cerca todas las fases de sus ensayos clínicos. Puesto que el Study Manager funciona sobre la web, puede ser accedida con seguridad de cualquier computadora con una conexión del Internet.

De acuerdo con Advanced Clinical Software (ACS, reveladores del Study Manager), más hospitales y universidades han elegido a Study Manager por encima de otros sistema de manejo de ensayos clínicos debido a su capacidad de reducir el coste y de hacer la investigación más eficiente, más productiva, más provechosa, y más agradable para los administradores de investigación, coordinadores, médicos, y otros miembros del personal.

Si usted es parte de un solo departamento o una oficina centralizada que desea manejar ensayos clínicos a través de departamentos múltiples, el Study Manager le da una sola base de dato, centralizada en la cual usted pueda seguir cualquier número de pacientes y de ensayos clínicos. La arquitectura flexible y de gran alcance del Study Manager se diseña de forma creciente para el uso multi-sitio y multi-departamento, pero tendría como inconveniente que crearía una dependencia directa con dicho sistema.

2.3 OpenClinica

OpenClinica es un sistema para manejar estudios clínicos, es de código abierto y basado en Java. Presenta una interfaz fácil de usar, permite la extracción de datos y filtrado de archivos, posibilita el intercambio de recursos de forma segura y transparente, admite exportación de herramientas para la migración de datos clínicos en las hojas de cálculo Excel y base de datos locales. OpenClinica presenta una infraestructura escalable, utiliza Java J2EE, con base de datos relacionales PostgreSQL incluidas y Oracle 10G, además usa el patrón de arquitectura Modelo Vista Controlador y Subversión como controlador de versiones. De las anteriores aplicaciones web para Sistemas de Manejo de Datos de EC, OpenClinica es la más aplicable a las necesidades del CIM, desde las tecnologías que usa hasta sus funcionalidades.

Con respecto a las funcionalidades que rigen el monitoreo dentro del sistema, el Centro de Inmunología Molecular no queda satisfecho con las mismas. Por lo que se hace necesario analizarlas

para su posterior modificación así como inclusión de otras nuevas de acuerdo con las necesidades de CIM.

3 Metodología y Herramientas a Utilizar

Se necesita analizar algunas de las herramientas y tecnologías que estén en correspondencia con los requisitos que se quiere, donde finalmente se llegara a escoger aquellas tecnologías que aporten más al buen funcionamiento del sistema luego de un estudio y análisis de las mismas.

La importancia que ha adquirido el procesamiento de datos de los ensayos clínicos a nivel mundial, ha provocado que en el mundo hayan surgido un sin número de sistemas automatizados dirigidos al control eficiente y veloz de dichos procesos. En este epígrafe se muestra un análisis de las principales tecnologías que se consultaron a la hora de desarrollar funcionalidades para el monitoreo de datos a partir del sistema OpenClinica.

3.1 Proceso de desarrollo de software

Una metodología es un conjunto ordenado de pasos a seguir para cumplir un objetivo. Dentro de la Ingeniería de Software, el objetivo es el desarrollo de software de alta calidad que cumpla con las necesidades del cliente, por lo que una metodología de ingeniería de software es: un conjunto de pasos ordenados a seguir para desarrollar software de alta calidad que cumpla con las necesidades del cliente.

En la actualidad si utilizan una gran cantidad de metodologías para el desarrollo de los procesos en el análisis y diseño, entre ellas se encuentra RUP.

3.2 RUP - Metodología de desarrollo de software OO (Orientado a Objetos)

Una metodología es un conjunto ordenado de pasos a seguir para cumplir un objetivo; dentro de la Ingeniería de Software, es un proceso, es decir, define como, cuando y que debo hacer para desarrollar un software de alta calidad que cumpla con las necesidades del cliente.

RUP es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del

más genérico, no es un sistema escalonado con pasos establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización que mejoran la productividad del equipo de trabajo, entregando las mejores prácticas del software a todos los miembros del mismo.

¿Por qué RUP?

Desde la consolidación de UML como lenguaje estándar para el modelado se ha definido un buen número de procesos para el desarrollo de aplicaciones orientadas a objetos que utilizan este lenguaje como medio de expresión de los diferentes modelos que se crean durante el ciclo de vida. Las características principales deseables en cualquier proceso software basado en UML están muy presentes en RUP, y son:

- RUP incluye cuatro fases: iniciación, elaboración, construcción y transición. Las fases de iniciación y elaboración incluyen las actividades de diseño del ciclo de vida del desarrollo, donde se hace énfasis al modelado del negocio, requisitos, análisis y diseño. Las fases de construcción y transición constituyen su producción, donde el énfasis recae sobre las actividades de implementación, pruebas y despliegue.

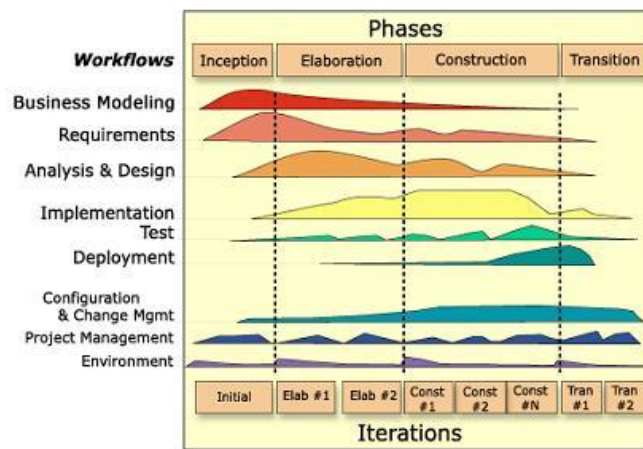


Ilustración 1 RUP

- ✓ Agrupa las actividades en grupos lógicos definiéndose flujos de trabajo principales (modelado del negocio, requerimientos, análisis y diseño, implementación, pruebas , instalación, administración del proyecto, administración de configuración y cambios y ambiente)
- ✓ RUP es un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos, permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini

proyecto se puede ver como una iteración del cual se obtiene un incremento que produce un crecimiento en el producto.

- ✓ En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo.
- ✓ RUP es Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño que constituyen la arquitectura del producto a desarrollar.
- ✓ En RUP se debe realizar control de cambios los que solucionan parte de las dificultades principales del desarrollo de software, como la comunicación entre equipos de desarrollo, la consistencia, la interferencia entre miembros de un equipo que trabajan en paralelo, etc.
- ✓ En RUP es importante verificar continuamente la calidad de un sistema con respecto a su funcionalidad, fiabilidad, rendimiento de la aplicación y rendimiento del sistema.
- ✓ RUP es más adaptable para proyectos a largo plazo, es basada en la documentación, en la que no son deseables todos esos cambios volátiles de las metodologías ágiles.
- ✓ Cuenta con diferentes elementos de planificación (plan de desarrollo, plan de iteración, plan de calidad, etc.) con los que se controla el desarrollo del software. A través de un esquema predefinido de escalabilidad y gestión de riesgos, se pueden reconocer previamente problemas y fallos de forma temprana y prevenirlos/corregirlos.

RUP es la metodología más rigurosa y la que mejor se adapta a la realización de proyectos grandes, en cuanto a tamaño y duración y a sistemas Orientados a Objetos. La utilización de RUP para el desarrollo del software conlleva a reflexionar y determinar que es la metodología que se requiere para el buen funcionamiento del sistema, luego de realizar el estudio de sus propias características.

3.3 Lenguaje de Modelado (UML Versión 2.0)

Según su definición, UML (Unified Modeling Language – Lenguaje Unificado de Modelado) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software desde una perspectiva orientada a objetos. En UML podemos encontrar elementos (abstracciones que constituyen los bloques básicos de construcción), relaciones (Ligan los elementos) y diagramas (Es la representación gráfica de un conjunto de elementos).

UML es un lenguaje que:

- ✚ Sus modelos pueden conectarse de forma directa con una gran variedad de lenguajes de programación.

- ✚ Proporciona un vocabulario y unas reglas que se centran en la representación conceptual y física de un sistema y que indican cómo crear y leer modelos bien formados.
- ✚ Mezcla gráficos y texto, de hecho, detrás de cada símbolo en la notación UML hay una semántica bien definida, de manera que un desarrollador puede escribir un modelo en UML, y otro desarrollador, o incluso otra herramienta, puede interpretar ese modelo sin ambigüedad.
- ✚ Cubre toda la documentación de la arquitectura de un sistema y todos sus detalles. También proporciona un lenguaje para expresar requisitos y pruebas del software.
- ✚ Especifica todas las decisiones de análisis, diseño e implementación que deben realizarse al desarrollar y desplegar un sistema con gran cantidad de software.
- ✚ Modela las actividades de planificación de proyectos y gestión de versiones.

¿Por qué UML?



Ilustración 2 UML

- ✚ UML es un lenguaje común que permite el crear una disciplina con estándares que tiene las siguientes características: Modela sistemas utilizando técnicas orientadas a objetos (OO).
- ✚ Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- ✚ Permite conectarse con lenguajes de programación (Ingeniería directa e inversa).
- ✚ Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- ✚ Cubre las cuestiones relacionadas con el tamaño propias de los sistemas complejos y críticos.
- ✚ Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- ✚ Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.

- ✚ UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir, no especificando en sí mismo qué metodología o proceso usar.

3.4 Herramienta CASE para la modelación del sistema.

Las Herramientas CASE (Computer Aided Software Engineering, del español "Ingeniería de Software Asistida por Ordenador") son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Visual Paradigm SUITE. (Versión 3.4)



Ilustración 3 Visual Paradigm

Es una potente herramienta para visualizar y diseñar elementos de software, para ello utiliza el lenguaje UML, proporciona a los desarrolladores una plataforma que les permita diseñar un producto con calidad de una forma rápida. Facilita la interoperabilidad con otras herramientas CASE y se integra con las siguientes herramientas Java: Eclipse/IBM WebSphere, Jbuilder, NetBeans IDE, Oracle Jdeveloper, BEA Weblogic. Está disponible en varias ediciones: Enterprise, Professional, Community, Standard, Modeler y Personal.

3.5 Herramienta para el desarrollo

Al fin de seguir con las mismas herramientas del sistema OpenClinica.

J2EE (Java Enterprise Edition)

OpenClinica está desarrollado en esta plataforma de programación, forma parte de la plataforma Java la que se encarga del desarrollo y ejecución de software y aplicaciones en el lenguaje Java, con una arquitectura distribuida en N niveles basada en componentes modulares de software ejecutándose sobre un servidor de aplicaciones. Java EE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans, servlets, portlets (siguiendo la especificación de Portlets Java), JavaServer Pages y varias tecnologías de servicios web.

Esto permite al desarrollador:

- ✚ Crear una Aplicación de Empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores.
- ✚ Concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel.
- ✚ La seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados.

Además de otros beneficios añadidos como es por ejemplo, que el servidor de aplicaciones puede manejar transacciones.

Java es un lenguaje neutral, portable, robusto, estable e independiente de la plataforma, fácil de aprender por los desarrolladores con una experiencia previa con otros lenguajes orientados a objetos, presenta compatibilidad con distintos sistemas operativos, incluye el uso de Servlets a la hora de gestionar conexiones con servidores, ya que tratan múltiples conexiones como una sola.

3.5.1 Eclipse (versión 3.4.2)



Ilustración 4 Eclipse

Es una Herramienta multiplataforma integrado gratuito y de código abierto, utilizado para varios lenguajes (Java, C, C++, PHP) .Originalmente desarrollado por International Business Machines (IBM) además utilizada para entornos integrados de desarrollo. Se le han añadido diversos plugins para proveer nuevas funcionalidades al programa enfocadas a las distintas necesidades de los desarrolladores, permite el control de versiones con Subversión, compila en tiempo real.

Instalar dentro de Eclipse los componentes necesarios para beneficiarnos de todas las ventajas para programar en PHP era una posibilidad, pero quitaría mucho más trabajo, tiempo de configuración al instalar el PDT Project. , distribución de Eclipse que incorpora, de casa, todas las utilidades creadas para simplificar la vida a los desarrolladores de PHP.

3.5.2 Apache Tomcat (versión 5.5.27)



Ilustración 5 Apache Tomcat (versión 5.5.27)

Muchas empresas lo usan en entornos de producción debido a su gran estabilidad, fácil de instalar con soporte de servlets y JSP's (Java Server Pages). Incluye el compilador llamado Jasper, que compila las JSP's y las convierte en servlets. Puede ejecutarse en maquinas más pequeñas y es compatible con las API más recientes de Java, escrito en Java por lo tanto requiere del sistema operativo que tenga instalada la Maquina Virtual de Java...Tomcat, ocupa muy poco espacio en total 1MB, por lo tanto se ejecuta muy veloz. La solidez de Tomcat se basa en que miles de desarrolladores contribuyen con código. Tomcat pone a disposición de todo el mundo las últimas actualizaciones de Java.

3.5.3 Sistema gestor de base de datos

En el mundo existen gran variedad de sistemas gestores de base de datos, los cuales son muy específicos en sus características de almacenamiento de datos, así como en el control de la integridad, confidencialidad y seguridad. Estos sistemas son el enlace entre los usuarios, la BD y las aplicaciones que se utilizan. Se pueden destacar MySQL que es un sistema de base de datos relacional y es uno de los más usados en el mundo, debido a su rapidez y su facilidad de uso, es multiplataforma. Oracle es un sistema de base de datos multiplataforma, permite implementar diseños con disparadores y procedimientos almacenados, tienes como elemento negativo que no es libre, y el precio de su licencia es excesivamente elevado, además de la necesidad de configurarlo para su correcto funcionamiento.

¿Por qué PostgreSQL?



Ilustración 6 PostgreSQL (Gestor de BD) (versión 8.2.9.1)

El Sistema Gestor de Base de Datos conocido como PostgreSQL se deriva del paquete Postgre que en un inicio se llamo Postgre95. Luego de más de una década de desarrollo él, PostgreSQL se ha convertido en el gestor de base de datos de **código abierto** más avanzado, ofreciendo control de concurrencia multi-versión, soportando en casi su totalidad la sintaxis SQL, incluyendo consultas, transacciones, tipos y funciones definidas por los usuarios; constando también con un amplio enlace con lenguajes de programación como PHP, C, C++, Python, **Java**, entre otros.

PostgreSQL está enmarcado dentro de la categoría de la Base de Datos identificada como objeto-relacional, ya que ofrece una potencia adicional al incorporar conceptos básicos que le permiten al usuario extender en gran medida el sistema. Presenta una alta concurrencia, mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés), permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

3.6 Herramienta de control de versiones

Las herramientas de control de versiones, facilitan el control, revisión o edición de las distintas versiones que pueda tener un producto en desarrollo o culminado.

¿Por qué Subversion?

Subversion es un software diseñado para el control de versiones, específicamente para reemplazar al popular CVS (Concurrent Versions System). Es libre y se conoce como (SVN). Una potente característica es que todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en un determinado momento en el tiempo. Mencionar algunas características que lo hacen más potente con respecto a otros sistemas de control de versiones:

- ✚ La Historia de los archivos la sigue a través de copias y renombrados.
- ✚ Las modificaciones son atómicas.
- ✚ La creación de ramas y etiquetas es una operación más eficiente; Tiene costo de complejidad constante ($O(1)$) y no lineal ($O(n)$) como en CVS.
- ✚ Se envían sólo las diferencias en ambas direcciones.
- ✚ Puede ser servido mediante Apache, sobre WebDAV/DeltaV.
- ✚ Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.
- ✚ Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos (SQL, LDAP, PAM).

Conclusiones

El presente capítulo es el resultado del análisis y búsqueda de la información vinculada al proceso de conducción de los Ensayos Clínicos Cubanos en el sistema OpenClinica, se realiza el análisis del estado del arte del tema tratado y se describe la justificación de las herramientas, tecnologías y metodologías utilizadas, para adaptar el monitoreo de los datos del sistema OpenClinica al proceso de conducción de los Ensayos Clínicos Cubanos.

CAPÍTULO 2: Características del Sistema

Introducción

En el presente capítulo se describe la modelación del negocio (Modelo de dominio) con el objetivo de comprender y describir las clases más importantes dentro del contexto del sistema y sus relaciones. Contiene además la captura de los requisitos funcionales y no funcionales, el diagrama de casos de uso del sistema, así como sus respectivas descripciones de los casos de uso considerados más significativos. Como parte de la solución del sistema alasClínicas a partir del estudio realizado se levantaron 19 requisitos funcionales agrupados en 9 Casos de Uso.

1 Modelo de Dominio

Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema, con el fin de establecer un entendimiento y tener un vocabulario común. La presente sección describe detalladamente los conceptos más importantes y sus relaciones. Para poder entender el contexto en que se desarrolla el sistema necesitamos definir conceptos que podemos agrupar en un Modelo de Dominio.

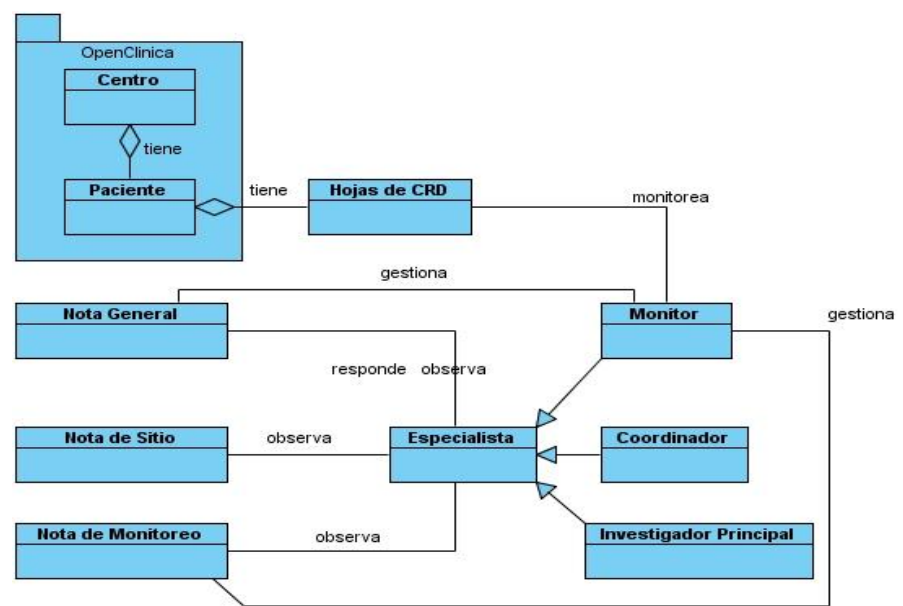


Ilustración 7 Modelo de Dominio

2 Requerimientos

Los requerimientos son condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. (IS1 2007-2008)

Es de vital importancia levantar los requerimientos del sistema para lograr un correcto funcionamiento del sistema y evitar inquietudes con el cliente. Durante una profunda investigación de OpenClinica y entrevistas con el cliente se levantaron tantos requisitos funcionales como no funcionales.

2.1 Requisitos funcionales

Son capacidades o condiciones que el sistema debe cumplir, son el punto de partida para identificar que debe hacer el sistema OpenClinica.

Requisitos Funcionales:

- RF1: Monitorear datos de una Hoja de CRD.
- RF2: Modificar estado de una hoja CRD
- RF3: Modificar estado a un Momento de Seguimiento
- RF4: Crear nota de monitoreo.
- RF5: Modificar estado de una nota de monitoreo.
- RF6: Responder nota de monitoreo.
- RF7: Crear nota general.
- RF8: Modificar estado de una nota general.
- RF9: Responder nota general.
- RF10: Visualizar Notas
- RF11: Listar notas del sitio en un centro.
- RF12: Listar notas de monitoreo de un centro.
- RF13: Listar notas generales de un centro.
- RF14: Mostrar informe de monitoreo para las Notas de Monitoreo.
- RF15: Mostrar informe de monitoreo para las Notas de Sitio.
- RF16: Mostrar informe de monitoreo para las Notas Generales.
- RF17: Mostrar detalles de la Nota de Monitoreo.
- RF18: Mostrar detalles de la Nota de Sitio.
- RF19: Mostrar detalles de la Nota General.

2.2 Requisitos no Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Una de las ventajas de OpenClinica es su capacidad de montarse en diferentes sistemas operativos con una fácil instalación. Destacándose requisitos no funcionales de Software, Soporte y Hardware.

2.2.1 RNF de Apariencia o Interfaz Externa

- ✓ Las páginas no tendrán muchas imágenes y poseerán pocos colores.
- ✓ Las páginas principales tendrán información que servirá de guía al usuario.
- ✓ Cada rol tendrá una interfaz diferente con las funciones que le corresponden.
- ✓ Se hará uso de simbología mediante íconos para indicar el estado de los elementos utilizados en el diseño. Además, los íconos contendrán funcionalidades específicas.

2.2.2 RNF de Usabilidad

- ✓ La aplicación tendrá un ambiente sencillo y será fácil de manejar para los usuarios.

2.2.3 RNF de Soporte

- ✓ Una vez terminada la aplicación se instalará en el CIM para realizar pruebas piloto del software y pruebas de despliegue.

2.2.4 RNF de Seguridad

- ✓ El acceso a cualquier manipulación del sistema, tanto entrada como análisis de datos debe estar sometido a un proceso de autenticación del usuario donde será especificado el rol, usuario y contraseña.
- ✓ Cada usuario tendrá asignado uno o varios roles en el sistema. Cada rol definido tendrá niveles de acceso al Software.
- ✓ Todo cambio o modificación en el sistema debe ser atribuible a un usuario particular según su autenticación.

2.2.5 RNF de Software

- ✓ Para la instalación de la aplicación se debe disponer del sistema operativo Windows o GNU Linux.
- ✓ Servidor de Base de Datos Postgres 8.2.x
- ✓ Navegador: Internet Explorer 6.* o superior, Mozilla Firefox 3.* o superior
- ✓ Web Server: Apache Tomcat 5.* o superior

- ✓ Debe estar instalado máquina virtual de Java SDK 1.5.x

2.2.6 RNF de Hardware

- ✓ Mínimo memoria RAM 256 (1Gb Recomendado)
- ✓ El servidor de Base de Datos debe tener 1 a 3 TB de espacio disponibles pues el volumen de información es bastante grande y perdura en el tiempo hasta 15 años.

2.2.7 RNF de Disponibilidad

- ✓ La aplicación debe mantenerse funcionando las 24 horas del día y los siete días de la semana.
- ✓ El servidor de aplicación debe soportar un aumento de usuarios concurrentes por minuto de 1 a 400.

2.2.8 RNF de Licencia



Ilustración 8 Licencia

Se estará usando una herramienta de software libre, licencia de código abierto GNU/GPL. La licencia GPL, al ser un documento que cede ciertos derechos al usuario, asume la forma de un contrato, por lo que usualmente se la denomina contrato de licencia o acuerdo de licencia.

2.2.9 RNF de Estándares Aplicables

Basado en estándares abiertos e independientes (CDISC ODM, HL7) o Robusto y modular diseñada con Java/J2EE incrementando el rendimiento del sistema y permitiendo la personalización o Base de Datos Independiente (Postgres (por defecto), Oracle, MySQL, SQL Server).(BLANCO 2009)

3 Patrones de Casos de Uso

3.1 Múltiples actores (roles comunes):

Los dos actores juegan el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol.

4 Descripción de los Casos de Usos del Sistema

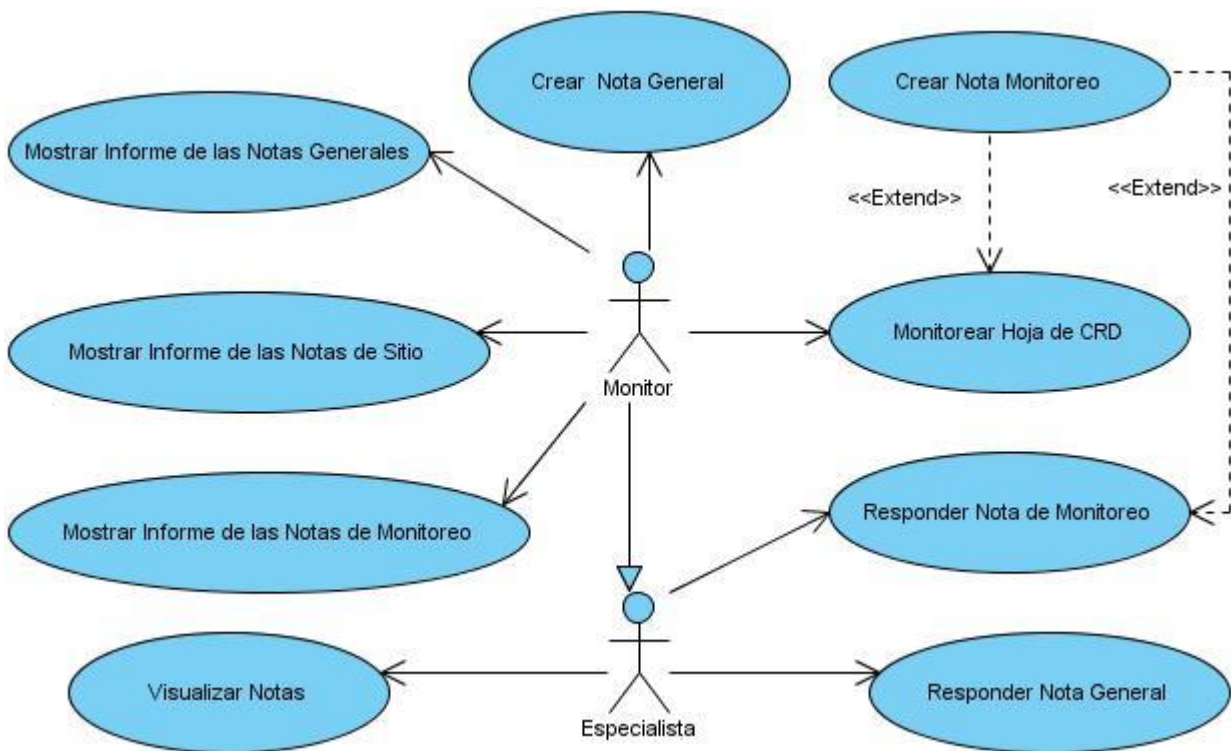


Ilustración 9 Diagrama de Casos de Uso del Sistema.

