

Universidad de las Ciencias Informáticas

Facultad 6



Título: “Sistema de Manejo de Datos de Ensayos Clínicos para el Centro de Inmunología Molecular. Módulo Publicador. Diseño del Sub-módulo: “Traceo y reporte de datos”.

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

Autores:

Ezequiel Martínez Justiz
Lorna Villafaña Betancourt

Tutor:

Ing. Elvira López Santos

Ciudad de la Habana

Junio 2008

“Piensa, cree, sueña y atrévete”

WALT DISNEY

DECLARACIÓN DE AUTORIA

Declaramos que somos los autores de este trabajo y autorizamos a la facultad 6 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Ezequiel Martínez Justiz

Lorna Villafaña Betancourt

Firma del autor

Firma del autor

Ing. Elvira López Santos

Firma del Tutor

DATOS DE CONTACTO

La compañera Elvira López Santos, actual tutora del trabajo de diploma: Sistema de Manejo de Datos de Ensayos Clínicos: Módulo Publicador. Diseño del sub-módulo "Traceo y reporte de datos", se graduó en la Universidad de las Ciencias Informáticas con el título de Ingeniera en Ciencias Informáticas en el año 2007, actualmente se desempeña como profesora de Práctica Profesional en la universidad donde se graduó y se desempeña como analista y jefa del módulo publicador del proyecto Ensayos Clínicos al cual pertenece.

AGRADECIMIENTOS

Queremos darte las gracias Elvira por ser nuestra tutora y amiga en todo momento. Por tus consejos precisos y tu apoyo infinito.

A la Universidad de las Ciencias Informáticas, por acogernos en su seno todos estos años que jamás podremos olvidar.

Queremos agradecerle además a los compañeros del proyecto Ensayos Clínicos que nos ayudaron en todo lo posible e imposible muchas veces. A Richard y Osvaldo por ser tan atentos y dispuestos a ayudar, a Yordel, el tiempo dedicado. A Marlien, por su eterna paciencia. A Liyanis, por su valiosísima ayuda. A todos en general por apoyarnos y estar juntos a nosotros en todo momento.

Annia y Daili: Gracias por ser las amigas que nunca pensé tener y estar a mi lado en todo momento. Por la paciencia infinita que tuvieron todos estos años con mis llantos y malcriadeces y el apoyo incondicional en cualquier circunstancia. Las quiero, por enseñarme que un amigo es familia también, por no decirme nunca que no y por su experiencia transmitida hacia mí.

Kenia: Por alegrarme la vida con sus ocurrencias y ser mi compañera de aventuras.

A las muchachas del apartamento 67203: Por ser fundamentales en momentos difíciles y ayudarme a seguir adelante. Martha, Indira, Milenys, Varna, Ana, muchas gracias.

A Yudiel: Por ser fiel compañero y escuchar con paciencia mis lamentos en cualquier situación.

A mi familia: Mariano, Baby, Felino, Yanet, Julita, Manuel, por estar siempre atentos y contentos.

A Maidel: Por el tiempo dedicado, las malas noches de sacrificio y los consejos oportunos.

A todos los que de una forma y otra nos ayudaron para que todo saliera.

A mi familia: Por siempre brindarme su apoyo incondicional, en cualquier momento de la vida.

A Maidel: Por prestarme todo su apoyo y brindarme su amor incondicional en todo momento.

A Alejandro: Por estar siempre presto a brindarme su ayuda, en cualquier circunstancia.

A María Elena: Por brindarme sus sabios consejos y ser una segunda madre para mí.

A todos aquellos que han hecho posible que yo esté aquí hoy. Que han marcado un pedazo de mi vida y que nunca olvidaré.

A mis amistades, por estar ahí cuando los necesité.

DEDICATORIA

A mi madre, Mayda, por ser la luz de mi vida. La consejera en todo momento, la persona más presente en mi vida, a la que le debe todo lo que soy y la persona que más admiro en el mundo. Por su sacrificio la vida entera, por su paciencia, abnegación y su fuerza de voluntad de levantarme del piso cada vez que caí. Por su dedicación, por cuidarme cuando estuve enferma, por darme todo el amor que una persona es capaz de dar. Te amo mamá.

A mi padre, Agustín, por mantenerme siempre firme. Por advertirme y aconsejarme, por decirme que no todo está bien y como puedo ser mejor. Por ser el profesional que es y darme su ejemplo diariamente. Te quiero papito.

A mi abuela, Adelaida, por su apoyo incondicional. Por estar siempre ahí para mí, mimarme, consentirme y escuchar pacientemente todas mis locuras. Por cuidarme en todo momento y esperarme con su café, cada vez que tenía un problema o necesité su compañía. Te quiero Abuela.

A mi mamá, Teresita, por ser la mano que me ha guiado a lo largo de la vida y me ha enseñado a persistir y luchar aunque todo parezca en mi contra.

A mi abuelo, Urbano, por siempre confiar en mí y ser una parte muy importante de mi vida.

A mi hermanita linda, Dayliunisk, por ser la alegría que revolotea a mi alrededor constantemente.

RESUMEN

La investigación surge en el marco de trabajo del proyecto Ensayos Clínicos del Centro de Inmunología Molecular (CIM), desarrollado conjuntamente con la Universidad de las Ciencias Informáticas (UCI), en la que se pretende realizar el diseño de un Sistema para el Manejo de Información de los Cuadernos de Recogida de Datos (CRD) para mejorar la recopilación y almacenamiento de la información de los CRD relacionada con la aprobación y certificación de la eficacia de los medicamentos para el tratamiento del cáncer, producidos en el CIM, lo que en la actualidad se realiza manualmente a través de complejos procesos de confección, llenado, revisión, aprobación y validación de los datos.

Como resultado de los trabajos de diploma efectuados durante el curso 2006-2007 se obtuvo el análisis y una primera versión del diseño del problema en cuestión, así como la posible solución encontrada. La parte de dicha solución reutilizable en la presente investigación está dividida en tres aspectos fundamentales: los requisitos funcionales, los requisitos no funcionales y los casos de uso del problema.

Por tanto, la investigación traducirá los requerimientos del módulo Publicador: sub-módulo "Traceo y reportes de datos" en un diseño de una aplicación web que permita recopilar y almacenar los procesos de realización de trazas y la generación de reportes de la información recogida en los CRD, con la calidad requerida, al lograr mantener la integridad de los datos para un mejor manejo de la información y la futura utilización en los estudios de análisis estadísticos, además de documentar todo el proceso de investigación y diseño.

Palabras Claves

Cuaderno de Recogida de Datos

Ensayos Clínicos

Centro de Inmunología Molecular

ÍNDICE

AGRADECIMIENTOS	I
DEDICATORIA	III
RESUMEN	IV
INTRODUCCIÓN	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	4
1.1 Ensayos Clínicos	4
¿Qué es un ensayo clínico?	4
CRD	5
1.2 Descripción del proceso actual	5
Módulo Publicador	6
En qué consiste el sub-módulo “Traceo y reportes de datos”	6
1.3 Generar reportes	7
MoReport	8
Queryx*Report	9
1.4 Realizar trazas del sistema	10
1.5 Sistemas informáticos vinculados al campo de acción	12
OpenClínica	12
Pivotal	14
Hipócrates	15
1.6 Fundamentación de las tecnologías y herramientas	16
1.6.1 Metodologías de desarrollo de software	16
1.6.2. Lenguaje Unificado de Modelado	19
1.6.3 Herramientas CASE	20
1.6.4 Herramienta Framework	21
1.6.5 Tecnologías de servidor web	23
1.6.6 Tecnologías de lenguaje de programación	24
1.6.7 Herramienta para el Control de Versiones	25
1.6.8 Herramientas para editar páginas Web	26
1.6.9 Plataforma de desarrollo	27
1.7 Conclusiones	28
CAPÍTULO 2: DISEÑO DEL SISTEMA	29
2.1 Diseño	29
2.2 Casos de uso y Requisitos Funcionales	30
Casos de Uso	30
Requisitos Funcionales	30
2.3 Patrones	31
Patrón MODELO-VISTA-CONTROLADOR (MVC)	32

Patrones GRASP	33
Patrón Decorator	34
Patrón Fachada	35
El uso de los patrones en la investigación:	35
2.4 Diagrama de paquetes de clases del Diseño	37
2.5 Diagramas de Clases del Diseño y Diagramas de Secuencia.	37
2.5.2 Descripción de las clases del Diseño	54
2.6 Prototipos no funcionales	60
2.7 Diagrama de Despliegue	71
2.8 Mapa de navegación	73
2.9 Diagrama de Clases Persistentes	74
2.10 Modelo de Datos	76
2.11 Validación del Diseño	78
Conclusiones	80
CONCLUSIONES	81
RECOMENDACIONES	82
REFERENCIA BIBLIOGRÁFICA	83
BIBLIOGRAFÍA	86
ANEXOS	90

ÍNDICE DE FIGURAS

Figura 1 Esquema representativo de la realización de trazas de la información	11
Figura 2 Diagrama de Desarrollo de RUP	17
Figura 3 Patrón Modelo-Vista-Controlador	33
Figura 4 Paquetes del Diseño	37
Figura 5 Diagrama de Clases del Diseño CU Visualizar_Queries	39
Figura 6 Diagrama de Secuencia CU Visualizar_Queries_Modelo	40
Figura 7 Diagrama de Secuencia CU Visualizar_Queries_Paciente	41
Figura 8 Diagrama de Clases del Diseño CU Visualizar_Trazas_Auditoria	42
Figura 9 Diagrama de Secuencia CU Visualizar_Trazas_Auditoria	43
Figura 10 Diagrama de Clases del Diseño CU Responder_Queries	44
Figura 11 Diagrama de Secuencia CU Responder_Queries_Modelo	45
Figura 12 Diagrama de Secuencia CU Responder_Queries_Paciente	46
Figura 13 Diagrama de Clases del Diseño CU Generar_Reporte	47
Figura 14 Diagrama de Secuencia CU Generar_Reporte_Modelo	48
Figura 15 Diagrama de Secuencia CU_Generar_Reporte_Variable	49
Figura 16 Diagrama de Clases del Diseño CU Visualizar_Inconsistencias	50
Figura 17 Diagrama de Secuencia CU Visualizar_Inconsistencias_Modelo	51
Figura 18 Diagrama de Secuencia CU Visualizar_Inconsistencias_Paciente	52
Figura 19 Capa de Modelo	53
Figura 20 Prototipo No Funcional “listar_Pacientes”	60
Figura 21 Prototipo No Funcional “mostrarListaQueriesModelo”	61
Figura 22 Prototipo No Funcional “mostrarListaQueriesPaciente”	62
Figura 23 Prototipo No Funcional “mostrarQuerieSel”	63
Figura 24 Prototipo No Funcional “selVerListaInconsistenciasModelo”	64
Figura 25 Prototipo No Funcional “selVerListaInconsistenciasPaciente”	65
Figura 26 Prototipo No Funcional “reportePor”	65
Figura 27 Prototipo No Funcional “selTipoReporteModelo”	66
Figura 28 Prototipo No Funcional “selTipoReporteVariable”	66
Figura 29 Prototipo No Funcional “asignarVariable”	67
Figura 30 Prototipo No Funcional “mostrarTablaModelo”	68
Figura 31 Prototipo No Funcional “mostrarTablaVariable”	68
Figura 32 Prototipo No Funcional “mostrarCriterioFiltrado”	69
Figura 33 Prototipo No Funcional “mostrarReporte”	69
Figura 34 Prototipo No Funcional “selTrazaAuditoria”	70
Figura 35 Prototipo No Funcional “introducirDatosBusqueda”	70
Figura 36 Prototipo No Funcional “mostrarTrazaSegunCriterios”	71
Figura 37 Modelo de Despliegue de la Aplicación	72
Figura 38 Mapa de Navegación	73
Figura 39 Diagrama de Clases Persistentes	75
Figura 40 Modelo Relacional de la Base de Datos	77
Figura 41 Diagrama de Casos de Uso del Sistema: Generalización_Actor_Visualizador	90
Figura 42 Diagrama de Casos de Uso del Sistema: Gerente de Datos	90
Figura 43 Diagrama de Casos de Uso del Sistema: Generalización_Actor_Especialista	91

ÍNDICE DE TABLAS

Tabla 1 Descripción de la Clase Pub_GenerarReporteActions.....	54
Tabla 2 Descripción de la Clase Pub_ResponderQueriesActions	55
Tabla 3 Descripción de la Clase Pub_VisualizarQueriesActions	56
Tabla 4 Descripción de la Clase Pub_VisualizarInconsistenciasActions	57
Tabla 5 Descripción de la Clase Pub_VisualizarTrazasActions	58
Tabla 6 Descripción de la Clase imprimir	59
Tabla 7 Descripción de la Clase exportar	59
Tabla 8 Descripción del CU Visualizar_Queries	91
Tabla 9 Descripción del CU Responder_Queries	92
Tabla 10 Descripción del CU Visualizar_Inconsistencias	93
Tabla 11 Descripción del CU Visualizar_Trazas	93
Tabla 12 Descripción del CU Generar_Reporte	94
Tabla 13 Trabajadores del Sistema.....	96

INTRODUCCIÓN

Introducción

El Centro de Inmunología Molecular (CIM), inaugurado en La Habana, Cuba el 5 de diciembre de 1994, es una institución biotecnológica dedicada a la investigación básica, desarrollo y fabricación de nuevos productos para el diagnóstico y tratamiento del cáncer y enfermedades relacionadas con el sistema inmune. Las líneas fundamentales de investigación de este centro son: la inmunoterapia del cáncer, especialmente en el desarrollo de vacunas moleculares terapéuticas y anticuerpos monoclonales.

El CIM ha venido desarrollando desde sus inicios, una serie de biomoléculas para el tratamiento de diferentes enfermedades mortales, principalmente el cáncer. Para ello se utilizan los Ensayos Clínicos (EC) o estudios clínicos que consisten en investigaciones diseñadas para determinar si nuevos medicamentos, vacunas u otros tratamientos contribuyen a prevenir, detectar o tratar una enfermedad.

A medida que se avanza en el desarrollo de estos productos, se progresa en las fases de los ensayos clínicos. 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.

Para la recogida de los datos de los ensayos, los científicos se auxilian de los Cuadernos de Recogida de Datos (CRD), los cuales contienen todas las variables recogidas durante el proceso. A nivel mundial se han logrado múltiples avances en este tema, pero aún, queda mucho por investigar. En respuesta a esta problemática, en Cuba se trabaja en la automatización del proceso de diseño y conducción de ensayos clínicos. El Polo Científico, en coordinación con la UCI, se ha propuesto desarrollar el Sistema de Manejo de Datos de Ensayos Clínicos que permitirá almacenar y gestionar el gran volumen de información resultante de los estudios. Actualmente, se encuentran en vías de desarrollo cinco módulos: Módulo de Administración del Sistema, Módulo de Validación, Módulo de Cronograma, Módulo de Monitoreo de los Cuadernos de Recogida de Datos y Módulo de Publicador de Cuadernos de Recogida de Datos, en el desarrollo de este último, se centra el presente Trabajo de Diploma.

Debido a que toda esta información recogida se encuentra en papeles, se corre el riesgo de que hayan muchos errores a la hora de entrar los datos, además, del complejo proceso de realizar trazas y desarrollar análisis estadísticos con los datos recogidos, por tal motivo, se desarrolló una investigación precedente, que propuso un resultado a estos problemas, y cuyo alcance fue el Diseño a partir de varios aspectos: Modelamiento del Negocio y el Levantamiento de Requerimientos Funcionales y no Funcionales. Esta primera propuesta de diseño tuvo que ser redefinida en la presente investigación,

INTRODUCCIÓN

debido a que se decidió usar una herramienta framework en la solución del problema y a la utilización de los patrones de arquitectura y diseño; que en la investigación anterior no se tuvieron en cuenta.

En el módulo Publicador de los Ensayos Clínicos, se realizará la inserción de los pacientes en el ensayo, así como, toda la entrada de información a los Cuadernos de Recogida de Datos, a partir de la Historia Clínica y los datos primarios. Este módulo por su complejidad está dividido en dos partes. Una se encarga de la gestión de la información y la otra de visualizar trazas y la realizar reportes de la información almacenada en el sistema. Este trabajo de Diploma se centra en los procesos de visualizar trazas y la realización de reportes de la información recogida en los Cuadernos de Recogida de Datos. Después de analizar los resultados de la investigación anteriormente citada, se formula el siguiente

Problema Científico:

¿Cómo traducir los requerimientos identificados para el módulo Publicador: Sub-módulo “Traceo y reporte de datos”, del Sistema de Manejo de Datos de Ensayos Clínicos (SIMDEC), en elementos que puedan ser implementados?

El **Objeto de Estudio** lo constituyen los procesos de gestión de la información de aplicaciones web para Ensayos Clínicos. El **Campo de Acción** abarca los procesos de gestión de la información de aplicaciones web para visualizar trazas y generar reportes de datos del Cuaderno de Recogida Datos de los Ensayos Clínicos.

El **Objetivo General** del trabajo de diploma es diseñar el sub-módulo “Traceo y reportes de datos” del módulo Publicador para el Sistema de Manejo de Datos de Ensayos Clínicos del CIM.

Para dar cumplimiento a dicho objetivo se diseñaron las siguientes **Tareas**:

- ✚ Revisión del levantamiento de requerimientos propuesto para la creación del módulo publicador: sub-módulo “Traceo y reportes de datos”, a partir de la investigación precedente.
- ✚ Estudio y selección de las herramientas a utilizar para el diseño del módulo publicador: sub-módulo “Traceo y reportes de datos” de los CRD del SIMDEC.
- ✚ Estudio y selección del Framework a utilizar.
- ✚ Estudio de los patrones de diseño y arquitectura a utilizar.
- ✚ Realización de los casos de uso del diseño.

INTRODUCCIÓN

- ✚ Elaboración del Diagrama de Clases Persistentes.
- ✚ Confección del Mapa de Navegación.
- ✚ Realización de los Prototipos no funcionales
- ✚ Elaboración del Diagrama de Despliegue.
- ✚ Validación del diseño.

El siguiente documento está estructurado en dos capítulos:

Capítulo 1: Fundamentación Teórica, incluye información de los Ensayos Clínicos, qué son y en qué consisten. Muestra información referente al Módulo Publicador: Sub-módulo “Traceo y reportes de datos” del Sistema de Manejo de Datos de Ensayos Clínicos, importancia y cómo se desarrollan los procesos internos del módulo. Recoge información sobre la visualización de trazas y la realización de reportes de una aplicación informática. Además incluye información sobre la visualización de inconsistencias y queries, y la forma de responder estas últimas. Se realiza un análisis de las herramientas que se utilizan para diseñar e implementar la propuesta de solución para la aplicación.

En el Capítulo 2: Se modelan artefactos correspondientes al diseño. Se presenta el modelo de diseño, donde son expuestas las realizaciones de los casos de uso definidos en el capítulo anterior, mediante diagramas de clases del diseño Web y las descripciones de las clases controladoras y sus operaciones. Se realizan los diagramas de interacción (secuencia) y el diagrama de despliegue. Se presenta una argumentación sobre los patrones de diseño y arquitectura a utilizar. Se muestra un mapa de navegación de las páginas correspondientes al sub-módulo “Traceo y reporte de datos”.

FUNDAMENTACIÓN TEÓRICA

Capítulo 1 Fundamentación Teórica

En este capítulo se brinda información referente al Módulo Publicador: Sub-módulo: “Traceo y reportes de datos” del Sistema de Manejo de Datos de Ensayos Clínicos, su importancia y como se desarrollan los procesos internos del sub-módulo, se incluye información acerca de los Ensayos Clínicos, qué es un Ensayo Clínico y en qué consisten. Proporciona información acerca del proceso de visualizar trazas y la realización de reportes de una aplicación informática, así como, de los procesos de visualizar inconsistencias y queries, y la forma de responder estas últimas. Se realiza un análisis de las herramientas que se utilizan para diseñar la propuesta de solución para la aplicación.

1.1 Ensayos Clínicos

En la actualidad se desarrollan a nivel mundial los Ensayos Clínicos como una vía de solución mediante el conocimiento científico, para contrarrestar el avance de diferentes enfermedades mortales, como por ejemplo el cáncer. La evaluación de vacunas y medicamentos se debe hacer de forma clara y concisa y que muestren los avances y riesgos en cada una de las etapas de la investigación, como resultado surgieron los Ensayos Clínicos.

¿Qué es un ensayo clínico?

Los ensayos clínicos son estudios de investigación que prueban el funcionamiento de los nuevos enfoques clínicos en las personas. Cada estudio responde preguntas científicas e intenta encontrar mejores formas de prevenir, explorar, diagnosticar o tratar una enfermedad. Los ensayos clínicos también pueden comparar un tratamiento nuevo con uno que ya se encuentra disponible.

Cada ensayo clínico tiene un protocolo o plan de acción para llevarlo a cabo. El plan describe lo que se hará en el estudio, cómo se hará y por qué cada parte del estudio es necesaria. Cada estudio tiene sus propias reglas acerca de quién puede participar. Algunos necesitan voluntarios con una determinada enfermedad. Algunos necesitan personas sanas. Otros solamente solicitan hombres o mujeres (1).

Los EC transitan por las siguientes fases, las cuales se centran en las siguientes tareas, esencialmente, en el proceso de aprobación de los medicamentos o tratamientos en estudio:

Primera fase: *Determinar la acción, seguridad y dosis en voluntarios.*

FUNDAMENTACIÓN TEÓRICA

Segunda fase: Su efectividad y seguridad en pocos casos.

Tercera fase: Seguridad y dosis en gran número de enfermos.

Cuarta fase: Seguridad y efectividad a largo plazo. (2).

CRD

Toda la información de los EC tiene que ser recogida para su posterior análisis y para esto se hace uso de los CRD que no son más que un documento o formulario impreso, óptico o automatizado, diseñado para anotar las variables recogidas durante un ensayo y transmitir al promotor toda la información requerida en el protocolo de cada sujeto del ensayo. Tanto la legislación cubana como las normativas internacionales requieren que el CRD sea incluido en la documentación sometida a aprobación por los comités éticos institucionales y las autoridades sanitarias. Este documento, siempre personalizado para cada proyecto de investigación clínica concreto, debe permitir registrar todos los datos que se solicitan en el protocolo, debe ser reflejo de la documentación clínica original, las historias clínicas (HC, informes de laboratorio y exploraciones complementarias), individual para cada paciente, debe facilitar la reproducción de los datos generados, el tratamiento estadístico, la preparación y corrección del informe final de un ensayo clínico y debe ser la fundamentación científica de un nuevo medicamento o producto sanitario. (3)

1.2 Descripción del proceso actual

En la actualidad el Centro de Inmunología Molecular (CIM) realiza este proceso de forma clásica, manualmente, a través de complejos procesos de confección, llenado, revisión, aprobación y validación de los datos (para más información acerca del tema remitirse a la investigación precedente). En este marco surge el Proyecto Ensayos Clínicos del CIM desarrollado conjuntamente con la Universidad de las Ciencias Informáticas (UCI), en el que se realiza el diseño e implementación de un Sistema para el Manejo de Información de los Cuadernos de Recogida de Datos (CRD) para mejorar la recopilación y almacenamiento de la Información de los CRD relacionada con la aprobación y certificación de la eficacia de los medicamentos para el tratamiento del cáncer, producidos en el CIM. Los resultados de los Trabajos de Diploma realizados durante el curso 2006-2007 desarrollaron el análisis e hicieron una primera propuesta de diseño, modificado en la investigación presente producto de la introducción de la herramienta framework en la solución del problema. La parte de la solución precedente reutilizable en la propuesta de solución actual está dividida en tres aspectos fundamentales: los requisitos funcionales, los requisitos no funcionales y los casos de uso del sistema

FUNDAMENTACIÓN TEÓRICA

Se definieron las herramientas a utilizar para el diseño de la aplicación informática, así como, la división en módulos:

- ✚ Módulo Publicador.
- ✚ Módulo Validación.
- ✚ Módulo Administración.
- ✚ Módulo Cronograma.
- ✚ Módulo Monitoreo.