4.1 Actores del Sistema

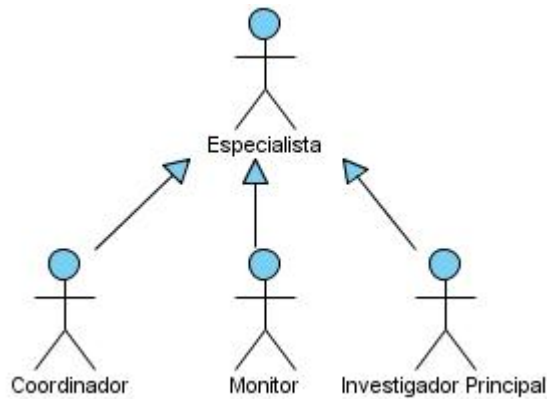


Ilustración 10 Diagrama de Actores del Sistema.

| Actor | Descripción |
|---|--|
| <p>Especialista:</p> <ul style="list-style-type: none"> ➤ Investigador Principal ➤ Coordinador ➤ Monitor | <p>Es el encargado de responder notas generales, de monitoreo y visualizar todas las notas del sistema. El Investigador Principal y el Coordinador solo pueden modificar el estado de las notas a <i>‘actualizado’</i>. Y el Monitor modifica las notas a estado <i>‘nuevo’</i>, <i>‘actualizado’</i> <i>‘cerrado’</i> y <i>‘resuelto’</i>.</p> <p>Un especialista puede tener rol de investigador principal, de coordinador o de monitor.</p> |
| <p>Monitor</p> | <p>Rol encargado de monitorear los datos de un centro, así como crear notas generales y de monitoreo. También realiza un informe de monitoreo.</p> |

4.2 CU: Monitorear Hoja de CRD

| | |
|---|---|
| Caso de Uso: | Monitorear Hoja de CRD |
| Actores: | Monitor (Inicia) |
| Resumen: | Desde una Hoja de CRD asociada a un Momento de Seguimiento de un Paciente el monitor revisa los datos, modifica el estado de la Hoja de CRD y del Momento de Seguimiento a ' <i>monitoreo iniciado</i> ' o ' <i>monitoreo completado</i> '. Si es necesario el Monitor crea una Nota de Monitoreo sobre una variable de la Hoja de CRD. |
| Precondiciones: | El Monitor debe haberse autenticado en el sistema. |
| Referencias | RF1, RF2, RF3 |
| Prioridad | Crítico |
| Flujo Normal de Eventos | |
| | |
| Acción del Actor | Respuesta del Sistema |
| 1- Desde una Hoja de CRD asociada a un Momento de Seguimiento de un Paciente el monitor selecciona ' <i>monitoreo iniciado</i> '. | 2- El sistema muestra un mensaje de advertencia indicando la necesidad de firmar con la contraseña del Monitor para cambiar a ' <i>monitoreo iniciado</i> '. |
| 3- El Monitor introduce su contraseña | 4- El sistema verifica la contraseña. |
| 5- El Monitor revisa los datos. | |
| 6- El Monitor guarda los cambios. | 7- El sistema modifica el estado de la Hoja de CRD y del Momento de Seguimiento. El sistema vuelve al listado de Hojas de CRD del actual paciente. Finaliza el Caso de Uso |
| Flujos Alternos | |

| Acción del Actor | Respuesta del Sistema |
|--|---|
| 1.1 El Monitor selecciona "monitoreo completado" | 7.2 El sistema muestra un mensaje de advertencia indicando la necesidad de firmar con la contraseña del Monitor para cambiar a "monitoreo completado". Ir al paso 3 |
| | 4.1 En caso que la contraseña sea incorrecta. El sistema muestra un mensaje de error .Desmarcando la opción seleccionada. Ir al paso 1 |
| 5.1 El Monitor escoge "Crear Nota de Monitoreo" | 5.2 Ir a la descripción del caso de uso " Crear Nota de Monitoreo ". Ir al paso 5 |

Prototipo de Interfaz

Monitoreo Iniciado para Examen Físico

Info hoja CRD-

| | |
|---|---|
| Examen Físico | Notas de Discrepancia: 0 Abierto, 0 Actualizado, 0 Resuelto, 0 Cerrado, 0 Not Applicable |
| ID del Sujeto del Estudio: nemo | ID de la Persona: nemo_oid |
| Estudio/Centro: Default Study | Edad en la Inclusión: 23 Años - 6 Meses - 18 Días |
| Momento de Seguimiento: asd(23/01/2009) | Fecha de Nacimiento: 06/07/1985 Sexo: V |
| Nombre del Entrevistador: * lalo <input type="text"/> | Fecha de la Entrevista: * 04/01/2009 <input type="text"/> |

Cron_CRD -- Seleccione para Ir --

| | | |
|--|-------------|---|
| Título: QQQQQQQ | | |
| Subtítulo: QQQQQQQ | | |
| Instrucciones: QQQQQQQ | | |
| <input type="radio"/> Monitoreo Comenzado <input type="radio"/> Monitoreo Completado | | <input type="button" value="Guardar"/> <input type="button" value="Salir"/> |
| 2 A | 2 A | |
| 170 * (12) | 140 * (12) | X |
| 170 * (12) | 140 * (12) | X |
| 170 * (12) | 170 * (12) | X |
| 170 * (12) | 170 * (12) | X |
| 170 * (12) | 170 * (12) | X |
| 170 * (12) | 170 * (12) | X |
| 170 * (12) | 170 * (12) | X |
| 170 * (12) | 170 * (12) | X |
| ANAD | | |
| <input type="radio"/> Monitoreo Comenzado <input type="radio"/> Monitoreo Completado | | <input type="button" value="Guardar"/> <input type="button" value="Salir"/> |

Poscondiciones

Queda monitoreada la Hoja de CRD

4.3 CU: Visualizar Notas

| | |
|---|--|
| Caso de Uso: | Visualizar Notas |
| Actores: | Especialista (Inicia) |
| Resumen: | El caso de uso se inicia cuando un especialista desde el menú principal escoge “ <i>Notas y Discrepancias</i> ”. El sistema muestra un listado de notas dividido en 3 secciones: Notas de Monitoreo. Notas de Sitio y Notas Generales. |
| Precondiciones: | Algún especialista debe estar autenticado en el sistema. |
| Referencias | RF10, RF11, RF12, RF13, RF18, RF19, RF17 |
| Prioridad | Crítico |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| 1- Del menú principal el Especialista selecciona “ <i>Notas y Discrepancias</i> ” | <p>2- El sistema muestra un listado de todas las notas del centro dividido en 3 secciones: <i>Notas de Monitoreo, Notas de Sitio y Notas Generales.</i></p> <p>Para las Notas de Monitoreo el sistema muestra:</p> <ul style="list-style-type: none"> ➤ El paciente con sus MS asociados ➤ La cantidad de Notas de Monitoreo para cada paciente en el MS. Las notas están agrupadas según su estado de resolución (“Nuevo”, “Actualizado”, “Resuelto”, “Cerrado”). <p>Para las Notas de Sitio el sistema muestra:</p> <ul style="list-style-type: none"> ➤ El paciente con sus MS asociados ➤ La cantidad de Notas de Sitio para cada paciente en el MS. |

| | |
|---|--|
| | <p>Para las Notas Generales el sistema muestra:</p> <ul style="list-style-type: none"> ➤ ID de la Nota ➤ Fecha de Creación ➤ Fecha de Actualización ➤ Descripción ➤ Notas Detalladas ➤ Numero de Notas ➤ Estado de Resolución ➤ Tipo de Nota ➤ Autor <p>Y el sistema permite al Especialista realizar una de las siguientes acciones:</p> <ul style="list-style-type: none"> ➤ Listar todas las Notas de Monitoreo. ➤ Listar todas las Notas de Sitio ➤ Listar todas las Notas Generales <p>Finaliza el Caso de Uso</p> |
| <p>3- El Especialista va a realizar un de las siguientes acciones:</p> <ol style="list-style-type: none"> a) Listar todas las Notas de Monitoreo. b) Listar todas las Notas de Sitio c) Listar todas las Notas Generales | <p>4- El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> ✓ Si el Especialista hace (a) el sistema muestra un listado de todas las notas de monitoreo del centro, ir a la sección: “Listar Notas de Monitoreo”. ✓ Si el Especialista hace (b) el sistema muestra un listado de todas las notas de sitio del centro, ir a la sección: “Listar Notas de |

Sitio”.

- ✓ Si el Especialista hace (c) el sistema muestra un listado de todas las notas generales de centro, ir a la sección: **“Listar Notas Generales”**

Prototipo de Interfaz

Administrar todas las notas del centro

Filtrar por estado : All Notes (Nuevo) (Actualizado) (Resuelto) (Cerrado) [#] = Momentos de seguimiento repetidos

Filtrar por tipo de nota : Todas las Notas | Anotación | Comprobación de Validación Fallida | Consulta

Notas de Monitoreo

Todos Momento Seguimiento 1 Momento Seguimiento 2 Momento Seguimiento 3

Página 1 de 1

| ID del Sujeto del Estudio | Estatus del Sujeto | Nombre del Centro | Momento Seguimiento #1 | Momento Seguimiento #2 | Momento Seguimiento #3 | Acciones |
|---------------------------|--------------------|-------------------|------------------------|------------------------|------------------------|----------|
| subject_1 | activo | default-study | (1) (1) [3] | | | |

Notas de Sitio

Todos Momento Seguimiento 1 Momento Seguimiento 2 Momento Seguimiento 3

Página 1 de 1

| ID del Sujeto del Estudio | Estatus del Sujeto | Nombre del Centro | Momento Seguimiento #1 | Momento Seguimiento #2 | Momento Seguimiento #3 | Acciones |
|---------------------------|--------------------|-------------------|------------------------|------------------------|------------------------|----------|
| subject_2 | activo | default-study | (4) | | | |
| subject_1 | activo | default-study | | (1) | | |

Notas Generales

Página 1 de 1 Crear Nota General

| Nombre del Centro | Fecha de Creación | Fecha de Actualización | Descripción | Notas Detalladas | Nº de Notas | Estado de resolución | Tipo | Autor | Acciones |
|-------------------|-------------------|------------------------|-------------|------------------|-------------|----------------------|------|-------|----------|
| | | | | | | | | | |

Sección “Listar Notas de Monitoreo”

Acción del Actor

Respuesta del Sistema

- 1- El sistema muestra una interfaz con todas las Notas de Monitoreo del centro. Permitiendo filtrarlas por su estado o su tipo. Para cada Nota de Monitoreo el sistema muestra:

| | |
|--|---|
| | <ul style="list-style-type: none"> • Paciente. • Fecha de Creación. • Fecha de Actualización. • Fecha del Momento de Seguimiento • Momento de Seguimiento • CRD. • Nombre del Dato. • Valor del Dato. • Número de notas. • Estado de Resolución. • Tipo. • Tipo de Entidad. • Descripción. • Notas Detalladas • Autor. |
| <p>2-El Especialista selecciona "Ver Nota"</p> | <p>3- El sistema crea una ventana con los detalles de la nota:</p> <ul style="list-style-type: none"> • Paciente. • Fecha de Creación. • Fecha de Actualización. • Fecha del Momento de Seguimiento. • Momento de Seguimiento. • CRD. • Nombre del Dato. • Valor del Dato. • Número de notas. • Estado de Resolución. • Tipo. • Tipo de Entidad. • Descripción. • Notas Detalladas. • Autor. |

| Finaliza el Caso de Uso | |
|---|--|
| Prototipo de Interfaz | |
| | |
| Sección “Listar Notas de Sitio” | |
| Acción del Actor | Respuesta del Sistema |
| | <p>1- El sistema muestra una interfaz con todas las Notas de Sitio del centro. Para cada Nota de Sitio el sistema muestra</p> <ul style="list-style-type: none"> • Paciente. • Fecha de Creación. • Fecha de Actualización. • Fecha del Momento de Seguimiento. • Momento de Seguimiento. • CRD. • Nombre del Dato. • Valor del Dato. • Tipo de Entidad. • Descripción. • Notas Detalladas. • Autor. |
| <p>2- El Especialista selecciona “Ver Nota”</p> | <p>3- El sistema crea una ventana con los detalles de la nota:</p> <ul style="list-style-type: none"> • Paciente. • Fecha de Creación. |

| | |
|--|--|
| | <ul style="list-style-type: none"> • Fecha de Actualización. • Fecha del Momento de Seguimiento. • Momento de Seguimiento. • CRD. • Nombre del Dato. • Valor del Dato. • Número de notas. • Tipo. • Estatus. • Tipo de Entidad. • Descripción. • Notas Detalladas. • Autor. <p>Finaliza el Caso de Uso</p> |
|--|--|

Prototipo de Interfaz

Ver Notas de Sitio

| ID del Sujeto del Estudio | Fecha de Creación | Fecha de Actualización | Fecha del Momento de Seguimiento | Momento de Seguimiento | CRD | Nombre de la Entidad | Valor de la Entidad | Descripción | Notas Detalladas | Tipo de Entidad por | Creado | Acciones |
|---------------------------|-------------------|------------------------|----------------------------------|------------------------|-----|----------------------|---------------------|-------------|------------------|---------------------|--------|----------|
| | | | | | | | | | | | | |

Ver Nota de Sitio

Fecha de la Entrevista = 13-abr-2009

| | |
|---------------------------------------|-----------------------------|
| Sujeto: | Momento de Seguimiento: N/A |
| Fecha del Momento de Seguimiento: N/A | CRD: N/A |

Detalles de la Nota

| | | |
|------------------|----------------------------------|-------------------------------------|
| sitio2 | | Last updated: 25/05/2009 by monitor |
| Tipo: Sitio | Estado de resolución: Comentario | # Notas: 1 |
| sitio2 | | 25/05/2009 by monitor |
| undefinidasdadsa | | Estatus: Comentario |

Sección “Listar Notas Generales”

| Acción del Actor | Respuesta del Sistema |
|------------------|--|
| | <p>1- El sistema muestra una interfaz con todas las Notas Generales del centro. Permitiendo filtrarlas por su estado o su tipo. Para cada Nota de Sitio el sistema muestra</p> <ul style="list-style-type: none"> • ID. • Fecha de Creación. • Fecha de Actualización. • Descripción. • Notas Detalladas. |

| | |
|--|---|
| | <ul style="list-style-type: none"> • Número de Notas • Autor. |
| 2- El Especialista selecciona "Ver Nota" | <p>3- El sistema crea una ventana con los detalles de la nota:</p> <ul style="list-style-type: none"> • Fecha de última Actualización. • Número de notas. • Tipo. • Descripción. • Notas Detalladas. • Autor. <p>Finaliza el Caso de Uso</p> |

Prototipo de Interfaz

Ver Notas Generales ? 📄

Filtrar por estado: All Notes (Nuevo) (Actualizado) (Resuelto) (Cerrado)

Filtrar por tipo de nota: Todas las Notas | Anotación | Consulta

Añadir Nota General

| ID | Fecha de Creación | Fecha de Actualización | Descripción | Notas Detalladas | # Notas | Estado de resolución | Tipo | Autor | Acciones |
|----|-------------------|------------------------|-------------|------------------|---------|----------------------|------|-------|----------|
| | | | | | | | | | 🔍 📄 |

Ver Nota General

Detalles de la Nota

📄 qweqwe Last updated: 28/05/2009 by monitor

| | | |
|-----------------|--------------------------------|------------|
| Tipo: Anotación | Estado de resolución: Resuelto | # Notas: 1 |
|-----------------|--------------------------------|------------|

qweqwe 28/05/2009 by monitor Estatus: Resuelto

4.4 CU: Responder Nota de Monitoreo

| | |
|------------------------|---|
| Caso de Uso: | Responder Nota de Monitoreo |
| Actores: | Especialista(Inicia) |
| Resumen: | El caso de uso inicia cuando el Especialista del listado de Notas de Monitoreo escoge "Resolver Nota". El sistema le muestra la interfaz correspondiente y ejecuta las acciones necesarias. El caso de uso finaliza cuando el Monitor guarda los cambios. |
| Precondiciones: | El Especialista debe haberse autenticado en el sistema. |

| Referencias | RF6, RF5,RF17 |
|---|--|
| Prioridad | Crítico |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| 1- Del listado de Notas de Monitoreo el Especialista selecciona “ <i>Resolver Nota</i> ”. | 2- El sistema muestra una interfaz con los datos de la nota: <ul style="list-style-type: none"> • Nombre de la Entidad • Valor de la entidad • Sujeto • Momento de Seguimiento • Fecha del Momento de Seguimiento • CRD • Descripción de la Nota • Tipo • Estado de Resolución • N° de Notas • Fecha de Creación • Autor • Estatus • Detalles y Descripción |
| 3- El Especialista escoge “Responder Nota” | 4- El sistema muestra una interfaz en la cual automáticamente llena los siguientes campos: <ul style="list-style-type: none"> • Sujeto • Momento de Seguimiento • CRD • Entidad/Campo • Nombre del Ítem • Nota padre • Fecha • Tipo El sistema deshabilita el campo “ <i>Tipo</i> ”. Y permite al Especialista llenar los siguientes: <ul style="list-style-type: none"> • Descripción • Nota Detallada • Estado de Resolución |

| | |
|---|---|
| <p>5- El Especialista edita los campos:</p> <ul style="list-style-type: none"> • Descripción • Nota Detallada <p>El campo “<i>Estado de Resolución</i>” es seleccionado, pudiendo ser:</p> <ul style="list-style-type: none"> ○ Actualizado ○ Resuelto ○ Cerrado | |
| <p>6- El Monitor guarda los cambios.</p> | <p>7- El sistema salva la nota en un registro de notas y emite un mensaje de éxito.</p> |

Prototipo de Interfaz

Flujos Alternos

| Acción del Actor | Respuesta del Sistema |
|---|---|
| <p>3.1 El Monitor escoge “<i>Nueva Nota de Monitoreo</i>”</p> | <p>3.2 Ir a la descripción del CU Crear Nota de Monitoreo.</p> |
| <p>Pos condiciones</p> | <p>Queda Respondida una Nota de Monitoreo</p> |

4.5 CU: Responder Nota General

| | |
|---|--|
| Caso de Uso: | Responder Nota General |
| Actores: | Especialista (Inicia). |
| Resumen: | El caso de uso inicia cuando el Especialista del listado de notas generales escoge responder una nota general. El sistema le muestra la interfaz correspondiente y ejecuta las acciones necesarias. El caso de uso finaliza cuando el Especialista guarda los cambios. |
| Precondiciones: | El Especialista debe haberse autenticado en el sistema. Debe existir alguna Nota General |
| Referencias | RF8, RF9, RF19 |
| Prioridad | Crítico |
| Flujo Normal de Eventos | |
| Acción del Actor | |
| Respuesta del Sistema | |
| 1- Del listado de Notas el Especialista escoge <i>“Resolver Nota General”</i> | 2- El sistema muestra una interfaz con los datos de la nota: <ul style="list-style-type: none"> • ID de la nota • Tipo • Estado de Resolución • Nº de Notas • Fecha de Creación • Autor • Estatus • Detalles y Descripción |
| 3- El Especialista escoge <i>“Responder Nota”</i> | 4- El sistema muestra una interfaz en la cual automáticamente llena los siguientes campos: <ul style="list-style-type: none"> • Nota padre • Fecha • Tipo El sistema deshabilita el campo <i>“Tipo”</i> |

| | |
|--|--|
| | <p>Y permite al Monitor llenar los siguientes:</p> <ul style="list-style-type: none"> • Descripción • Nota Detallada • Estado de Resolución <ul style="list-style-type: none"> ○ Actualizado ○ Resuelto ○ Cerrado |
| <p>5- El Especialista edita los campos:</p> <ul style="list-style-type: none"> • Descripción • Nota Detallada <p>El campo Estado de Resolución es seleccionado, pudiendo ser:</p> <ul style="list-style-type: none"> ○ Actualizado ○ Resuelto ○ Cerrado | |
| <p>6- El Especialista guarda los cambios.</p> | <p>7- El sistema salva la nota en un registro de notas y emite un mensaje de éxito.</p> <p>Finaliza el Caso de Uso</p> |
| <p>Prototipo de Interfaz</p> | |

Responder Nota General

Tipo Anotación Consulta

Descripción

Nota Detallada

Estado de resolución Actualizado

Fecha 16/04/2009

Nota Padre dddddd

Ver Notas de Discrepancia

Datos de la Nota

| | | | |
|--------|----------------|--|----------------|
| asd | | Last updated: 16/04/2009 by asotolongo | |
| ID: 51 | Tipo: Consulta | Estado de resolución: Nuevo | Nº de Notas: 1 |
| asd | | 16/04/2009 by asotolongo | Estatus: Nuevo |
| dddd | | | |
| | | Responder Nota | |

[Crear Nueva Nota](#)

| Flujos Alternos | |
|--|---|
| Acción del Actor | Respuesta del Sistema |
| 3.1 El Monitor escoge "Crear Nota General" | 4.1 Si el especialista es el Monitor 3.2 Ir a la descripción del CU Crear Nota General. |
| Pos condiciones | Queda respondida una Nota General |

4.6 CU: Crear Nota de Monitoreo

| | |
|------------------------|--|
| Caso de Uso: | Crear Nota de Monitoreo |
| Actores: | Monitor(Inicia) |
| Resumen: | El caso de uso se inicia cuando el monitor selecciona "Crear Nota de Monitoreo" El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando el Monitor decide guardar los cambios. |
| Precondiciones: | El monitor debe haberse autenticado en el sistema. |
| Referencias | RF4,RF5 |

| | |
|---|--|
| Prioridad | Crítico |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| | <p>1- El sistema muestra una interfaz en la cual automáticamente llena los siguientes campos:</p> <ul style="list-style-type: none"> • Sujeto • Momento de Seguimiento • CRD • Entidad/Campo • Nombre del Ítem • Fecha • Nota Padre <p>Y permite al monitor llenar los siguientes:</p> <ul style="list-style-type: none"> • Tipo <ul style="list-style-type: none"> ○ De anotación ○ Comprobación de Validación Fallida ○ Consulta • Descripción • Nota Detallada • Estado de Resolución <ul style="list-style-type: none"> ○ Nuevo ○ Actualizado ○ Resuelto ○ Cerrado |
| <p>2- El Monitor escoge el tipo, pudiendo ser:</p> <ul style="list-style-type: none"> ○ De anotación ○ Comprobación de Validación Fallida ○ Consulta | |
| <p>3- El Monitor edita los campos:</p> | |

| | |
|---|---|
| <ul style="list-style-type: none"> • Descripción • Nota Detallada | |
| <p>4- El campo Estado de Resolución es seleccionado por el Monitor, pudiendo ser:</p> <ul style="list-style-type: none"> ○ Nuevo ○ Actualizado ○ Resuelto ○ Cerrado | |
| <p>5- El Monitor guarda los cambios</p> | <p>6- El sistema salva la nota en un registro de notas y emite un mensaje de éxito.</p> <p>Finaliza el Caso de Uso</p> |

Prototipo de Interfaz

| | |
|------------------------|---|
| | <div style="border: 1px solid #ccc; padding: 10px;"> <p style="color: #4CAF50; margin: 0;">Añadir Nota de Monitoreo</p> <p>Sujeto <input type="text"/></p> <p>Momento de Seguimiento <input type="text"/></p> <p>Fecha del Momento de Seguimiento <input type="text"/></p> <p>CRD <input type="text"/></p> <p>Entidad Tipo/Campo <input type="text"/></p> <p>Nombre del Item <input type="text"/></p> <p>Tipo <input type="radio"/> Comprobación de Validación Fallida <input type="radio"/> Anotación <input type="radio"/> Consulta</p> <p>Descripción <input type="text"/></p> <p>Nota Detallada <input style="height: 40px;" type="text"/></p> <p>Estado de resolución <input type="text" value="Nuevo"/></p> <p>Fecha <input type="text"/></p> <p>Nota Padre <input type="text"/></p> <p style="text-align: center;"><input type="button" value="Enviar"/></p> </div> |
| Pos condiciones | Queda creada una nota de Monitoreo |

4.7 CU: Crear Nota general

| | |
|--|--|
| Caso de Uso: | Crear Nota General |
| Actores: | Monitor(Inicia) |
| Resumen: | <p>El caso de uso se inicia cuando el monitor selecciona “<i>Crear Nota General</i>”</p> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando el Monitor decide guardar los cambios.</p> |
| Precondiciones: | El monitor debe haberse autenticado en el sistema. |
| Referencias | RF7,RF8 |
| Prioridad | Crítico |
| Flujo Normal de Eventos | |
| Acción del Actor | |
| Respuesta del Sistema | |
| 1- El Monitor selecciona “ <i>Crear Nota General</i> ” | <p>2- El sistema muestra una interfaz en la cual automáticamente llena los siguientes campos:</p> <ul style="list-style-type: none"> • Fecha • Nota Padre <p>Y permite al monitor llenar los siguientes:</p> <ul style="list-style-type: none"> • Tipo <ul style="list-style-type: none"> ○ De anotación ○ Consulta • Descripción • Nota Detallada • Estado de Resolución <ul style="list-style-type: none"> ○ Nuevo ○ Actualizado ○ Resuelto |

| | |
|---|---|
| | <ul style="list-style-type: none"> ○ Cerrado |
| <p>3- El Monitor escoge el tipo, pudiendo ser:</p> <ul style="list-style-type: none"> ○ De anotación ○ Consulta | |
| <p>4- El Monitor edita los campos:</p> <ul style="list-style-type: none"> • Descripción • Nota Detallada | |
| <p>5- El campo Estado de Resolución es seleccionado por el Monitor, pudiendo ser:</p> <ul style="list-style-type: none"> ○ Nuevo ○ Actualizado ○ Resuelto ○ Cerrado | |
| <p>6- El Monitor guarda los cambios</p> | <p>7- El sistema salva la nota en un registro de notas.</p> <p>Finaliza el Caso de Uso</p> |
| <p>Prototipo de Interfaz</p> | |

Añadir Nota General

Tipo Anotación Consulta

Descripción

Nota Detallada

Estado de resolución

Fecha

Nota Padre

Pos condiciones | Queda creada una Nota de General

Conclusiones

En el capítulo se expuso: el modelo de dominio, los requerimientos del sistema y la descripción de los casos de uso. Con el modelo de dominio se logró un vocabulario común y un mejor entendimiento del sistema. Los 19 requisitos funcionales que se enumeraron dieron origen a 9 casos de usos. Haciéndose una propuesta del sistema que será diseñada e implementada en posteriores capítulos.

CAPÍTULO3: Análisis y Diseño del Sistema

Introducción

En el presente capítulo tiene como objetivo la descripción de estilos arquitectónicos y patrones de diseño, evidenciado su utilización dentro del sistema, dar una muestra de los diagramas de clases (breve descripción), de interacción (Secuencia) así como Modelo de Despliegue. Haciendo uso además de patrones de diseño y de arquitectura.

1 Patrones

Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.(GARCÍA 2005)

1.1 Patrones de Diseño

Entre las características para que una solución sea considerada un patrón de diseño, es que debe haber comprobado su efectividad resolviendo situaciones similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable diferentes problemas de diseño en distintas circunstancias. Pretenden proporcionar catálogos de elementos reusables en el diseño de de sistemas de software, evitar que se repitan las soluciones y puedan utilizarse ya conocidas, para así formar un vocabulario común entre los diseñadores ,estandarizar el modo en que se realiza el diseño, facilitando el aprendizaje a las nuevas generaciones de diseñadores condensando el conocimiento ya existente.

1.1.1 DAO

Los Objetos de Acceso a Datos son un Patrón de Diseño Core J2EE y considerados una buena práctica. La ventaja de usar objetos de acceso a datos es que cualquier objeto de negocio (aquel que contiene detalles específicos de operación o aplicación) no requiere conocimiento directo del destino final de la información que manipula. Los Objetos de Acceso a Datos pueden usarse en Java para aislar a una aplicación de la tecnología de persistencia Java subyacente(API de Persistencia Java), la cual podría ser JDBC, JDO, EJB CMP(Persistencia controlada por el Contenedor), TopLink, Hibernate, iBATIS, o cualquier otra tecnología de persistencia. Usando Objetos de Acceso de Datos significa que la tecnología subyacente puede ser actualizada o cambiada sin cambiar otras partes de la aplicación.

A continuación un ejemplo de una clase de acceso a datos que usa el patrón DAO. Esta clase maneja la información de las notas siendo el puente de interacción entre la base de datos y la aplicación.

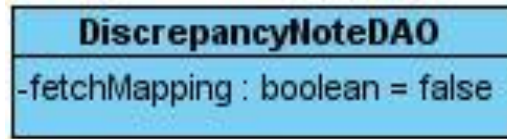


Ilustración 11 Patrón DAO

1.1.2 Iterador (Iterator)

Permite realizar recorridos sobre objetos compuestos independientemente de la implementación de estos. En la implementación se usa mucho este patrón para recorrer colecciones de datos.

Específicamente, como lo muestra la imagen, recorre colecciones de datos que contienen las tuplas que se obtienen de las consultas a la base de datos.

```
Iterator it = itemDataNotes.iterator();
while (it.hasNext()) {
    DiscrepancyNoteBean dn = (DiscrepancyNoteBean) it.next();
    if (dn.isSitio()) {
        it.remove();
    }
}
```

Ilustración 12 Patrón Iterador

1.1.3 Fachada de Sesión (Session Façade)

El Session Façade maneja los objetos de negocio y proporciona un servicio de acceso uniforme a los clientes. Se evidencia en la creación de objetos bean y que se almacena en las sesiones permitiendo usar este objeto en distintas clases del flujo.