Módulo Publicador

El Módulo Publicador del Sistema de Manejo de Datos de Ensayos Clínicos se encarga de insertar, gestionar, realizar reportes y visualizar trazas de la información de los CRD. El módulo se divide en dos partes: una que se encarga de las dos primeras funcionalidades y la segunda de las restantes. Este Trabajo de Diploma se centra específicamente en la visualización de trazas y la realización de reportes de la información.

En qué consiste el sub-módulo “Traceo y reportes de datos”

Una vez insertados los pacientes y los modelos, y gestionada la información sobre los mismos, lo cual se realiza en la primera parte del módulo, se hace necesario realizar reportes sobre la información relativa a los mismos. Existen dos formas de realizar los reportes, por variable independiente o por modelo. Para realizar los reportes se filtran un conjunto de pacientes, a los que se les quiere realizar el reporte; el objetivo del mismo es que se muestren un conjunto de datos, para exportarlos y así poder realizar los análisis estadísticos correspondientes.

Otra funcionalidad importante que debe cumplir el sistema, es permitir a los especialistas correspondientes realizar el seguimiento de todas las acciones ejecutadas durante el ensayo sobre los datos introducidos, lo cual permitirá tener un estricto control sobre las modificaciones que se le puedan realizar a los datos, lo que proporciona seguridad a la aplicación, ya que se podrá conocer con certeza quién, qué y cómo lo hizo. Otra ventaja de la realización de trazas en el sistema es que permitirá saber a qué versión es necesario volver en caso de modificación errónea de los datos introducidos.

FUNDAMENTACIÓN TEÓRICA

Otras funcionalidades importantes se pueden ver en visualizar inconsistencias, responder y visualizar queries, estas funcionalidades le dan una mayor seguridad al sistema. Las queries generadas son vistas en todo momento por la persona encargada de responderlas, así como las inconsistencias por modelo y pacientes, posibilitando un mejor manejo de la información. La visualización de estas queries es de vital importancia, ya que controla de manera eficiente el llenado de los datos dentro del ensayo, evitando errores tales como datos fuera de fecha, entre otros.

En el sub-módulo se diseñan los siguientes casos de uso, atendiendo a los requisitos funcionales y no funcionales del análisis del problema, específicamente los que están relacionados con la visualización de trazas y la realización de reportes.

- ✚ Visualizar_Queries.
- ✚ Responder_Queries.
- ✚ Visualizar_Inconsistencias.
- ✚ Visualizar_Trazas_Auditorías.
- ✚ Generar_Reportes.

La investigación precedente arrojó un total de 23 requisitos funcionales agrupados en los cinco casos de uso anteriores.

1.3 Generar reportes

(...)Los generadores de reportes son programas para crear informes sobre diseño en una amplia variedad de formatos que no son rutinariamente producidos por un sistema de información. Extraen datos de los archivos o de las bases de datos y crean reportes de acuerdo con muchos formatos, proporcionan más control, pueden manejar datos de cálculos y lógica compleja antes de darles la salida (...) (4)

A nivel internacional se utilizan diferentes variantes a la hora de solventar la necesidad de generar reportes de la información recogida en una base de datos, con el objetivo de realizar cualquier tipo de estudios con estos. Han surgido una serie de aplicaciones que incluyen dentro de sus funcionalidades generadores de reportes de la información recogida en las bases de datos de las empresas que han

FUNDAMENTACIÓN TEÓRICA

desarrollado y alojan dichas aplicaciones. Estos generadores de reportes están programados en diferentes lenguajes de programación como por ejemplo ASP, C#, PHP.

Colateralmente al desarrollo de estas aplicaciones la comunidad internacional del software libre ha ido desarrollando una serie de códigos, en PHP principalmente, que permiten realizar consultas a las bases de datos alojadas en casi cualquier sistema gestor de bases de datos y generar reportes con estas consultas. Ofrecen diferentes funcionalidades que permiten exportar dichos datos a disímiles formatos, como es el caso de XML, PDF o TXT, y posteriormente imprimirlos.







En el módulo Publicador: sub-módulo "Traceo y reportes de datos", se utilizan estos códigos PHP para realizar una serie de reportes de la información contenida en el sistema y posteriormente, imprimir estos datos y extraerlos para su utilización en otras aplicaciones, con el objetivo de realizar estudios posteriores.

MoReport

Es un generador de Reportes basado en las librerías GTK y liberado con licencia GNU/GPL. Algunas de sus características abarcan soporte para base de datos PostgreSQL, uso del estándar XML como formato de entrada, y salidas en PostScript. Actualmente se encuentra en la versión 0.1.

Las definiciones de las estructuras y componentes del informe se hacen en XML el cual es procesado por MoReport. Tanto impresiones como pre visualizaciones se hacen con un archivo PostScript. Está dividido en dos partes, una es el motor el cual podrá procesar los XML desde sus aplicaciones brindando un interfaz cómoda, que permite desarrollar las configuraciones de los informes y soporta multi-secciones. (5)

MoReport, para el diseño de un informe hace uso de las siguientes estructuras básicas:

-  Encabezado de Página.
-  Pie de Página.
-  Encabezado de Reporte.
-  Pie de Reporte.
-  Sección Detalles.
-  Admite la creación de más de un grupo.

FUNDAMENTACIÓN TEÓRICA

En el cual se pueden definir componentes como:

- ✚ Campos relacionados con una consulta SQL a una Base de Datos
- ✚ Etiquetas de Texto.
- ✚ Inserción de Líneas.
- ✚ Gráficos (no Soporta).

MoReport es un generador de reportes con soporte para PostgreSQL (por el momento, está pensado añadirle soporte para más Servidores de Base de Datos) con salida en formato PostScript (también se añadirá soporte para otros formatos de salida). El formato de entrada para MoReport se basa en el estándar XML. Por una parte sería lo más adecuado utilizar este software para realizar los reportes puesto que es compatible con la herramienta que se utiliza para gestionar la información de la base de datos, pero su mayor inconveniente a parte de no ser un producto terminado es que da las salidas en PostScript y se necesita en Pdf, doc o txt.

Queryx*Report

*El Generador de Reportes Queryx*Report es un producto desarrollado por SQL Software, utilizando herramientas Oracle™. El sistema busca satisfacer las necesidades de personas del área operativa y administrativa de diferentes organizaciones, en cuanto a obtención de Reportes y/o archivos planos que muestren la información almacenada en sus bases de datos con presentaciones tabulares. (6)*

En el momento de realizar un reporte, Queryx*Report registra el nombre que el usuario establece al reporte y con el cual quedará almacenado el correspondiente procedimiento en el sistema. Además del nombre, el usuario puede establecer un título principal, un subtítulo y comentarios que considere de utilidad. El programa, mediante listas de valores, muestra los nombres de cada una de las tablas y vistas que componen la base de datos, con el objetivo de que el usuario defina aquellas de las cuales desea alguna información.

Una vez almacenada toda la definición del reporte, el usuario puede generar el informe cada vez que lo necesite.

FUNDAMENTACIÓN TEÓRICA

Queryx*Report ofrece al usuario las siguientes funcionalidades:

- Establecer un nombre a cada columna del reporte.
- Especificar filtros para limitar la información de cada tabla y columna que se incluirá en el reporte.
- Especificar agrupamientos sobre la información que se incluirá en el reporte.
- Definir ordenamientos sobre la información que se incluirá en el reporte.
- Definir rompimientos con salto de página o de línea.

Esta herramienta ofrece una serie de facilidades que serían de mucha ayuda a la hora de realizar los reportes, pero tiene implícito, una serie de limitaciones que descartan su posible uso:

- ✚ Es un software desarrollado con herramientas privativas.
- ✚ No es compatible con el sistema gestor de base de datos propuesto en el documento de arquitectura del proyecto Ensayos Clínicos.
- ✚ Al no tener dominio sobre el código fuente, no se puede modificar y adecuar a cualquier tipo de necesidad particular de la aplicación.
- ✚ No permite las salidas necesarias para el usuario.

1.4 Realizar trazas del sistema

El acto de realizar una traza indica por donde pasó el programa cada vez que se ejecuta una instrucción. Imprime o muestra en la pantalla el valor de las variables, en una porción o en todo el programa.

A continuación se muestra un esquema que explica en qué consiste el acto de realizar trazas de la información.

FUNDAMENTACIÓN TEÓRICA

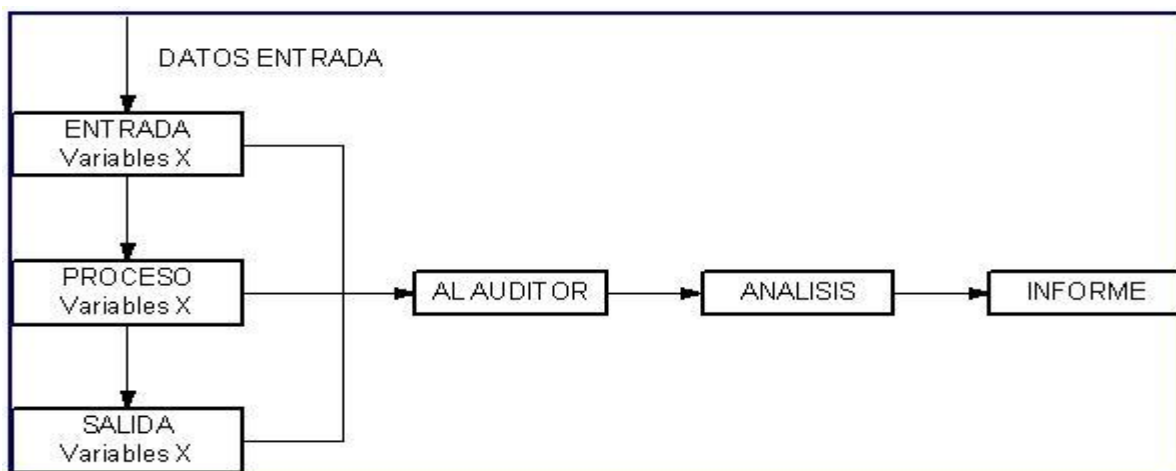


Figura 1 Esquema representativo de la realización de trazas de la información

A nivel internacional se han desarrollado disímiles variantes a la hora de solventar la necesidad de realizar trazas que permitan tener un control de las acciones realizadas por los usuarios sobre la información recogida en un determinado sistema informático. Han surgido una serie de aplicaciones que incluyen dentro de sus funcionalidades generadores de trazas que permiten hacer un seguimiento detallado de cualquier acción que se realice sobre el sistema, obteniendo, entre otros datos de seguridad, el número IP (Protocolo de internet) desde el que se realizó la acción, el usuario y el tiempo que estuvo conectado a la aplicación.

Muchas aplicaciones de bases de datos seguras requieren que se mantenga una traza de auditoría. Ésta, es un registro histórico de todos los cambios (inserciones, borrados o actualizaciones) de la base de datos, junto con la información sobre el usuario que realizó y cambió y en qué momento, lo que ayuda a la seguridad de la aplicación.

Mediante las trazas se tiene la posibilidad de medir, asegurar, demostrar, monitorear y registrar los accesos a la información almacenada en la base de datos incluyendo la capacidad de determinar:

- ✚ Quién accede a los datos.
- ✚ Cuándo se accedió a los datos.
- ✚ Desde qué tipo de dispositivo o aplicación.
- ✚ Desde qué ubicación en la red.
- ✚Cuál fue la sentencia SQL ejecutada.

FUNDAMENTACIÓN TEÓRICA



Cuál fue el efecto del acceso a la base de datos.

Lo explicado anteriormente, se realiza en la aplicación guardando los datos en una tabla específica de la base de datos con el nombre de Trazas, como consecuencia de una acción realizada. Esta información se mantiene disponible para ser vista en cualquier momento y a partir de ésta, adoptar medidas de carácter estratégico para los usuarios de la aplicación.

En el módulo Publicador: sub-módulo “Traceo y Reporte de Datos”, se utilizan un conjunto de herramientas para visualizar una serie de trazas, que van a permitir tener un control de las acciones realizadas por los usuarios sobre la información recogida en el sistema y en caso de daño intencional o accidental, de cualquier tipo de dato, proceder a la recuperación del mismo e investigar el suceso en detalle.

1.5 Sistemas informáticos vinculados al campo de acción

A nivel mundial existen una serie de aplicaciones informáticas que se encargan de la gestión de la información de los CRD de los Ensayos Clínicos, sin embargo, no implementan un modo de realización de reportes tan completo como el sistema a desarrollar requiere, ni un sistema de realización de trazas a un nivel tan elevado de detalle, entre otras desventajas. Por lo cual, se hace necesario desarrollar un sistema informático nacional que responda plenamente a las necesidades emanadas de las actividades realizadas en los centros científicos, en este caso específico el CIM.

Algunos de los sistemas internacionales, mencionados con anterioridad, son estudiados a continuación:

OpenClínica

Este producto es un software de manejo de datos clínicos (CDMS por sus siglas en inglés), desarrollado por Akaza Research y liberado bajo la licencia LGPL. Este software está pensado para administrar estudios clínicos, por lo que se apega a los estándares internacionales existentes. (7)

En esencia, es una plataforma de código abierto para la recogida de datos electrónicos usados en investigaciones clínicas y el diseño de cuadernos para la captura de datos de forma tal que pueden ser creados como se desee sin tener conocimientos de programación. Proporciona una interfaz fácil de

FUNDAMENTACIÓN TEÓRICA

usar para la presentación y validación de los datos facilitando su uso a los médicos e investigadores que participan en la inscripción de los pacientes. Permite la extracción de datos y filtrado de archivos para ser empleados por los investigadores, estadísticos, y los directores de estudio, así como la administración de cuentas de usuario, presentación de informes por los administradores, auditorías, y configuración. Posibilita el intercambio de recursos de una forma segura y transparente y el manejo de los datos de importación y exportación de herramientas para la migración de datos clínicos en las hojas de cálculo Excel y bases de datos locales. Ofrece amplias interfaces para la consulta y recuperación de datos, el uso de guías de estudios específicos de las funciones y privilegios de usuario, la encriptación SSL y el control de acceso y cambios por los usuarios. Cuenta con una robusta infraestructura de tecnología escalable y desarrollada que utiliza Java J2EE, con bases de datos relacionales PostgreSQL incluidas (de código abierto) y Oracle 10G para apoyar las necesidades de la empresa de investigación clínica.

Desventajas de OpenClínica:

- ✚ OpenClínica no cuenta con un sistema de realización de reporte tan completo como el que el sistema abordado en la investigación demanda, debido a que es necesario que sea capaz de exportar los reportes a una serie de formatos que este software no realiza.
- ✚ Aparte de que para el diseño de los CRD las variables son exportadas de un Excel, el cual es un software propietario y por lo tanto contrario a las políticas de migración al software libre.
- ✚ Según los requerimientos del negocio del cliente, se demanda la existencia de un Investigador Principal que supervise todo el proceso del llenado de los datos, los certifique y los firme, privilegio que no está concebido para el rol de Investigador Principal en OpenClínica.
- ✚ OpenClínica está desarrollado en Java, el cual es un lenguaje de programación que no está comprendido en la arquitectura definida en el proyecto.
- ✚ Además no implementa un sistema de realización de trazas a nivel de acciones realizadas en los datos sobre los módulos y los pacientes insertados, tal y como el sistema en desarrollo demanda.

FUNDAMENTACIÓN TEÓRICA

Pivotal

Pivotal es una consultoría médico-farmacéutica y de investigación clínica, especializada y organizada en las siguientes áreas de la investigación médica: oncología, cardio-vascular, endocrinología, medicina interna, anti-infecciosos, psiquiatría, neurología, reumatología y enfermedades óseas

Servicios que ofrece:

- + Diseño e implementación de bases de datos de ensayos clínicos*
- + Diseño y programación de rutinas de validación de datos con PL/SQL y Perl-SQL*
- + Diseño de CRD en papel o electrónicos*
- + Entrada de datos: doble entrada, con validación por tercera persona.*
- + Adaptación a los sistemas del cliente entrando en su intranet mediante una conexión por red privada segura y autenticada, con sistemas de emulación de terminal.*

Sistemas que usa:

- + Conjunto de rutinas en PL/SQL para validación de datos clínicos.*
- + Sistema de manejo de diccionarios y listas de codificación, usando diccionarios de cliente o estándar, ICD-9-CM, entre otros.*
- + SAS®8.2, para informes y manejo estadístico de datos clínicos*
- + Interface entre SAS y Oracle Clinical. (8)*

Desventajas de Pivotal:

- + Este sistema no cumple con lo requerido por los especialistas del CIM, ya que utiliza software privativo, lo cual lo hace inaccesible a cambios futuros en sus funcionalidades.*
- + La empresa que presta los servicios de este software nunca le da el sistema al cliente, sino que recoge los datos que se quieren procesar y en sus instalaciones los procesan, lo cual convertiría a los especialistas del CIM en dependientes de la referida empresa, con los consabidos gastos de tiempo y dinero.*
- + No implementa un sistema de trazas de las acciones realizadas sobre la información.*

FUNDAMENTACIÓN TEÓRICA

- ✚ La confidencialidad de los datos de los ensayos clínicos se perderían, al tener terceros, dominio de los mismos.

Hipócrates

Hipócrates es un sistema de archivo informático de Historias Clínicas.

Este sistema se ha creado para el control burocrático y archivamiento total de las consultas médicas, adaptado a la dinámica habitual de una consulta médica y se puede utilizar sin la necesidad de tener conocimientos informáticos.

Aporta un sistema de registro adaptado, perfectamente, al tiempo y mecánica de la consulta médica. Es capaz de registrar absolutamente toda la actividad de consulta, datos, historia, pruebas complementarias, prescripciones de los enfermos. Enriquece la historia en papel al poder incluir imágenes, sonidos y videos. Ayuda a la codificación de diagnósticos y actuaciones de una forma inmediata y natural sin ninguna sobrecarga a la actividad.

Con sus pautas, protocolos, percentiles, alarmas, normas técnicas mínimas, riesgos, calendarios vacúnales, dietas, supone un "manual operativo" complementario de la consulta. Suministra en tiempo real toda la información necesaria para la prescripción farmacológica. Además ofrece un total desarrollo de la ofimática de la consulta, impresión de documentos, recetas, volantes, etc.

Si se dispone de todos los datos archivados, se facilitan las posibilidades de emisión de informes, estudios estadísticos o revisiones a posteriori de enfermedades o procesos (9).

Desventajas de Hipócrates:

- ✚ Sin embargo, este software no se adapta a los requerimientos de los especialistas en el CIM, ya que no tiene el alcance requerido por el sistema. Solamente se encarga de algunos datos que aparecen en las historias clínicas, pero no abarca la enorme cantidad de información requerida para realizar un CRD.
- ✚ Además de que es un software privativo, y por lo tanto, no se le puede realizar ningún cambio en su estructura interna.

FUNDAMENTACIÓN TEÓRICA

- ✚ No realiza ningún tipo de gestión de trazas sobre las acciones realizadas por los usuarios en la información almacenada.
- ✚ No permite exportar los reportes a los formatos requeridos por los especialistas del CIM.

1.6 Fundamentación de las tecnologías y herramientas

A nivel mundial existen diferentes herramientas y tecnologías para desarrollar software, desde lenguajes de programación hasta herramientas que controlen el desarrollo del mismo. Las tecnologías y herramientas presentadas a continuación fueron escogidas por el grupo de arquitectura del proyecto Ensayos Clínicos en conjunto con todos los integrantes del mismo; partiendo de un análisis de la investigación precedente y teniendo en cuenta la necesidad de la nación cubana de realizar el proceso de migración a software libre. Para profundizar más en el tema se recomienda remitirse al documento de arquitectura del proyecto Ensayos Clínicos.

1.6.1 Metodologías de desarrollo de software

En el mundo se utilizan diferentes metodologías para el desarrollo de una aplicación informática. Dependiendo de la naturaleza del proyecto, se utiliza la más apropiada para satisfacer las necesidades del cliente, desde las tradicionales hasta las más actuales metodologías ágiles. Después de hacer un estudio de la investigación precedente y analizar las metodologías propuestas, se determinó utilizar para el desarrollo del sistema el Proceso Unificado de Desarrollo de Software (RUP).

Proceso Unificado de Desarrollo de Software (RUP)

(...) el Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software (...) Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto (10).

RUP está basado en componentes y utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software.

Los aspectos más significativos de RUP se pueden resumir en:

FUNDAMENTACIÓN TEÓRICA

- ✚ Dirigido por casos de uso.
- ✚ Centrado en la arquitectura.
- ✚ Iterativo e incremental.

RUP se divide en cuatro fases:

- ✚ **Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- ✚ **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen.
- ✚ **Construcción:** Se obtiene un producto listo para su utilización que se está documentado y tiene un manual de usuario.
- ✚ **Transición:** El reléase ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

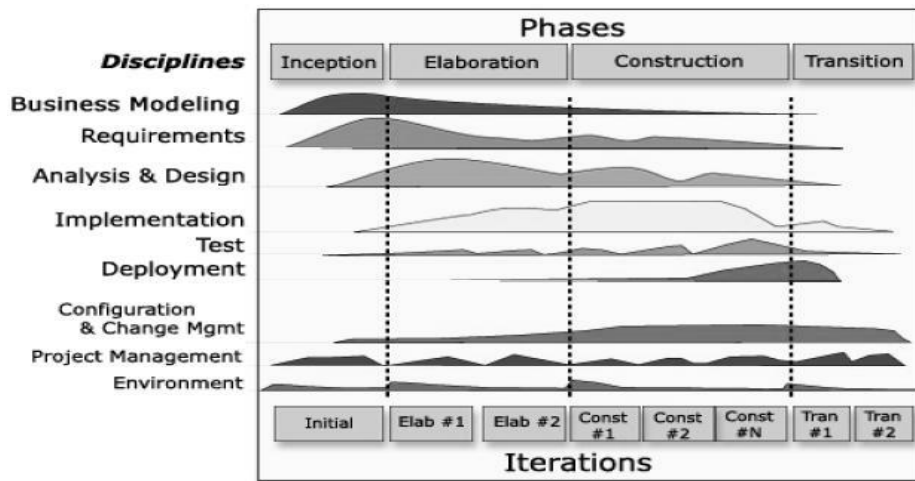


Figura 2 Diagrama de Desarrollo de RUP

Los elementos de RUP son:

- ✚ **Actividades:** Son los procesos que se llegan a determinar en cada iteración.
- ✚ **Trabajadores:** Son las personas o entes involucrados en cada proceso.

FUNDAMENTACIÓN TEÓRICA

✚ **Artefactos:** Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

RUP define nueve actividades a realizar en cada fase del proyecto:

1. Modelado del negocio.
2. Análisis de requisitos.
3. Análisis y diseño.
4. Implementación.
5. Prueba.
6. Distribución.
7. Gestión de configuración y cambios.
8. Gestión del proyecto.
9. Gestión del entorno.

Roles y artefactos

La metodología propone el desarrollo de una serie de artefactos por parte de los trabajadores que participan. En la investigación se escogen los siguientes:

Flujo de Trabajo de Análisis y Diseño:

Trabajadores del Diseño:

- ✚ Diseñador.
- ✚ Diseñador de interfaz de usuario.