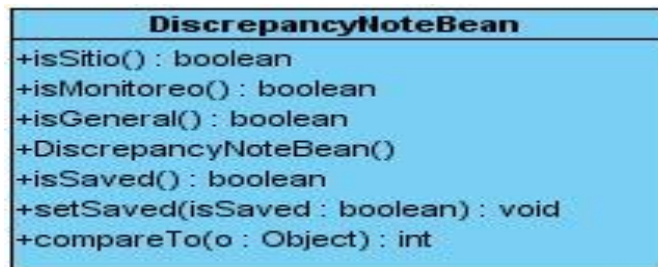


Ilustración 13 Patrón Fachada de Sesión

1.1.4 Vista compuesta (Composite View)

Un patrón Vista Compuesta es un objeto vista que está compuesto de otros objetos vista. Por ejemplo, una página JSP que incluye otras páginas JSP y HTML usando la directiva include o el action include. Este patrón se evidencia por ejemplo en la página menú.jsp la cual incluye otras páginas JSP además de código HTML para construir la página final que muestra el contenido al usuario.

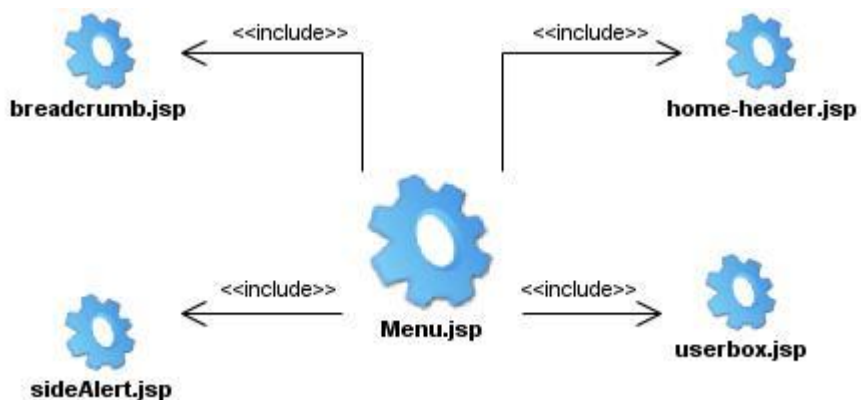


Ilustración 14 Patrón Vista Compuesta

1.1.5 Experto

Asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Es la solución a ¿cuál es el principio fundamental en virtud del cual se asignan las responsabilidades en el diseño orientado a objetos?

La clase `DiscrepancyNoteBean` es la máxima responsable de manejar la información de una nota de discrepancia; ver ilustración 13.

1.1.6 Creador:

Guía la asignación de responsabilidades relacionadas con la creación de objetos (una tarea muy común). La intención básica del patrón es encontrar un creador que necesite conectarse al objeto creado en alguna situación. Logrando así mayor reutilización.

La ilustración 12 ejemplifica el uso del patrón creador, donde se crea un objeto de tipo `DiscrepancyNoteBean` para reutilizar su información.

1.1.7 Bajo acoplamiento:

Surge para dar solución a la dependencia logrando un aumento en la reutilización. El acoplamiento mide qué tan fuerte está una clase conectada con otras. Una clase con bajo o débil acoplamiento no depende de "muchas otras" clases. Una clase con alto o fuerte acoplamiento recurre a muchas otras clases. Este tipo de clase no es conveniente, pues: cambios en las clases relacionadas ocasionan cambios en la clase local; son más difíciles de entender; son más difíciles de reutilizar. Por lo que la solución planteada por este patrón es asignar una responsabilidad para mantener bajo acoplamiento.

1.1.8 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. Una clase con baja cohesión hace un trabajo excesivo. Las clases con baja cohesión a menudo representan un alto grado de abstracción o han asumido responsabilidades que deberían haber delegado a otros objetos. La solución es asignar una responsabilidad de modo que la cohesión siga siendo alta.

2 Patrones arquitectura

Son los que definen la estructura de un sistema, los cuales a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de tal sistema..

2.1 El patrón de arquitectura de software Modelo – Vista – Controlador (MVC).

La arquitectura del software alude a la estructura global del software y a las formas en que la estructura proporciona la integridad conceptual de un sistema. En su forma más simple, la arquitectura es la estructura jerárquica de los componentes del programa (módulos), la manera en que los componentes interactúan y la estructura de datos que van a utilizar los componentes. Sin embargo, en un sentido más amplio, los "componentes" se pueden generalizar para presentar los elementos principales del sistema y sus interacciones. El diseño arquitectónico define la relación entre los elementos estructurales principales del software, los patrones de diseño que se pueden utilizar para lograr los requisitos que se han definido para el sistema, y las restricciones que afectan a la manera en que se pueden aplicar los patrones de diseño arquitectónicos.

El Modelo Vista Controlador divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

Modelo (Model): Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.

Vista (View): Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.

Controlador (Controller): Reciben las entradas, usualmente 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.

Se definen las clases dominio (Modelo) para que no tengan acoplamiento ni visibilidad directa respecto a las clases ventana (Vista) y para que los datos de la aplicación y de la funcionalidad se conserven en las clases de dominio, no en las de ventana. Definir las clases manejadores (Controlador) para que

procesen los eventos (peticiones) al sistema y redireccionen a las clases dominio y ventana tanto el procesamiento como la visualización de resultados respectivamente.

Algunos de sus principales beneficios son:

- Menor acoplamiento.
 - Desacopla las vistas de los modelos.
 - Desacopla los modelos de la forma en que se muestran e ingresan los datos.

- Mayor cohesión.
 - Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo de negocio).

- Las vistas proveen mayor flexibilidad y agilidad.
 - Se puede crear múltiples vistas de un modelo.
 - Las vistas pueden anidarse.
 - Se puede cambiar el modo en que una vista responde al usuario sin cambiar su representación visual.
 - Se puede sincronizar las vistas.
 - Las vistas pueden concentrarse en diferentes aspectos del modelo.

- Mayor facilidad para el desarrollo de clientes ricos en múltiples dispositivos y canales
 - Una vista para cada dispositivo que puede variar según sus capacidades.
 - Una vista para la Web y otra para aplicaciones de escritorio.

- Más claridad de diseño.

- Facilita el mantenimiento.

- Mayor escalabilidad.

3 Uso del Digester

Es una librería de clases que se utiliza para procesar archivos XML a través de la búsqueda de reglas. Nos oculta el parseo y nos permite pasar directamente de XML a colecciones de objetos. Nuestro trabajo es simplemente es definir simplemente la correspondencia entre las entidades XML y objetos Java. Su propósito inicial fue procesar el fichero de configuración de una aplicación web basada en Struts (struts-config.xml). Posteriormente, dada su evidente utilidad para una gran heterogeneidad de aplicaciones, fue incluido como un framework independiente.

4 Diagramas de clases del diseño

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro. A continuación se muestran algunos diagramas realizados agrupándose en clase de acuerdo con sus funcionalidades.

Paquetes:

Vista:



"X".Jsp Se encarga de generar dinámicamente las paginas clientes.

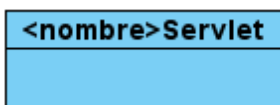


"X".html creada por la jsp interactúa con el cliente.



form A través de él se envían los datos que entra el cliente al servidor.

Control:

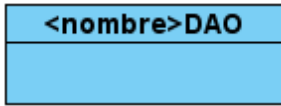


Estas clases se encargan del manejo de los datos del modelo estructurarlos y mandarlos a la vista o los envía al modelo para ser guardados.

Modelo:



Contienen los datos del sistema cada objeto de ella constituye una tupla en las tablas de la base de datos.



Ocultan la forma en que el sistema salva o accede a los datos, facilitando la abstracción de este con la base de datos.

Inclusiones:



Utilizado para validar la información y construir la vista a mostrar en las páginas *.html mostradas a los usuarios.



Grupo de *.jsp de las que dependen algunas *.jsp desarrolladas para crear la vista que se mostrará en las páginas *.html.

4.1 Diagrama de Clases de Diseño del CU Crear Nota General

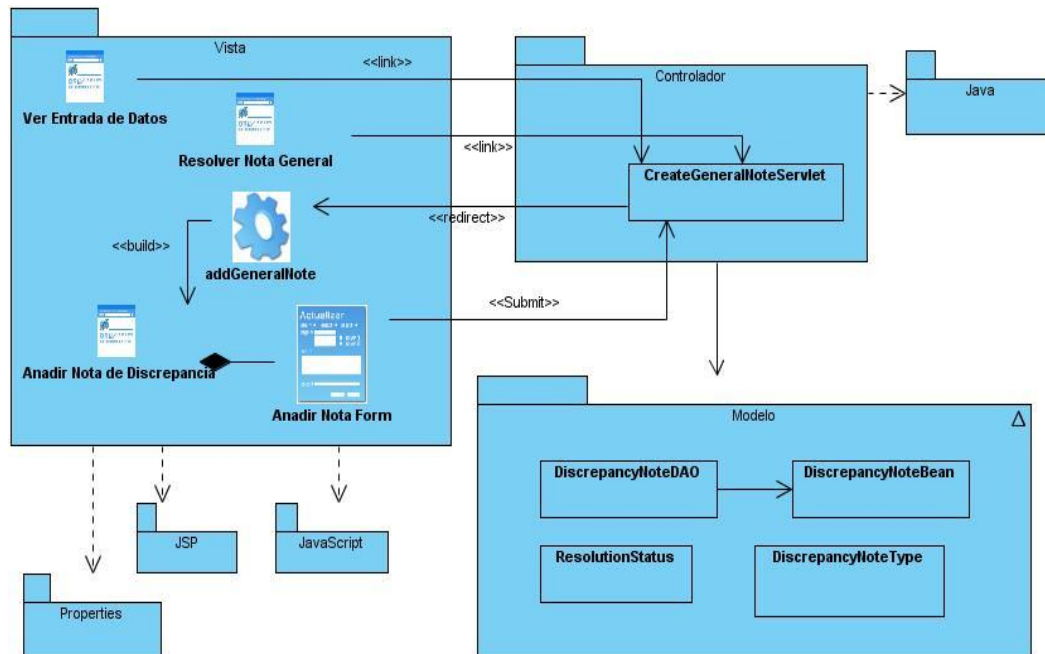


Ilustración 15 Crear Nota General.

| | | |
|---|-------------|--|
| Caso de Uso: Crear Nota General | | |
| Nombre: CreateGeneralNoteServlet | | |
| Tipo: Es la clase encargada del control al crear una Nota General. | | |
| Atributo | Tipo | |
| locale | Locale | |
| DIS_NOTE | String | |
| Dis_Types | String | |
| Res_statuses | String | |
| NEW_Notes | String | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| myProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| processRequest() | void | Método que se encarga de gestionar los datos a la nota y guardarla a la base de datos. |

4.2 Diagrama de Clases del Diseño del CU Responder Nota General

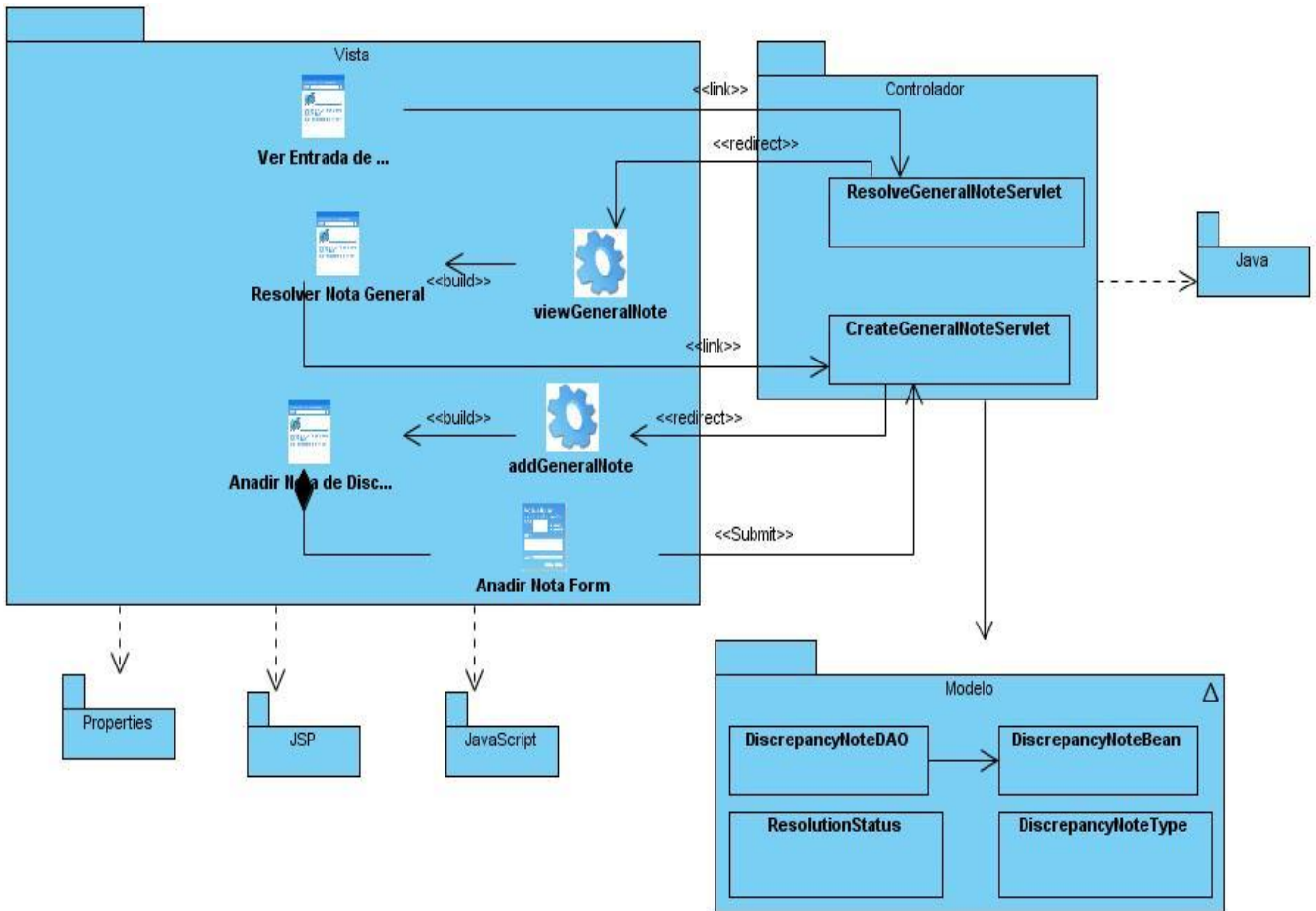


Ilustración 16 Responder Nota General.

| | | |
|--|-------------|---|
| Caso de Uso: Responder Nota General | | |
| Nombre: ResolveGeneralNoteServlet | | |
| Tipo: Se encarga del manejo de los datos para posteriormente mostrar los detalles de la nota general en cuestión, pueda ser resuelta. | | |
| Atributo | Tipo | |
| locale | Locale | |
| Note_id | String | |
| Dis_Note | String | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| myProceed() | void | Método que se encarga de verificar los roles para |

| | | |
|--|-------------|--|
| | | permitir o no el envío de datos. |
| processRequest() | void | Método que se encarga de gestionar los datos a mostrar en los detalles de la nota, la cantidad de notas hijas sus descripciones. |
| Nombre: CreateGeneralNoteServlet | | |
| Es la clase encargada del control al crear una Nota General. | | |
| Atributo | Tipo | |
| locale | Locale | |
| DIS_NOTE | String | |
| Dis_Types | String | |
| Res_statuses | String | |
| NEW_Notes | String | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| mayProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| processRequest() | void | Método que se encarga de gestionar los datos a la nota y guardarla a la base de datos. |

4.3 Diagrama de Clases del Diseño del CU Monitorear Hoja de CRD

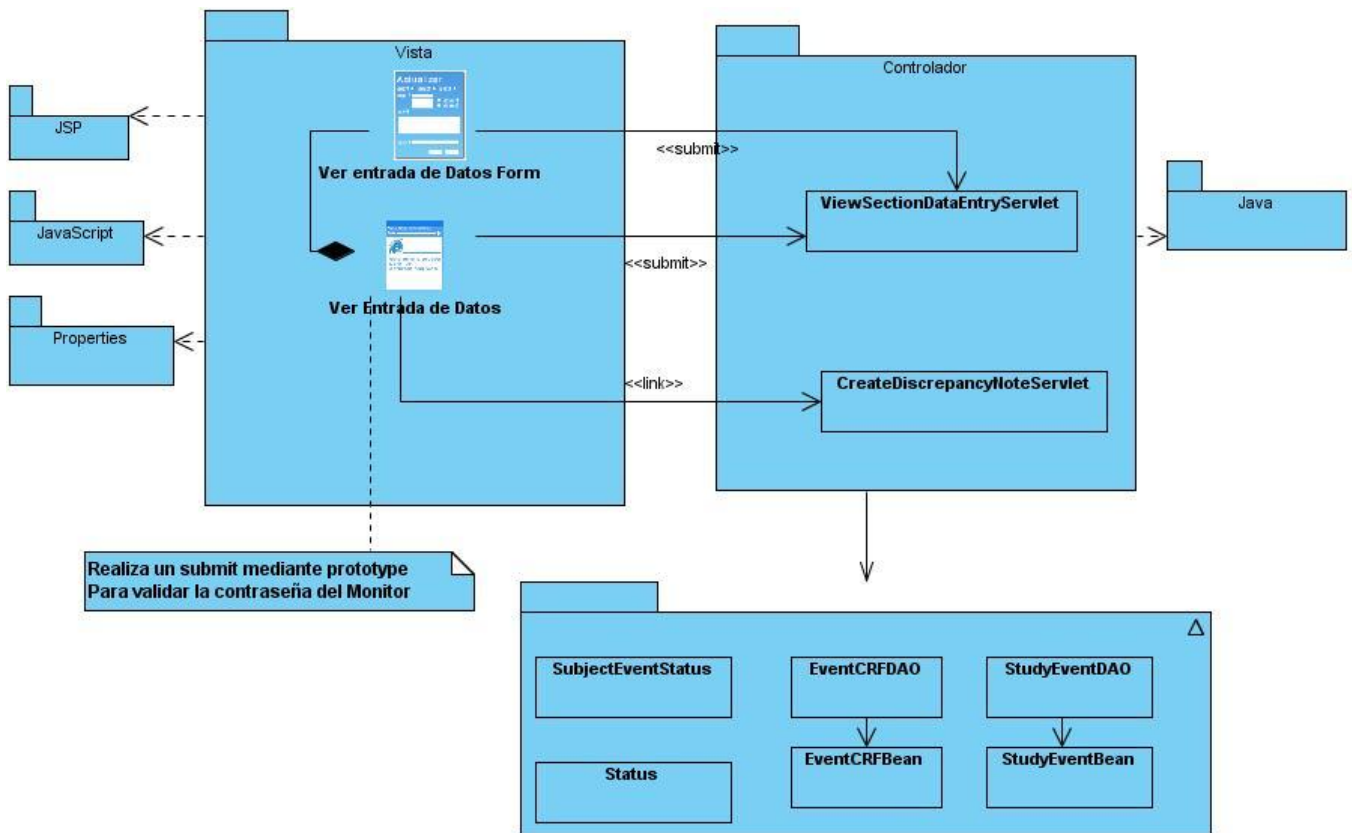


Ilustración 17 Responder Nota General.

| | | |
|--|-------------|--|
| Caso de Uso: Monitorear Hoja de CRD | | |
| Nombre: ViewSectionDataEntryServlet | | |
| Tipo: Controladora de datos. | | |
| Atributo | Tipo | |
| locale | Locale | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| myProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| processRequest() | void | Método que se encarga de gestionar los datos para la conformación de una Hoja de CDR |
| Nombre: CreateDiscrepancyNoteServlet | | |
| Es la clase encargada del control al crear una Nota General. | | |
| Atributo | Tipo | |

| | | |
|---------------------------|-------------|--|
| locale | Locale | |
| DIS_NOTE | String | |
| Dis_Types | String | |
| Res_statuses | String | |
| NEW_Notes | String | |
| Parent_id | String | |
| Entity_type | String | |
| FromDiscrepancy_Note_Name | String | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| mayProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| processRequest() | void | Método que se encarga de gestionar los datos a la nota y guardarla a la base de datos. |

4.4 Diagrama de Clases del Diseño del CU Visualizar Notas

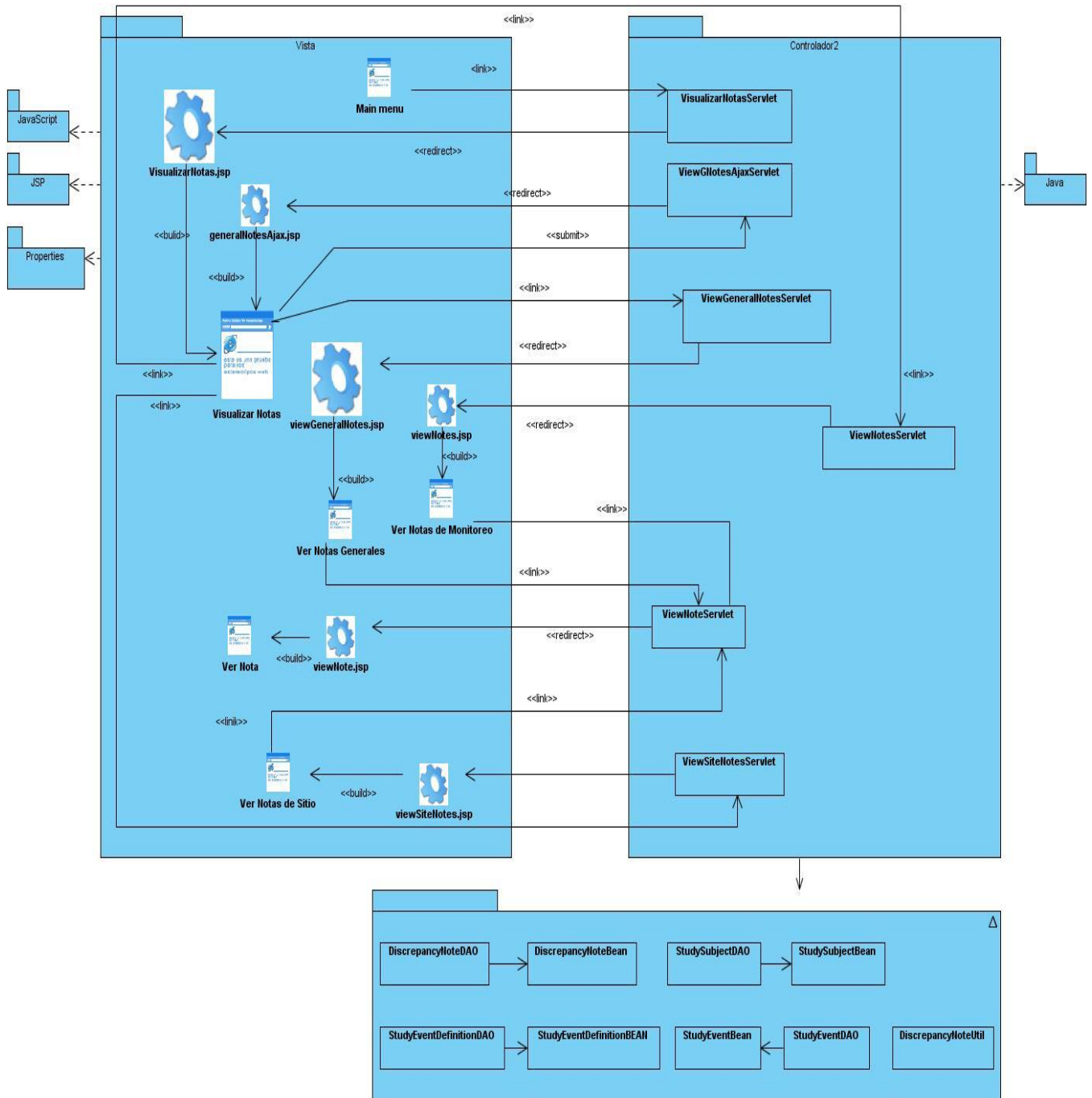


Ilustración 18 Visualizar Notas.

| | | |
|---|-------------|---|
| Caso de Uso: Visualizar Notas | | |
| Nombre: VizualizarNotasServlet | | |
| Tipo: Controladora de datos. Listado por bloques de notas. | | |
| Atributo | Tipo | |
| locale | Locale | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| myProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| processRequest() | void | Método que se encarga de gestionar los datos para la conformación del listado de las notas. |
| Nombre: ViewGNotesAjaxServlet | | |
| Tipo: Controladora de Datos, que muestra el listado de notas generales en la interfaz visualizar notas. | | |
| Atributo | Tipo | |
| locale | Locale | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| mayProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| processRequest() | void | Este método confecciona el listado de las notas generales. |
| Nombre: ViewGeneralNotesServlet | | |
| Tipo: Controladora de Datos, que muestra el listado de notas generales en la interfaz ver notas generales. | | |
| Atributo | Tipo | |
| locale | Locale | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| mayProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| processRequest() | void | Este método confecciona el listado de las notas generales. |
| Nombre: ViewNotesServlet | | |

| | | |
|---|-------------|--|
| Tipo: Controladora de Datos, que muestra el listado de las notas de Monitoreo en la interfaz ver notas de monitoreo. | | |
| Atributo | Tipo | |
| locale | Locale | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| processRequest() | void | Este método confecciona el listado de las notas generales. |
| mayProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| Nombre: ViewNoteServlet | | |
| Tipo: Controladora de Datos, que muestra una nota en dependencia de su tipo. | | |
| Atributo | Tipo | |
| locale | Locale | |
| Dis_Note | String | |
| Note_id | String | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| processRequest() | void | Muestra los detalles de una nota. |
| mayProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| Nombre: ViewSiteNotesServlet | | |
| Tipo: Controladora de Datos, que muestra una nota en dependencia de su tipo. | | |
| Atributo | Tipo | |
| locale | Locale | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| mayProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| processRequest() | void | Este método confecciona el listado de las notas de sitio. |

4.5 Diagrama de Clases del Diseño del CU Crear Nota de Monitoreo

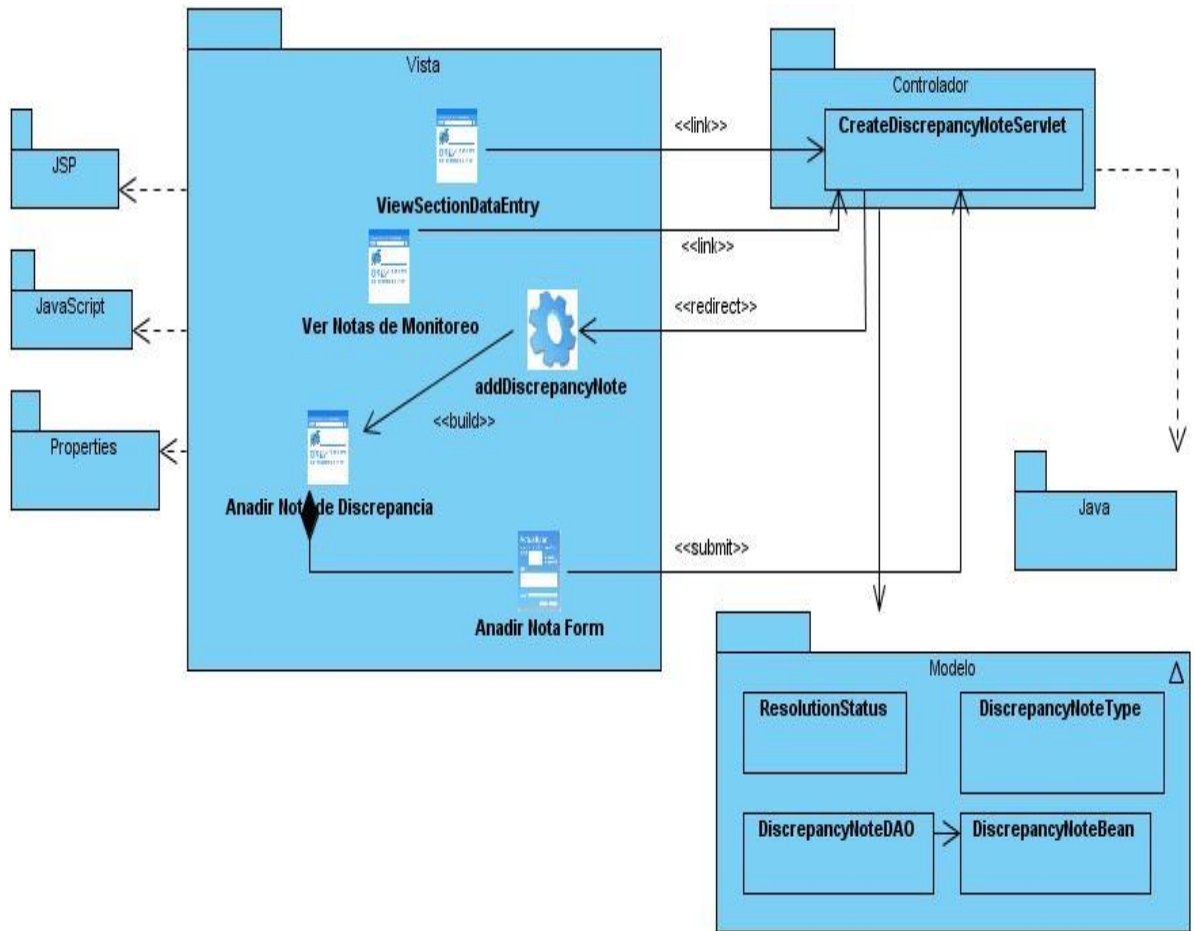


Ilustración 19 Crear Nota de Monitoreo

| Caso de Uso: Crear Nota de Monitoreo | |
|--|--------|
| Nombre: CreateDiscrepancyNoteServlet | |
| Tipo: Es la clase encargada del control al crear una Nota de Monitoreo. | |
| Atributo | Tipo |
| locale | Locale |
| DIS_NOTE | String |
| Dis_Types | String |
| Res_statuses | String |
| NEW_Notes | String |
| Parent_id | String |
| FromDiscrepancy_Note_Name | String |

| Entity_type | String | |
|-------------------|--------|--|
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| myProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| processRequest() | void | Método que se encarga de gestionar los datos a la nota y guardarla a la base de datos. |