Artefactos:

- ✚ Realización de los casos de uso del diseño.
- ✚ Clases de diseño.
- ✚ Subsistemas de diseño.
- ✚ Paquetes de diseño.
- ✚ Mapa de navegación.
- ✚ Prototipos de interfaz de usuario
- ✚ Vista lógica.

FUNDAMENTACIÓN TEÓRICA

✚ Diagrama de despliegue

1.6.2. Lenguaje Unificado de Modelado

Teniendo en cuenta los resultados aportados por la investigación precedente y la decisión adoptada por el grupo de arquitectura del proyecto, se determinó utilizar como lenguaje de modelado el Lenguaje Unificado de Modelado (UML). Se partió de la enorme necesidad existente de especificar, visualizar y construir los artefactos del sistema de software. Y además se tuvo en cuenta la necesidad de dar soporte a la metodología de software propuesta.

UML (versión 2.0)

UML son las siglas de Unified Modeling Language (Lenguaje Unificado de Modelado), notación (esquemática en su mayor parte) con que se construyen sistemas por medio de conceptos orientados a objetos.

El UML (Lenguaje Unificado de Modelado) se define como un “lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software (...)”. Es un sistema notacional (que entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos (11).

Es un lenguaje que se utiliza para construir modelos, no constituye una guía para el desarrollador, que le indique la forma en que debe realizar el análisis y diseño orientado a objetos ni le muestra cuál proceso de desarrollo escoger.

UML para el modelado UML posee varios diagramas:

- ✚ Diagrama de casos de uso.
- ✚ Diagrama de clases.
- ✚ Diagrama de estados.
- ✚ Diagrama de secuencias.
- ✚ Diagrama de actividades.
- ✚ Diagrama de colaboraciones.
- ✚ Diagrama de componentes.

FUNDAMENTACIÓN TEÓRICA

✚ Diagrama de distribución.

1.6.3 Herramientas CASE

A la hora de desarrollar un software se hace patente la necesidad de contar con una herramienta que permita aumentar la productividad y la calidad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo; producto de esta necesidad surgieron las herramientas CASE.

Para la presente investigación fue necesario una herramienta CASE que utilizara como lenguaje de modelado UML.

Teniendo en cuenta los resultados aportados por la investigación precedente y la decisión adoptada por el grupo de arquitectura del proyecto, se determinó modificar la herramienta CASE propuesta en la investigación precedente (Rational Rose Enterprise Edition), presentándose la utilización del Visual Paradigm en su lugar. Se tuvo en cuenta, para esta decisión, que la herramienta CASE Rational Rose es un software propietario y no es multiplataforma.

VISUAL PARADIGM (versión 3.1)








Es una herramienta CASE que utiliza como lenguaje de modelado a UML, siendo a la vez multiplataforma.

Ofrece una serie de ventajas muy útiles:








- ✚ *Entorno de creación de diagramas para UML 2.0*
- ✚ *Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.*
- ✚ *Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.*
- ✚ *Capacidades de ingeniería directa (versión profesional) e inversa.*
- ✚ *Modelo y código que permanece sincronizado en todo el ciclo de desarrollo*
- ✚ *Disponibilidad de múltiples versiones, para cada necesidad.*
- ✚ *Disponibilidad de integrarse en los principales IDEs.*
- ✚ *Disponibilidad en múltiples plataformas.*

FUNDAMENTACIÓN TEÓRICA

Los tipos de diagramas que se pueden incluir son:

-  Componentes.
-  Despliegue.
-  Secuencia.
-  Casos de Uso.
-  Clase.
-  Actividad.
-  Estado (12).

Entre sus principales características se pueden citar:

-  Licencia: Gratuita y Comercial.
-  Producto de calidad.
-  Soporta aplicaciones Web.
-  Varios idiomas.
-  Generación de código para Java y exportación como HTML.
-  Fácil de instalar y actualizar.
-  Compatibilidad entre ediciones.

Tiene disponible distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community. Facilita licencias especiales para fines académicos.

1.6.4 Herramienta Framework

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes (13).

Una de las grandes ventajas del uso de un framework es que proporciona estructura al código fuente, impulsa al implementador a crear código más legible y más fácil de mantener. Finalmente, un framework facilita la programación de aplicaciones, ya que realiza un proceso de encapsulación de las operaciones complejas en instrucciones sencillas.

FUNDAMENTACIÓN TEÓRICA

Teniendo en cuenta las enormes facilidades que ofrecen el uso de los framework, se decidió utilizar uno para desarrollar el Sistema de Manejo de Datos de Ensayos Clínicos. El grupo de arquitectura del proyecto, en conjunto con el resto de los integrantes del proyecto, decidió que el más indicado era el framework Symfony. Esta decisión se basó, entre otros factores, en que dicho framework implementa el patrón de arquitectura MVC, tiene una gran comunidad de desarrollo en constante retroalimentación, es multiplataforma, es compatible con la gran mayoría de los gestores de bases de datos y su proyecto de desarrollo está respaldado por compañías como Yahoo, lo cual le proporciona una gran estabilidad de desarrollo futuro. Para más información remitirse al documento de la arquitectura del proyecto, donde están reflejadas las comparaciones que fundamentan la decisión del uso del mismo.



SYMFONY (versión 1.0.10)

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.

Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

*Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. A continuación se muestran algunas de sus características (13).*

Características de Symfony

-  Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
-  Independiente del sistema gestor de bases de datos.

FUNDAMENTACIÓN TEÓRICA

- ✚ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✚ Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- ✚ Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- ✚ Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ✚ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Por defecto incorpora varios entornos de desarrollo diferentes e incluye varias herramientas que permiten automatizar las tareas más comunes de la ingeniería del software.

1.6.5 Tecnologías de servidor web

Un servidor web es un programa que implementa el protocolo HTTP (Protocolo de transferencia de hipertexto). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.






Teniendo en cuenta los resultados aportados por la investigación precedente y la decisión adoptada por el grupo de arquitectura del proyecto, se determinó utilizar como servidor web a Apache. Esta decisión se basó, entre otras razones, en la calidad de servicio del servidor Apache, su robustez y estabilidad. Además de que es capaz de soportar una gran cantidad de conexiones y es multiplataforma.

Servidor HTTP APACHE (versión 2.2.3)

Es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

FUNDAMENTACIÓN TEÓRICA

Ventajas de Apache:

-  *Apache es un servidor altamente configurable de diseño modular.*
-  *Apache es una tecnología gratuita de código fuente abierto.*
-  *Apache trabaja con gran cantidad de Perl, PHP y otros lenguajes de script.*
-  *Apache funciona en Linux y en otros sistemas de Unix.*
-  *Apache también funciona en Windows (14).*

Es muy sencillo incrementar las posibilidades del servidor web Apache; existen una gran cantidad de módulos implementados para realizar acciones concretas. El hecho de tener el dominio, sobre el código fuente del servidor, le da a los implementadores una ventaja enorme, ya que pueden adaptar el servidor para sus necesidades particulares.

Al ser multiplataforma, una gran cantidad de usuarios de disimiles sistemas operativos apuestan por su uso, ya que no los restringe a un sector determinado. Existen gran cantidad de documentos y materiales de ayuda en internet en todos los idiomas referido al servidor web apache. Trabaja con gran cantidad de Lenguajes de Programación interpretados como PHP, Perl, soporte con CGI (Common Gateway Interface), Java y otros lenguajes de script.

1.6.6 Tecnologías de lenguaje de programación

A nivel global se utilizan diferentes herramientas para el desarrollo de una aplicación informática. Teniendo en cuenta los resultados aportados por la investigación precedente y la decisión adoptada por el grupo de arquitectura del proyecto, se determinó utilizar como lenguaje de programación PHP 5. Esta decisión se basó, entre otras razones, en que es multiplataforma, proporciona un alto rendimiento, presenta posibilidades de interactuar con una gran cantidad de gestores de bases de datos, implementa una poderosa orientación a objetos y posee una enorme comunidad a nivel global, que se encarga de la retroalimentación y los procesos de mejora del software.

PHP 5

PHP son las siglas de PHP Hypertext Preprocessor (Preprocesado de Hypertexto), un lenguaje interpretado de alto nivel que es insertado en documentos HTML y que es ejecutado en el lado del

FUNDAMENTACIÓN TEÓRICA

servidor, a diferencia de otros lenguajes como Java Script, que son interpretados en el lado del cliente (el navegador de usuario) (15).

Entre sus características fundamentales están:

- ✚ **Gratuito.** Puede descargarse y utilizarse en cualquier aplicación de manera completamente libre.
- ✚ **Gran popularidad.** Existe una gran comunidad de desarrolladores y programadores que continuamente implementan mejoras en su código.
- ✚ **Enorme eficiencia.** Con escaso mantenimiento y un servidor, puede soportar sin problema millones de visitas diarias. Sencilla integración con múltiples bases de datos. Esencial para una página Web verdaderamente dinámica, es una correcta integración con base de datos. Puede conectarse a PostgreSQL, Oracle, dbm, filePro, interbasem, MySQL o cualquier otra base de datos compatible con ODBC (Iopen Database Connectivity Standard).
- ✚ **Versatilidad.** PHP puede usarse con la mayoría de sistemas operativos.
- ✚ **Gran número de funciones predefinidas.** Está dotado de un gran número de funciones que simplificarán enormemente tareas habituales como descargar documentos, enviar correos, trabajar con cookies y sesiones, etc.

1.6.7 Herramienta para el Control de Versiones

Una versión, revisión o edición de un producto, es el estado en que se encuentra en un momento dado en su desarrollo o modificación. Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas (por ejemplo, para algún cliente específico).

Teniendo en cuenta los resultados aportados por la investigación precedente y la decisión adoptada por el grupo de arquitectura del proyecto, se determinó utilizar como herramienta para el control de versiones a Subversion. Esta decisión se basó, entre otras razones, en que es multiplataforma, presenta una fuerte integración con el servidor web seleccionado, es capaz de manejar ficheros y directorios a través del tiempo y representa una garantía de la calidad de un trabajo, ya que en caso extremo de pérdida de la información se puede virar hacia una versión correcta.

FUNDAMENTACIÓN TEÓRICA

Subversion

Principales características:

- ✚ **Fuerte integración con Apache:** Esto permite definir controles de acceso avanzados y navegación vía web para consultar el depósito de archivos.
- ✚ **Transparencia al eliminar y cambiar nombres de archivos:** Si ha intentado este último proceso en CVS, seguramente sabrá que requiere intervención manual en el depósito para lograrlo, Subversion contempla esta deficiencia y la corrige con éxito.
- ✚ **Copias ligeras sobre ramificaciones:** La generación de ramificaciones en CVS además de ser un proceso involucrado implica la generación de una copia nueva en el depósito, mismo mecanismo que hace crecer exponencialmente el tamaño del depósito, Subversion independientemente del número de ramificaciones creadas mantiene un árbol diferencial de cambios, minimizando así el espacio consumido en el depósito.
- ✚ **Copias diferenciales de archivos binarios:** Basado en el mismo principio de copias ligeras, Subversion es capaz de mantener un control diferencial sobre cualquier archivo binario del depósito así reduciendo el consumo de espacio, esto contrastado con CVS que requiere archivar copias completas de un archivo binario cada vez que éste cambia (16).

1.6.8 Herramientas para editar páginas Web.

El editor de páginas Web permite la confección de la aplicación.

Teniendo en cuenta los resultados aportados por la investigación precedente y la decisión adoptada por el grupo de arquitectura del proyecto, se determinó modificar la herramienta propuesta en la investigación precedente (Macromedia DreamWeaver) como herramienta para editar los prototipos de interfaz, presentándose la utilización del Kompozer en su lugar. Se tuvo en cuenta, para esta decisión, que la herramienta propuesta en la investigación precedente era un software propietario y no es multiplataforma. Mientras que la nueva propuesta de herramienta de edición es un software libre, con gran facilidad de uso y presenta cierta similitud con la propuesta de la investigación precedente.

Kompozer

FUNDAMENTACIÓN TEÓRICA

Kompozer es una herramienta de creación de páginas Web en la que no se requiere ningún conocimiento de HTML por parte de los usuarios. Es un software gratuito y tiene muchas características parecidas al Dreamweaver (17).

Consta de cuatro pestañas: Normal (editables WYSIWYG), Etiquetas HTML, Source (código fuente) y Preview (vistas).

Puede administrar los sitios y publicar directamente desde el editor. Posee un editor de CSS, consola Javascript, iconos para construir las etiquetas HTML. Es bastante estable y fácil de usar. Permite editar archivos de gran tamaño. Incluye funcionalidades como: manejador de FTP, una nueva paleta de colores, entre otras

1.6.9 Plataforma de desarrollo

En la investigación precedente, se decidió usar como plataforma de desarrollo para la construcción de la aplicación, el software Zend Studio; debido a la necesidad de utilizar un framework para el desarrollo del producto, se realizó un estudio más exhaustivo de la propuesta anterior y se llegó a la conclusión de utilizar como plataforma de desarrollo Eclipse, producto de las múltiples ventajas que ofrece, en vez de Zend Studio. Se tuvo en cuenta, para esta decisión, que la nueva propuesta es OpenSource, multiplataforma, posee de una comunidad de millones de usuarios, posee una elevada cantidad de plugins, presenta soporte multilenguaje en una única herramienta, posee la característica de poder realizar la compilación en tiempo real. Mientras que la propuesta anterior es un software privativo, no es multiplataforma y no se tiene acceso a su código fuente.

Eclipse (versión 3.3.1 con plugins 1.0.1)

Eclipse es una plataforma de desarrollo de código abierto. Es usado para múltiples aplicaciones como por ejemplo de escritorio, aplicaciones basadas en framework así como para la construcción de entornos de desarrollo (IDEs) que puedan ser utilizados para la construcción de aplicaciones web, aplicaciones Java de todo tipo y programas C++, además ofrece un autocompletado de código para el desarrollo con el framework Symfony. Es un IDE de desarrollo multiplataforma ejecutándose en varios sistemas operativos como Windows y Linux (18).

FUNDAMENTACIÓN TEÓRICA

1.7 Conclusiones

En el presente capítulo se incluyó información acerca de los Ensayos Clínicos, los CRD, que son y en qué consisten. Se muestra una breve fundamentación del flujo actual de los procesos en el CIM, así como la propuesta de automatización del proceso. Se brindó información referente al Módulo Publicador: sub-módulo “Traceo y reportes de datos”, del Sistema de Manejo de Datos de Ensayos Clínicos, qué es, en qué consiste y cómo funciona. Así como también se dio un breve bosquejo sobre algunas aplicaciones informáticas a nivel mundial relacionadas con el campo de acción de la investigación.

Por otra parte, se expone de manera breve la fundamentación de cada una de las herramientas propuestas en el análisis previo: la herramienta CASE, el lenguaje de modelado y el lenguaje de programación; incorporándose el análisis de otras herramientas: servidor web y Framework.

Producto de esto se propone desarrollar el diseño de la aplicación, utilizando RUP como metodología para el proceso de desarrollo de software, UML como lenguaje de modelado, como herramienta CASE Visual Paradigm, PHP 5 como lenguaje de programación, como servidor web Apache, Symfony como framework, Kompozer como herramienta para la edición de prototipos de interfaz, Eclipse como IDE de desarrollo y como herramienta para el control de versiones Subversion.

Teniendo en cuenta estos conceptos y conociendo las herramientas a utilizar en el diseño de las funcionalidades del Módulo Publicador: sub-módulo “Traceo y reportes de datos” del Sistema de Manejo de Datos de Ensayos Clínicos, se da continuidad al desarrollo de la propuesta.

DISEÑO DEL SISTEMA

Capítulo 2: Diseño del Sistema

En el capítulo que se presenta a continuación se modelan artefactos correspondientes al diseño. Se presenta el modelo de diseño, donde son expuestas las realizaciones de los casos de uso definidos en el capítulo anterior, mediante diagramas de clases del diseño Web. Se realizan los diagramas de interacción (secuencia) y el diagrama de despliegue. Se presenta el diagrama de clases persistentes, así como el modelo relacional de la base de datos. Se analiza el diseño realizado, a partir de la validación del diseño, mediante una entrevista. Se muestran además los prototipos no funcionales de la aplicación, para el módulo en cuestión, así como el mapa de navegación de dichas páginas.

2.1 Diseño

El diseño del software es un proceso iterativo mediante el cual los requisitos se traducen en un plano para construir el software.

El diseño de software es tanto un proceso como un modelo. El proceso de diseño es una secuencia de pasos que hacen posible que el diseñador describa todos los aspectos del software que se va construir. Sin embargo, es importante destacar que el proceso de diseño simplemente no es un recetario. Un conocimiento creativo, experiencia en el tema, un sentido de lo que hace que un software sea bueno, y un compromiso general con la calidad son factores críticos de éxito para un diseño competente.

(...) El diseño es la única forma de convertir exactamente los requisitos de un cliente en un producto o sistema de software finalizado. El diseño del software sirve como fundamento para todos los pasos siguientes del soporte del software y de la ingeniería del software. Sin un diseño, corremos el riesgo de construir un sistema inestable -un sistema que fallará cuando se lleven a cabo cambios; un sistema que puede resultar difícil de comprobar; y un sistema cuya calidad no puede evaluarse hasta muy avanzado el proceso, sin tiempo suficiente y con mucho dinero gastado en él (19).

El diseño del presente trabajo de diploma se realizará a partir de los artefactos arrojados por la investigación precedente referida en el capítulo anterior. De ellos, se cuenta con 23 requisitos funcionales y cinco casos de uso del sistema, los cuales se muestran a continuación:

2.2 Casos de uso y Requisitos Funcionales

Casos de Uso

Se encargan fundamentalmente de realizar trazas de la información recogida en los CRD, así como realizar reportes de dicha información que después serán impresos y archivados para documentar cada uno de los ensayos clínicos por paciente.

- ✚ **Visualizar_Queries:** El caso de uso comienza cuando el Coordinador de la investigación clínica, Investigador promotor o Investigador principal deciden ver las queries del ensayo, de un paciente o de un modelo en particular.
- ✚ **Responder_Queries:** El caso de uso comienza cuando el Coordinador de la investigación clínica o Investigador Principal decide responder query de un modelo, paciente o del ensayo. El caso de uso finaliza cuando el coordinador o investigador principal responde una query.
- ✚ **Visualizar_Inconsistencias:** El caso de uso comienza cuando el Coordinador de la investigación clínica, Investigador promotor o Investigador principal deciden ver las inconsistencias de un modelo o paciente.
- ✚ **Visualizar_Trazas_Auditorias:** El caso de uso comienza cuando el Coordinador de la investigación clínica, Investigador Promotor o Investigador principal, deciden ver las trazas de auditoría de un ensayo, por determinado criterio de búsqueda.
- ✚ **Generar_Reportes:** El caso de uso comienza cuando el Gerente de datos decide realizar algún reporte para conocer el comportamiento de diferentes variables en determinados pacientes del ensayo, el mismo se puede realizar de dos formas por modelos o por variables independientes. Obteniéndose una tabla con el reporte generado. Que se puede imprimir o exportar a diferentes formatos.

Requisitos Funcionales

Para llevar a cabo estas funcionalidades se levantaron una serie de requisitos funcionales. Su implementación permitirá no solo trabajar con la base de datos, realizando consultas a distintas tablas a la vez sino que permitirá contar con la información archivada no solo en formato digital sino también en papel. Logrando una mejor organización de la información.

Los requisitos funcionales del análisis previo son:

DISEÑO DEL SISTEMA

- R10) Mostrar queries por paciente.
- R11) Mostrar queries por modelos.
- R12) Mostrar queries por ensayo.
- R13) Actualizar Query.
- R14) Mostrar inconsistencia por paciente.
- R15) Mostrar inconsistencia por modelos.
- R19) Buscar trazas de auditoría.
- R20) Mostrar trazas de auditoría.
- R23) Asignar variables.
- R24) Generar reporte.
- R25) Mostrar reporte generado.
- R26) Filtrar pacientes.
- R27) Imprimir listado de pacientes.
- R28) Imprimir listado de visitas de un paciente.
- R29) Imprimir datos de un modelo.
- R30) Imprimir datos de un paciente.
- R31) Imprimir cronograma de ejecución del ensayo general.
- R32) Imprimir cronograma de ejecución del ensayo para un paciente específico.
- R33) Imprimir listados de inconsistencias
- R34) Imprimir query.
- R35) Imprimir reporte generado.
- R36) Imprimir trazas de auditorías.
- R37) Exportar reporte.

2.3 Patrones

El patrón es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas (...)
Muchos patrones ofrecen orientación sobre cómo asignar las responsabilidades a los objetos ante determinada categoría de problemas (11).

DISEÑO DEL SISTEMA

El objetivo básico que se persigue con la idea de patrón es no tener que reinventar la rueda cada vez que se tenga necesidad de utilizarla. Los patrones capturan soluciones a situaciones de especial interés, bien por ser muy comunes, bien por ser especialmente complejas.

La llegada de los patrones al mundo de la construcción de sistemas software ha sido una aportación fundamental, llevando a esta disciplina a abandonar su tradicional status de artesanía y acercándola al nivel de sistematización y robustez del que gozan otras ingenierías. (20)

Symfony propone como patrón de arquitectura a usar el Modelo-Vista-Controlador y como patrones de diseño el Decorator y los GRASP (Patrones de Asignación de Responsabilidades). Dichos patrones se explican a continuación.

Patrón MODELO-VISTA-CONTROLADOR (MVC)

(...)La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.”

“(...) MVC, está formado por tres niveles:

- ✚ El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.*
- ✚ La vista transforma el modelo en una página web que permite al usuario interactuar con ella.*
- ✚ El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista (13).*

En la investigación se hará uso del MVC debido a las ventajas que se derivan de su uso y además por ser el patrón de arquitectura que propone el framework Symfony.

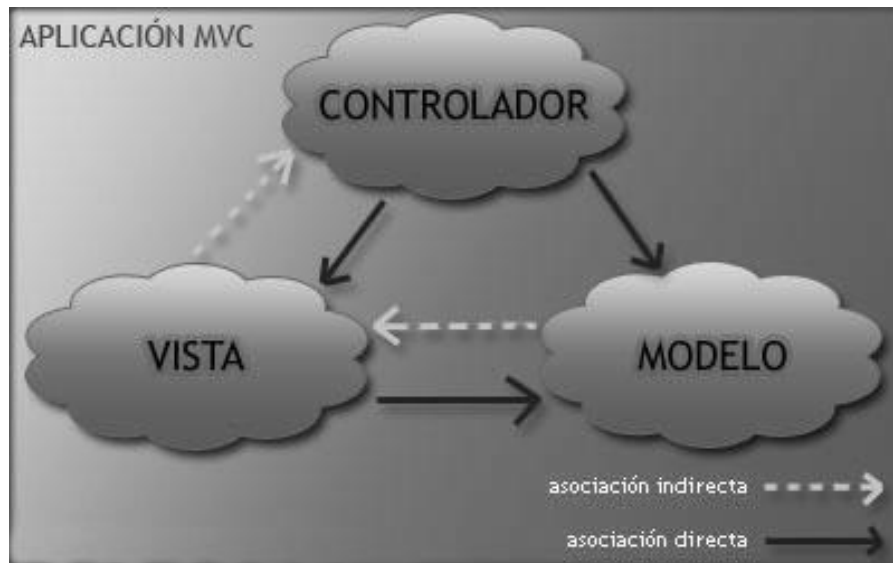


Figura 3 Patrón Modelo-Vista-Controlador

Patrones GRASP

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades).

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (11).

Existen 5 patrones GRASP principales, los cuales son:

Experto: *Asignar una responsabilidad al experto en información; la clase que posee la información necesaria para cumplir con la responsabilidad.*

Creador: *Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento.*

Bajo acoplamiento: *El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con qué las conoce y con qué recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras (...)*

DISEÑO DEL SISTEMA

Alta Cohesión: La cohesión (o, más exactamente, 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.

Controlador: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones:

- ✚ el "sistema" global (controlador de fachada).
- ✚ la empresa u organización global (controlador de fachada).
- ✚ algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas).
- ✚ un manejador artificial de todos los eventos del sistema de un caso de uso, generalmente denominados "**Manejador<NombreCasodeUso>**" (o controlador de casos de uso).

Utilice la misma clase de controlador con todos los eventos del sistema en el mismo caso de uso (11).

Y cuatro patrones GRASP adicionales que son:

- ✚ Fabricación Pura.
- ✚ Polimorfismo.
- ✚ Indirección.
- ✚ No hables con extraños.

Patrón Decorator

El patrón Decorator responde a la necesidad de añadir dinámicamente funcionalidad a un objeto.

A veces se desea adicionar responsabilidades a un objeto pero no a toda la clase. Las responsabilidades se pueden adicionar por medio de los mecanismos de Herencia, pero este mecanismo no es flexible porque la responsabilidad es adicionada estáticamente. La solución flexible es la de rodear el objeto con otro objeto que es el que adiciona la nueva responsabilidad. Este nuevo objeto es el Decorator (21).

El patrón Decorator se debe usar para:

- ✚ Adicionar responsabilidades a objetos individuales dinámicamente sin afectar otros objetos.
- ✚ Para agregar responsabilidades que pueden ser retiradas

DISEÑO DEL SISTEMA

- ✚ Cuando no es práctico adicionar responsabilidades por medio de la herencia (...)

Patrón Fachada

El patrón fachada es usado para definir una sola clase que unifique las interfaces y asignarle la responsabilidad de colaborar con el subsistema.

El uso de los patrones en la investigación:

Modelo-Vista-Controlador

Symfony como herramienta Framework realiza una implementación del patrón MVC. La vista está compuesta por tres capas más: Vista, Plantilla y Layout; el modelo está compuesto por dos capas: Abstracción de los datos y Acceso a los datos y el controlador está compuesto por dos capas también: Controlador Frontal y Acción.

- ✚ **Vista:** Contiene toda la lógica de la vista, el código necesario para enlazar la plantilla y el layout.
- ✚ **Layout:** Denominado también plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación. Su función es integrarse con la plantilla y decorarla.
- ✚ **Plantillas:** Contiene el código típico de una plantilla. Está compuesta por código HTML y Código PHP para acceder a las variables en la acción de cada página.
- ✚ **Abstracción de los datos:** Se encarga fundamentalmente de adecuar el lenguaje del acceso a los datos a cualquier Sistema Gestor de Base de Datos.
- ✚ **Acceso a los datos:** Se encarga de acceder a los datos específicos almacenados en cada una de las tablas de una forma específica definida por la herramienta Symfony.
- ✚ **Controlador frontal:** Punto de entrada único para toda la aplicación, se encarga de conectar las peticiones de la vista con el modelo.
- ✚ **Acción:** Define el código específico del controlador de cada página.

-Decorator

Symfony implementa el patrón Decorator ya que este añade funcionalidad a una clase, dinámicamente.

DISEÑO DEL SISTEMA

En el diagrama de clases se tiene un layout, el cual es el encargado de almacenar todo el código HTML común a todas las páginas, mientras que la plantilla es la encargada de almacenar el código referente a una sola en específico. Por lo que en cada página mostrada la plantilla se encarga de decorar el código HTML común a todas las páginas, añadiéndole código dinámico, específicamente para la página mostrada en ese momento.

-GRASP

Symfony implementa los patrones GRASP, ejemplo de ello es el patrón bajo acoplamiento ejemplo de ello es el patrón bajo acoplamiento debido a que las clases utilizadas presentan poca dependencia entre ellas y no presentan herencias profundas. Se utilizó el patrón controlador con el propósito de centralizar las actividades y mantener una alta cohesión ya que mantiene una labor única dentro del sub-módulo. Se utiliza además el patrón fabricación pura, debido a la presencia de una clase artificial la cual da soporte a la reutilización; así como la existencia de una clase intermedia que permitirá la comunicación entre los módulos.

-Fachada

Es usado el patrón Fachada proporcionando una clase que unifica la interfaz del sub-módulo y tiene la responsabilidad de interactuar con el mismo.

2.4 Diagrama de paquetes de clases del Diseño

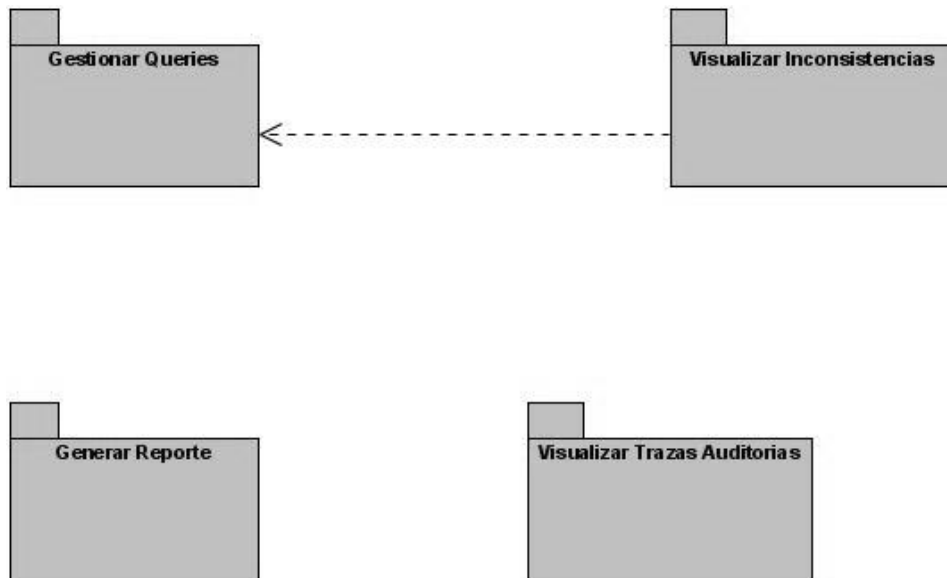


Figura 4 Paquetes del Diseño

2.5 Diagramas de Clases del Diseño y Diagramas de Secuencia.

Un diagrama de clases describe gráficamente las especificaciones de las clases de software y de las interfaces de la aplicación. La modelación se optimiza en este caso representando las páginas, las relaciones entre ellas y las clases que utilizan en el sistema dividido en paquetes y estos a su vez por escenarios.

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de las veces, esto implica modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. Los diagramas de interacción pueden utilizarse para visualizar, especificar, construir y documentar la dinámica de una sociedad particular de objetos, o se pueden utilizar para modelar un flujo de control particular de un caso de uso.

La herramienta Framework Symfony propone dos tipos de clase dentro del modelo de datos: las clases bases y las clases personalizadas. Las clase bases son las que se generan directamente a partir del esquema de la base de datos. No se deben modificar, pues cada vez que se genera el modelo se borran todas las clases. Las clases personalizadas o también llamadas clases objetos heredan de las

DISEÑO DEL SISTEMA

clases bases y se almacenan en otro archivo del Symfony. Se generan dos de cada tipo, por lo que la estructura quedaría de la siguiente manera: ClaseObjeto, ClaseObjetoPeer, BaseClaseObjeto y BaseClaseObjetoPeer.

Las ClasesObjeto y BaseClaseObjeto representan un registro de la base de datos, permiten acceder a las columnas de un registro y a los registros relacionados. Mientras que las clases peer poseen métodos estáticos para trabajar con las tablas de la base de datos, que devuelven normalmente un objeto o una colección de objetos de la clase objeto relacionada.

Cuando se construye el modelo, crea una clase objeto base para cada una de las tablas definidas en la base de datos. Cada una de estas clases poseen constructores, y métodos de acceso por defecto, teniendo en cuenta la definición de cada columna: get() y set() (para obtener las propiedades de los objetos).

La ClaseObjetoBasePeer contiene los métodos retrieveByPk() y doselect(), el primero devuelve el objeto relacionada con la clase, pasándole por parámetro el id que define la clase, la clave primaria en la base de datos, mientras que el segundo también obtiene datos pero teniendo en cuenta un objeto de la clase Criteria, utilizada para definir consultas simples sin utilizar las sentencias propias del PostgreSQL, consiguiendo así la abstracción de la base de datos.