4.6 Diagrama de Clases del Diseño del CU Responder Nota de Monitoreo.

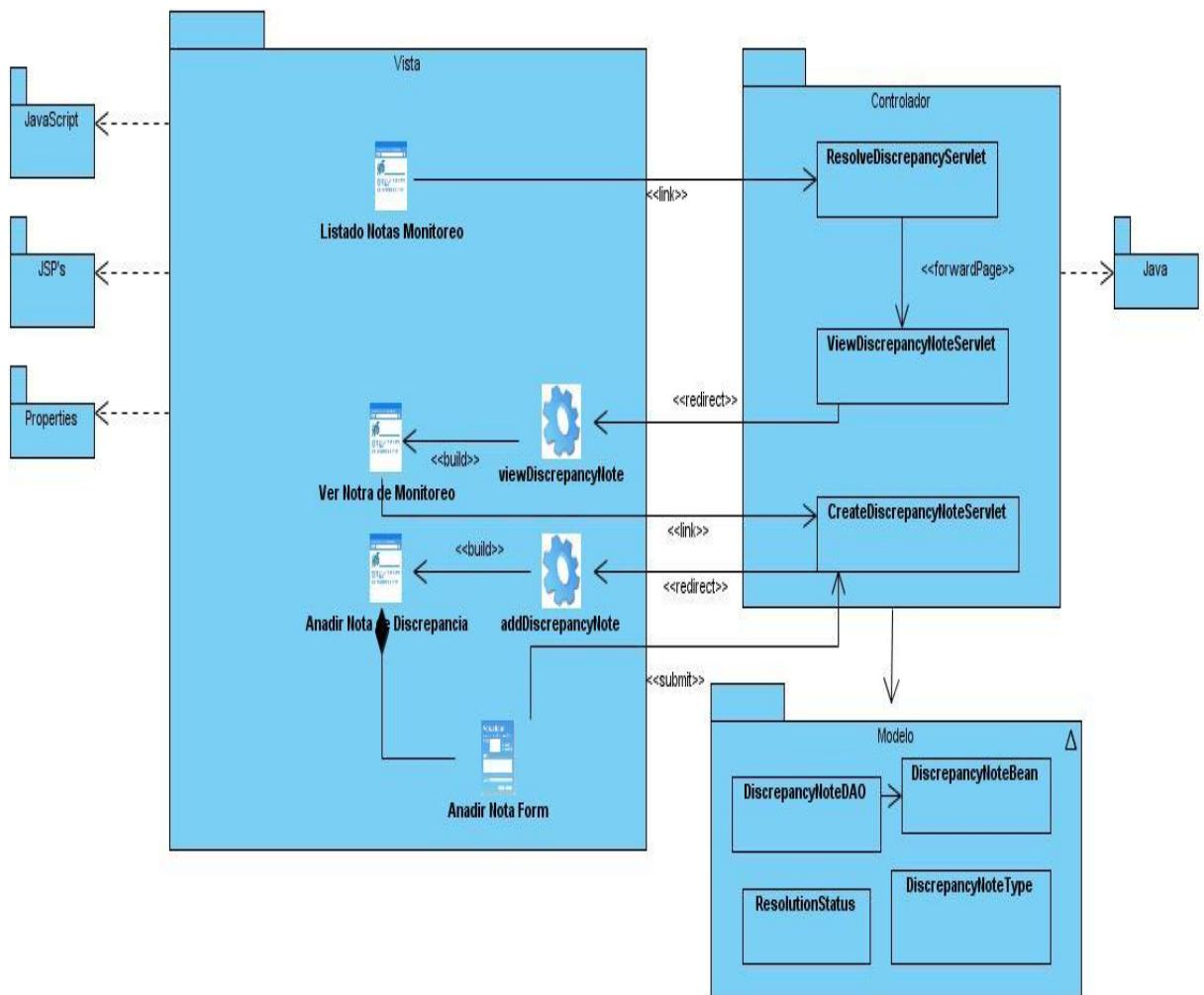


Ilustración 20 Responder Nota de Monitoreo

| | | |
|--|-------------|--|
| Caso de Uso: Responder Nota de Monitoreo | | |
| Nombre: ResolveDiscrepancyNoteServlet | | |
| Tipo: Maneja el link re direccionando hacia otro servlet. | | |
| Atributo | Tipo | |
| locale | Locale | |
| Resolving_Note | String | |
| Nombre | Tipo | Descripción |
| myProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| processRequest() | void | Redirecciona para la servlet. |
| Nombre: ViewDiscrepancyNoteServlet | | |
| Tipo: Se encarga del manejo de los datos para posteriormente mostrar los detalles de la nota de monitoreo en cuestión, para que pueda ser resuelta. | | |
| Atributo | Tipo | |
| locale | Locale | |
| Entity_types | String | |
| Parent_id | String | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| mayProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| processRequest() | void | Método que se encarga de gestionar los datos a mostrar en los detalles de la nota, la cantidad de notas hijas sus descripciones. |
| Nombre: CreateDiscrepancyNoteServlet | | |
| Tipo: Es la clase encargada del control al crear una Nota de Monitoreo. | | |
| Atributo | Tipo | |
| locale | Locale | |
| DIS_NOTE | String | |
| Dis_Types | String | |
| Res_statuses | String | |
| NEW_Notes | String | |

| | | |
|---------------------------|-------------|--|
| Parent_id | String | |
| FromDiscrepancy_Note_Name | String | |
| Entity_type | String | |
| Responsabilidades | | |
| Nombre | Tipo | Descripción |
| myProceed() | void | Método que se encarga de verificar los roles para permitir o no el envío de datos. |
| processRequest() | void | Método que se encarga de gestionar los datos a la nota y guardarla a la base de datos. |

5 Diagramas de interacción

Estos diagramas describen las interacciones entre los objetos del diseño para los casos de uso en el mismo. Esto se hace mediante diagramas de secuencia que contienen las instancias de los actores, los objetos del diseño y las transmisiones de mensajes entre éstos, que participan en el caso de uso. Si los casos de uso tienen varios lujos o subflujos distintos, suele ser útil crear un diagrama de secuencia para cada uno de ellos. Esto puede hacer más clara la realización del caso de uso, y también permite extraer diagramas de secuencia que representan iteraciones generales reutilizables.

5.1 Diagrama de Secuencia del Caso de Uso Crear Nota de General.

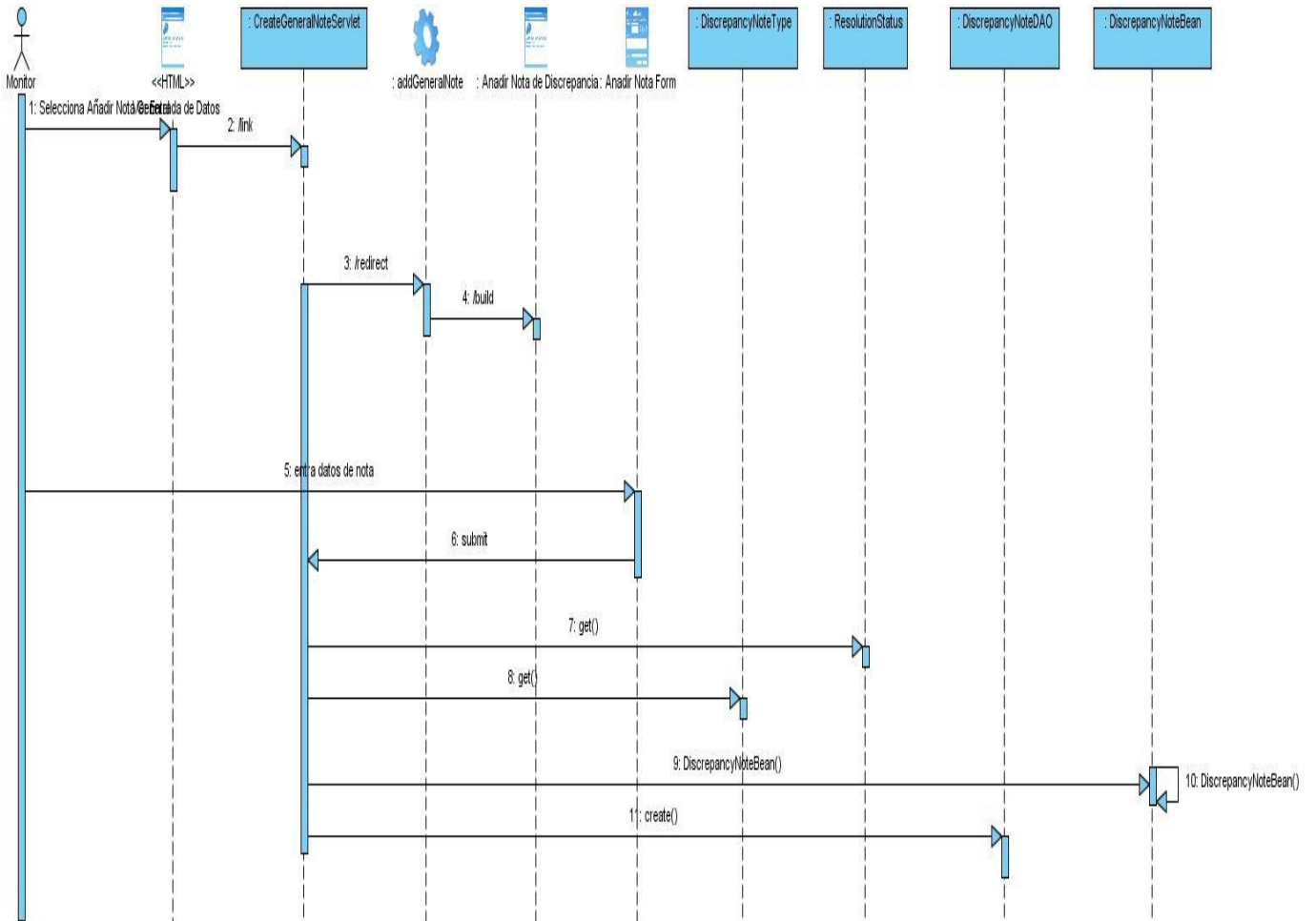


Ilustración 21 Crear Nota General # 1.

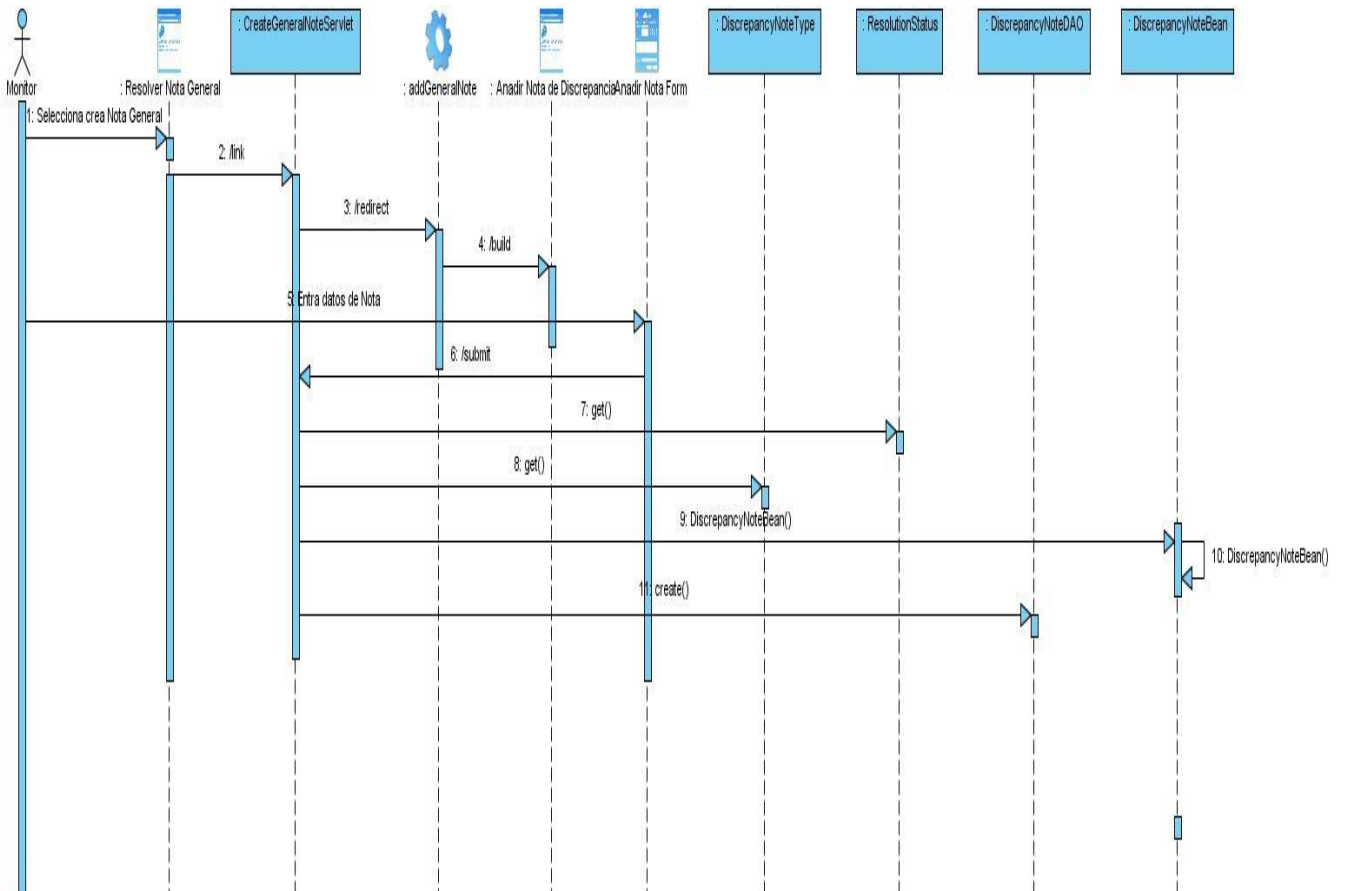


Ilustración 22 Crear Nota General #2.

5.2 Diagrama de Secuencia del Caso de Uso Responder Nota de General.

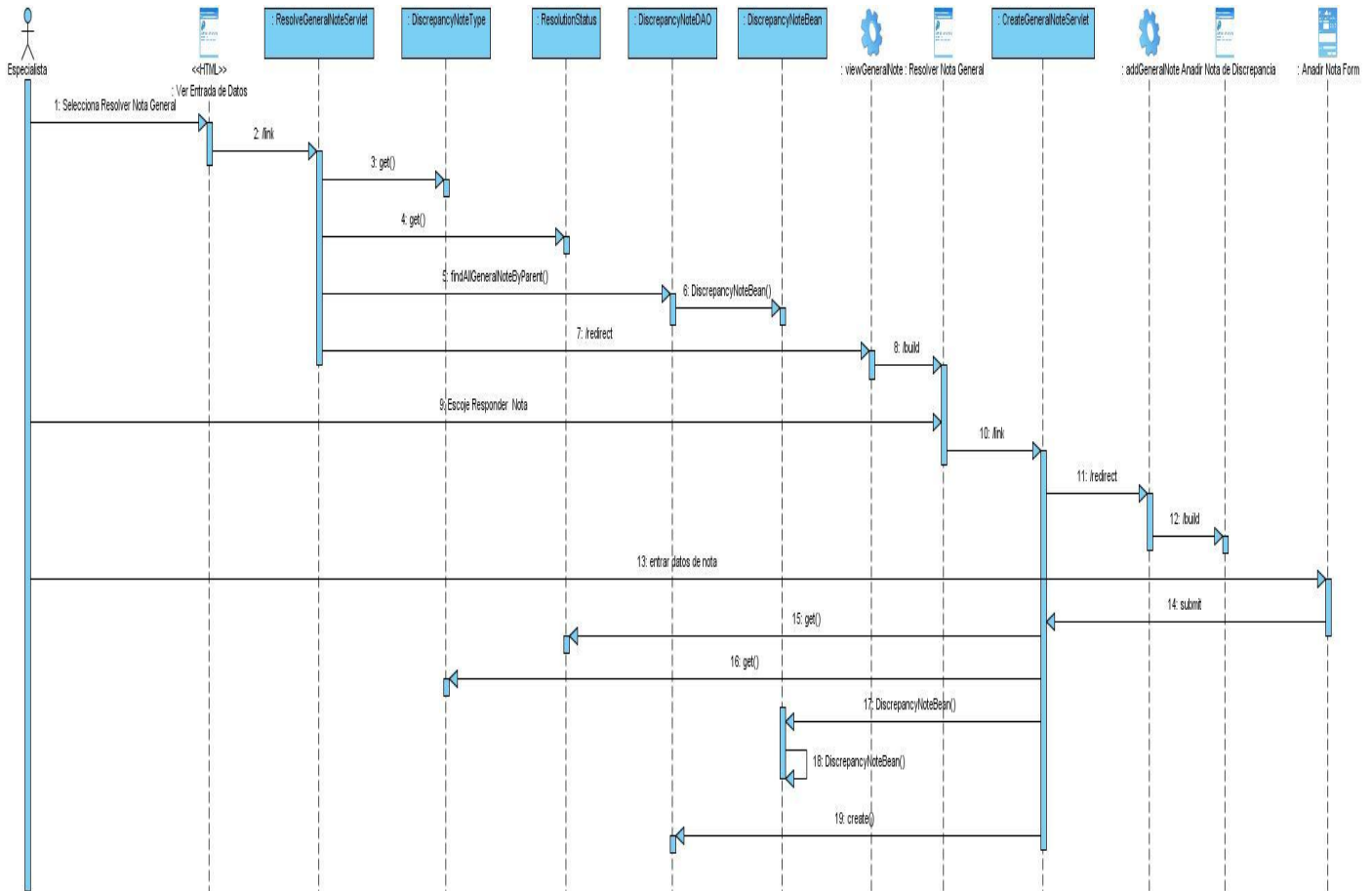


Ilustración 23 Responder Nota General.

5.3 Diagrama de Secuencia del Caso de Uso Crear Nota Monitoreo.

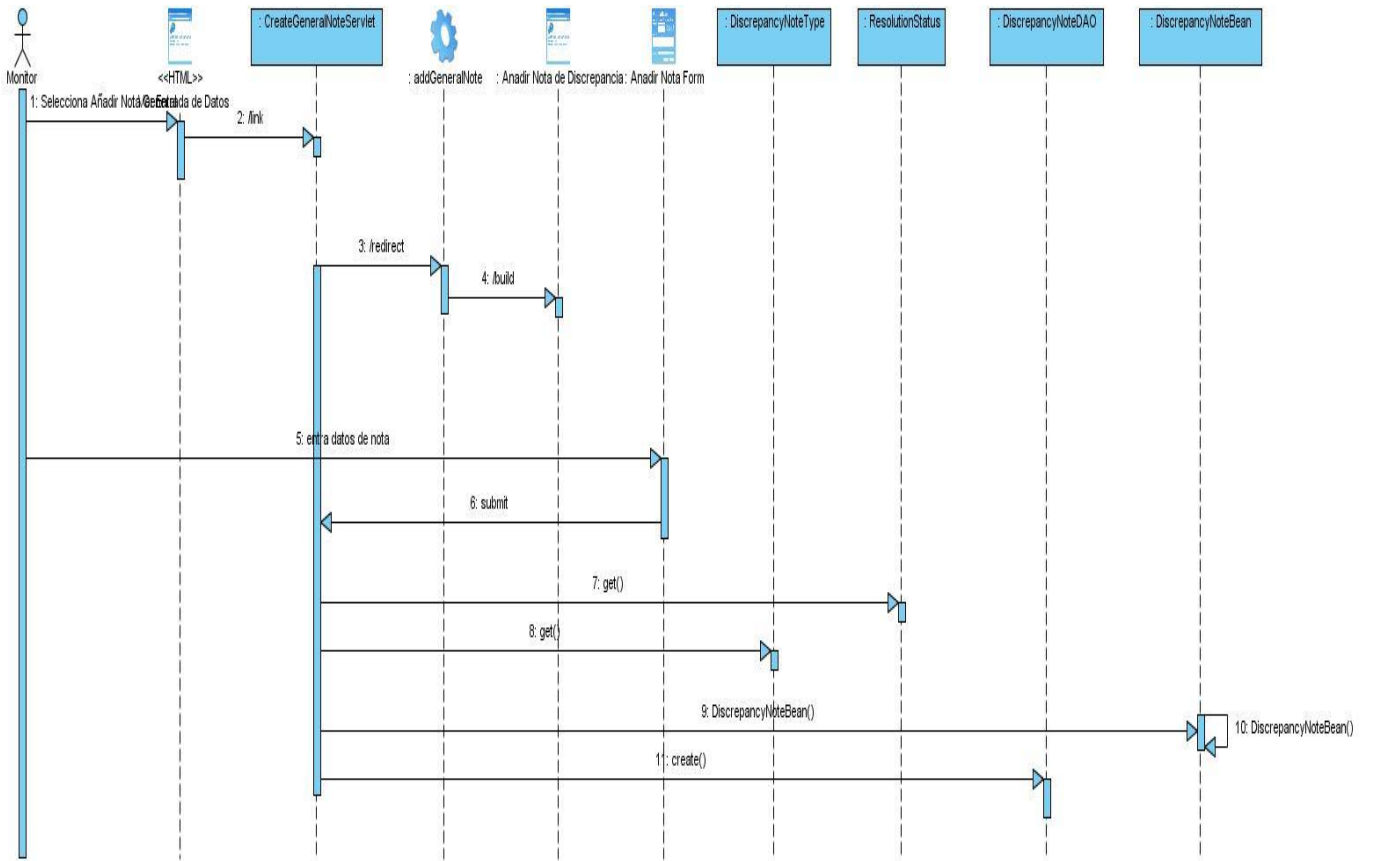


Ilustración 24 Crear Nota Monitoreo # 1.

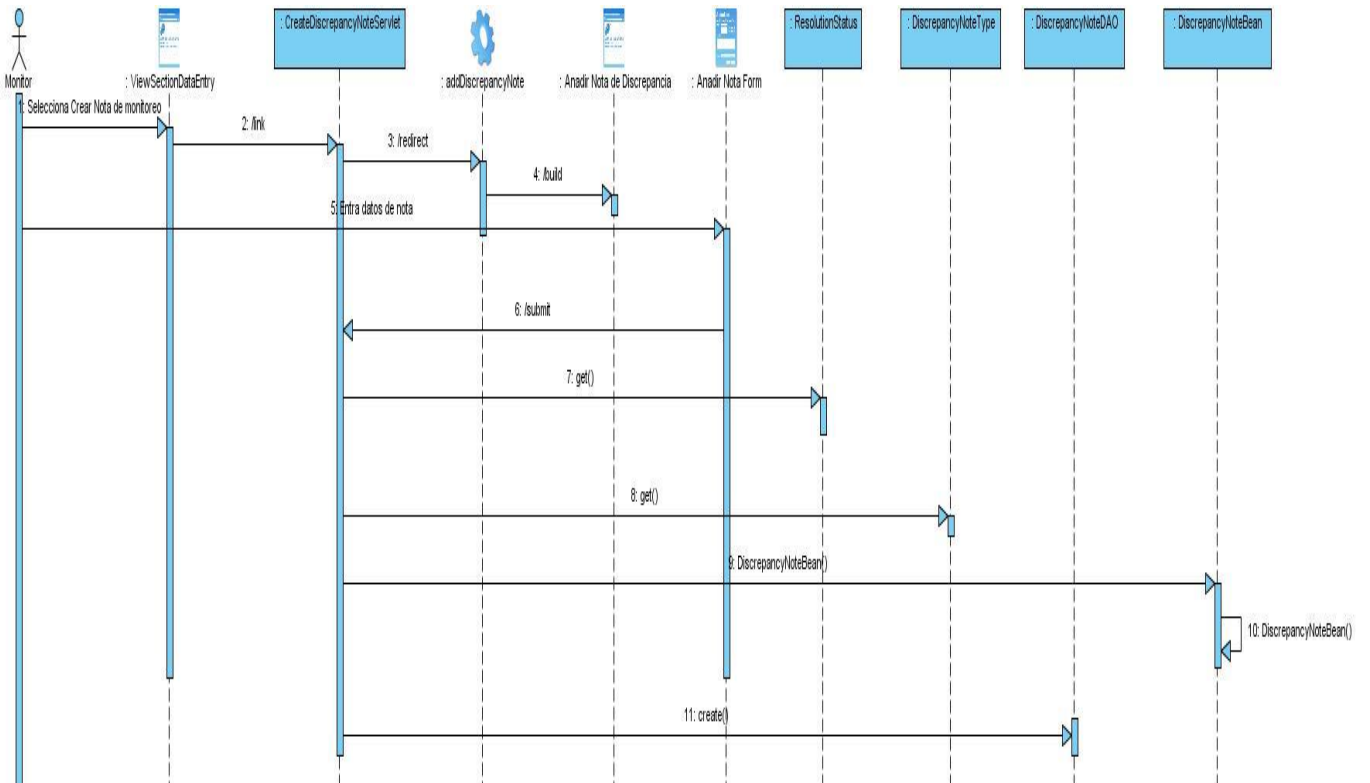


Ilustración 25 Crear Nota Monitoreo # 2.