DISEÑO DEL SISTEMA

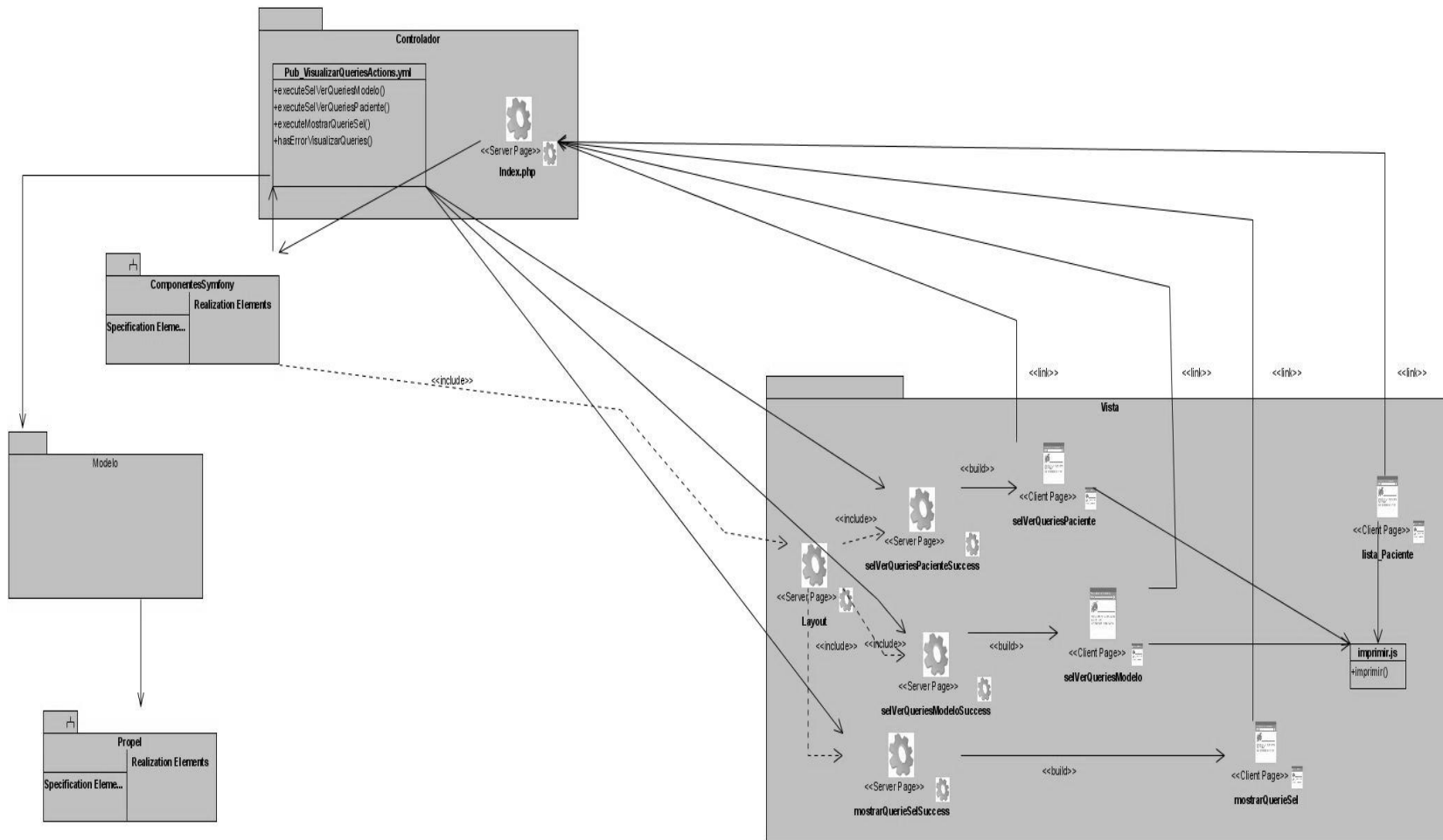


Figura 5 Diagrama de Clases del Diseño CU Visualizar_Queries

DISEÑO DEL SISTEMA

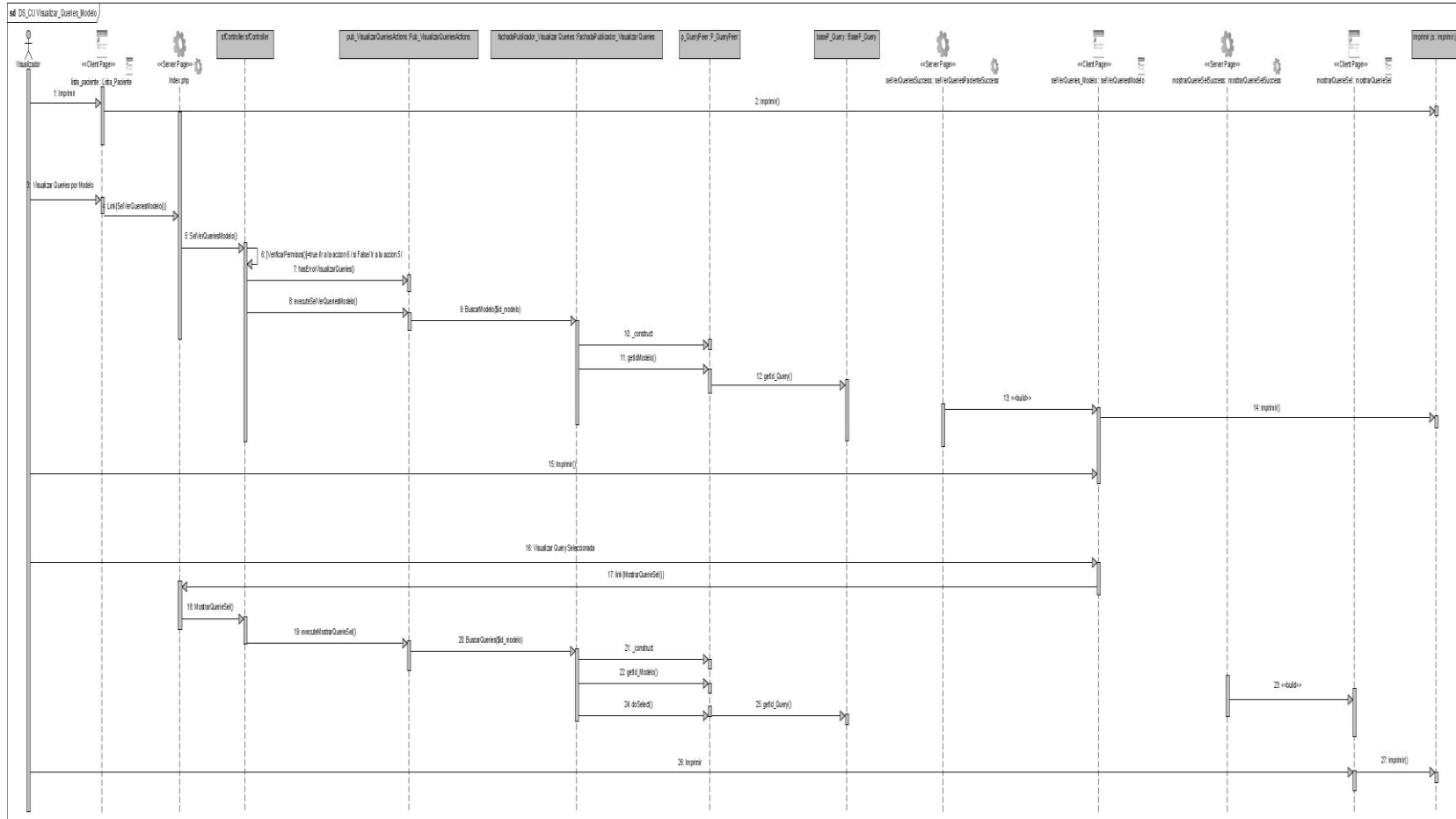


Figura 6 Diagrama de Secuencia CU Visualizar_Queries_Modelo

DISEÑO DEL SISTEMA

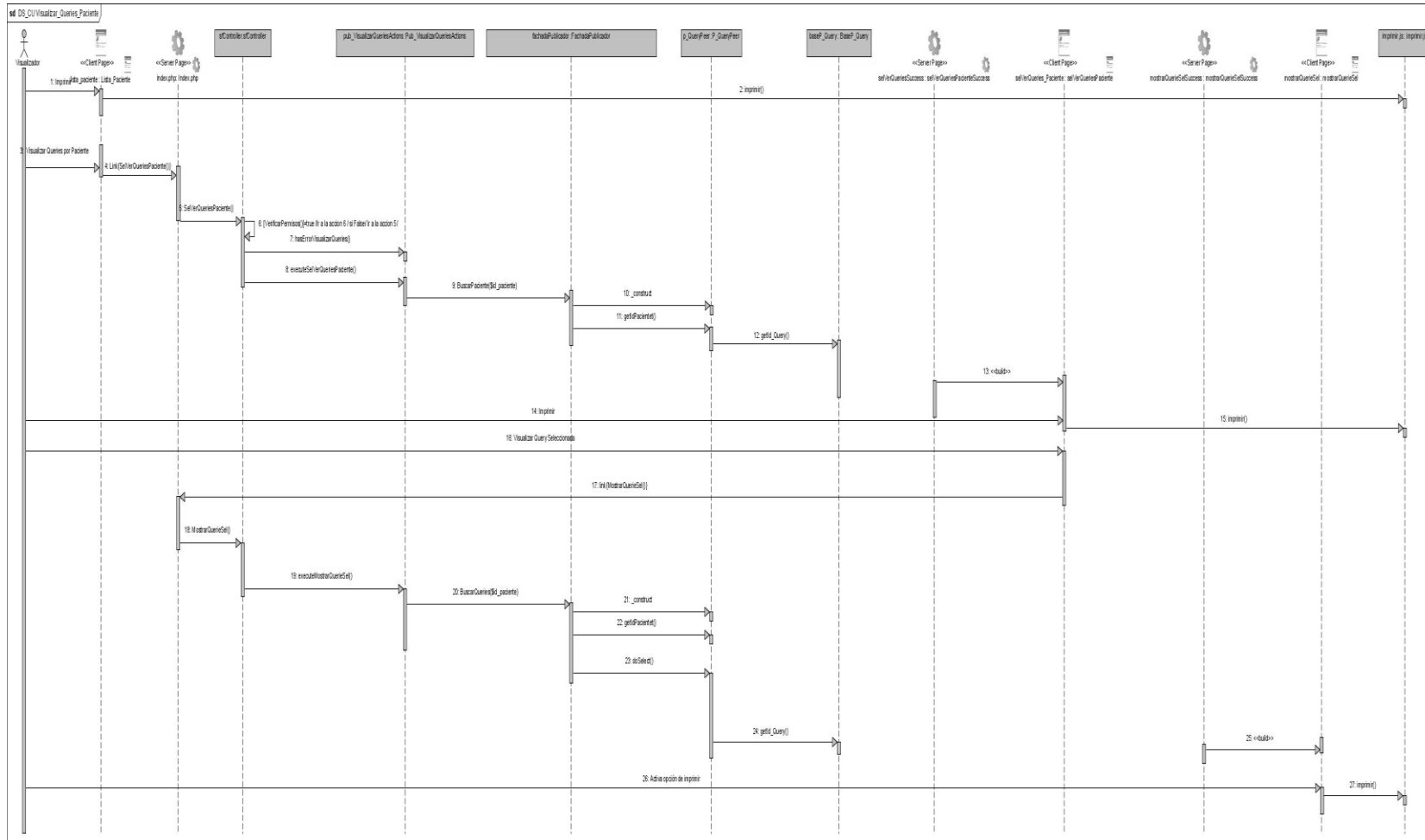


Figura 7 Diagrama de Secuencia CU Visualizar_Queries_Paciente

DISEÑO DEL SISTEMA

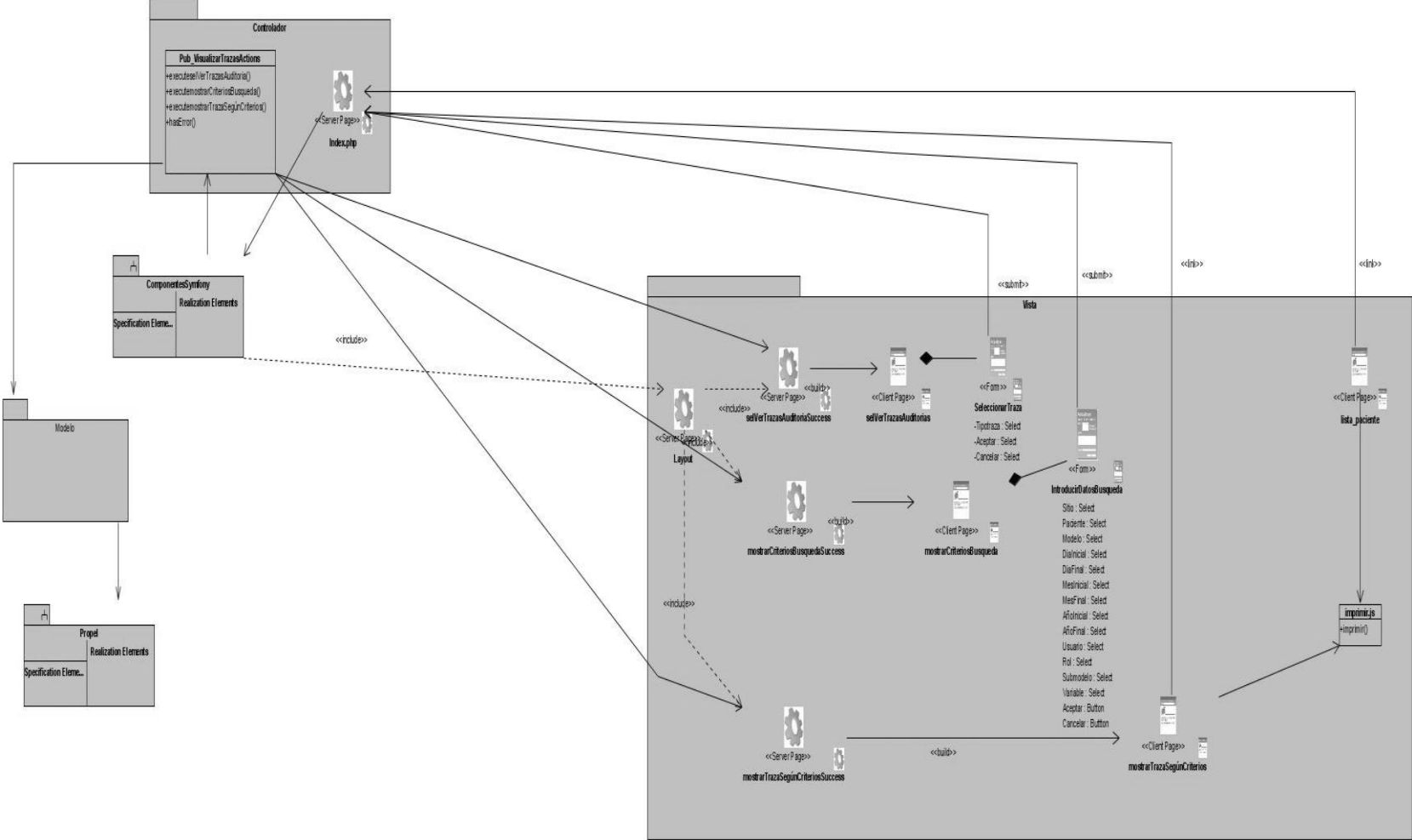


Figura 8 Diagrama de Clases del Diseño CU Visualizar_Trazas_Auditoria

DISEÑO DEL SISTEMA

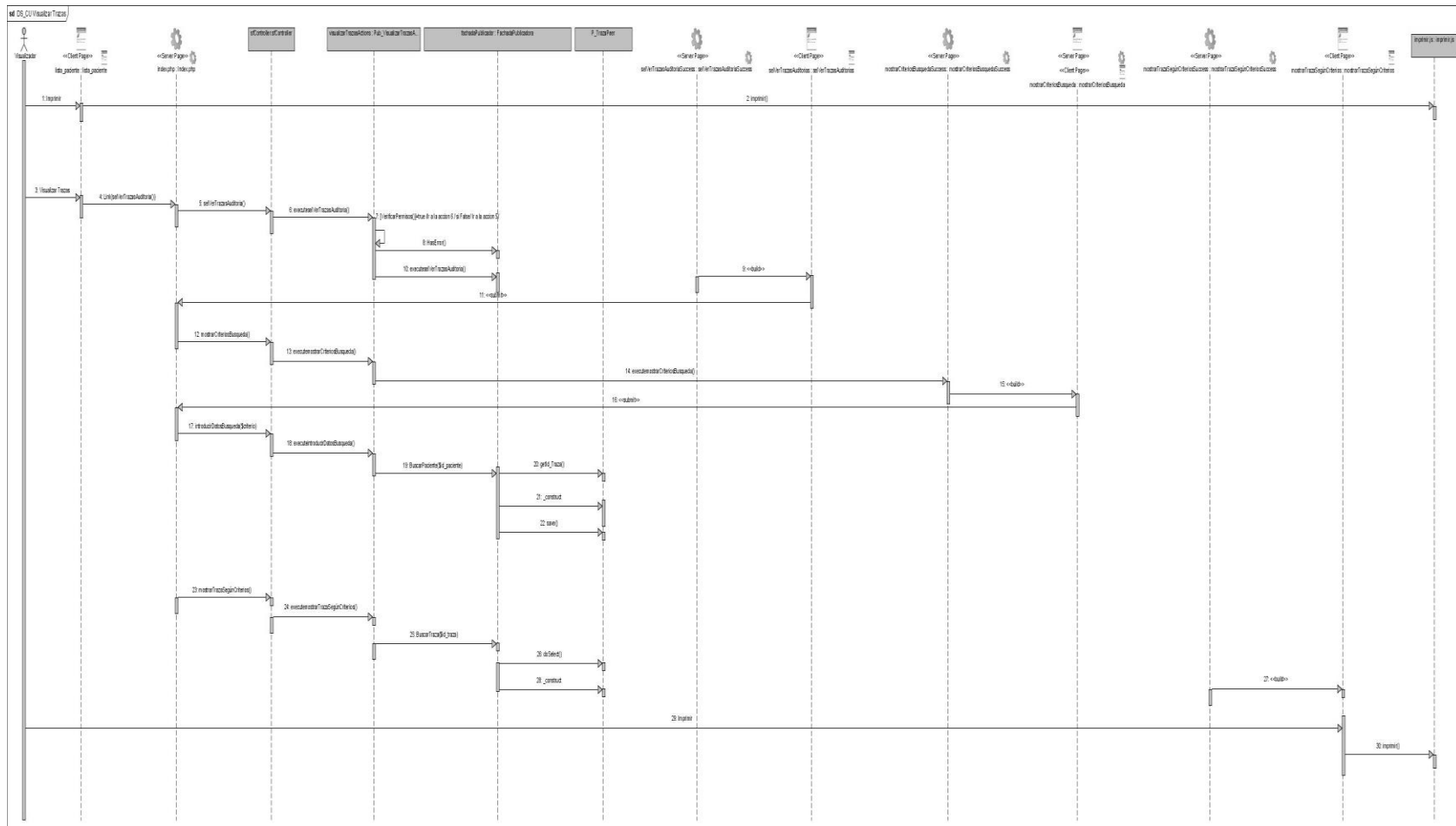


Figura 9 Diagrama de Secuencia CU Visualizar_Trazas_Auditoria

DISEÑO DEL SISTEMA

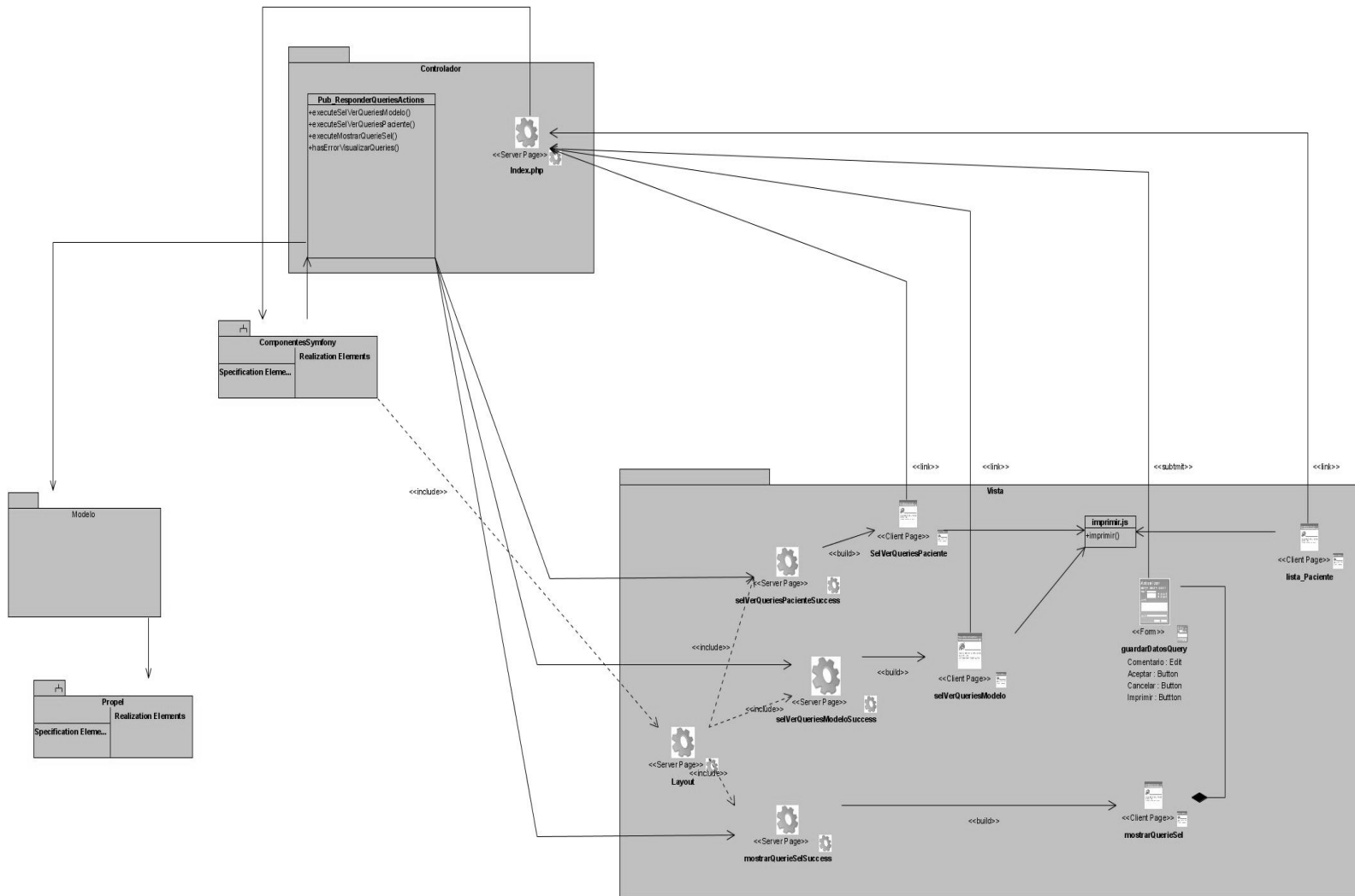


Figura 10 Diagrama de Clases del Diseño CU Responder_Queries

DISEÑO DEL SISTEMA

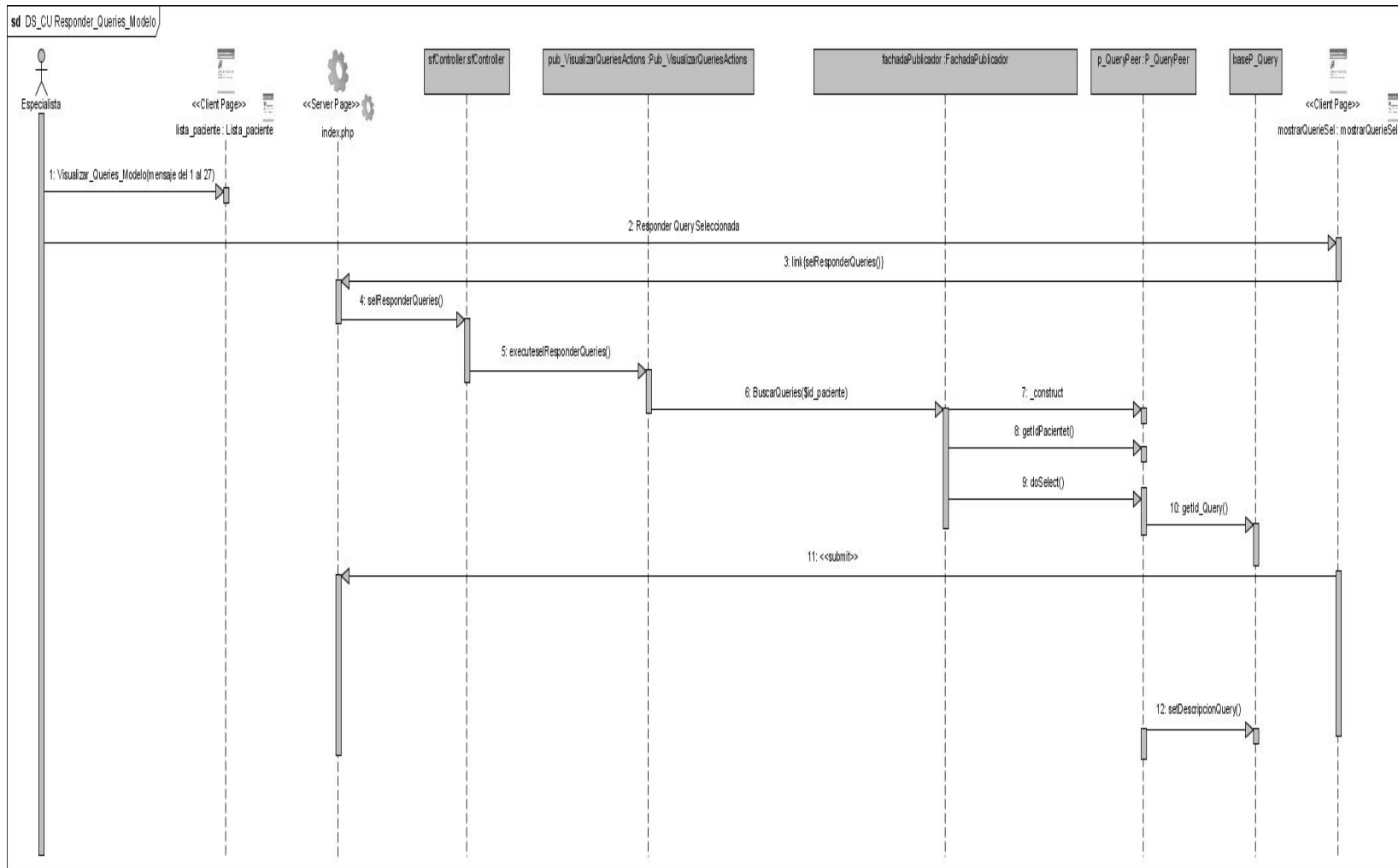


Figura 11 Diagrama de Secuencia CU Responder_Queries_Modelo

DISEÑO DEL SISTEMA

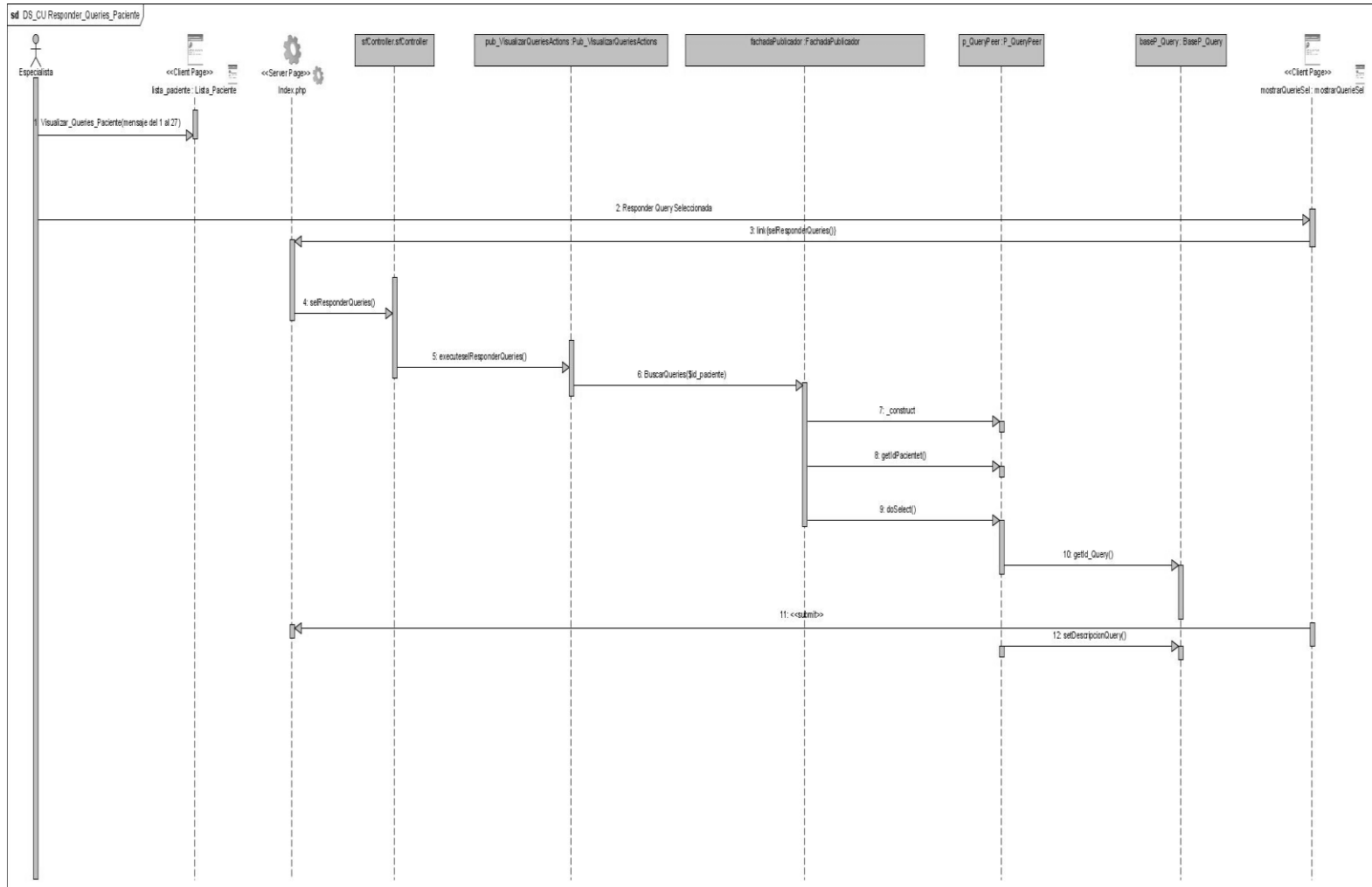


Figura 12 Diagrama de Secuencia CU Responder_Queries_Paciente

DISEÑO DEL SISTEMA

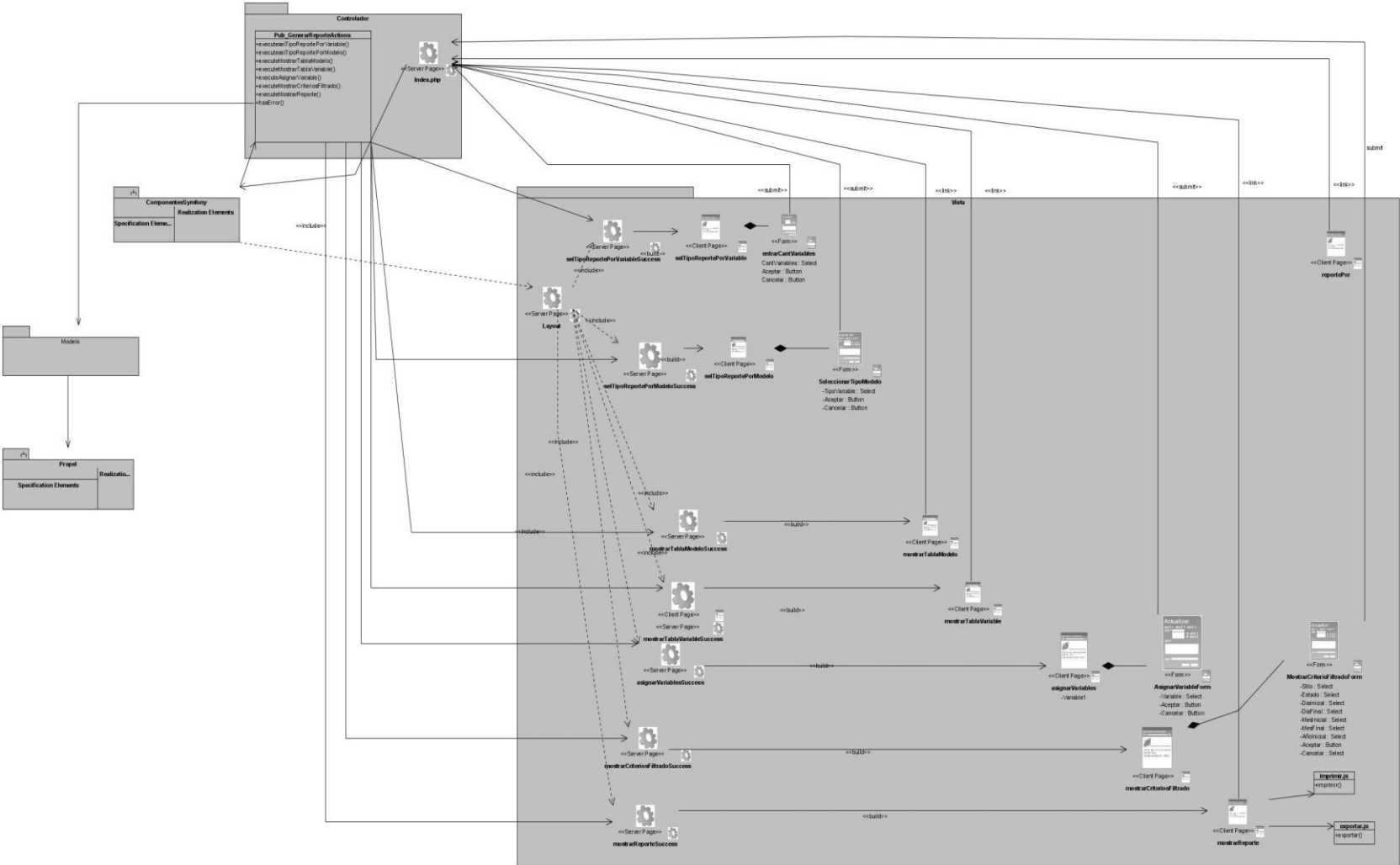


Figura 13 Diagrama de Clases del Diseño CU Generar_Reporte

DISEÑO DEL SISTEMA

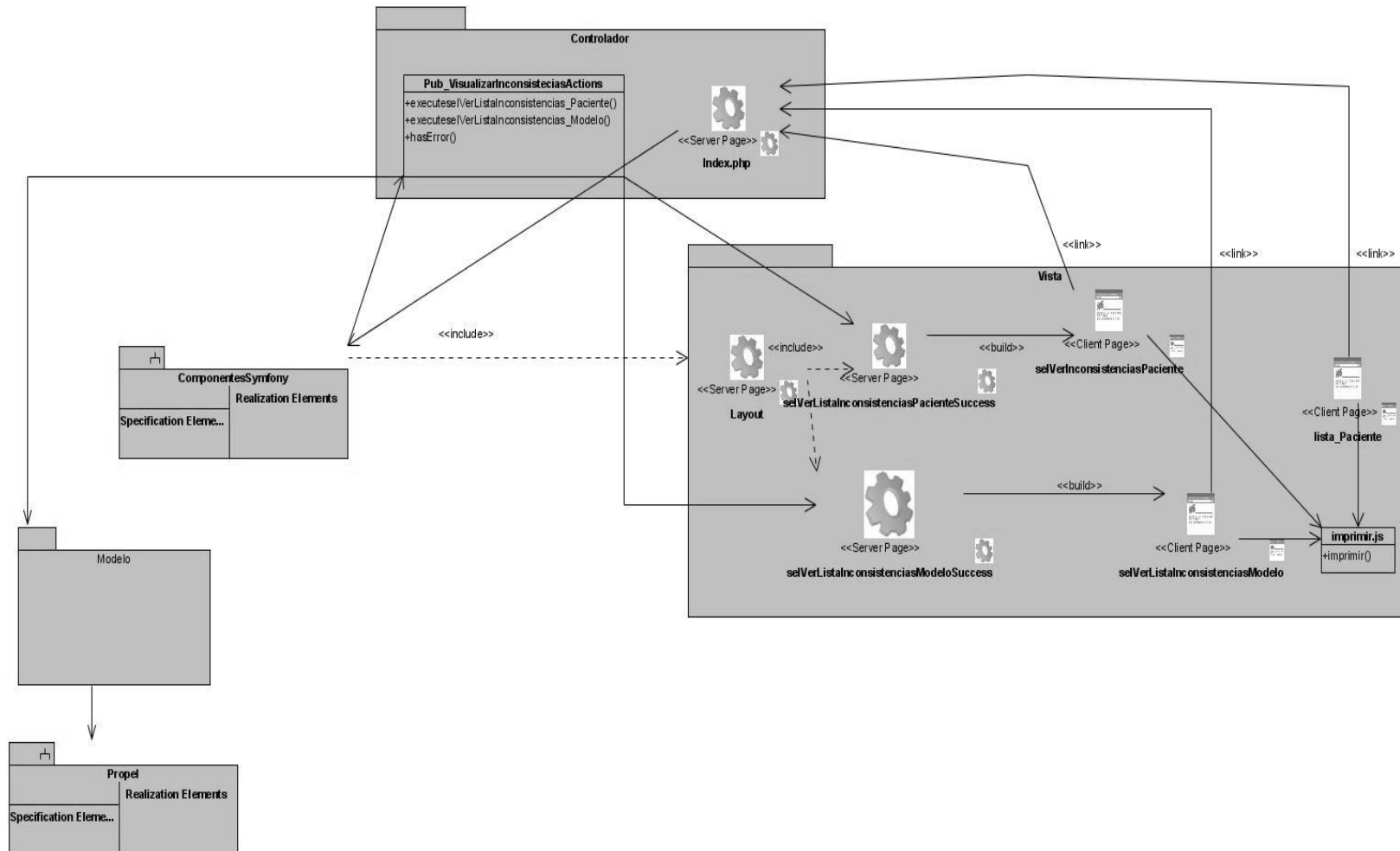


Figura 16 Diagrama de Clases del Diseño CU Visualizar_Inconsistencias

DISEÑO DEL SISTEMA

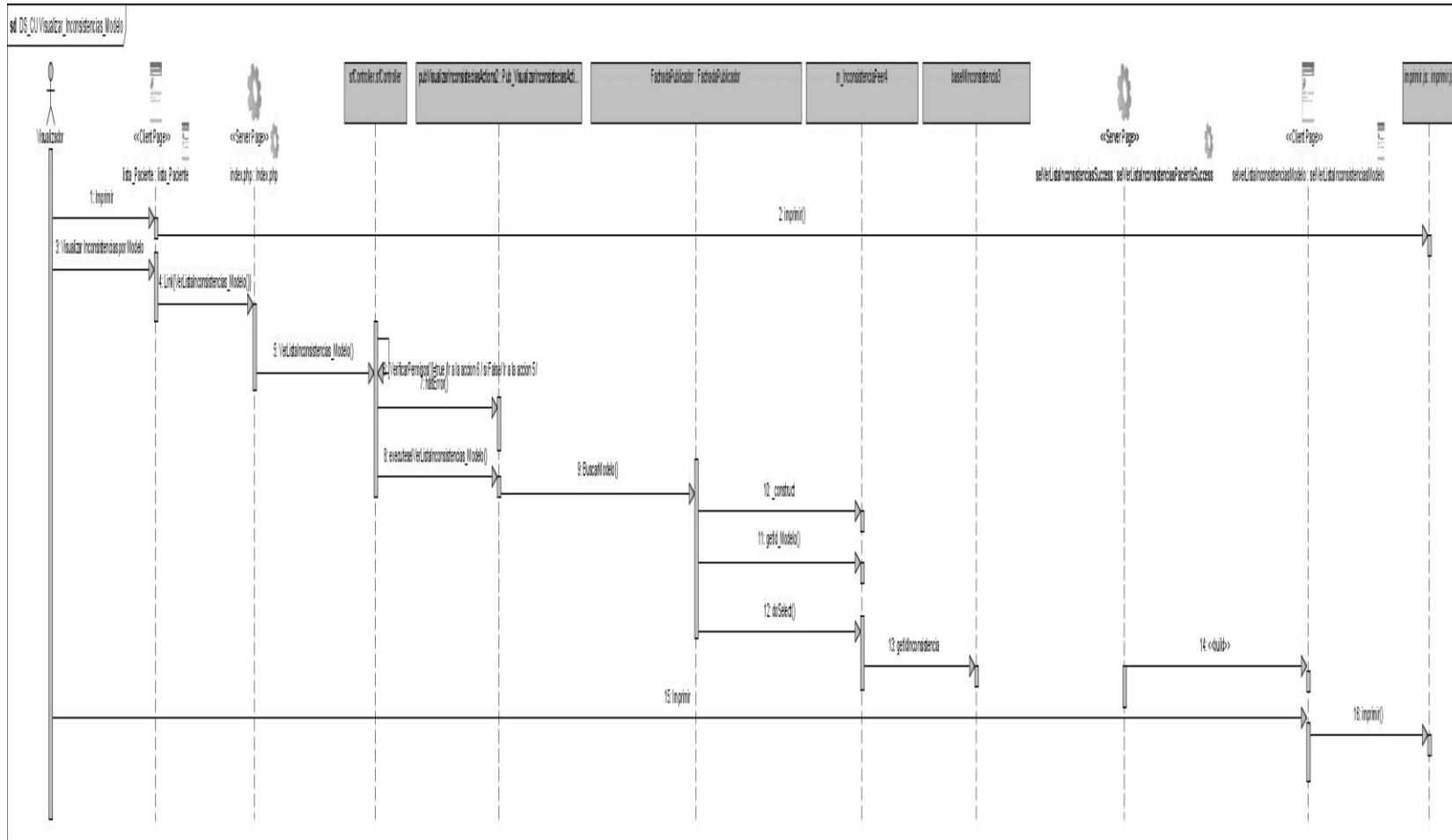


Figura 17 Diagrama de Secuencia CU Visualizar_Inconsistencias_Modelo

DISEÑO DEL SISTEMA

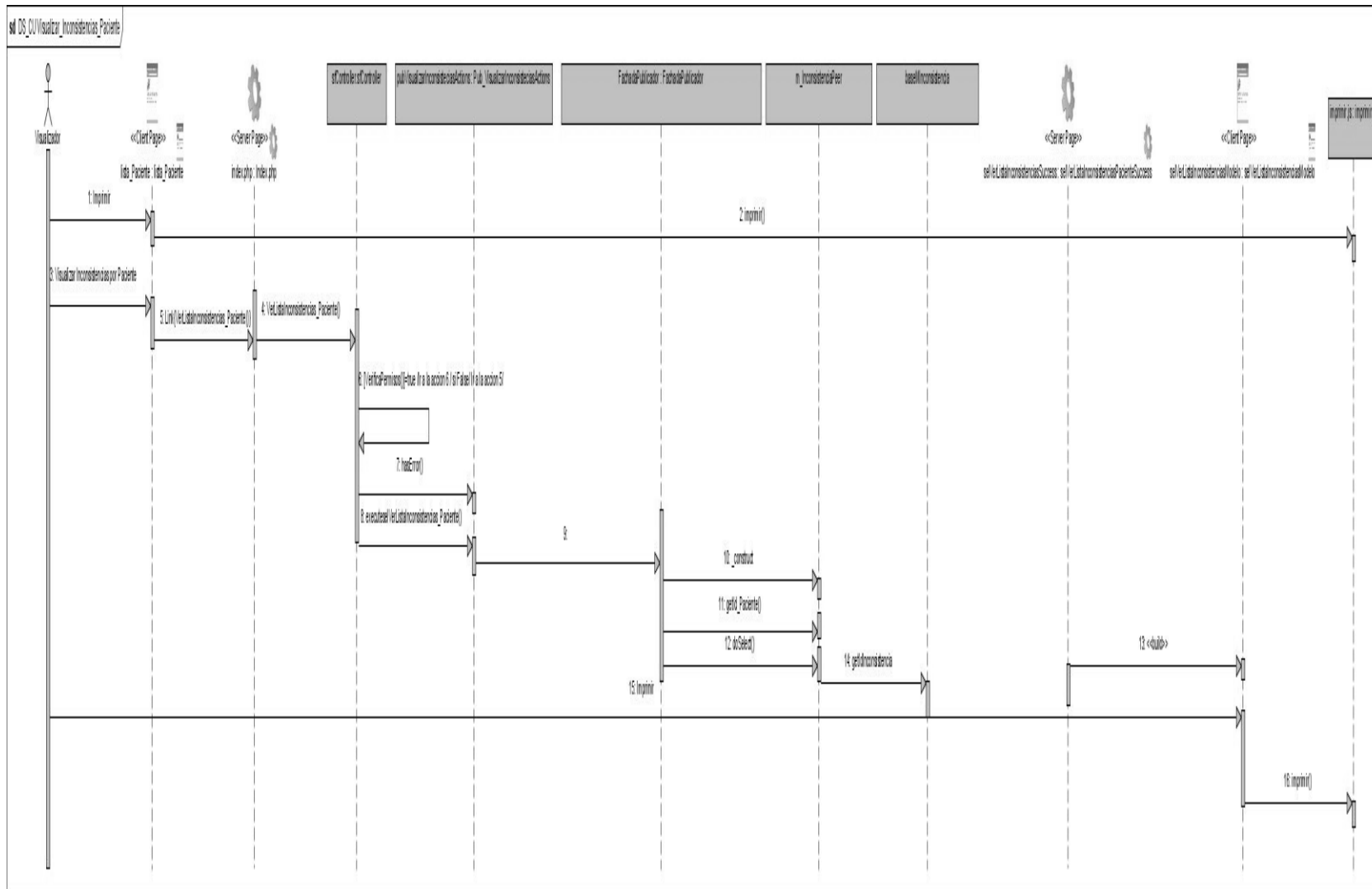


Figura 18 Diagrama de Secuencia CU Visualizar_Inconsistencias_Paciente

DISEÑO DEL SISTEMA

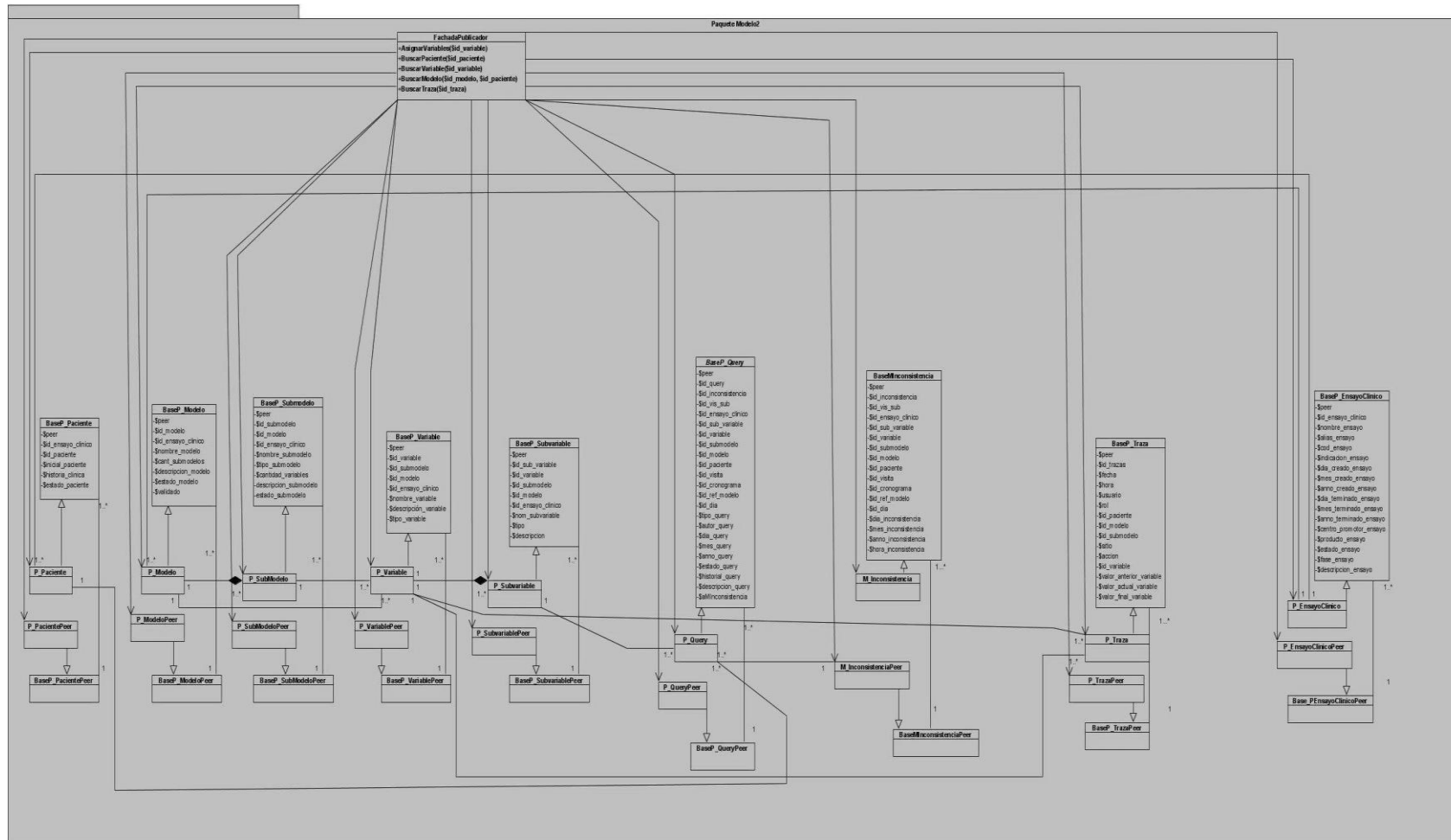


Figura 19 Capa de Modelo

DISEÑO DEL SISTEMA

2.5.2 Descripción de las clases del Diseño

Tabla 1 Descripción de la Clase Pub_GenerarReporteActions

Nombre: Pub_GenerarReporteActions	
Tipo: Clase (Esta clase contiene todas las funciones encargadas de la gestión de los reportes, utilizándose la clase FachadaPublicador para acceder a las tablas de la Base de Datos y obtener la información mostrada en el reporte)	
Atributo	Tipo
Responsabilidades	
Nombre:	Descripción:
executeselTipoReportePorVariable()	Muestra la página con las opciones que permiten entrar la cantidad de variables que contendrá el reporte generado.
executeMostrarTablaModelo()	Muestra la Tabla de reportes por modelo para lo cual ejecuta la función de la fachada del modelo BuscarModelo(\$id_modelo).
hasError()	Función que se ejecuta cuando existe error en la validación de los permisos.
executeAsignarVariable()	Se muestra la página con las opciones para escoger los valores de las variables del reporte generado, mostrándolo en una tabla que contendrá las variables a medida que se van asignando, para esto ejecuta el método de la fachada del modelo AsignarVariables(\$id_variable).

DISEÑO DEL SISTEMA

executeMostrarCriteriosFiltrado()	Muestra las opciones para el filtrado del reporte que se va a generar, dando la posibilidad de hacer una selección de la extensión del reporte en cuanto a la información.
executeMostrarReporte()	Muestra una página con el reporte realizado ya a la Base de Datos como tal, la información pedida por el usuario.
executeMostrarTablaVariable()	Muestra la Tabla de reportes por variables para lo cual ejecuta la función de la fachada del modelo BuscarVariables(\$id_variable).
executeselTipoReportePorModelo()	Es la página encargada de permitir la opción de escoger el modelo al que se le realizará el reporte.

Tabla 2 Descripción de la Clase Pub_ResponderQueriesActions

Nombre: Pub_ResponderQueriesActions	
Tipo: Clase (Esta clase contiene las funcionalidades de correspondientes al proceso de de responder una querie, englobando la funcionalidades de mostrar los datos específicos de ella y listarlas por paciente o por modelo).	
Atributo	Tipo
Responsabilidades	
Nombre:	Descripción:
execute SelVerQueriesModelo()	Muestra la Tabla de queries por modelo para lo cual ejecuta la función de la fachada del modelo BuscarModelo(\$id_modelo, \$id_paciente). Muestra la

DISEÑO DEL SISTEMA

	página que contiene una tabla, que está determinada a la visualización de queries por paciente.
executeMostrarQueryeSel()	Muestra una página con la queries seleccionada en la lista de queries por paciente o modelo, donde se ve todos los datos de la querye y permite la opción en responderla. Posteriormente se guarda la respuesta a dicha querye.
executeSelVerQueriesPaciente()	Muestra la Tabla de queries por paciente para lo cual ejecuta la función de la fachada del modelo BuscarPaciente(\$id_paciente).
hasErrorResponderQueries()	Función que se ejecuta cuando existe error en la validación de los permisos.
hasErrorVisualizarQueries()	Función que se ejecuta cuando existe error en la validación de los permisos.

Tabla 3 Descripción de la Clase Pub_VisualizarQueriesActions

Nombre: Pub_VisualizarQueriesActions	
Tipo: Clase (Esta clase contiene las funcionalidades correspondientes al proceso de visualización de queries, permitiendo verlas listadas por paciente y modelo, además de ver sus datos específicos).	
Atributo	Tipo
Responsabilidades	
Nombre:	Descripción:
execute SelVerQueriesModelo()	Muestra la Tabla que contiene la lista de queries por modelo para lo cual ejecuta la función de la fachada del

DISEÑO DEL SISTEMA

	modelo BuscarModelo(\$id_modelo).
executeMostrarQuerieSel()	Muestra una página con la queries seleccionada en la lista de queries por paciente o modelo, donde se ve todos los datos de la querie.
executeSelVerQueriesPaciente()	Muestra la Tabla de queries por paciente para lo cual ejecuta la función de la fachada del modelo BuscarPaciente(\$id_paciente).
hasErrorVisualizarQueries()	Función que se ejecuta cuando existe error en la validación de los permisos.

Tabla 4 Descripción de la Clase Pub_VisualizarInconsistenciasActions

Nombre: Pub_VisualizarInconsistenciasActions	
Tipo: Clase (Esta clase contiene todas las funciones correspondientes al proceso de visualización de inconsistencias por paciente o por modelo, además de los datos específicos de cada una de ellas, posibilitando imprimirlos).	
Atributo	Tipo
Responsabilidades	
Nombre:	Descripción:
executeselVerListaInconsistencias_Modelo()	Muestra la Tabla de inconsistencias por modelo para lo cual ejecuta la función de la fachada del modelo BuscarModelo(\$id_modelo).
executeselVerListaInconsistencias_Paciente()	Muestra la Tabla de inconsistencias por paciente para lo cual ejecuta la función

DISEÑO DEL SISTEMA

	de la fachada del modelo BuscarPaciente(\$id_paciente).
hasError()	Función que se ejecuta cuando existe error en la validación de los permisos.

Tabla 5 Descripción de la Clase Pub_VisualizarTrazasActions

Nombre: Pub_VisualizarTrazasActions	
Tipo: Clase (Esta clase contiene todas las funciones correspondientes al proceso de visualización de trazas de información de la base de datos almacenadas anteriormente, dando la posibilidad de imprimirlas).	
Atributo	Tipo
Responsabilidades	
Nombre:	Descripción:
executeselVerTrazasAuditoria()	Muestra la Tabla de trazas lo cual ejecuta la función de la fachada del modelo BuscarTraza(\$id_modelo).
executemostrarCriteriosBusqueda()	Muestra un formulario donde se escoge de qué tipo de trazas se quiere la visualización.
hasError()	Función que se ejecuta cuando existe error en la validación de los permisos.
executeintroducirDatosBusqueda()	Se muestra una página con un formulario para introducir las restricciones de la visualización de la traza: paciente, modelo, fecha, entre otros. Para lo cual se utilizan los métodos de la fachada BuscarPaciente(\$id_paciente) y BuscarModelo(\$id_modelo).

DISEÑO DEL SISTEMA

Tabla 6 Descripción de la Clase imprimir

Nombre: imprimir.js	
Tipo: Clase (Esta clase contiene la funcionalidad de imprimir, se encuentra en un fichero de códigos java script utilizados por la aplicación).	
Atributo	Tipo
Responsabilidades	
Nombre:	Descripción:
imprimir()	Función que se ejecuta cuando se pincha en el botón de imprimir y se conecta con la aplicación de imprimir del sistema.

Tabla 7 Descripción de la Clase exportar

Nombre: exportar.js	
Tipo: Clase (Esta clase contiene la funcionalidad de exportar, se encuentra en un fichero de códigos java script utilizados por la aplicación)	
Atributo	Tipo
Responsabilidades	
Nombre:	Descripción:
exportar()	Función que se ejecuta cuando se pincha en el botón de exportar y permite exportar el reporte mostrado al formato doc, pdf o txt.

DISEÑO DEL SISTEMA

2.6 Prototipos no funcionales

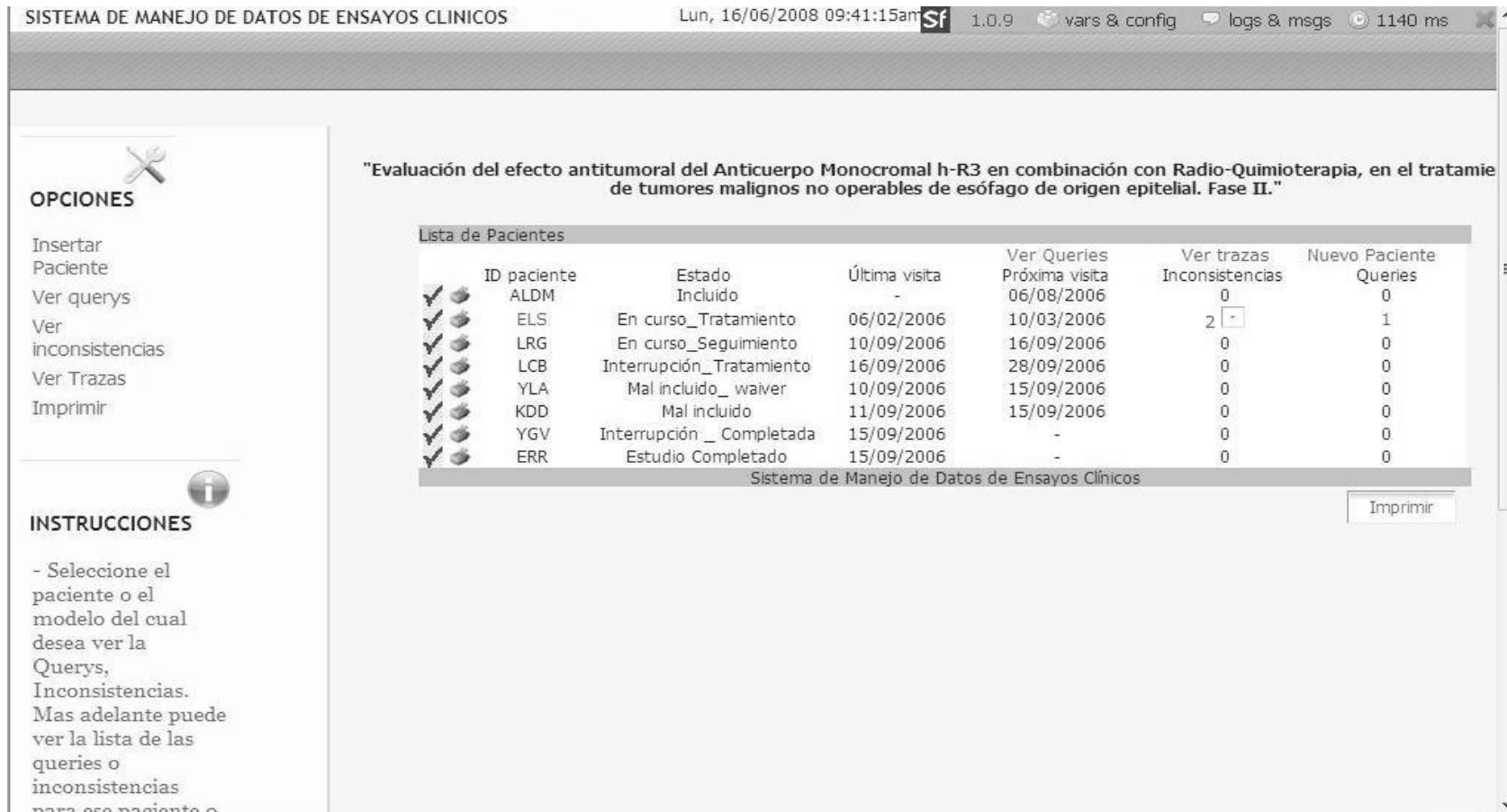




Figura 20 Prototipo No Funcional "Istar_Pacientes"

DISEÑO DEL SISTEMA

SISTEMA DE MANEJO DE DATOS DE ENSAYOS CLINICOS Lun, 16/06/2008 09:43:23am Sf 1.0.9 vars & config logs & msgs 1172 ms


OPCIONES

Mostrar
Querie
Seleccionada
Imprimir


INSTRUCCIONES

- Seleccione la posibilidad de ver la querie seleccionada.













MÓDULOS

Publicador
Cronograma
Administración
Validación
Monitoreo

"Evaluación del efecto antitumoral del Anticuerpo Monocromal h-R3 en combinación con Radio-Quimioterapia, en el tratamiento de tumores malignos no operables de esófago de origen epitelial. Fase II."