5.4 Diagrama de Secuencia del Caso de Uso Responder Nota Monitoreo.

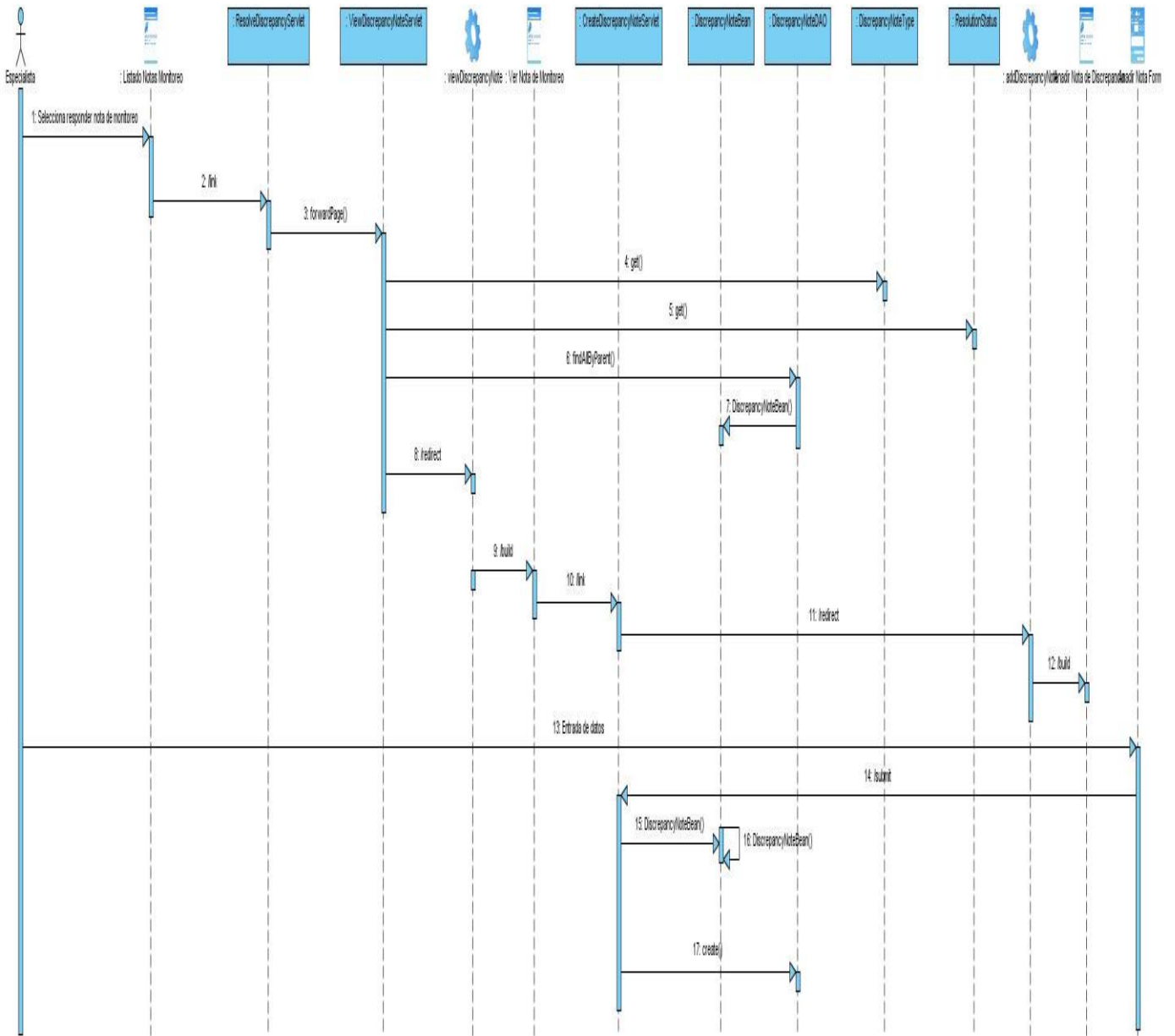


Ilustración 26 Responder Nota Monitoreo.

5.5 Diagrama de Secuencia del Caso de Uso Monitorear Hoja de CRD.

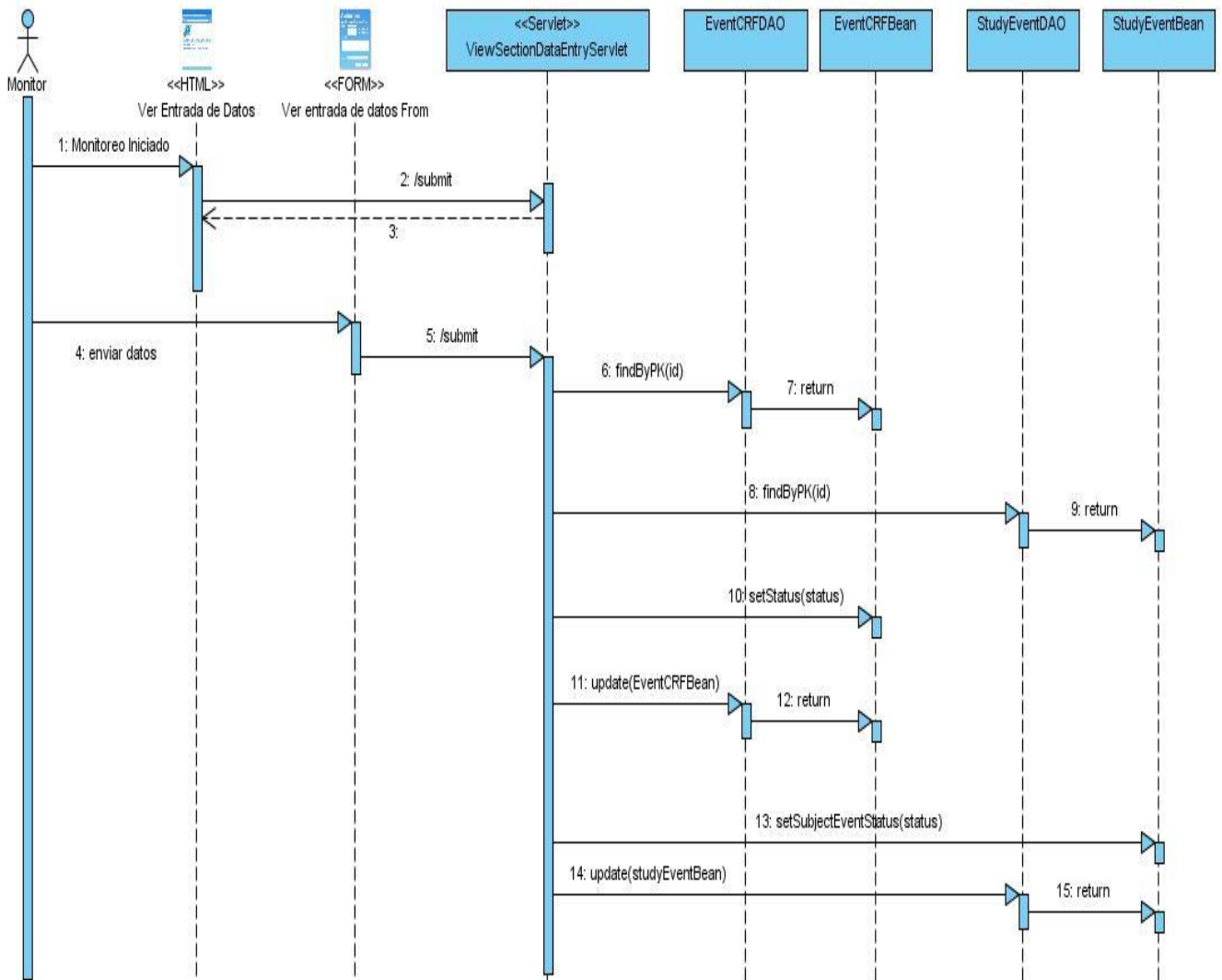


Ilustración 27 Monitorear Hoja de CRD

5.6 Diagrama de Secuencia del Caso de Uso Visualizar Notas.

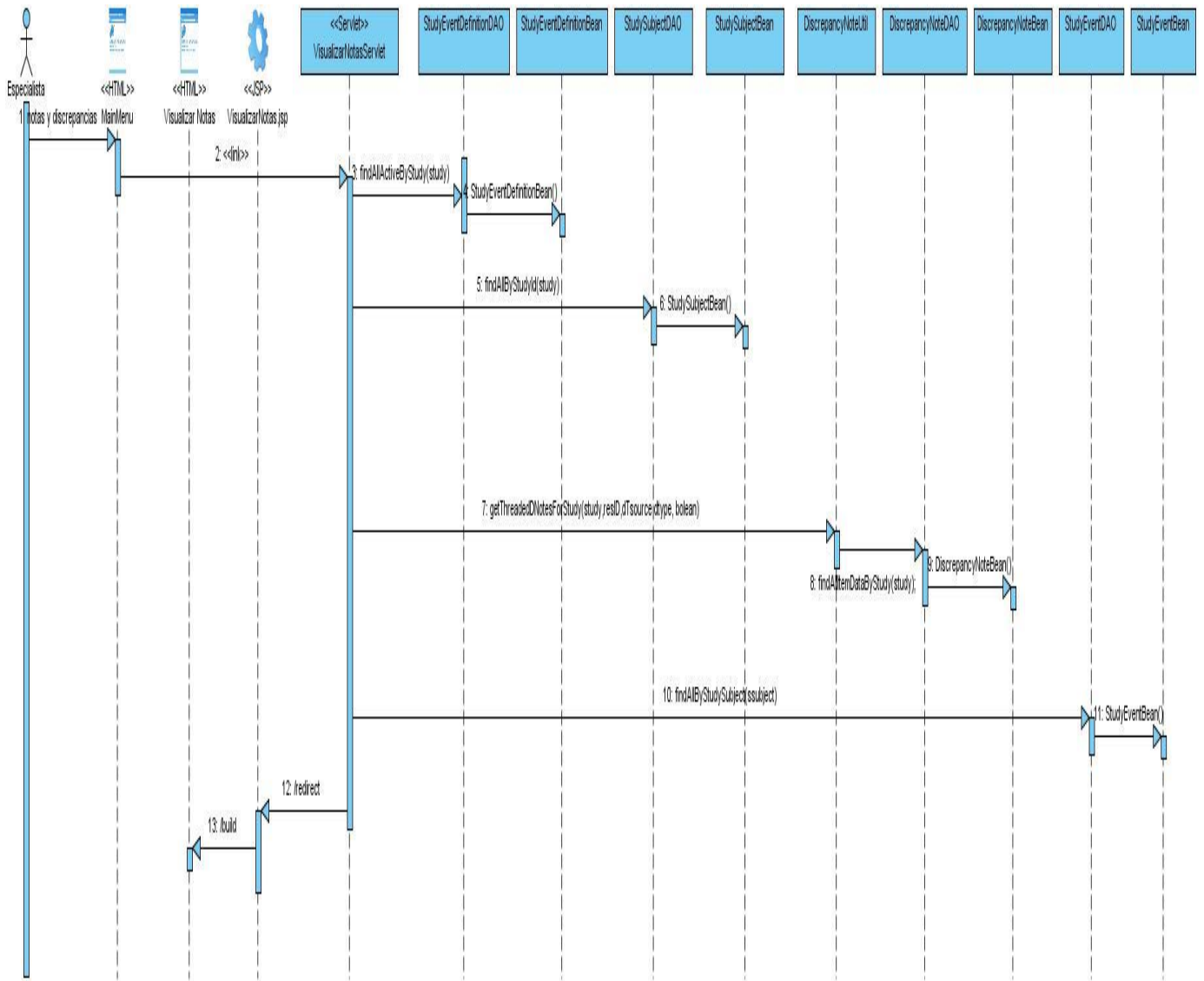


Ilustración 28 Visualizar Notas Sección#1.

6 Modelo de Datos.

Un modelo de datos es un conjunto de conceptos que describe la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. El modelo de datos aporta la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información.

Más adelante se muestra el modelos de datos para adaptar funcionalidades al monitoreo de los datos. Contando con 18 entidades.

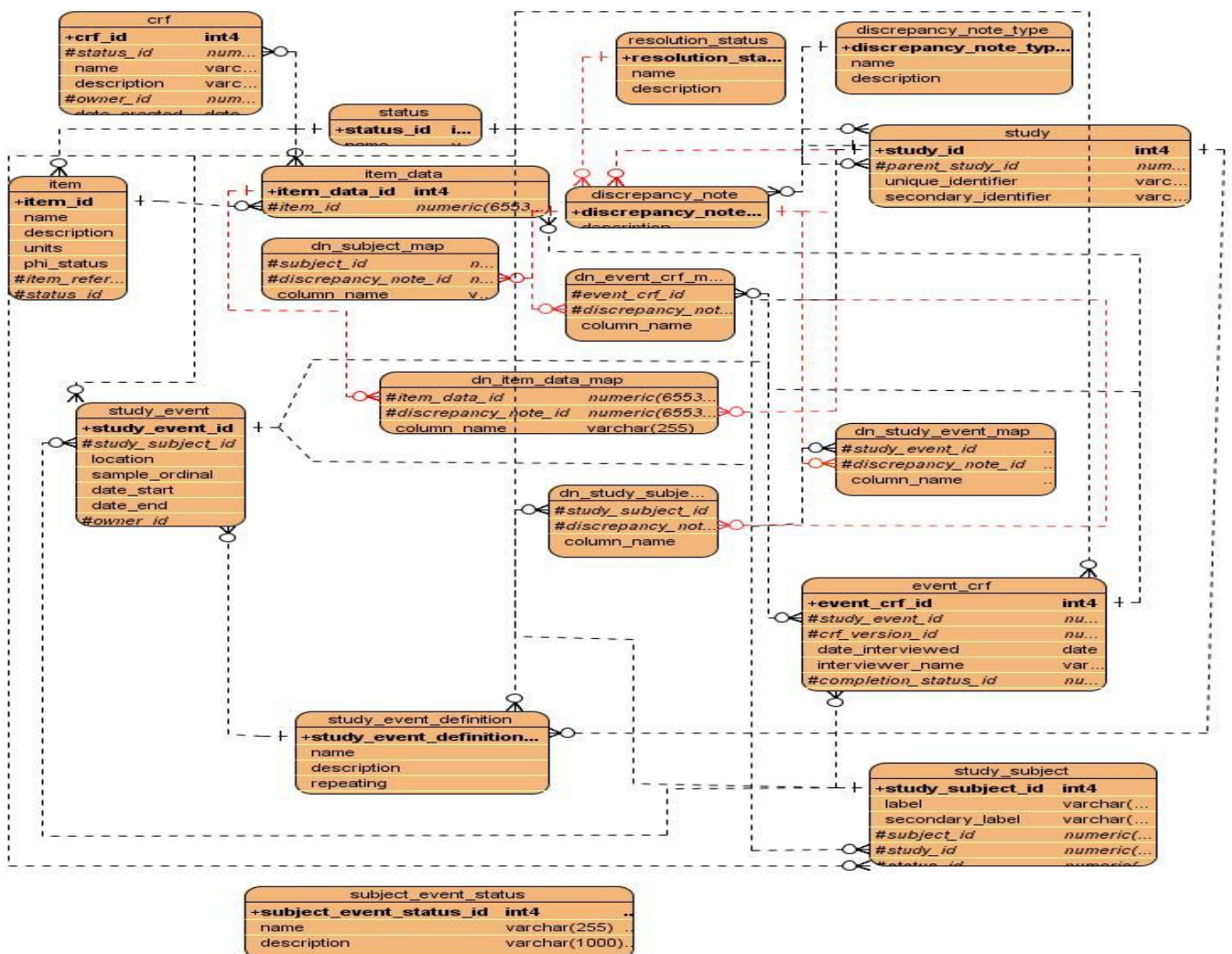


Ilustración 29 Modelo de Datos.

| No. | Entidad | PK | Tipo de dato | Número de campos | Descripción de Entidad |
|-----|-------------------|----------------------|--------------|------------------|--|
| 1 | subject | subject_id | integer | 12 | Entidad que define todos los sujetos asociados al estudio, dígase “padre de sujeto”, “madre de sujeto” y “sujeto” en sí mismo. |
| 2 | study_subject | study_subject_id | integer | 17 | Entidad que contiene todos los “sujetos” que se deciden estudiar en un Ensayo y pasan a ser “pacientes”. |
| 3 | resolution_status | resolution_status_id | integer | 3 | Entidad contiene los estados de resolución de las Notas. |
| 4 | dn_subect_map | | - | 3 | Contiene el id de la nota asociada a un paciente. |
| 5 | dn_event_crf | | - | | Contiene id de la nota asociada al momento de seguimiento “X”. |
| 6 | status | status_id | integer | 3 | Entidad que contiene los tipos de “estados” definidos para “pacientes”, “momentos de seguimientos” y “hojas de crd”. |
| 7 | study_event | study_event_id | integer | 16 | Entidad que contiene los “momentos de seguimiento” existentes para un estudio. |

| | | | | | |
|----|------------------------|---------------------------|---------|----|---|
| 8 | event_crf | event_crf_id | integer | 20 | Entidad que contiene las "hojas de crd" asociadas a un "momento de seguimiento". |
| 9 | crf | crf_id | integer | 9 | Entidad que contiene las "hojas de crd" existentes. |
| 10 | dn_item_data_map | | | 3 | Contiene el id de la nota asociada a la variable. |
| 11 | study_event_definition | study_event_definition_id | integer | 14 | Entidad que contiene las definiciones de "momentos de seguimientos" que aún no han sido asociados a "pacientes" y "hojas de crd" y pertenecen al "cronograma general" pero no al "cronograma específico". |
| 12 | dn_study_event_map | | | 3 | Contiene el id de la nota asociada al estudio y momento de seguimiento. |
| 13 | item | item_id | integer | 13 | Entidad que contiene las "variables" existentes en una "hoja de crd". |
| 14 | item_data | item_data_id | integer | 10 | Entidad que contiene los valores asociados a las "variables" de una "hoja de crd". |
| 15 | discrepancy_note | discrepancy_note_id | integer | 10 | Entidad que contiene las "notas de monitoreo" existentes. |
| 16 | discrepancy_note_type | discrepancy_note_type | integer | 3 | |

| | | | | | |
|----|----------------------|-------------------------|---------|---|---|
| | | ype_id | | | |
| 17 | dn_study_subect | | | 3 | Contiene el id de la nota asociada al estudio y al paciente. |
| 18 | subject_event_status | subject_event_status_id | integer | 3 | Contiene los estados de los pacientes y momentos de seguimientos. |

7 Diagrama de Despliegue.

La aplicación Web, resultado de la presente investigación estará distribuida de la siguiente forma:

En el Centro de Inmunología Molecular se encontrará la aplicación servidora, a la que estarán conectadas todas las PCs clientes del centro, y también se conectarán a esa aplicación otras PCs de otras partes del país con la debida autorización del centro y configurado en el servidor Firewall + Proxy. Existirán impresoras para la impresión de los modelos y otros datos de importancia, esto fue un requisito del cliente, pero en caso de faltar este dispositivo no afectará en nada el funcionamiento de la aplicación. Habrá un servidor de bases de datos, que será la base de datos primario, donde se encontrarán todos los datos del sistema, y se realizarán copias de resguardo de la información en otro servidor.

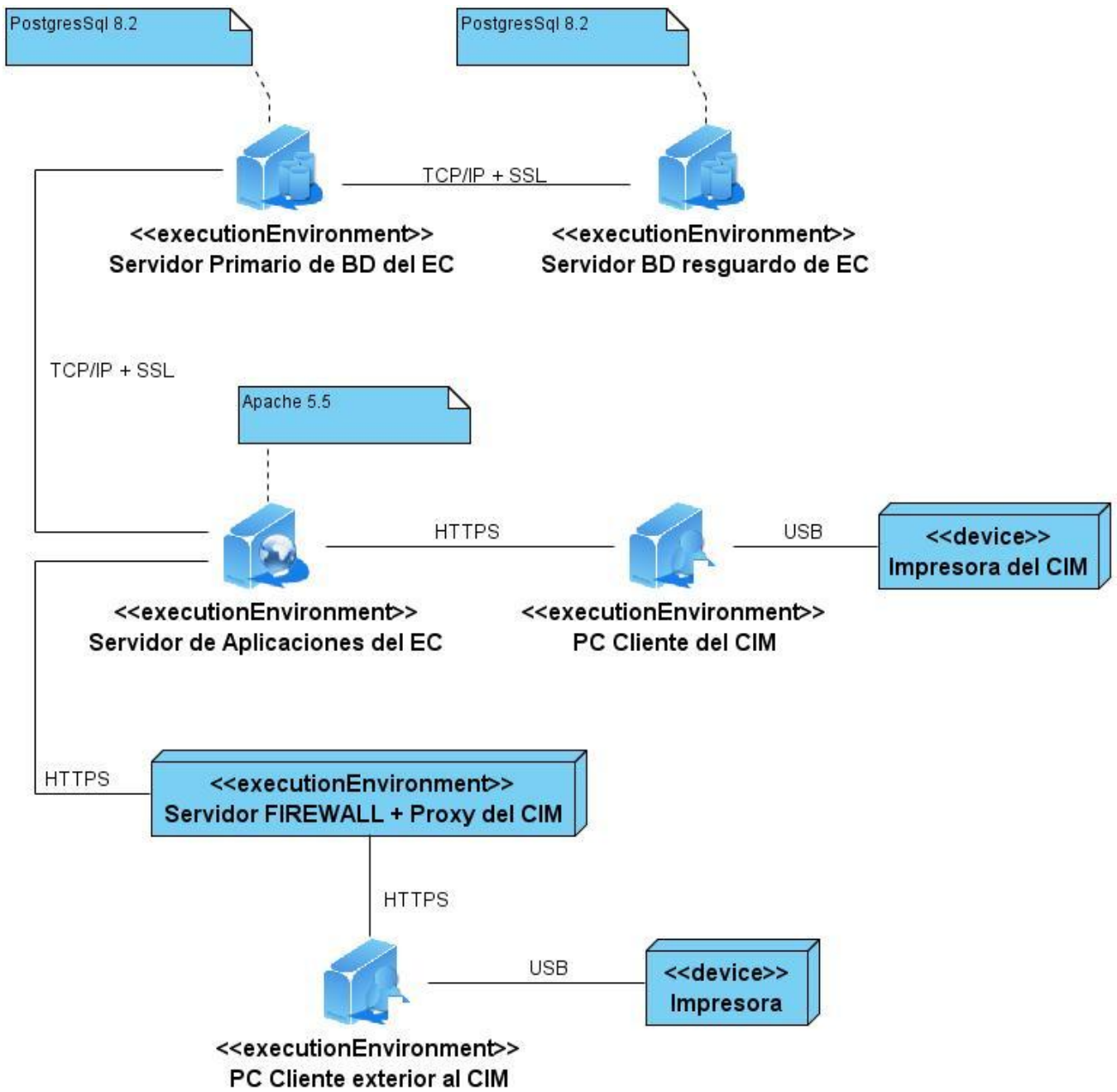


Ilustración 30 Diagrama de Despliegue.

Conclusiones

Como resultado del capítulo se han obtenido las clases del diseño, organizadas por capas debido a la utilización de patrón Modelo Vista Controlador así como los demás patrones que utiliza internamente el sistema cumpliendo con la arquitectura planteada por el proyecto. Se realizan además descripciones de los diferentes diagramas de diseño e interacción. Además se muestra la estructura del sistema desde su modelo de datos hasta su despliegue.

CAPÍTULO 4: Implementación del Sistema

Introducción

El presente capítulo modela los artefactos referentes al flujo de trabajo de implementación, diagramas de componentes, ejemplos de código y principales interfaces de la aplicación así como una breve explicación de los mismos.

1 Componentes

Identifican objetos físicos que hay en un tiempo de ejecución, compilación o desarrollo además de tener interfaz propia, los componentes incluyen código en cualquiera de sus formatos (código fuente o ejecutable) e imágenes; pero pueden ser documentos manuales cuando se describen partes no informatizadas de un sistema de información.

1.1 Diagrama de componentes.

El diagrama de componente describe la descomposición física del sistema de software (y eventualmente su entorno organizativo) en componentes a efectos de construcción y funcionamiento. La descomposición del diagrama de componentes se realiza en términos de componentes y de relación entre los mismos.

De acuerdo a la arquitectura que presenta OpenClinica. En los diagramas de componentes realizados, se muestran como están distribuidos según el patrón arquitectónico Modelo-vista-Controlador.

En la **Vista**: Pueden encontrarse paquetes de componentes formados por las paginas .jsp, hojas de estilos .css, JavaScripts.js además de un paquete Internacionalización del Lenguaje.

En el **Controlador**: Pueden encontrarse paquetes de componentes formados por archivos .java que no son más que las controladoras dentro del sistema , además de un paquete formado por las librerías utilizadas por estas clases java.

En el **Modelo**: Se encuentran paquetes de componentes formados por archivos .java que se encargan de la abstracción del sistema con la BD. Llamados "X" Bean y "DAO".

Otros Paquetes: Se reflejan paquetes como el de base de datos, el paquete de las propiedades DAO.xml el cual contiene en formato XML las consultas en SQL necesarias para la búsqueda y edición de las entidades de la base de datos.

Observándose la relación de dependencia entre los paquetes:

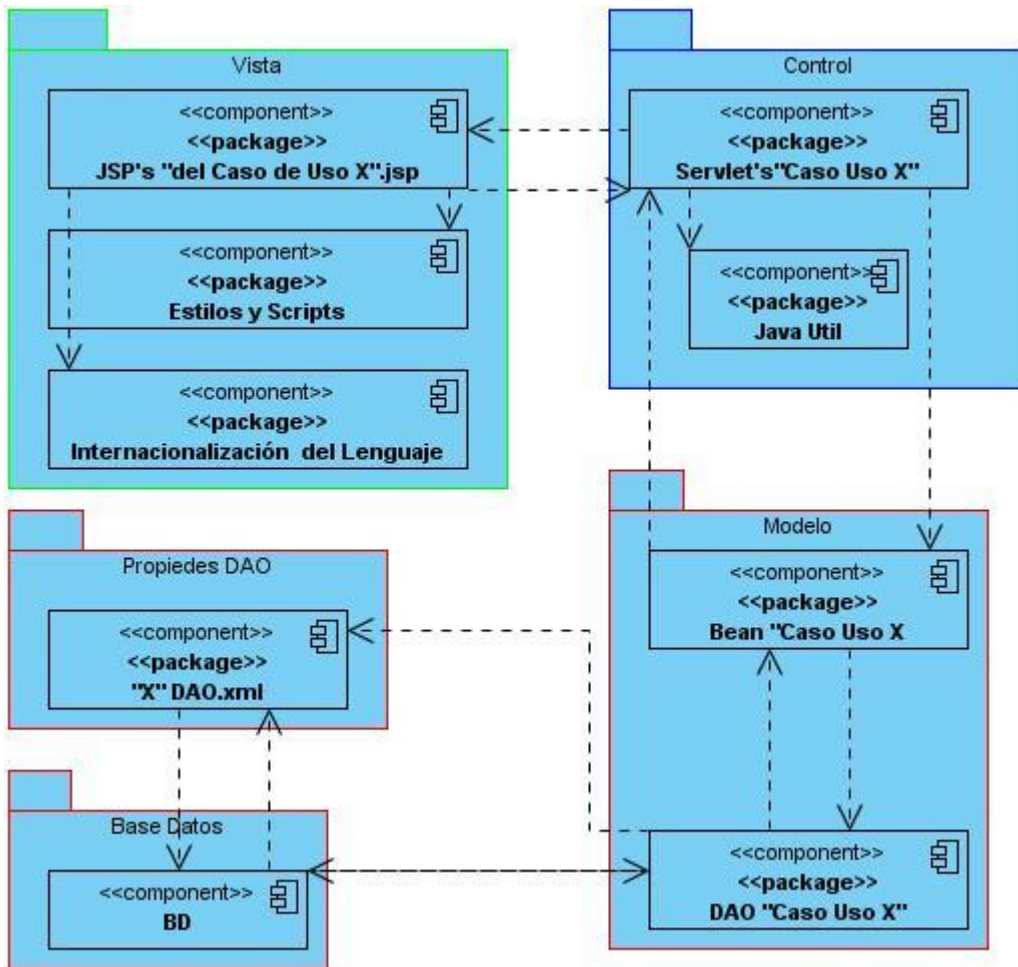


Ilustración 31 Diagrama de Componentes

1.2 Diagrama de componente del caso de uso Crear Nota de Monitoreo.

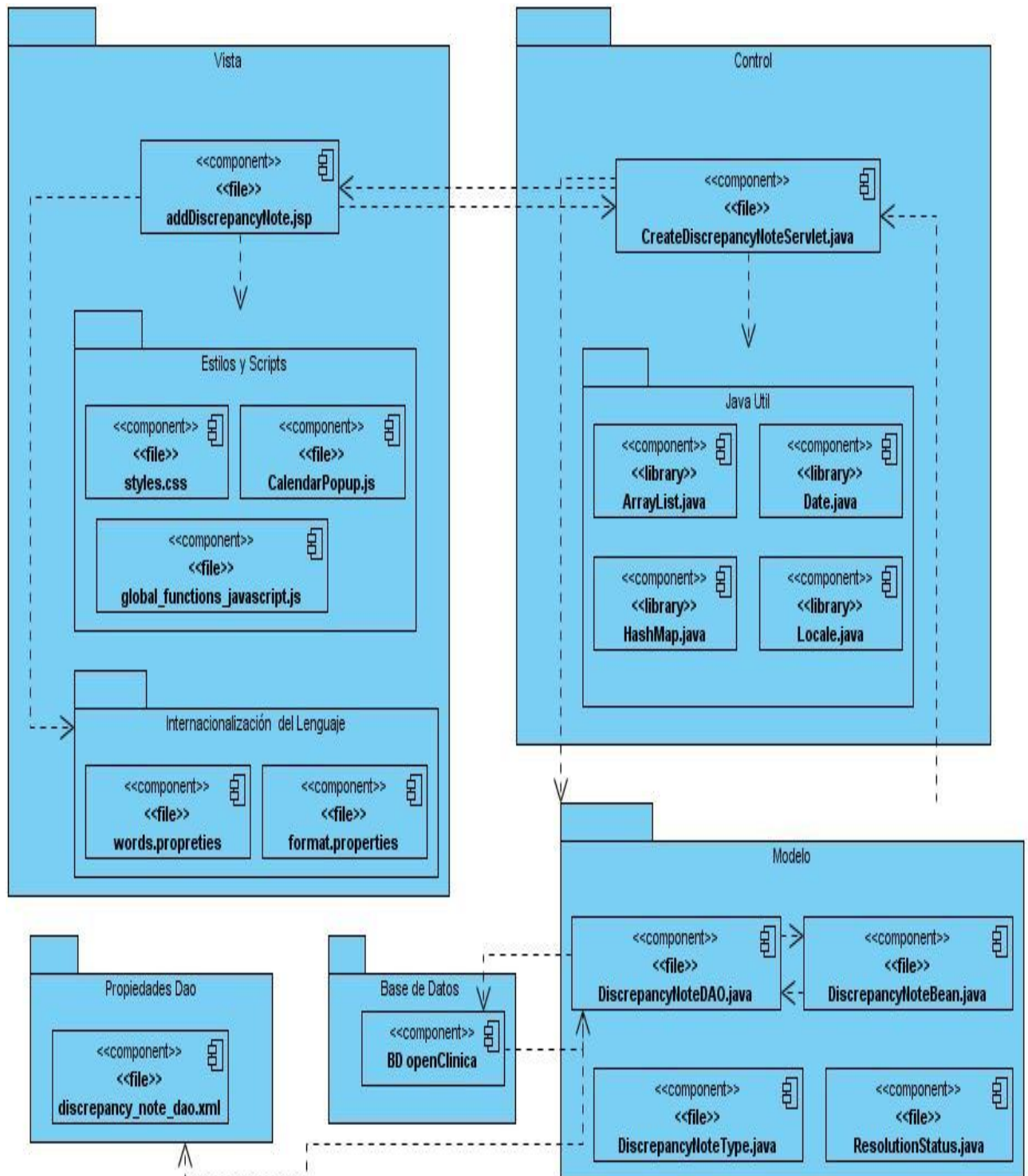


Ilustración 32 DC Crear Nota de Monitoreo

1.3 Diagrama de componente del caso de uso Responder Nota de Monitoreo.

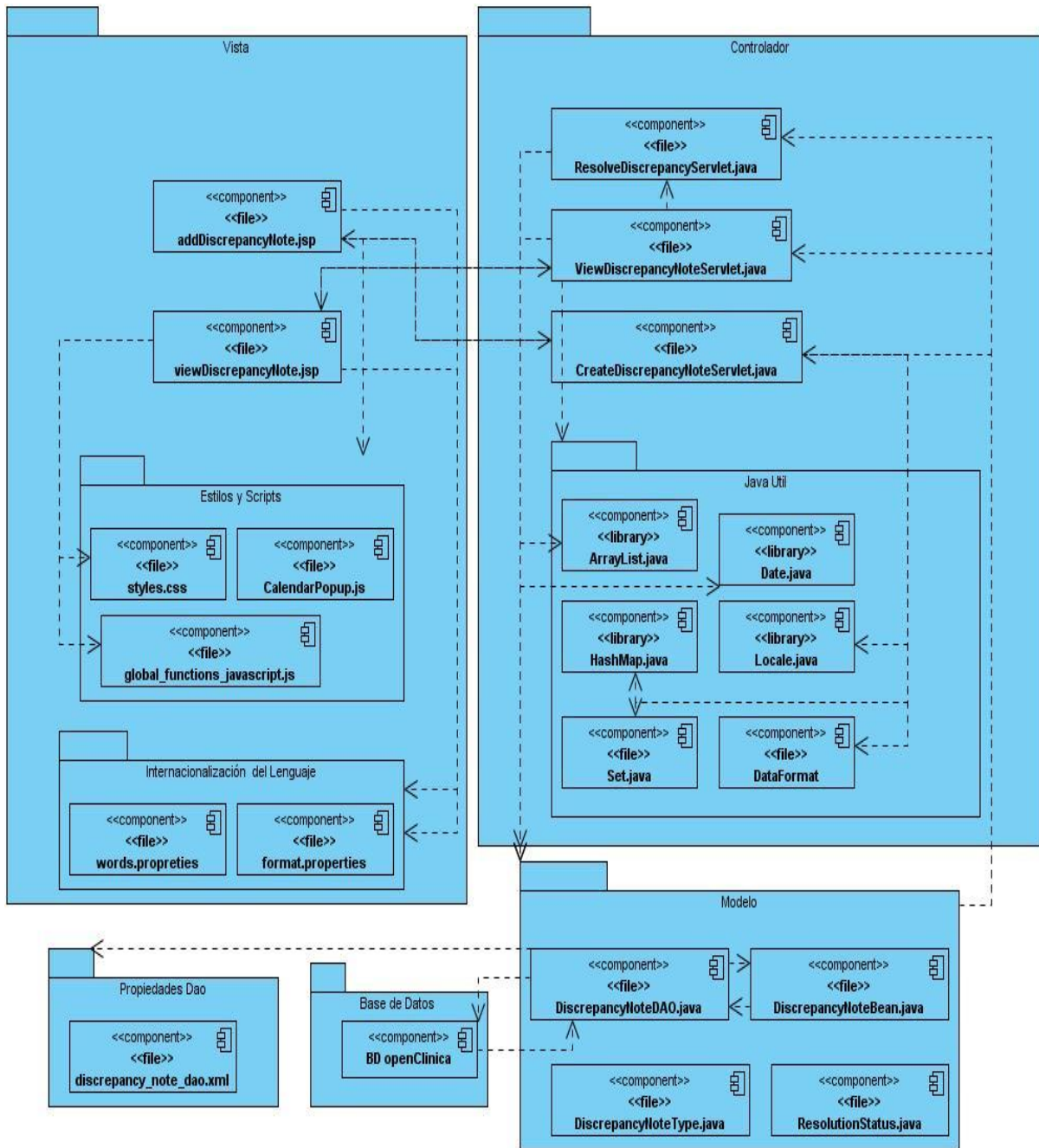


Ilustración 33 DC Responder Nota de Monitoreo

1.4 Diagrama de componente del caso de uso Crear Nota de General.

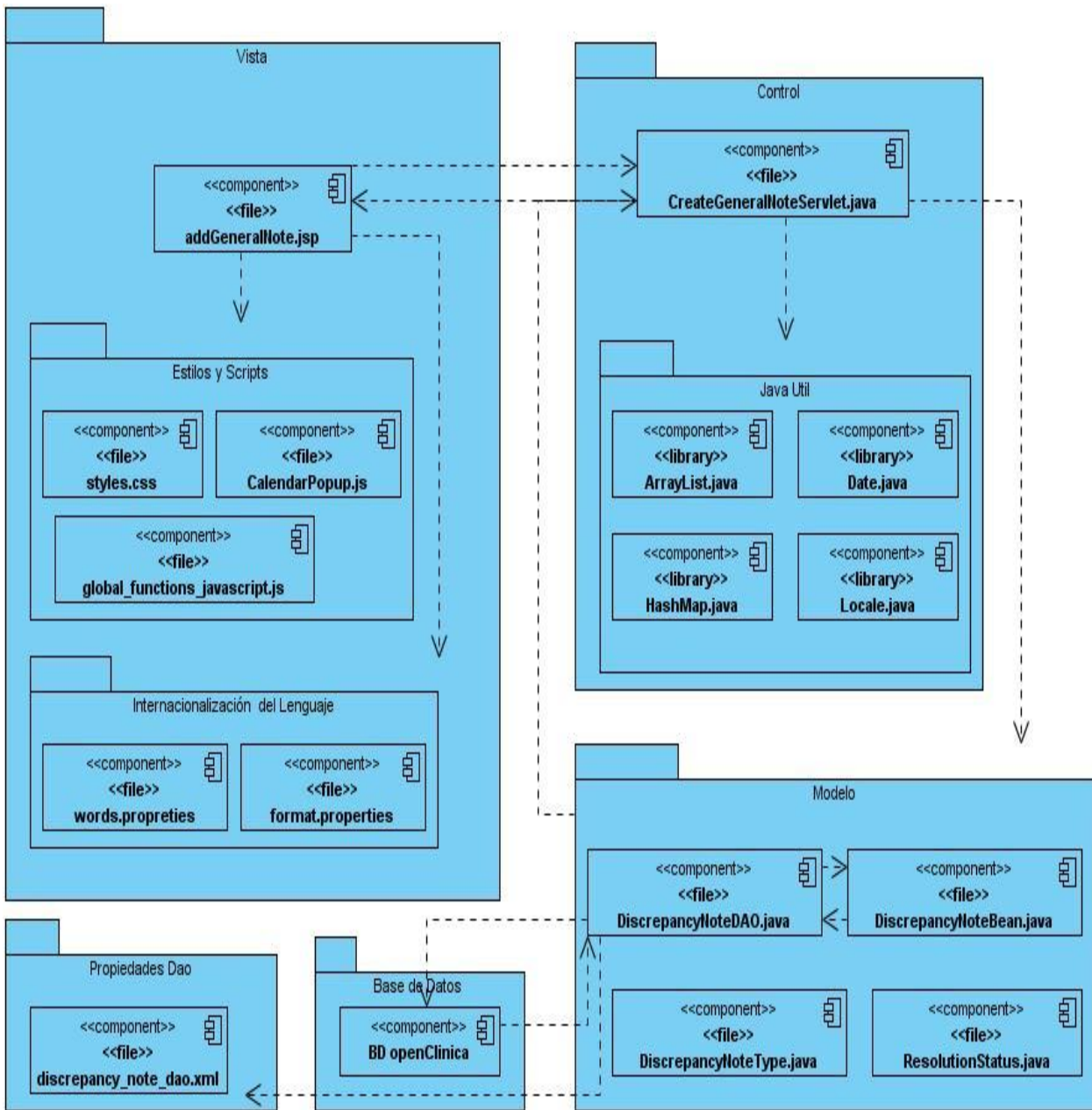


Ilustración 34 DC Crear Nota General

1.5 Diagrama de componente del caso de uso Responder Nota de Monitoreo.

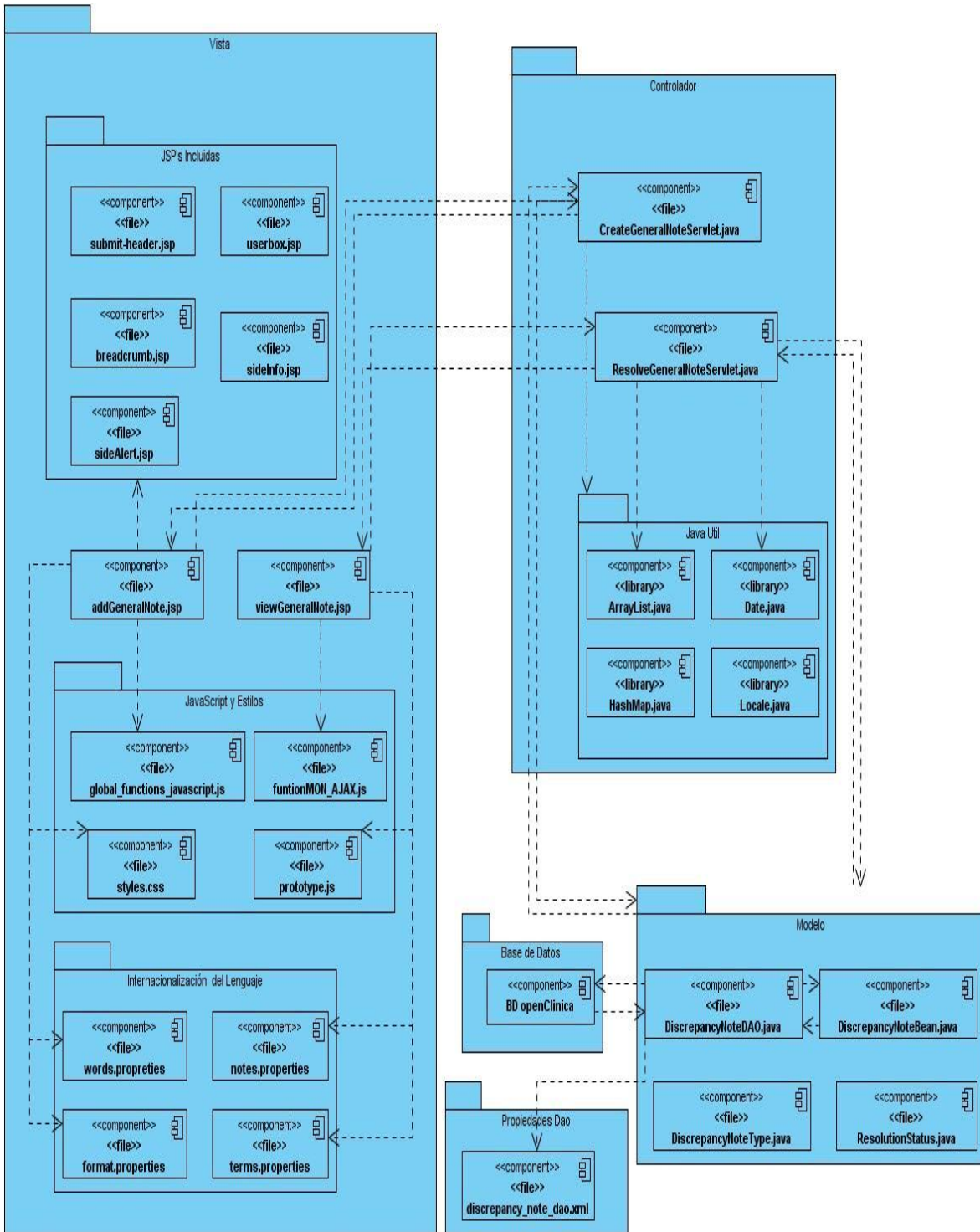


Ilustración 35 DC Responder Nota de Monitoreo

1.6 Diagrama de componente del caso de uso Monitorear Hoja de CRD.

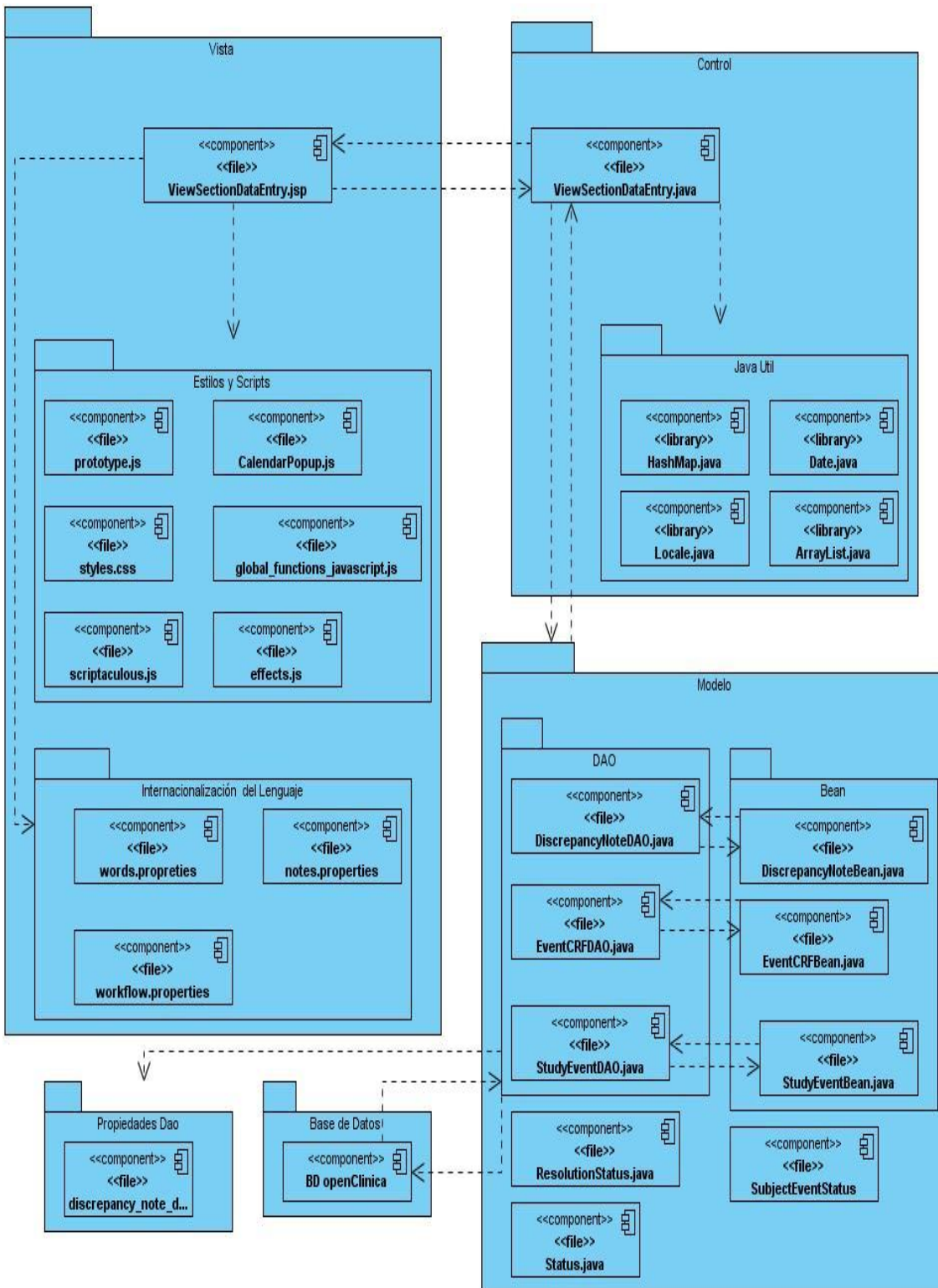


Ilustración 36 DC Monitorear Hoja de CRD

1.7 Diagrama de componente del caso de uso Visualizar Notas.

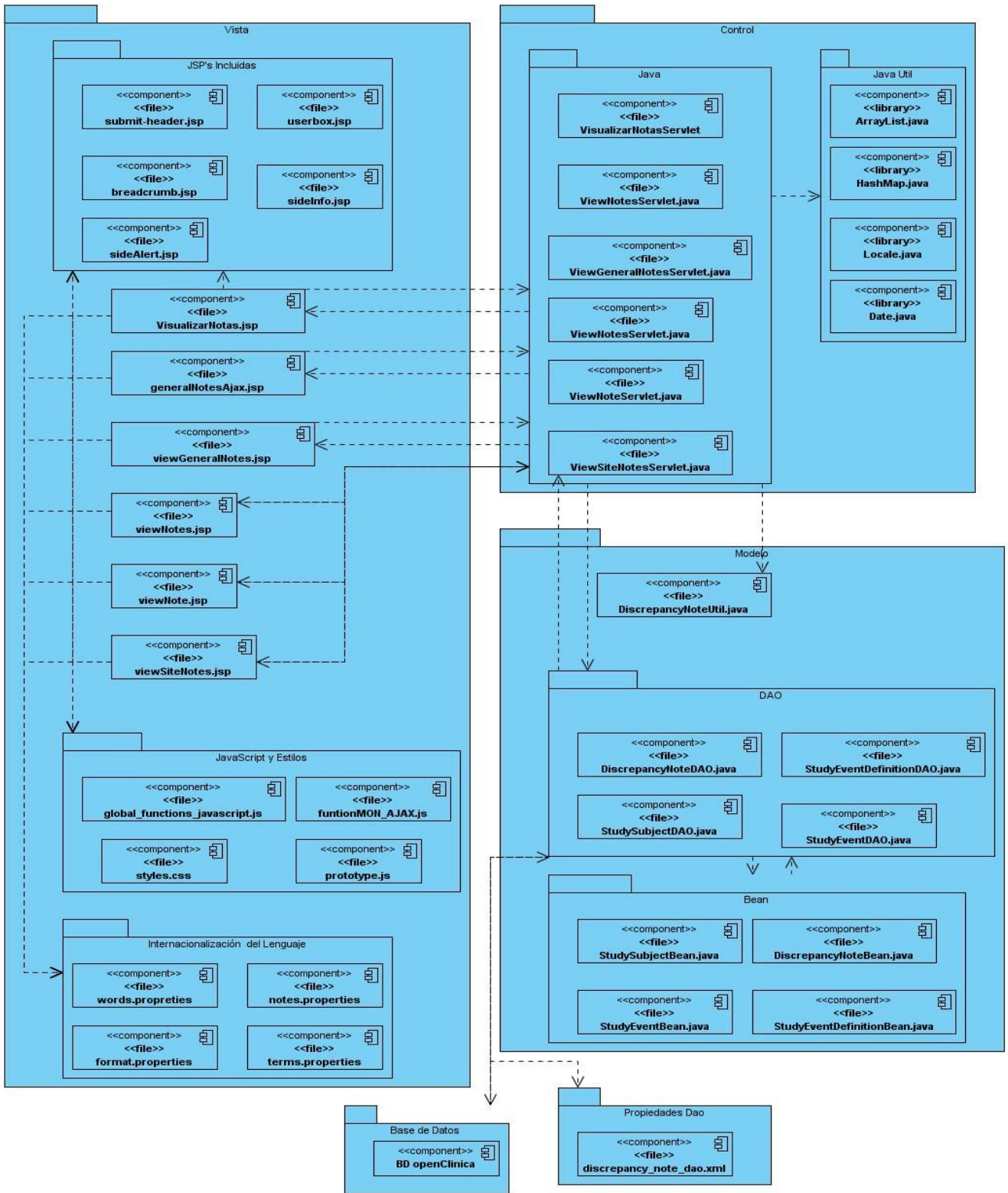


Ilustración 37 DC Vizualsizar Notas

2 Ejemplos de Códigos Relevantes.

2.1 CU Visualizar Notas

El siguiente código evidencia el uso de los objetos DAO para facilitar la abstracción a la base de datos. Por ejemplo:

- Según el valor de la variable “parentStudyId” realiza la correspondiente búsqueda en la base de datos y guarda el resultado de la búsqueda en la variable “allDefs” que es una lista de Momentos de Seguimiento.

```
int parentStudyId = currentStudy.getParentStudyId();
ArrayList studyGroupClasses = new ArrayList();
ArrayList allDefs = new ArrayList();
StudyGroupClassDAO sgcdao = new StudyGroupClassDAO(sm.getDataSource());
StudyEventDefinitionDAO seddao = new StudyEventDefinitionDAO(sm.getDataSource());
if (parentStudyId > 0) {
    StudyBean parentStudy = (StudyBean) stdao.findByPK(parentStudyId);
    studyGroupClasses = sgcdao.findAllActiveByStudy(parentStudy);
    allDefs = seddao.findAllActiveByStudy(parentStudy);
} else {
    parentStudyId = currentStudy.getId();
    studyGroupClasses = sgcdao.findAllActiveByStudy(currentStudy);
    allDefs = seddao.findAllActiveByStudy(currentStudy);
}
```

Ilustración 38 Ejemplo de código del CU Vizualizar Notas

2.2 CU Crear Nota General

El próximo segmento de código ejemplifica el uso de los objetos DAO y Bean para crear un nuevo registro de una Nota General.

```
DiscrepancyNoteDAO dndao = new DiscrepancyNoteDAO(sm.getDataSource());
DiscrepancyNoteBean note = new DiscrepancyNoteBean();
note.setDescription(description);
note.setDetailedNotes(detailedDes);
note.setOwner(ub);
note.setCreatedDate(new Date());
note.setResolutionStatusId(resStatusId);
note.setDiscrepancyNoteTypeId(typeId);
note.setStudyId(currentStudy.getId());
note.setParentDnId(parent.getId());
note.setEntityType(DiscrepancyNoteType.get(typeId).getNameWithOutBundle());
dndao.create(note);
```

Ilustración 39 Ejemplo de código del CU Crear Nota General

Como primer paso se crea un objeto Bean que representará la nota que posteriormente será guardada en la base de datos y otro objeto DAO que será el responsable de salvar dicha nota. Los métodos set son usados para cambiar el valor de algún atributo de la nota por ejemplo: setDescription (método encargado de modificar la descripción de la nota). Finalmente el objeto ejecuta la sentencia create(), para salvar la nota en la base de datos, pasándole como parámetro el objeto Bean nombrado note.

2.3 CU Crear Nota de Monitoreo

Este caso es muy similar al anterior con la diferencia que aquí se refleja los pasos para la validación de los datos provenientes de la vista.

```
Validator v = new Validator(request);
String description = fp.getString("description");
int typeId = fp.getInt("typeId");
int resStatusId = fp.getInt(RES_STATUS_ID);
String detailedDes = fp.getString("detailedDes");
v.addValidation("description", Validator.NO_BLANKS);
v.addValidation("description", Validator.LENGTH_NUMERIC_COMPARISON,
NumericComparisonOperator.LESS_THAN_OR_EQUAL_TO, 255);
v.addValidation("detailedDes", Validator.LENGTH_NUMERIC_COMPARISON,
NumericComparisonOperator.LESS_THAN_OR_EQUAL_TO, 1000);

v.addValidation("typeId", Validator.NO_BLANKS);

HashMap errors = v.validate();
DiscrepancyNoteBean note = new DiscrepancyNoteBean();
note.setDescription(description);
note.setDetailedNotes(detailedDes);
note.setOwner(ub);
note.setCreatedDate(new Date());
note.setResolutionStatusId(resStatusId);
note.setDiscrepancyNoteTypeId(typeId);

if (errors.isEmpty()) {
    dndao.create(note);
}
```

Ilustración 40 Ejemplo de código de CU Crear Nota de Monitoreo

Como primer paso se crea un objeto de tipo Validator que será el responsable de ejecutar las validaciones. El objeto de tipo HashMap contiene el resultado de la validación. Si dicho objeto está vacío se ejecuta la sentencia create() para salvar la nota en la base de datos.