Lista de Visitas

ID Paciente: ELS **Nueva Queries**

	Modelo	Fecha	Estado	Inconsistencias	Queries
 	V1_Inclusión	06/02/2006	Firmado	0	0
 	V2_Laboratorio Clínico	10/03/2006	Monitoreado	2	1
 	V3_Examen Físico	25/04/2006	Incompleto	0	0
 	V4_Laboratorio Clínico	18/05/2006	Monitoreado	0	0
 	Eventos Adversos		Completo	0	0
 	Interrupción		Vacío	0	0

Sistema de Manejo de Datos de Ensayos Clínicos

Figura 21 Prototipo No Funcional "mostrarListaQueriesModelo"

DISEÑO DEL SISTEMA

SISTEMA DE MANEJO DE DATOS DE ENSAYOS CLINICOS Lun, 16/06/2008 09:43:48am Sf 1.0.9 vars & config logs & msgs 1098 ms

OPCIONES

Mostrar
Querie
Seleccionada
Imprimir

INSTRUCCIONES

- Seleccione la posibilidad de ver la querie seleccionada.

MÓDULOS

Publicador
Cronograma
Administración
Validación
Monitoreo

"Evaluación del efecto antitumoral del Anticuerpo Monocromal h-R3 en combinación con Radio-Ruimioterapia, en el tratamiento de tumores malignos no operables de esófago de origen epitelial. Fase II."

Lista de Querías

ID Paciente: ELS

Estado	Modelo	Autor	Descripción	Inconsistencia asociada
Asignado	Lab. Clínico	elopez	Valor de hemoglobina incorrecto	1

Sistema de Manejo de Datos de Ensayos Clínicos

Cancelar

Figura 22 Prototipo No Funcional "mostrarListaQueriesPaciente"

DISEÑO DEL SISTEMA

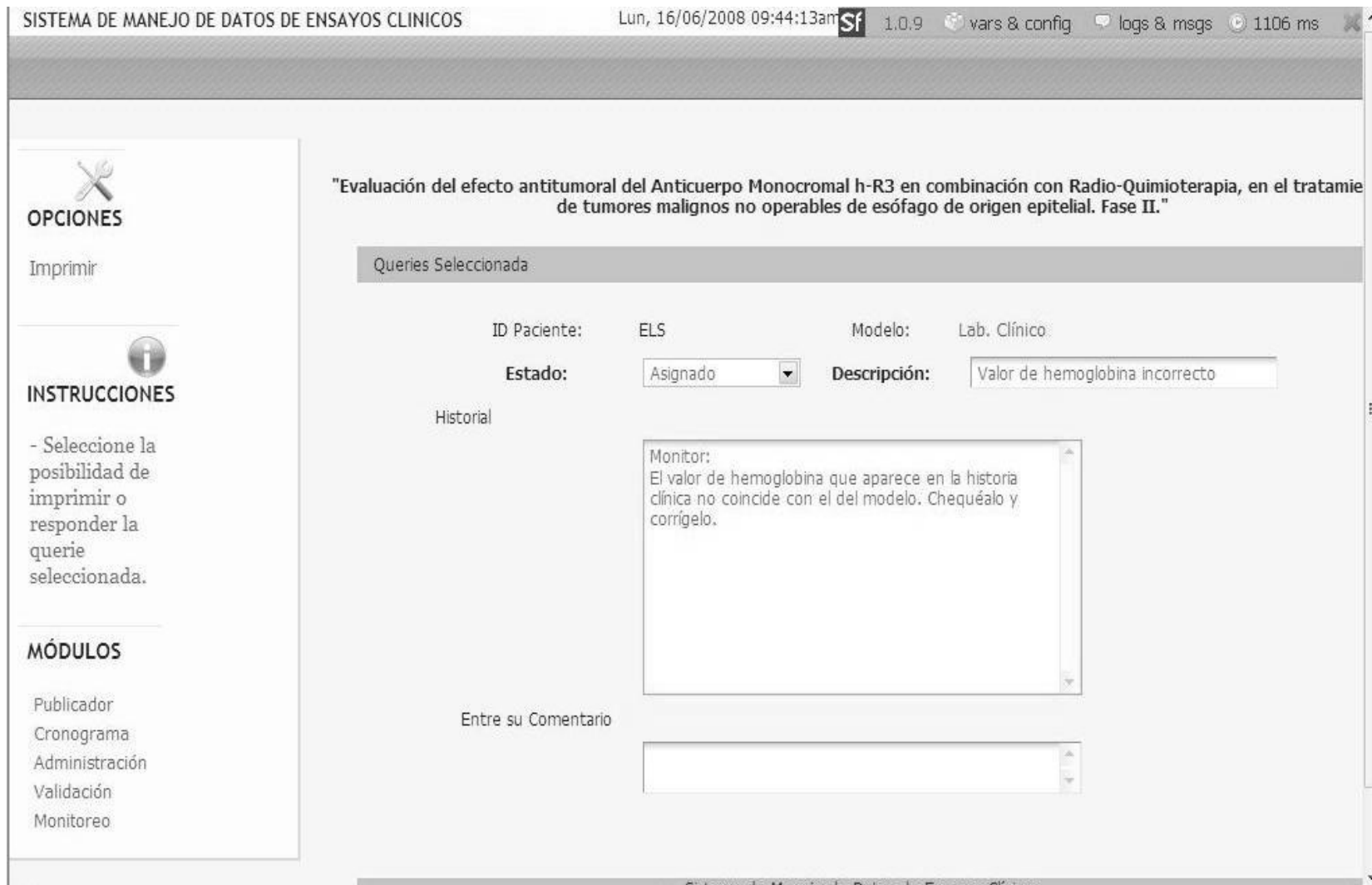


Figura 23 Prototipo No Funcional "mostrarQuerieSel"

DISEÑO DEL SISTEMA

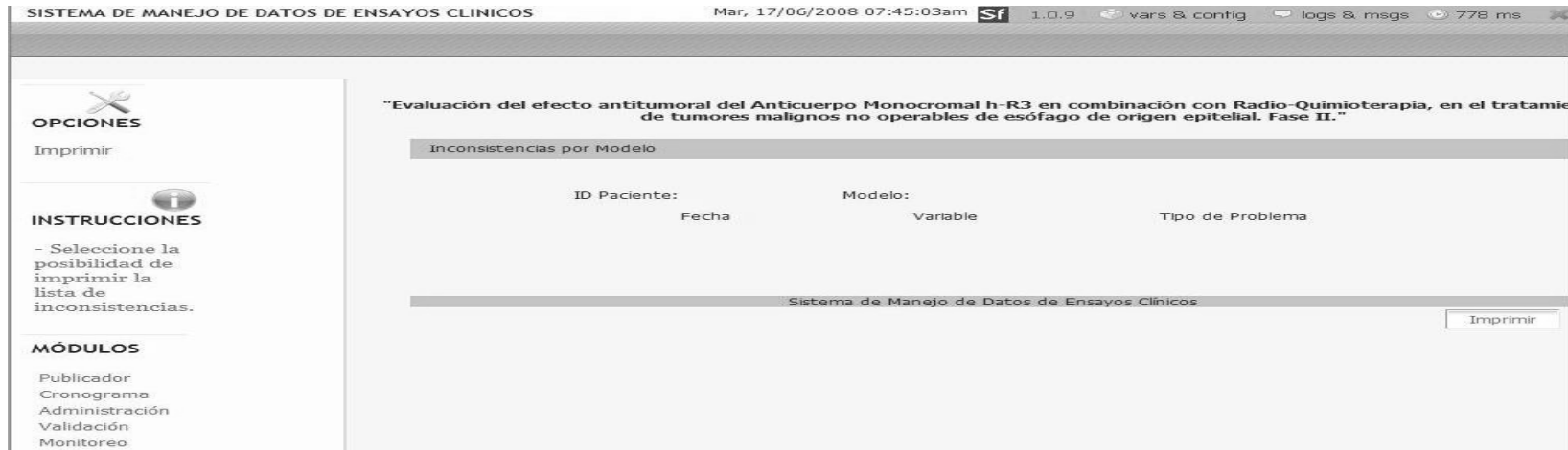


Figura 24 Prototipo No Funcional "seVerListaInconsistenciasModelo"

DISEÑO DEL SISTEMA

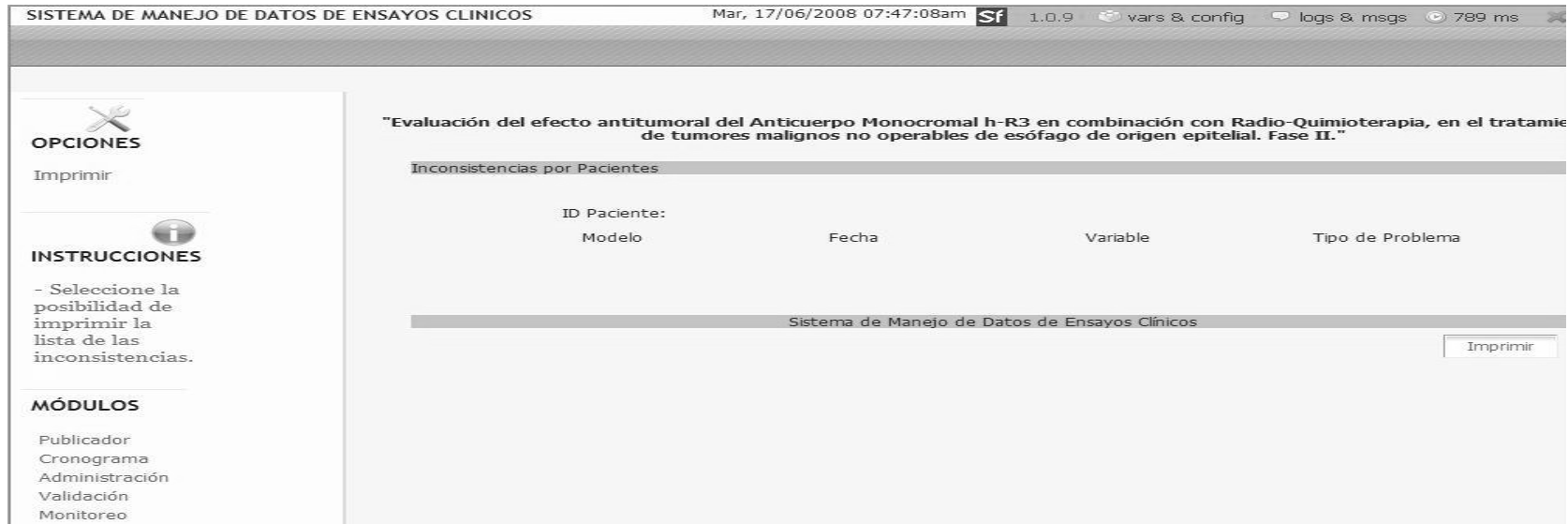


Figura 25 Prototipo No Funcional "seVerListaInconsistenciasPaciente"



Figura 26 Prototipo No Funcional "reportePor"

DISEÑO DEL SISTEMA



Figura 27 Prototipo No Funcional “selTipoReporteModelo”



Figura 28 Prototipo No Funcional “selTipoReporteVariable”

DISEÑO DEL SISTEMA

The screenshot shows a web browser window titled "SISTEMA DE MANEJO DE DATOS DE ENSAYOS CLINICOS". The browser's address bar shows the date and time "Lun, 16/06/2008 09:38:34am" and the version "1.0.9". The page content is divided into two main sections:

- INSTRUCCIONES:** A sidebar on the left containing an information icon and the text: "Seleccione la posibilidad de asignar valores a las variables."
- MÓDULOS:** A sidebar on the left listing several modules: "Publicador", "Cronograma", "Administración", "Validación", and "Monitoreo".

The main content area is titled "Asignar Variable" and contains three rows of input fields for variables V1, V2, and V3. Each row includes dropdown menus for "Visita", "Modelo", and "Submodelo", a "Ver" button, and a "Variable" dropdown menu. At the bottom of the form, there are "Aceptar" and "Cancelar" buttons. The footer of the page reads "Sistema de Manejo de Datos de Ensayos Clínicos".

Figura 29 Prototipo No Funcional "asignarVariable"

DISEÑO DEL SISTEMA



Figura 30 Prototipo No Funcional “mostrarTablaModelo”

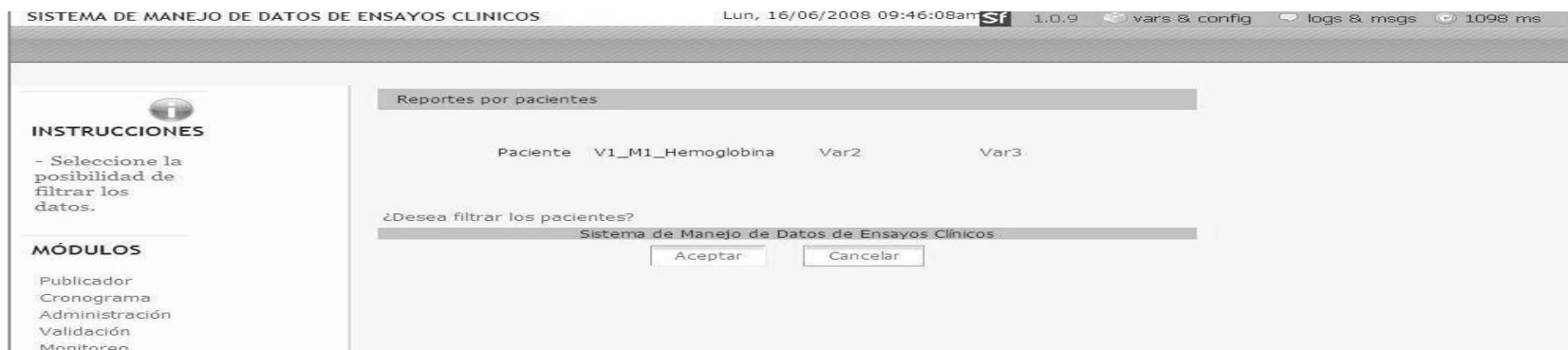


Figura 31 Prototipo No Funcional “mostrarTablaVariable”

DISEÑO DEL SISTEMA



Figura 32 Prototipo No Funcional "mostrarCriterioFiltrado"



Figura 33 Prototipo No Funcional "mostrarReporte"

DISEÑO DEL SISTEMA



Figura 34 Prototipo No Funcional “selTrazaAuditoria”



Figura 35 Prototipo No Funcional “introducirDatosBusqueda”



Figura 36 Prototipo No Funcional “mostrarTrazaSegunCriterios”

2.7 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.

DISEÑO DEL SISTEMA

La aplicación principal estará en el CIM. (Modelo de despliegue, Proyecto Ensayos Clínicos, fecha 02/02/08)

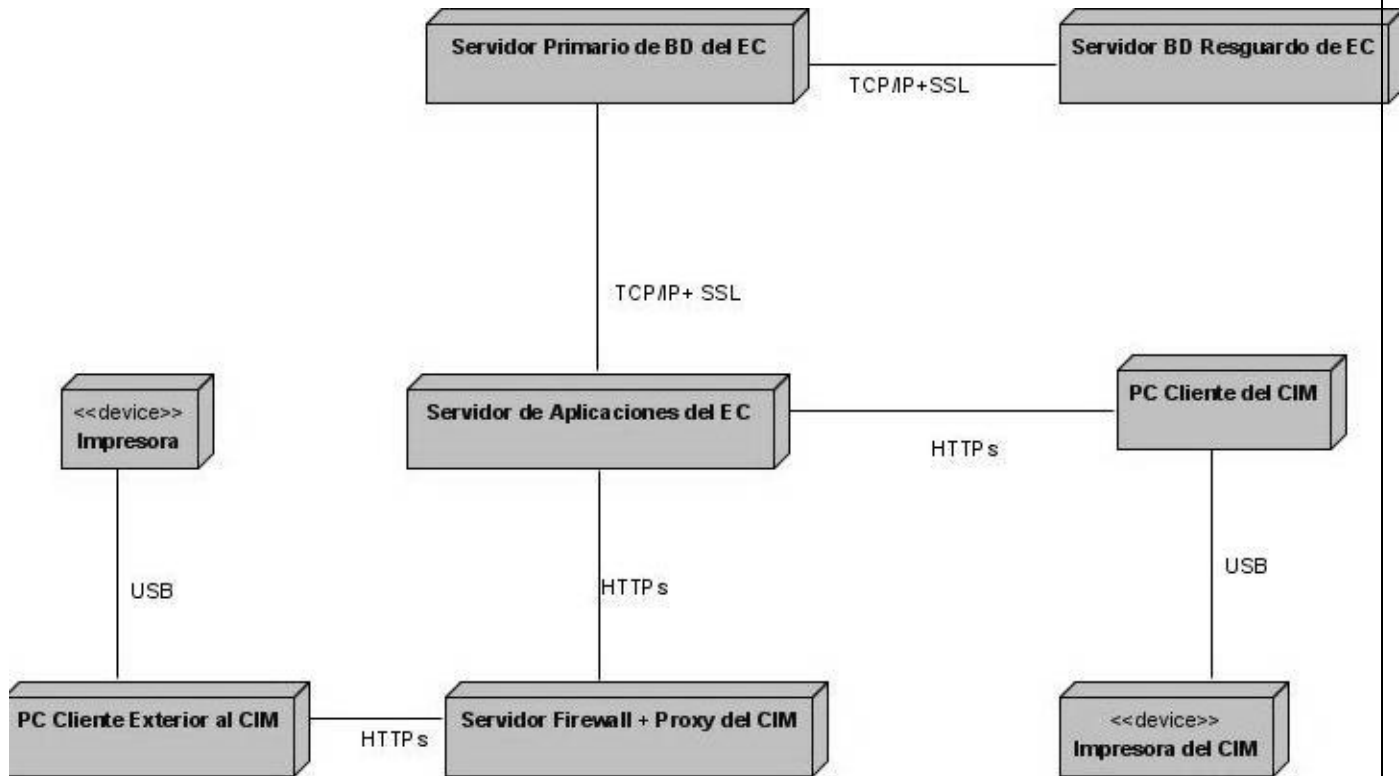


Figura 37 Modelo de Despliegue de la Aplicación

A partir del diagrama de despliegue anteriormente expuesto se define la distribución de las capas lógicas en los nodos de procesamiento de la siguiente forma:

- ✚ En el nodo Servidor de Aplicaciones del EC son ubicadas las capas lógicas Vista, Controlador, Modelo y Servicios.
- ✚ En los nodos PC Cliente del CIM y PC Cliente Exterior al CIM se presentaría el resultado de la capa Vista en cada petición, producido mediante código HTML.

DISEÑO DEL SISTEMA

- + Dentro de los nodos Servidor Primario de BD del EC y Servidor BD Resguardo de EC estaría la Base de datos y salvadas, respectivamente (no forman parte de las capas lógicas, se presentan para una mayor claridad).

2.8 Mapa de navegación

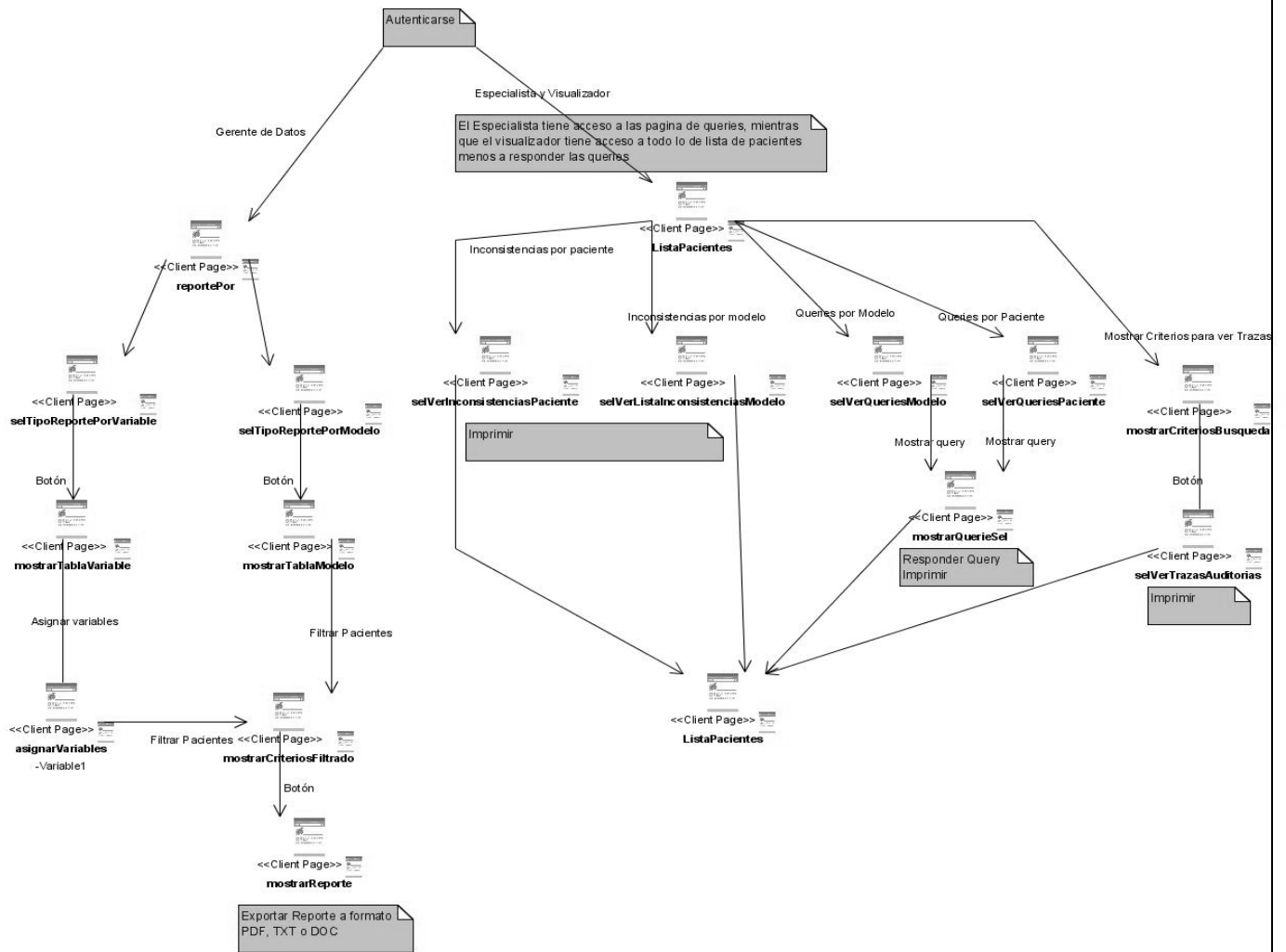


Figura 38 Mapa de Navegación

2.9 Diagrama de Clases Persistentes

Todas las clases identificadas en el dominio del análisis no son persistentes. La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Lo contrario son las clases temporales (**transient**) que son manejadas y almacenadas por el sistema en tiempo de ejecución por lo que dejan de existir cuando termina el programa. Es responsabilidad del diseñador definir cuáles clases son las que deben ser persistentes.

En fases anteriores dentro del ciclo de vida del desarrollo de un proyecto, puede que se hayan definido clases que coordinan el trabajo de varias clases, las llamadas clases controladoras. Se recomienda no definir las como persistentes porque conceptualmente no lo son.

Las clases persistentes obtenidas en la investigación son:

- ✚ BaseP_Variable.
- ✚ BaseP_Modelo.
- ✚ BaseP_Submodelo.
- ✚ BaseP_Subvariable.
- ✚ BaseP_Paciente.
- ✚ BaseP_Query.
- ✚ BaseP_Traza.
- ✚ BaseMInconsistencia.