2.4 Monitorear una Hoja de CRD

Los objetos DAO además de salvar la información también pueden realizar actualizaciones sobre una entidad. Para eso dichos objetos poseen la sentencia `update()`. La siguiente imagen muestra como se actualiza la información de un MS y de un CRD.

```
EventCRFDAO eventCrfDao = new EventCRFDAO (sm.getDataSource());
EventCRFBean eventCrfBean = (EventCRFBean) eventCrfDao.findByPK(eventCRFId);

StudyEventDAO studyEventDAO = new StudyEventDAO(sm.getDataSource());
int studyEventID = eventCrfBean.getStudyEventId();
StudyEventBean studyEventBean =(StudyEventBean)studyEventDAO.findByPK(studyEventID);

int id_statusMON = fp.getInt("statusMON");
if(id_statusMON == 9 || id_statusMON == 10){

    studyEventBean.setSubjectEventStatus(SubjectEventStatus.get(id_statusMON));
    studyEventDAO.update(studyEventBean);
}
```

Ilustración 41 Ejemplo de código del CU Monitorear Hoja de CRD

2.5 Función JavaScript que ejecuta funciones de AJAX. CU Visualizar Notas

Esta función actualiza por peticiones de AJAX el listado de notas generales en la interfaz del CU Visualizar Notas permitiéndole al listado la paginación sin tener que actualizar el navegador. La función "Updater" contiene las funciones de AJAX para realizar la petición y la función "okFunc" es quien muestra en la interfaz el listado si y solo si la petición de AJAX tuvo éxito. Ver ilustración 41.

```
function Updater(param) {  
  
    new Ajax.Request('ViewGNotesAjax', {  
        method: 'post',  
        postBody: param,  
        asynchronous: true,  
        onSuccess: okFunc  
    });  
}  
  
function okFunc(t) {  
    t.responseText.evalScripts();  
    $('dg_table').innerHTML=t.responseText;  
}
```

Ilustración 42 Función AJAX

3 Principales Interfaces

3.1 CU Crear Nota de Monitoreo.

Añadir Nota de Monitoreo Cerrar Ventana

| | |
|----------------------------------|--|
| Sujeto | Pac_12 |
| Momento de Seguimiento | Momento_A |
| Fecha del Momento de Seguimiento | 20/02/2009 |
| CRD | Fallecimiento |
| Entidad Tipo/Campo | itemData/value |
| Nombre del Item | tumor |
| Discrepancy Thread ID | |
| Tipo | <input type="radio"/> Comprobación de Validación Fallida <input type="radio"/> Anotación <input checked="" type="radio"/> Consulta |
| Descripción | <input type="text"/> |
| Nota Detallada | <input type="text"/> |
| Estado de resolución | <input type="text" value="Nuevo"/> |
| Fecha | 07/06/2009 |
| Nota Padre | Ninguna |

3.2 Responder Nota de Monitoreo

Ver Notas de Monitoreo
[Cerrar Ventana](#)

tumor = 1

| | | | |
|--|------------|--------------------------------|---------------|
| Sujeto: | Pac_12 | Momento de Seguimiento: | Momento_A |
| Fecha del Momento de Seguimiento: | 20/02/2009 | CRD: | Fallecimiento |

Detalles de la Nota

| | | | |
|-------------------------------|-----------------------|---|----------------------|
| <input type="checkbox"/> kkkk | | Last updated: 05/06/2009 by monitor | |
| ID: 336 | Tipo: Consulta | Estado de resolución: Actualizado | # Notas: 2 |
| kkkk | | 05/06/2009 by monitor | Estatus: Nuevo |
| respeta | | 05/06/2009 by monitor | Estatus: Actualizado |
| asd | | | |
| | | Responder Nota | |

[Nueva Nota de Monitoreo](#)

Añadir Nota de Monitoreo

[Cerrar Ventana](#)

Sujeto Pac_12
 Momento de Seguimiento Momento_A
 Fecha del Momento de Seguimiento 20/02/2009
 CRD Fallecimiento
 Entidad Tipo/Campo itemData/value
 Nombre del Item tumor

Discrepancy Thread ID

Tipo
 Comprobación de Validación Fallida
 Anotación
 Consulta

Descripción

Nota Detallada

Estado de resolución

Fecha 07/06/2009

Nota Padre Ninguna

3.3 Crear Nota General

Añadir Nota General Cerrar Ventana

Tipo Anotación
 Consulta

Descripción

Nota Detallada

Estado de resolución

Fecha 01/06/2009

Nota Padre Ninguna

3.4 Responder Nota General

Resolver Nota General

[Cerrar Ventana](#)

Detalles de la Nota

| | | | |
|--------------------------------|------------------------|---|--------------------------------|
| <input type="checkbox"/> tesis | | Last updated: 05/06/2009 by monitor | |
| ID: 338 | Tipo: Anotación | Estado de resolución: Actualizado | # Notas: 2 |
| xx | | | |
| tesis | | 05/06/2009 by monitor | Estatus: Nuevo |
| tesis | | 05/06/2009 by monitor | Estatus: Actualizado |
| ihnuihnuh8nuh8n8unj | | | |
| | | | Responder Nota |

[Nueva Nota General](#)

Añadir Nota General

[Cerrar Ventana](#)

Nota General ID 34

Tipo Anotación
 Consulta

Descripción

Nota Detallada

Estado de resolución

Fecha 01/06/2009

Nota Padre xxxx

[Ver nota padre y nota/s relacionada/s](#)

3.5 Monitorear Hoja de CRD

Ver Entrada de Sección de Datos de Fallecimiento 1


| Info hoja CRD- | |
|--|---|
| Fallecimiento 1  | Notas de Discrepancia: 0 Abierto, 0 Actualizado, 0 Resuelto, 0 Cerrado |
| ID del Sujeto del Estudio: Pac_2 | ID de la Persona: Barte |
| Estudio/Centro: Cancer_Pulmon | Edad en la Inclusión: |
| Momento de Seguimiento: Momento_G(20/02/2009) | Fecha de Nacimiento: 01/01/1254 Sexo: M |
| Número de Ocurrencias: 1 | |
| Nombre del Entrevistador: * <input type="text" value="Dr Perez"/> | Fecha de la Entrevista: * <input type="text" value="12/04/2009"/> |

| Monitoreo | |
|--|--|
| <input checked="" type="radio"/> Monitoreo Comenzado | <input type="radio"/> Monitoreo Completado |
| <input type="button" value="Monitorear"/> <input type="button" value="Salir"/> | |

◀ **Falleci...(5/5)** ▶ -- Seleccione para Ir -- ▼

| Título: Fallecimiento | |
|---|---|
| Página: 1 | |
| Fecha de fallecimiento | <input type="text" value="07/01/2009"/> |
| Fallecimiento relacionado con el tumor de mamas | <input type="text" value="No"/> |
| Se realizó necropsia | <input type="text" value="No"/> |
| Fecha de Necropsia | <input type="text" value="30/01/2009"/> |
| Causa de fallecimiento | <input type="text" value="asdasdadsa"/> |

3.6 Visualizar Notas



Inicio Enviar Datos Extraer Datos
Ver Todos Sujetos
Notas y Discrepancias

[Informar Problema](#) | [Soporte](#)

Usuario: monitor
 Estudio: Cancer_Pulmon
 ID: CP
 Rol: monitor
[Cambiar Centro/Estudio](#)
[Salir](#)

Alertas y Mensajes

Instrucciones
 Seleccione un sujeto si desea ver más detalles, añadir o revisar sus datos.

Información
 Estudio: Cancer_Pulmon
 Fecha de Inicio: 20/02/2009
 Fecha de Fin: 24/02/2009
 IP: Evangelio
 Fecha de Aprobación IRB / Verificación del Protocolo: 23/10/2006
 ¿Recopilar Fecha de Nacimiento del Sujeto? Sólo Año de Nacimiento
 Número de Sujetos : 3

Gestionar todas las Notas de Discrepancia en Cancer_Pulmon ?

Estatus : (Nuevo) (Actualizado) (Resuelto) (Cerrado) [#] = Momentos de seguimiento repetidos

Notas de Monitoreo

Todos los Momentos de Seguimiento
Momento_A
Momento_F
Momento_G

Página 1 de 1 Busca

| ID del Sujeto del Estudio | Estatus del Sujeto | Nombre del Centro | Momento_A | Momento_F | Momento_G | leandro | Acciones |
|---------------------------|--------------------|-------------------|------------|------------|-----------|---------|----------|
| Pac_12 | activo | CP | (2) (2) | (2) (2) | | | |
| Pac_2 | signed | CP | | | | | |

Notas de Sitio

Página 1 de 1 Busca

| ID del Sujeto del Estudio | Estatus del Sujeto | Nombre del Centro | Momento_A | Momento_F | Momento_G | leandro | Acciones |
|---------------------------|--------------------|-------------------|------------|------------|-----------|---------|----------|
| Pac_12 | activo | CP | (2) (2) | (2) (2) | | | |
| Pac_2 | signed | CP | | | | | |

Notas Generales

Página 1 de 3 Ver Más Añadir Nota General

| ID | Fecha de Creación | Fecha de Actualización | Descripción | Notas Detalladas | # Notas | Acciones |
|----|-------------------|------------------------|--------------|------------------|---------|----------|
| 34 | 25/05/2009 | 25/05/2009 | xxxx | rrrrrr | 2 | |
| 37 | 25/05/2009 | 25/05/2009 | H | H | 1 | |
| 76 | 26/05/2009 | 26/05/2009 | Rciahr | Rcihard | 2 | |
| 81 | 27/05/2009 | 27/05/2009 | zxxx | xxxx | 2 | |
| 88 | 27/05/2009 | 27/05/2009 | asd | asdaasd | 1 | |
| 90 | 27/05/2009 | 27/05/2009 | xxxxxxxxxx | xxxxxxxxxx | 1 | |
| 92 | 27/05/2009 | 27/05/2009 | asdas | dasdasdasda | 1 | |
| 94 | 27/05/2009 | 27/05/2009 | 444444444444 | 444444444444 | 1 | |
| 96 | 27/05/2009 | 27/05/2009 | asd | asdasdasdasd | 1 | |
| 98 | 27/05/2009 | 27/05/2009 | asdas | dasdasdasdasda | 1 | |

Flujo de Trabajo

Enviar Datos ➡ List Discrepancy Notes

Conclusiones

Con el desarrollo del capítulo se crearon los Diagramas de Componentes para cada Caso de Uso, se representaron fragmentos de código utilizados en la implementación del sistema alasClínicas y se visualizaron las principales interfaces de dicha aplicación. Se evidencia la disposición física del sistema y se muestra su estructura interna.

Conclusiones Generales

Durante el desarrollo del presente trabajo, se pudo arribar a aspectos concluyentes después del profundo análisis de los principales conceptos de Proceso de Conducción de Ensayos Clínicos Cubanos, analizando el comportamiento de estos en el sistema OpenClinica. Se capturaron requisitos funcionales y no funcionales agrupándose los funcionales en 9 Casos de Uso. Dando cumplimiento a los objetivos trazados logrando identificar las funcionalidades a agregar o modificar implementando clases que se agregaron y modificaron dentro del sistema para las Clínicas. Con la culminación de este trabajo se espera un sistema capaz de garantizar de forma segura y rápida la Gestión de Datos de Ensayos Clínicos Cubanos.

Recomendaciones:

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

- Realizar las pruebas de la aplicación
- Realizar la integración del sistema
- Mejorar el proceso de búsqueda que implementa OpenClinica en el listado de todas las notas.
- Capacitar al personal encargado de monitorear el sistema alasClínicas en el CIM.

Referencias Bibliográficas

1. ELEC. *Cáncer en español-National Cancer Institute*, 2008. [2009]. Disponible en: <http://www.icic.nci.nih.gov/espanol>
2. OMS. *Organización Mundial de la Salud*, 2009. [2009]. Disponible en: <http://www.who.int/mediacentre/factsheets/fs297/es/index.html>
3. GEORGIA, M. C. O. *Study Manager*, 2008. [2009]. Disponible en: <http://www.mcg.edu/OCIS/studymanager.html>
4. IS1, C. *Tema 2: Fase de Inicio. Negocio y Requerimiento. Conferencia 4: Fase de Inicio. Flujo de trabajo de requerimientos*. <http://eva.uci.cu>, 2007-2008.
5. BLANCO, D. *Estándares de OpenClinica*, 2009. [2009]. Disponible en: <http://www.slideshare.net/davidblanco/presentacin-openclinica>
6. GARCÍA, J. *Análisis y Diseño*, 2005. [2009]. Disponible en: <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>

Bibliografía

- BLANCO, D. *Estándares de OpenClinica*, 2009. [2009]. Disponible en: <http://www.slideshare.net/davidblanco/presentacin-openclinica>
- CARLA LAMBERTINI and C. CORTÉS. *Oracle*, 2006. [2009]. Disponible en: http://www.oracle.com/global/lad/corporate/press/2007_jan/oracle_remote_data_capture_onsite.html
- CIM. *Centro de Inmunología Molecular*, 2008. [2009]. Disponible en: <http://www.cim.sld.cu/>
- DUNIA RUBIO ALVARES and K. D. DOMÍNGUEZ. *Sistema de manejo de datos de los Ensayos Clínicos: Módulo de Monitoreo*. La Habana, UCI, junio de 2007. p.
- ELEC. *Cáncer en español-National Cancer Institute*, 2008. [2009]. Disponible en: <http://www.wicic.nci.nih.gov/espanol>
- GARCÍA, J. *Análisis y Diseño*, 2005. [2009]. Disponible en: <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>
- GEORGIA, M. C. O. *Study Manager*, 2008. [2009]. Disponible en: <http://www.mcg.edu/OCIS/studymanager.html>
- IS1, C. *Tema 2: Fase de Inicio. Negocio y Requerimiento. Conferencia 4: Fase de Inicio. Flujo de trabajo de requerimientos*. <http://eva.uci.cu>, 2007-2008.
- IVAR JACOBSON; GRADY BOOCH, et al. *El Proceso Unificado de Desarrollo de Software, Volumen*. Editorial Félix Varela. La Habana, 2004. p.
- LARMAN, C. *UML y Patrones, Introducción al análisis y diseño orientado a objetos*. Editorial Félix Varela. La Habana, 2004. p.
- MARRERO, R. A. C. *Sistema de Manejo de Datos de Ensayos Clínicos Cubano*. La Habana, UCI, junio de 2008. p.
- MENDOZA, M. A. S. *Informatízate*, 2002-2004. [2009]. Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
- OBJECT, D. A., 2009. [2009]. Disponible en: http://es.wikipedia.org/wiki/Data_Access_Object
- OMS. *Organización Mundial de la Salud*, 2009. [2009]. Disponible en: <http://www.who.int/mediacentre/factsheets/fs297/es/index.html>
- OPENCLINICA. *OpenClinica, Open Source of Clinical Research 2003-2009*. [2009]. Disponible en: <http://www.openclinica.org/page.php?pid=58>
- PRESSMAN, R. S. *Ingeniería del Software, un enfoque práctico*. Editorial Félix Varela. 2005. p.
- UCI. *Tema 1: Introducción a la Ingeniería de Software. Conferencia 1: Introducción a la Ingeniería de Software*. <http://eva.uci.cu>, 2007-2008.

Anexos

A continuación se muestran documentos e imágenes asociadas al presente trabajo de diploma.

1 CU: Mostrar Informe de las Notas Generales

| | |
|------------------------|--|
| Caso de Uso: | Mostrar Informe de las Notas Generales |
| Actores: | Monitor (Inicia) |
| Resumen: | El caso de uso inicia cuando el Monitor selecciona " <i>Mostrar Informe de las Notas Generales</i> ". EL sistema muestra listado con todas las notas del centro en un formato adecuado a la impresión. |
| Precondiciones: | El Monitor debe haberse autenticado al sistema. |
| Referencias | RF16, RF13 |
| Prioridad | Crítico |

Flujo Normal de Eventos

| Acción del Actor | Respuesta del Sistema |
|---|--|
| 1- Del listado de todas las Notas Generales del centro el Monitor selecciona " <i>Mostrar Informe de las Notas Generales</i> ". | 2- El sistema muestra un mensaje de alerta indicando al Monitor los pasos necesarios para realizar la impresión. |
| 3- El Monitor cierra la ventana del mensaje. | 4- El sistema visualiza un listado con todas las Notas Generales del centro en un formato adecuado para la impresión. Cada nota tiene: <ul style="list-style-type: none"> • Autor • Fecha de Creación • Fecha de Actualización • Estado de Resolución • Descripción • Notas Detalladas Finaliza el Caso de |

Prototipo de Interfaz

Notas Generales

| Autor | Fecha de Creación | Fecha Actualización | Estado de Resolución | Descripción | Notas Detalladas |
|-------|-------------------|---------------------|----------------------|-------------|------------------|
| | | | | | |

2 CU: Mostrar Informe de las Notas de Sitio

| | |
|------------------------|---|
| Caso de Uso: | Mostrar Informe de las Notas de Sitio |
| Actores: | Monitor (Inicia) |
| Resumen: | El caso de uso inicia cuando el Monitor selecciona " <i>Mostrar Informe de las Notas de Sitio</i> ". EL sistema muestra listado con todas las notas del centro en un formato adecuado a la impresión. |
| Precondiciones: | El Monitor debe haberse autenticado al sistema. |
| Referencias | RF15, RF11 |
| Prioridad | Crítico |

Flujo Normal de Eventos

| Acción del Actor | Respuesta del Sistema |
|---|---|
| 1- Del listado de todas las Notas de Sitio del centro el Monitor selecciona " <i>Mostrar Informe de las Notas de Sitio</i> ". | 2- El sistema muestra un mensaje de alerta indicando al Monitor los pasos necesarios para realizar la impresión. |
| 3- El Monitor cierra la ventana del mensaje. | 4- El sistema visualiza un listado con todas las Notas de Sitio del centro en un formato adecuado para la impresión. Cada nota tiene: <ul style="list-style-type: none"> • Paciente • Fecha de Creación • Momento de Seguimiento (estado) • CRD • Nombre de Dato • Valor de Dato • Descripción • Notas Detalladas • Autor |

Finaliza el Caso de Uso

Prototipo de Interfaz

Notas de Sitio

| Paciente | Fecha de Creación | Momento de Seguimiento (estado) | CRD | Nombre del Dato | Valor del Dato | Descripción | Notas Detalladas | Autor |
|----------|-------------------|---------------------------------|-----|-----------------|----------------|-------------|------------------|-------|
| | | | | | | | | |

3 CU: Mostrar Informe de las Notas de Monitoreo

| | | |
|--|--|--|
| Caso de Uso: | Mostrar Informe de las Notas de Monitoreo | |
| Actores: | Monitor (Inicia) | |
| Resumen: | El caso de uso inicia cuando el Monitor desde el listado de todas las Notas de Monitoreo del centro selecciona " <i>Mostrar Informe de las Notas de Monitoreo</i> ". EL sistema muestra listado con todas las notas del centro en un formato adecuado a la impresión. | |
| Precondiciones: | El Monitor debe haberse autenticado al sistema. | |
| Referencias | RF14, RF12 | |
| Prioridad | Crítico | |
| Flujo Normal de Eventos | | |
| Acción del Actor | | |
| Respuesta del Sistema | | |
| 1-Del listado de todas las Notas de Monitoreo del centro el Monitor selecciona " <i>Mostrar Informe de las Notas de Monitoreo</i> ". | 2-El sistema muestra un mensaje de alerta indicando al Monitor los pasos necesarios para realizar la impresión. | |
| 3-El Monitor cierra la ventana del mensaje. | 4-El sistema visualiza un listado con todas las Notas de Monitoreo del centro en un formato adecuado para la impresión. Cada nota tiene: <ul style="list-style-type: none"> • Paciente • Fecha de Creación • Fecha de Actualización • Momento de Seguimiento (estado) • CRD • Nombre del Dato • Valor del Dato • Estado de Resolución • Tipo | |

| | | | | | | | | | | <ul style="list-style-type: none"> • Descripción • Notas Detalladas • Autor <p>Finaliza el Caso de Uso</p> | |
|------------------------------|-------------------|------------------------|---------------------------------|-----|-----------------|----------------|----------------------|------|-------------|--|-------|
| Prototipo de Interfaz | | | | | | | | | | | |
| Notas de Monitoreo | | | | | | | | | | | |
| Paciente | Fecha de Creación | Fecha de Actualización | Momento de Seguimiento (estado) | CRD | Nombre del Dato | Valor del Dato | Estado de Resolución | Tipo | Descripción | Notas Detalladas | Autor |
| | | | | | | | | | | | |

4 Diagramas de secuencias

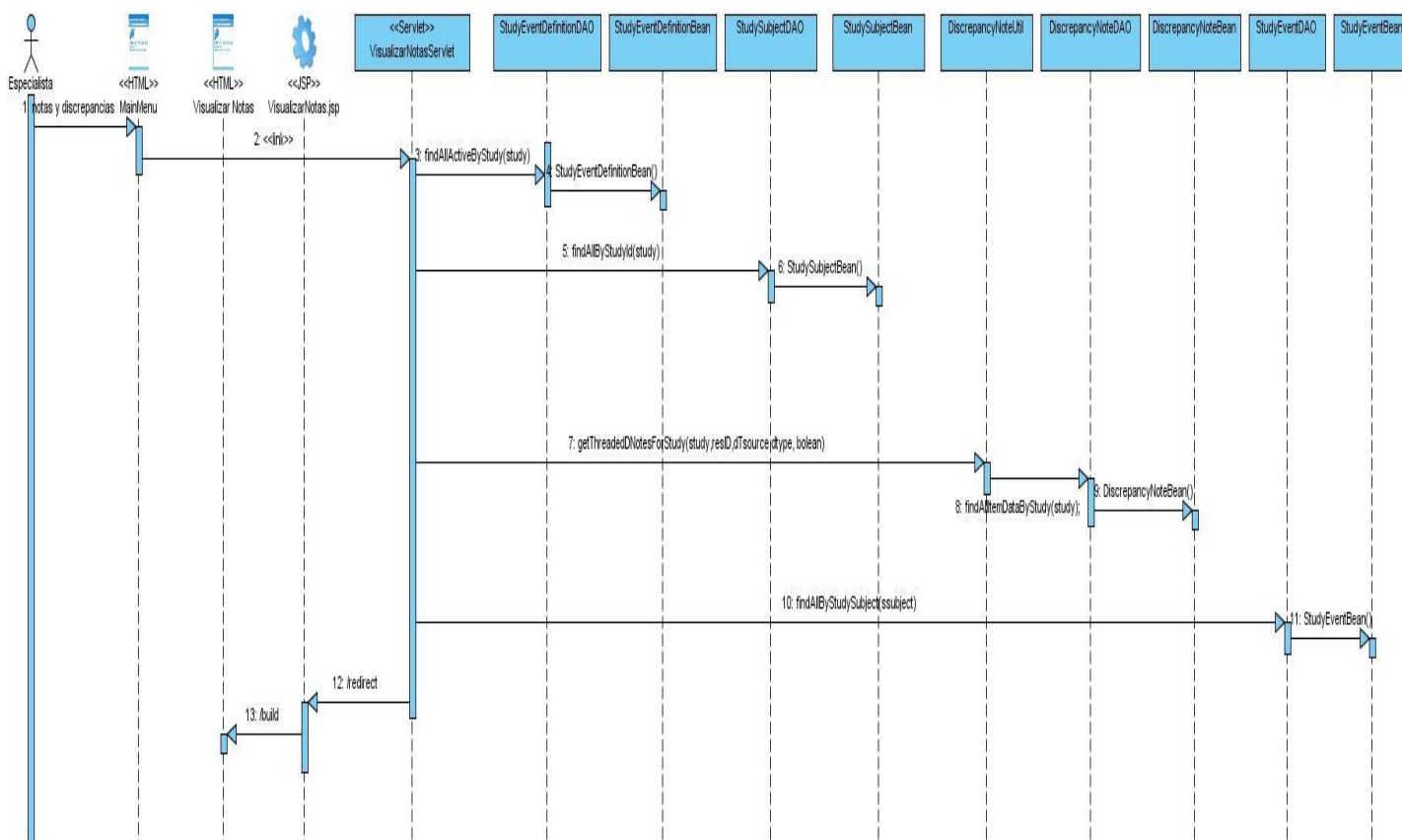


Ilustración 43 Diagrama de secuencia del CU Visualizar Notas: ver Notas de Monitoreo y Sitio

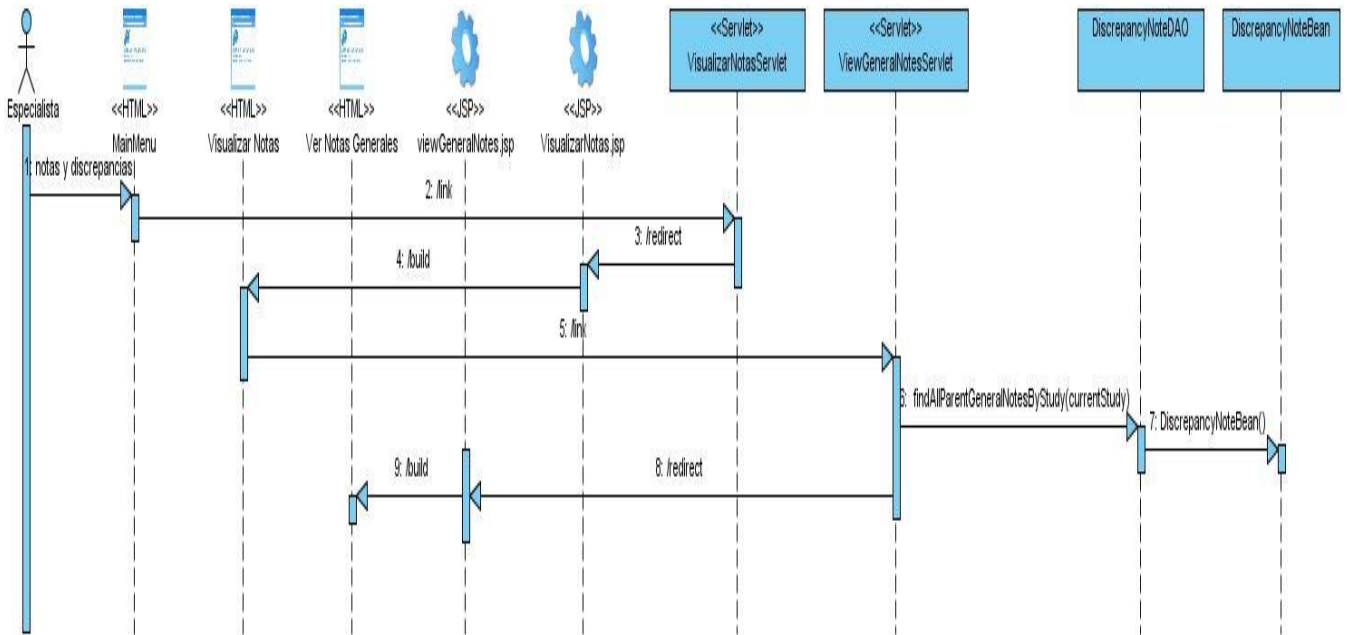


Ilustración 44 Diagrama de secuencia del CU Visualizar Notas: listar Notas Generales

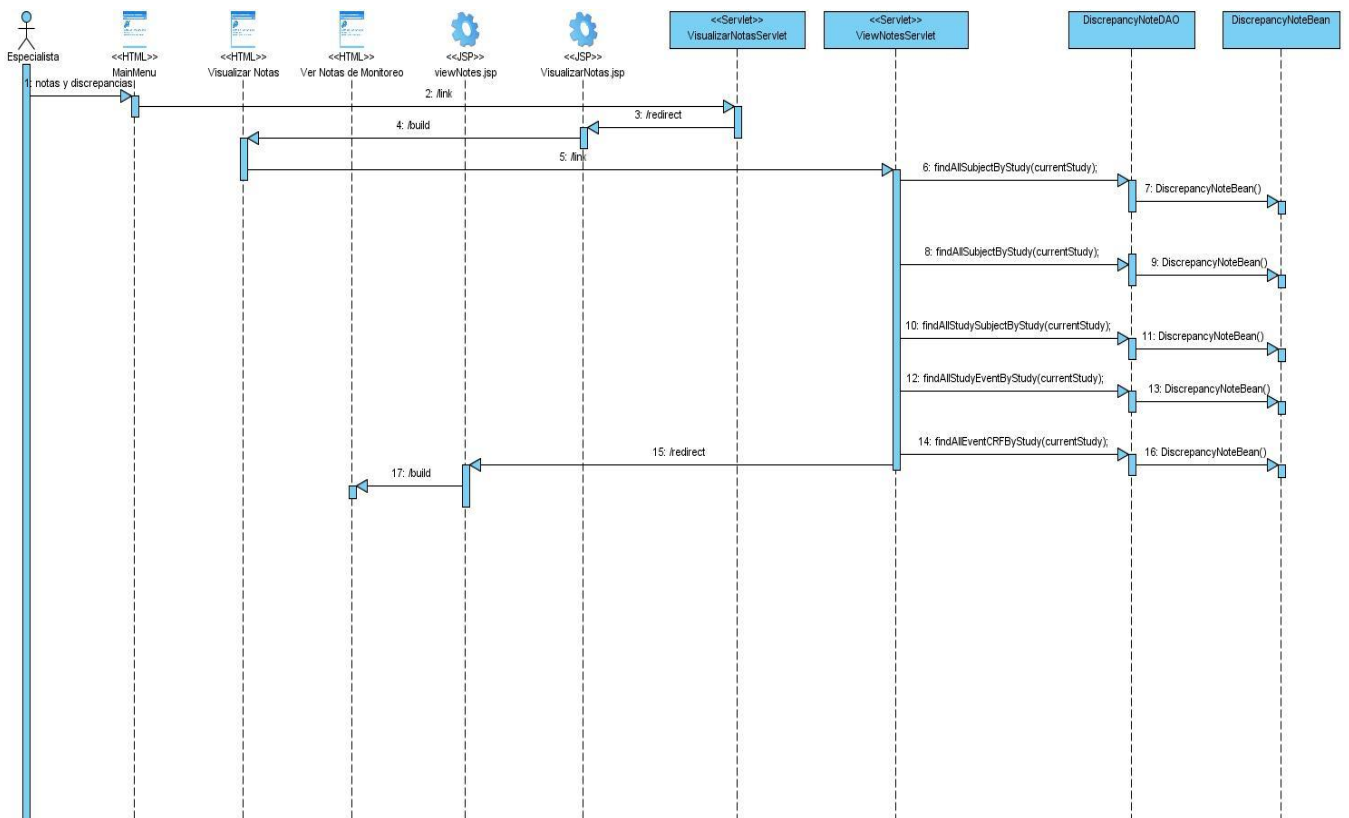


Ilustración 45 Diagrama de secuencia del CU Visualizar Notas: listar Notas de Monitoreo

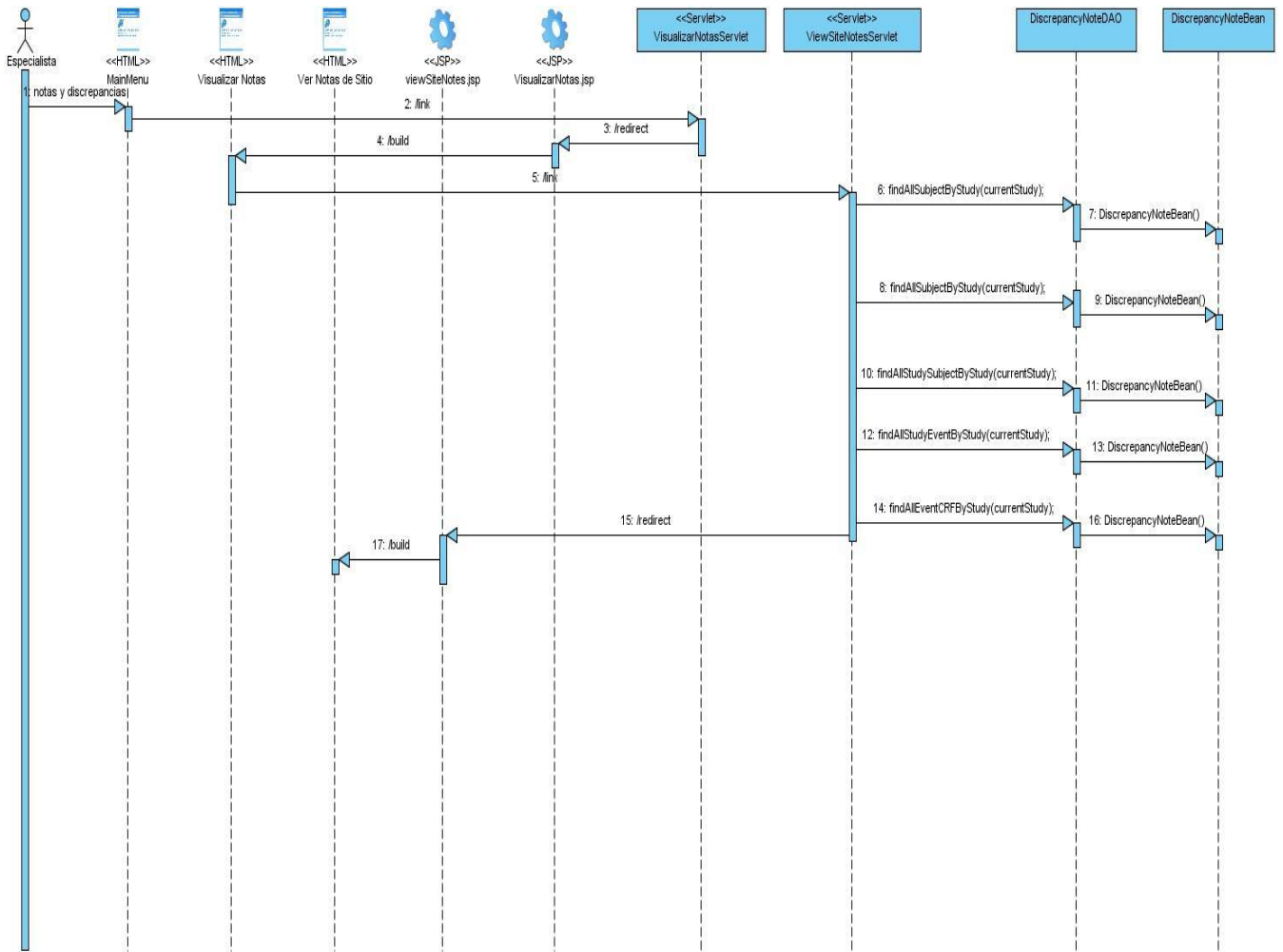


Ilustración 46 Diagrama de secuencia del CU Visualizar Notas: listar Notas de Sitio

5 Diagramas de Componentes

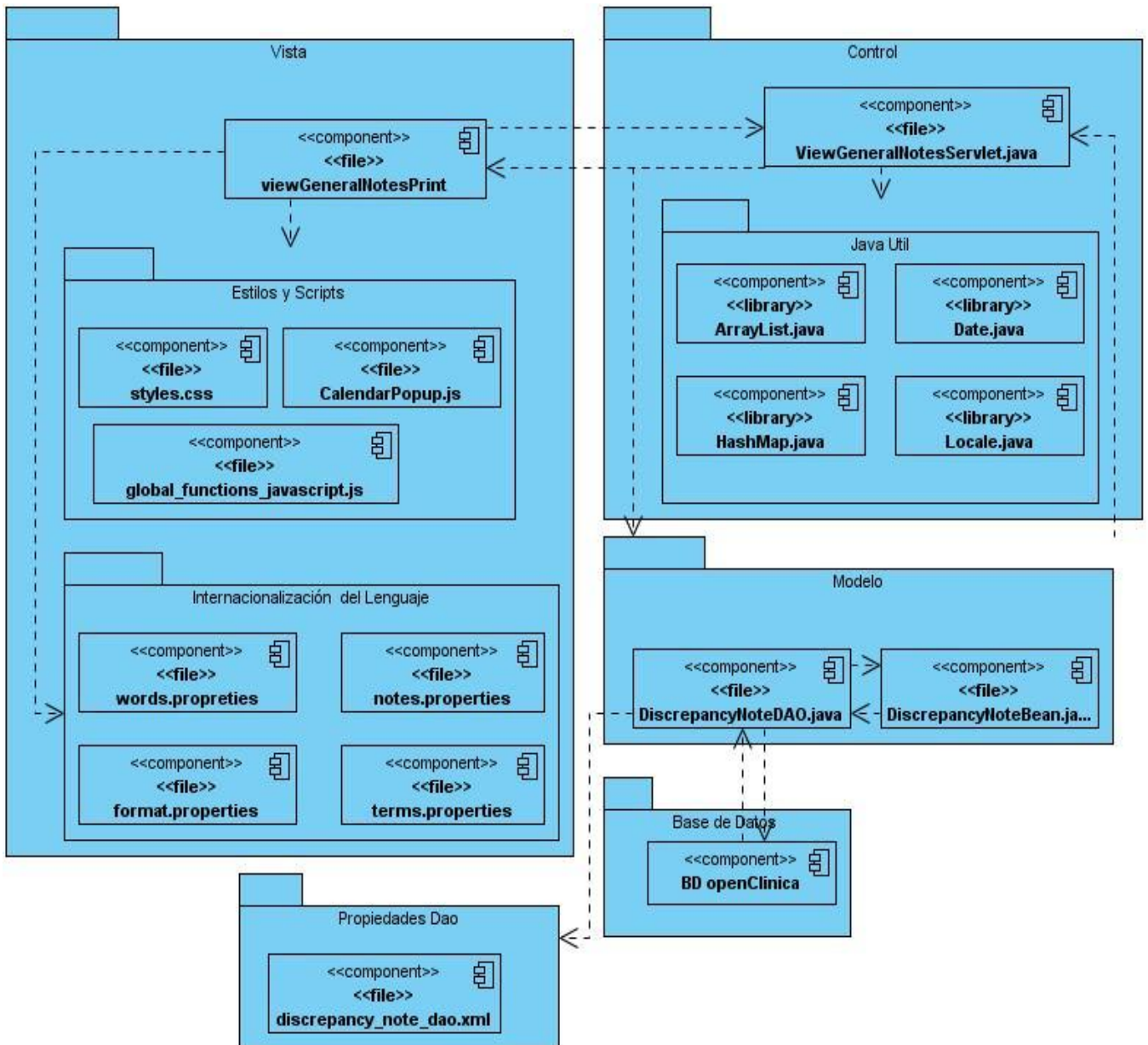


Ilustración 47 Mostrar informe de Notas Generales

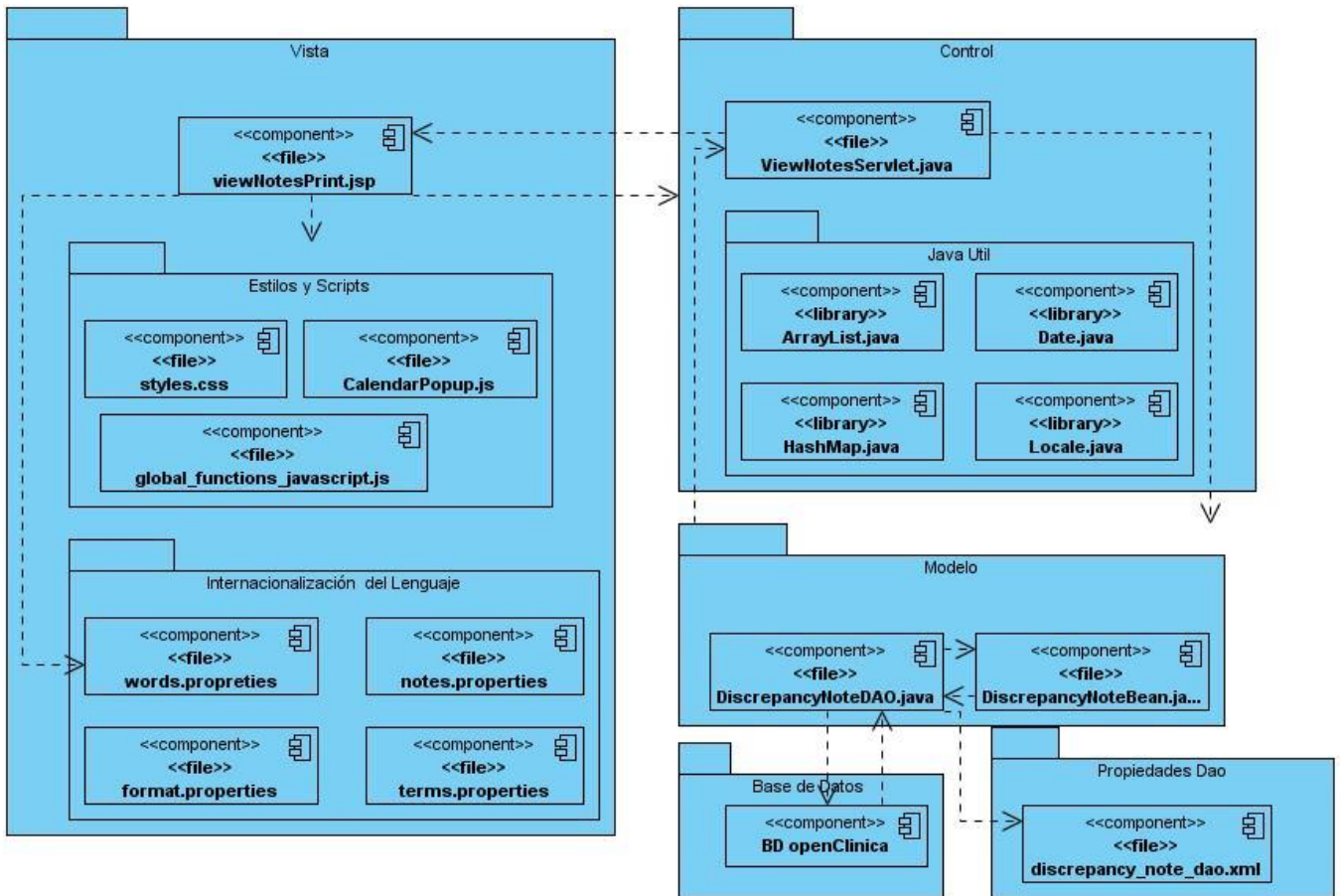


Ilustración 48 Mostrar informe de Notas de Monitoreo

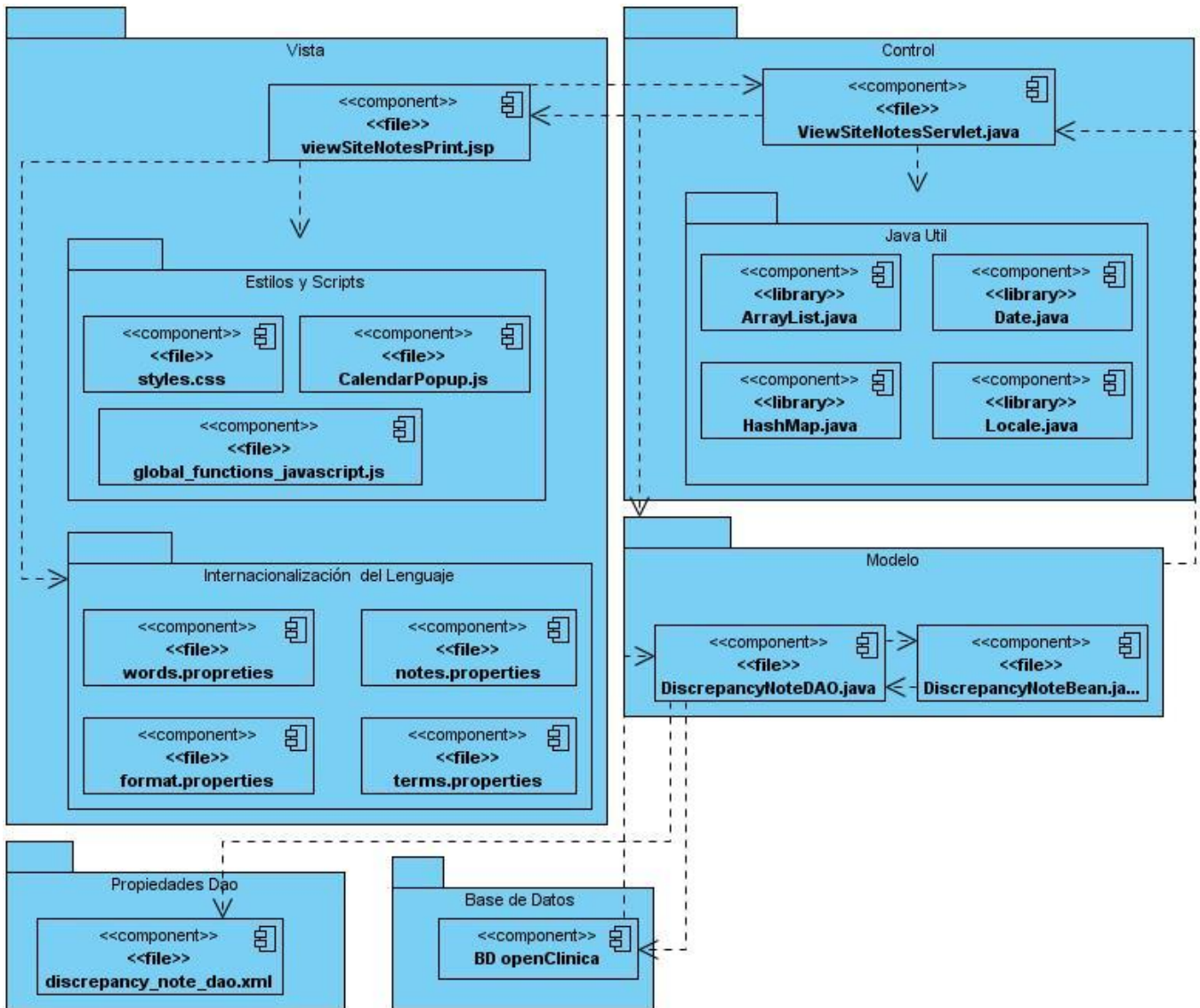


Ilustración 49 Mostrar informe de Notas de Sitio

6 Diagramas de Clase de Diseño

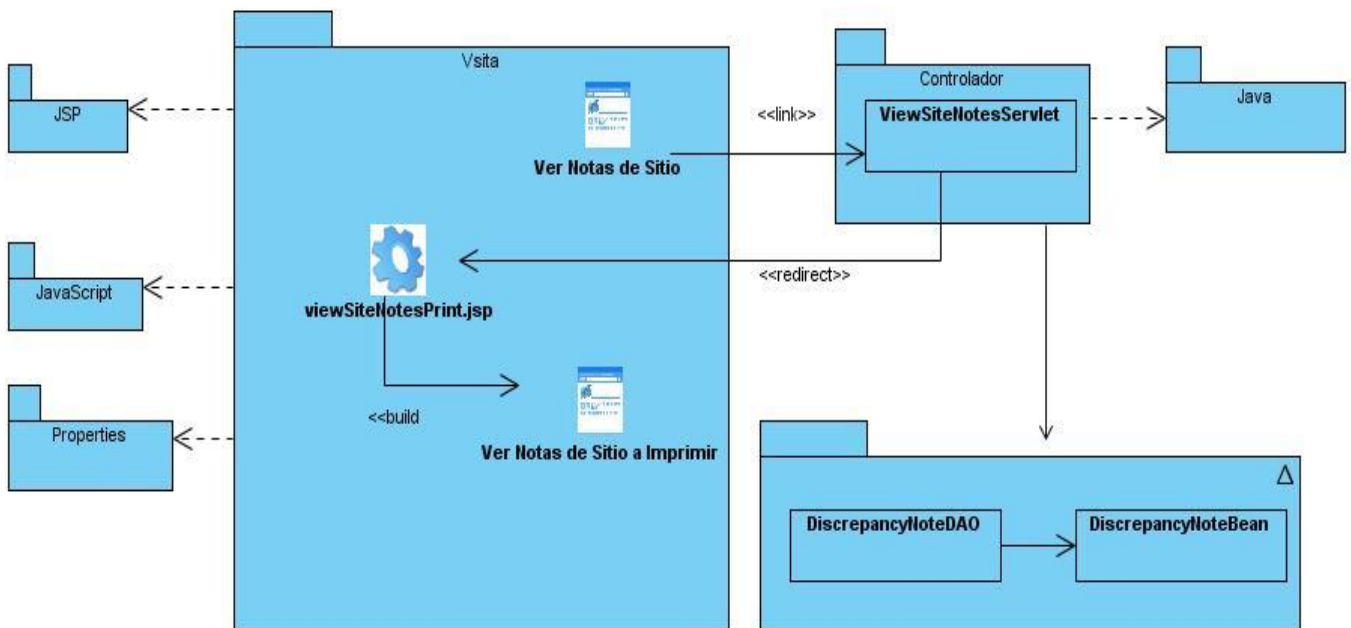


Ilustración 50 Diagrama de clase del diseño de las Notas de Sitio

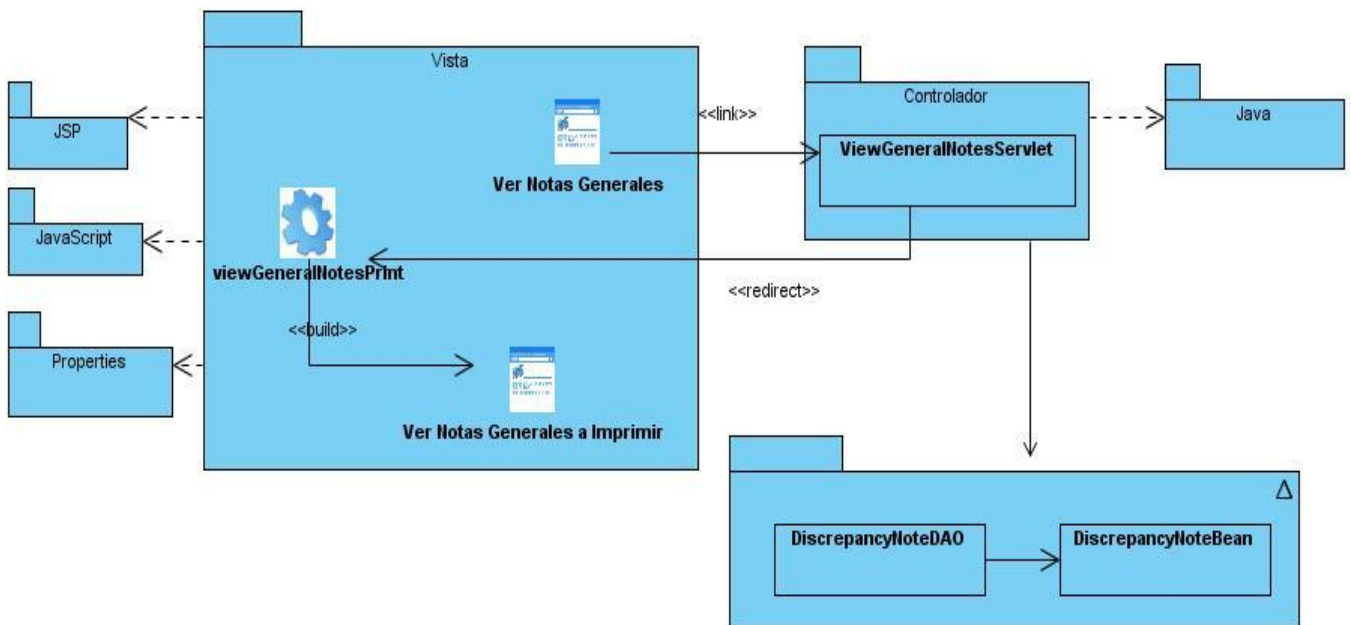


Ilustración 51 Diagrama de clase del diseño de las Notas Generales

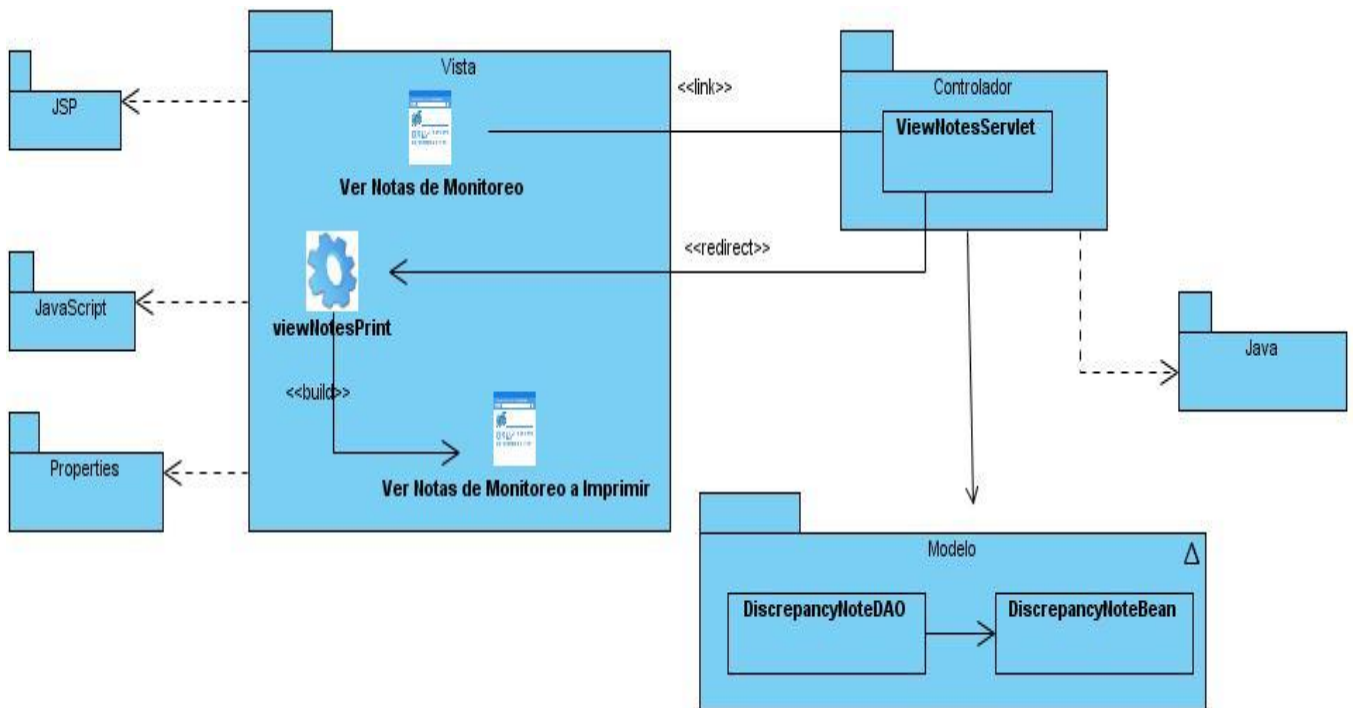


Ilustración 52 Diagrama de clase del diseño de las Notas de Monitoreo

Glosario de Términos

A

B

Bean: Componente de software reutilizable que contiene la información de un elemento en la base de datos.

C

CIC: Coordinador de Investigación Clínica. Persona implicada en la Gestión de los Datos de un Ensayo Clínico.

CIM: Centro de Inmunología Molecular.

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

CU: Caso de Uso

CUS: Caso de Uso del sistema.

D

Definición de Momento de Seguimiento: Son aquellas que se crean en el Cronograma General con un conjunto de parámetros entre los que se encuentran fecha, plazo de llenado, tipo y Hojas de CRD asociadas a la misma, no están asociadas a Paciente alguno. Su asociación a Paciente constituye un Momento de Seguimiento.

E

EC: Ensayo Clínico.

Estudio: Ensayo Clínico.

Especialista: Agrupa un conjunto de roles dentro del sistema. Tales como Investigador Principal, Coordinador y Monitor.

F

G

GDEC: Gestión de Datos de ensayos clínicos. Conjunto de procesos de un Ensayo Clínico guiado por el CIC para gestionar información asociada a Pacientes, Hojas de CRD y Momentos de Seguimiento de un estudio.

H

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

I

J

K

L

M

Momento de Seguimiento: Asociación de un paciente a una o más Hojas de CRD en una fecha planificada. Representa una visita y los datos que serán recogidos de la misma en una fecha para un paciente.

Monitor: Rol dentro del sistema.

N

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

O

P

Paciente: Sujeto asociado a un Estudio.

Parser: Analizador sintáctico y es una de las partes de un compilador.

Q

R

S

Sujeto: Persona registrada en la BD que no ha sido asignada a un Estudio. Cada Paciente es un sujeto. Los familiares de Pacientes son sujetos.

T

U

UML: Lenguaje de modelado.

V

Variable: (Referente a "Hoja de CRD") Cada uno de los datos que se recogen en una Hoja de CRD como por ejemplo: el nombre del paciente, la edad del paciente o la provincia.