DISEÑO DEL SISTEMA

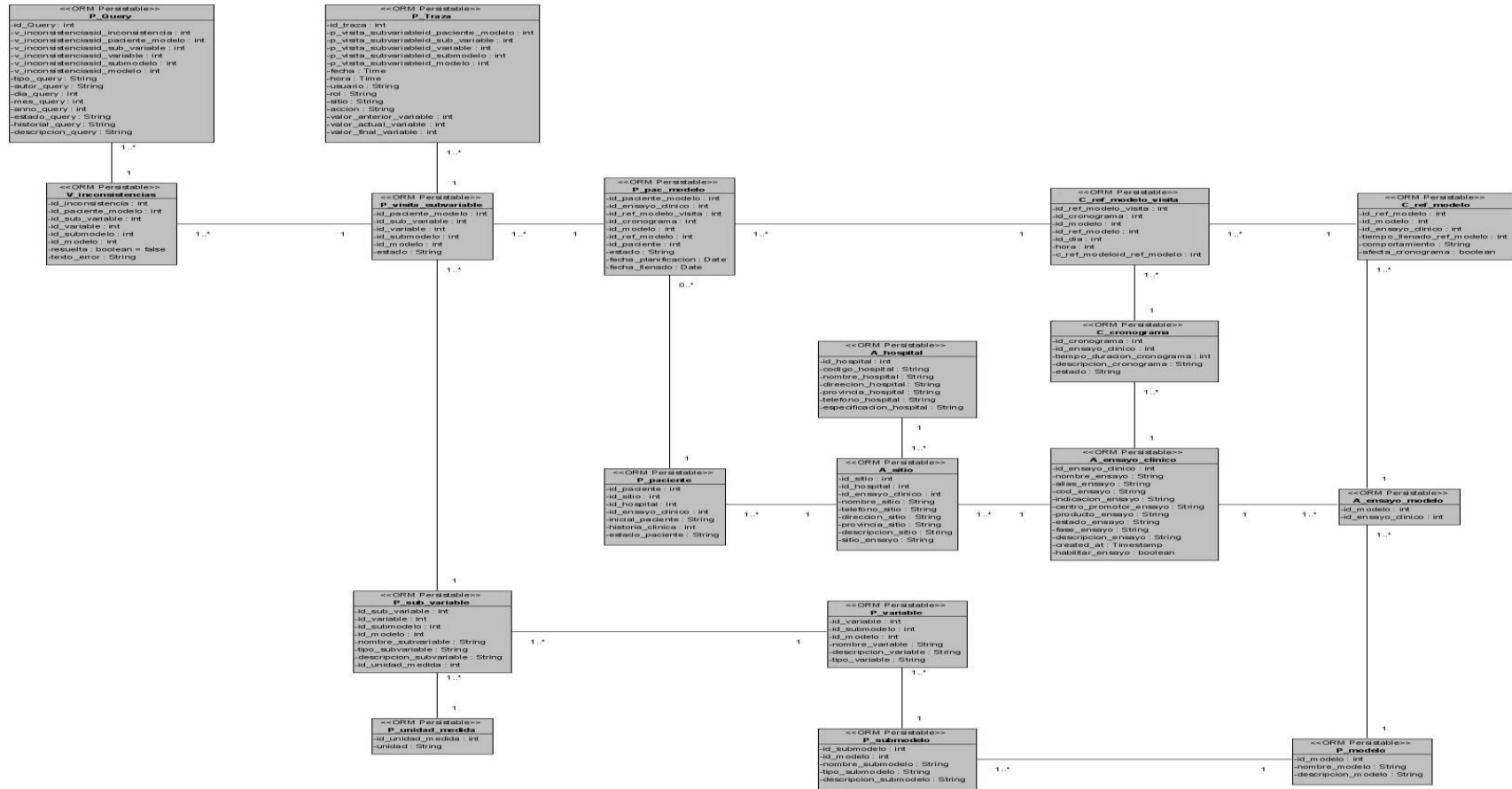


Figura 39 Diagrama de Clases Persistentes

2.10 Modelo de Datos

El diseñador de Base de datos es quien realiza la actividad Diseñar Base de Datos a partir de las clases del diseño y las realizaciones de casos de uso del diseño. El artefacto resultante de esta actividad será el Modelo de Datos, que describe la representación lógica y física de los datos persistentes.

Pasos que guían el Diseño de la Base de datos a partir de un modelo orientado a objetos.

Los pasos en el diseño de la BD son:

1. Definir las clases persistentes.
2. Refinar las clases.
3. Clasificar las clases y los atributos.
4. Realizar el diagrama de clases.
5. Realizar el diagrama de transición de estado.
6. Obtener las restricciones estáticas y las fórmulas dinámicas.
7. Convertir las clases al medio de almacenamiento.

DISEÑO DEL SISTEMA

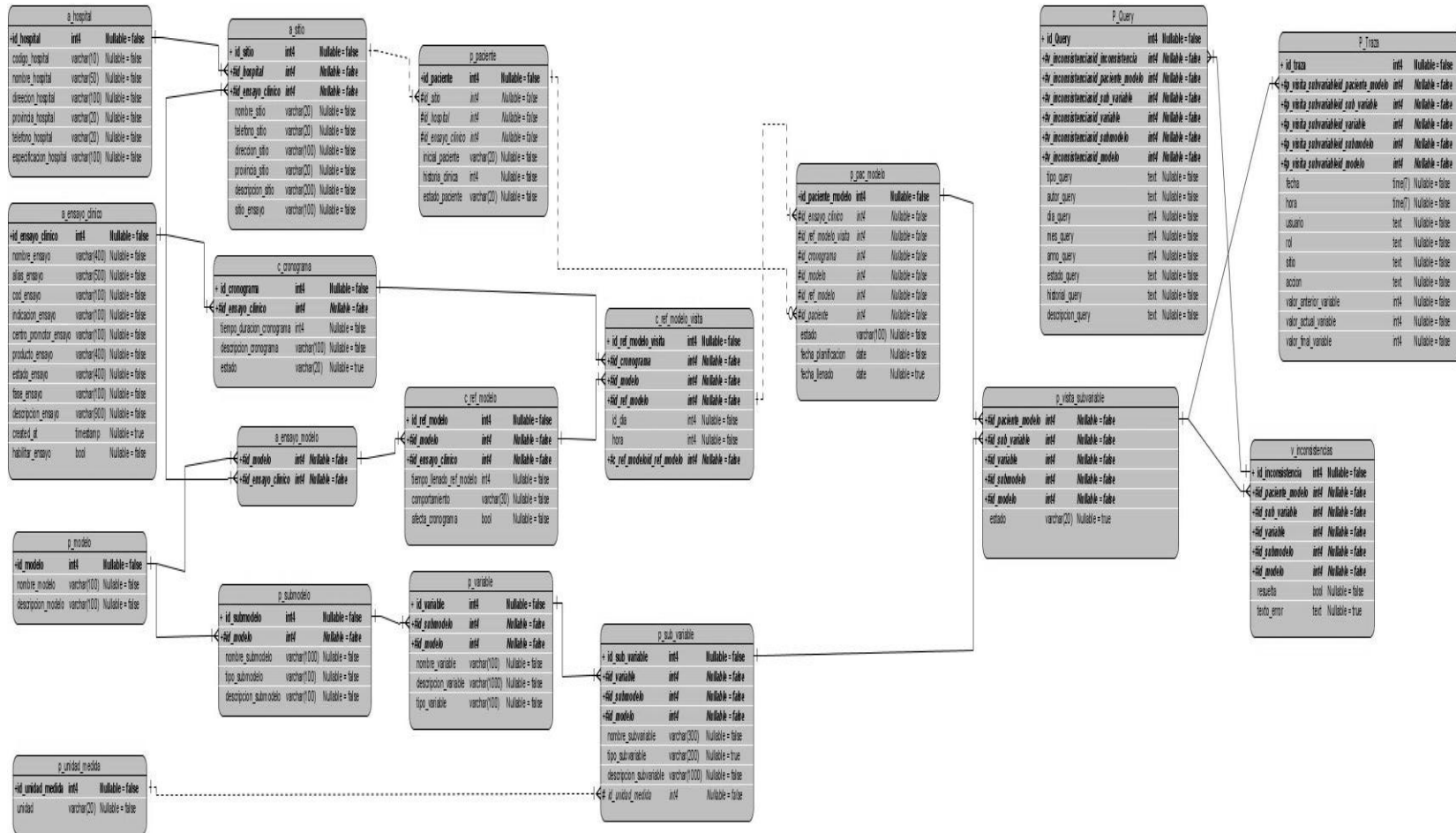


Figura 40 Modelo Relacional de la Base de Datos

DISEÑO DEL SISTEMA

2.11 Validación del Diseño

Se desarrolló el siguiente epígrafe mediante una entrevista a los implementadores que desarrollarán el diseño propuesto.

La entrevista es una forma de investigación en la cual las personas manifiestan oralmente y a través de una relación interpersonal sus opiniones, creencias, sentimientos, puntos de vista, actitudes y conductas a un entrevistador capacitado y que debe apoyarse en una guía que le dé estructuración. Cuando se aplica en forma estructurada y consistente es un tipo muy valioso de investigación. La entrevista permite cubrir con profundidad una mayor cantidad de áreas y profundizar en ellas. Además, posee una gran flexibilidad para aplicarse en diferentes circunstancias y obtener una visión global del problema que se estudia. En términos generales, una entrevista se diferencia de una encuesta o cuestionario por su carácter no estandarizado, lo cual, al mismo tiempo que es su mayor debilidad, representa también su mayor fuerza puesto que le proporciona una gran flexibilidad para aplicarse en una gama muy amplia de circunstancias (22).

Por lo anteriormente planteado se justifica la utilización de esta forma de investigación para validar de manera eficiente el diseño realizado en el presente Trabajo de Diploma, dado por la estrecha relación entre los implementadores y diseñadores del módulo y la falta de código para realizar dicha validación a través de otros métodos.

Aquí se muestran los resultados arrojados por las entrevistas realizadas a los programadores, los aspectos medidos fueron:

1. Comprensión del modelo de diseño. El diseño estuvo sujeto a cambios, introducidos a partir de la decisión de utilizar un framework, ya que en las primeras iteraciones no se contaba con una buena preparación sobre la herramienta, resultando un proceso iterativo, donde la versión final que se obtuvo

DISEÑO DEL SISTEMA

fue comprendida por los dos programadores del sub-módulo "Traceo y reporte de datos". Los artefactos obtenidos establecerán el punto de partida para la implementación permitiendo entender la organización de la información, las clases a implementar, los atributos, la identificación de los métodos relacionados con las reglas del negocio, así como la lógica de los mensajes.

2. Correspondencia de los requerimientos no funcionales con el diseño.

Se evidenció la correspondencia de los requerimientos no funcionales, con la utilización de PHP 5 como lenguaje de programación. El nombramiento de clases, atributos y métodos cumplieron con los estándares del lenguaje de programación utilizado. El requisito no funcional de extensibilidad también se evidenció con los diagramas del diseño ya que el mismo cumple con ser adaptable, con la capacidad de poder soportar funcionalidades adicionales o modificar las funcionalidades existentes sin impactar el resto de los requerimientos contemplados en el sistema, además de la gran ventaja ofrecida por el framework para cambiar fácilmente de Sistema Gestor de Base de Datos.

3. Descomposición del diseño en partes más manejables. Los diagramas del diseño están estructurados en paquetes lo que propiciará la organización en la implementación, pues será posible descomponer los trabajos de implementación en partes más manejable llevadas a cabo por diferentes programadores.

4. Inclusión de todos los requerimientos relacionados con la funcionalidad.

Se incluyeron todos los requerimientos de forma general, aunque los desarrolladores para lograr un mayor nivel de estandarización en la captura de datos y disminuir la probabilidad de introducir errores al transcribir los datos agregarán los requisitos funcionales que les sean necesarios para una mejor implementación.

DISEÑO DEL SISTEMA

5. Especificación de todas las interfaces utilizadas. Se especificaron todos los prototipos no funcionales con los implementadores, como una vía de mejor conexión entre el trabajo desarrollado por los diseñadores y la futura implementación.

Los implementadores relacionados con la aplicación, en especial con el sub-módulo publicador “Traceo y reporte de datos” desarrollan de una manera acertada su trabajo con la herramienta Symfony y PHP, aunque se superarán en su trabajo con la herramienta CASE propuesta, en vistas a un mejor manejo del diseño realizado.

Sus valoraciones acerca del diseño realizado plantean que se mantiene una estrecha relación con la arquitectura que propone Symfony; además de hacer un buen uso de los patrones de diseño y de arquitectura, adecuándose de manera satisfactoria a lo propuesto por la herramienta anteriormente citada. En caso de una reestructuración serían sencillos los cambios, debido a que está separado por módulos y de manera bien estructurada. Cualquier cambio en la base de datos no afectaría las demás capas. El diagrama de clases persistentes responde a una buena estructura de la base de datos. En caso de que aparezca algún otro módulo sería fácil adecuarlo a lo ya existente. En conclusión, los programadores relacionados con el diseño plantean que el diseño realizado es entendible y fácil de manipular e implementar.

Conclusiones

En el capítulo se presentó el modelado de artefactos correspondientes al diseño. Se mostró el modelo de diseño, donde son expuestas las realizaciones de los casos de uso definidos en el capítulo anterior, mediante diagramas de clases del diseño Web. Se realizaron los diagramas de interacción (secuencia) y el diagrama de despliegue. Se presentó el diagrama de clases persistentes y el modelo relacional de la base de datos, analizando posteriormente el diseño realizado a partir de una validación del diseño propuesto en el capítulo.

CONCLUSIONES

CONCLUSIONES

Luego del estudio realizado sobre los resultados obtenidos, en el análisis previo de esta investigación, y la generación de los artefactos relacionados con el Diseño, se permitirá a los implementadores programar un sistema que dé como resultado realizar una mejor recogida de los datos de los pacientes incluidos en los EC, incrementando además, la seguridad del sistema contra fallas, generando queries a partir de inconsistencias de los datos y visualizando trazas como vía de obtener una mejor información de los cambios efectuados en estos.

Se realizó una investigación sobre los diferentes patrones de diseño y arquitectura que intervendrán en el sistema; así como se delineó un conjunto de herramientas a utilizar. Se realizaron un conjunto de descripciones de las clases del Diseño, que se ajustan a los principios y patrones del diseño orientado a objetos.

Teniendo en cuenta la magnitud del sistema a implementar se obtuvo una serie de artefactos como resultado de la etapa de diseño, que facilitarán la comunicación entre los diseñadores y el equipo de desarrollo.

RECOMENDACIONES

RECOMENDACIONES

- ✚ Utilizar los artefactos desarrollados en el diseño propuesto para una futura implementación sobre la base de componentes reutilizables.
- ✚ Realizar un sistema de cursos de superación dirigidos a los futuros implementadores, que les haga más sencillo y comprensible implementar el sistema.

REFERENCIA BIBLIOGRÁFICA

REFERENCIA BIBLIOGRÁFICA

1. MedlinePlus: Información de salud para usted. [En línea] Biblioteca Nacional de Medicina de Estados Unidos e Institutos de Salud, 2007. [Citado el: 29 de noviembre de 2007.] <http://www.nlm.nih.gov/medlineplus/spanish/clinicaltrials.html>.
2. **Franco, Leopoldo Vega.** Ensayos Clínicos. [En línea] 2007. [Citado el: 29 de noviembre de 2007.] <http://www.facmed.unam.mx/deptos/salud/spdos/presenta/enscli.pdf>
3. **Estrada, M y Lopez, E.** *Sistema de manejo de Datos: Módulo Publicador.* s.l. : Universidad de las Ciencias Informáticas, 2007.
4. Definición de software. [En línea] 2008. [Citado el: 15 de Diciembre de 2008.] <http://www.cosaslibres.com/software.html>
5. **Moreno, C.** Manual del Usuario de moReport Versión 0.1. [En línea] 2003. [Citado el: 15 de Diciembre de 2007.] <http://bo.unsa.edu.ar/docacad/morep/>
6. **SQL Software.** Queryx*Report. [En línea] 2007. [Citado el: 15 de Diciembre de 2007.] <http://www.sqlsoftware.com.co/Tools/QueryxReport.htm>.
7. OpenClinica 2.0 liberado y gratuito (y ético). [En línea] 2008. [Citado el: 4 de 1 de 2008.] <http://www.sociedad20.com/20061110-openclinica-20-liberado-y-gratuito-y-etico.html>
8. Pivotal. [En línea] 2008. [Citado el: 15 de Marzo de 2008.] <http://www.pivotal.es/extranet/servicios>.

REFERENCIA BIBLIOGRÁFICA

9. Hipocrates. [En línea] 2008. [Citado el: 15 de Marzo de 2008.]
<http://www.hipocrates.com/informa/descri.htm>.
10. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. s.l. : Adilison Wesley, 2000.
11. **Larman, Craig.** *UML y Patrones*. s.l. : Felix Varela, 1999. págs. 4-19. Vol. I
12. **Vizcaíno, Aurora, García, Felix Oscar y Caballero, Ismael.** LabTR1_VP. [En línea] 2008. [Citado el: 1 de Junio de 2008.] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf
13. **Potencier, Fabien y Zaninotlo, Francois.** *Symfony, la guía definiiva*. s.l. : Apress, 2007.
14. **Kaber, Mohammed.** *La biblia del Servidor Apache 2*. s.l. : Anaya Multimedia, 2003.
15. **Heras, de la Cruz Daniel y Zumbado, Carlos.** *Flash, PHP y MySQL*. s.l. : Anaya Multimedia, 2004.
16. Subversion. [En línea] 2007. [Citado el: 30 de Marzo de 2008.]
<http://www.osmosislatina.com/subversion/basico.htmMySQL>
17. Kompozer. [En línea] 2008. [Citado el: 1 de Abril de 2008.]
<http://blog.taragana.com/index.php/archive/recommended-free-windows-linux-html-editor-open-source/es>.

REFERENCIA BIBLIOGRÁFICA

18. Eclipse. [En línea] 2008. [Citado el: 1 de Abil de 2008.] <http://www.eclipse.org>
19. **Pressman, Roger.** *Ingeniería del Software.Un enfoque práctico.* s.l. : Mc Graw Hill, 2003.
20. **Souto, Abel Marcos.** Patrones Software: Breve Introducción. [En línea] 2008. [Citado el: 15 de Abril de 2008.]
<http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=12>
21. **García, Walter y Ospina, Gustavo.** Decorator. [En línea] 2008. [Citado el: 15 de Abril de 2008.]
<http://agamenon.uniandes.edu.co/~pfiguero/soo/PatronesDiseno/Decorator/Decorator.htm>.
22. **Rodríguez, Irene Trelles.** *Comunicación Organizacional. Selección de Lecturas.* s.l. : "Felix Varela", 2001.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

1. MedlinePlus: Información de salud para usted. [En línea] Biblioteca Nacional de Medicina de Estados Unidos e Institutos de Salud, 2007. [Citado el: 29 de noviembre de 2007.] <http://www.nlm.nih.gov/medlineplus/spanish/clinicaltrials.html>
2. **Franco, Leopoldo Vega.** Ensayos Clínicos. [En línea] 2007. [Citado el: 29 de noviembre de 2007.] <http://www.facmed.unam.mx/deptos/salud/spdos/presenta/enscli.pdf>
3. **Estrada, M y Lopez, E.** *Sistema de manejo de Datos: Módulo Publicador.* s.l. : Universidad de las Ciencias Informáticas, 2007.
4. Definición de software. [En línea] 2008. [Citado el: 15 de Diciembre de 2008.] <http://www.cosaslibres.com/software.html>
5. **Moreno, C.** Manual del Usuario de moReport Versión 0.1. [En línea] 2003. [Citado el: 15 de Diciembre de 2007.] <http://bo.unsa.edu.ar/docacad/morep/>
6. **SQL Software.** Queryx*Report. [En línea] 2007. [Citado el: 15 de Diciembre de 2007.] <http://www.sqlsoftware.com.co/Tools/QueryxReport.htm>.
7. OpenClinica 2.0 liberado y gratuito (y ético). [En línea] 2008. [Citado el: 4 de 1 de 2008.] <http://www.sociedad20.com/20061110-openclinica-20-liberado-y-gratuito-y-etico.html>
8. Pivotal. [En línea] 2008. [Citado el: 15 de Marzo de 2008.] <http://www.pivotal.es/extranet/servicios>.

BIBLIOGRAFÍA

9. Hipocrates. [En línea] 2008. [Citado el: 15 de Marzo de 2008.]
<http://www.hipocrates.com/informa/descr.htm>.
10. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. s.l. : Adilison Wesley, 2000.
11. **Larman, Craig.** *UML y Patrones*. s.l. : Felix Varela, 1999. págs. 4-19. Vol. I
12. **Vizcaíno, Aurora, García, Felix Oscar y Caballero, Ismael.** LabTR1_VP. [En línea] 2008. [Citado el: 1 de Junio de 2008.] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf
13. **Potencier, Fabien y Zaninotlo, Francois.** *Symfony, la guía definiiva*. s.l. : Apress, 2007.
14. **Kaber, Mohammed.** *La biblia del Servidor Apache 2*. s.l. : Anaya Multimedia, 2003.
15. **Heras, de la Cruz Daniel y Zumbado, Carlos.** *Flash, PHP y MySQL*. s.l. : Anaya Multimedia, 2004.
16. Subversion. [En línea] 2007. [Citado el: 30 de Marzo de 2008.]
<http://www.osmosislatina.com/subversion/basico.htmMySQL>
17. Kompozer. [En línea] 2008. [Citado el: 1 de Abril de 2008.]
<http://blog.taragana.com/index.php/archive/recommended-free-windows-linux-html-editor-open-source/es>.

BIBLIOGRAFÍA

18. Eclipse. [En línea] 2008. [Citado el: 1 de Abril de 2008.] <http://www.eclipse.org>
19. **Pressman, Roger.** *Ingeniería del Software. Un enfoque práctico.* s.l. : Mc Graw Hill, 2003.
20. **Souto, Abel Marcos.** Patrones Software: Breve Introducción. [En línea] 2008. [Citado el: 15 de Abril de 2008.] <http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=12>
21. **García, Walter y Ospina, Gustavo.** Decorator. [En línea] 2008. [Citado el: 15 de Abril de 2008.] <http://agamenon.uniandes.edu.co/~pfiguero/soo/PatronesDiseno/Decorator/Decorator.htm>.
22. **Rodríguez, Irene Trelles.** *Comunicación Organizacional. Selección de Lecturas.* s.l. : "Felix Varela", 2001.
23. **Gallego, José Antonio.** *Desarrollo Web con PHP y MySQL.* s.l. : Anaya Multimedia, 2003
24. ¿Qué es un ensayo clínico? - Instituto del Corazón de Texas (Texas Heart Institute). [En línea] Centro de Información Cardiovascular del Texas Heart Institute , 2007. http://www.texasheartinstitute.org/HIC/Topics_Esp/FAQ/clinical_trials_span.cfm.

BIBLIOGRAFÍA

25. vivaphp. [En línea] 2008. [Citado el: 2 de 5 de 2008.] <http://www.vivaphp.com.ar/frameworks/zend-framework-1.0.html>.
26. Desarrollo en PHP-Los framework agilizan tu trabajo. [En línea] 2008. [Citado el: 2 de Mayo de 2008.] http://blog.enicaragua.org.ni/roller/php/entry/crea_tu_radioblog_sin_php.
27. **Pérez, Damian**. Los Frameworks de PHP agilizan tu trabajo. [En línea] 2008. [Citado el: 2 de Mayo de 2008.] <http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo>
28. **Charte, Francisco**. *Proyectos profesionales. PHP 5*. s.l. : Anaya Multimedia, 2004.
29. Internet Information Services. [En línea] 2008. [Citado el: 15 de Marzo de 2008.] <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>
30. **Ballester Marsal, Andres**. Documento de Arquitectura de Software (Ensayos clínicos). Ciudad de la Habana : s.n., 2008.

ANEXOS

1 Diagrama de Casos de Uso del Sistema

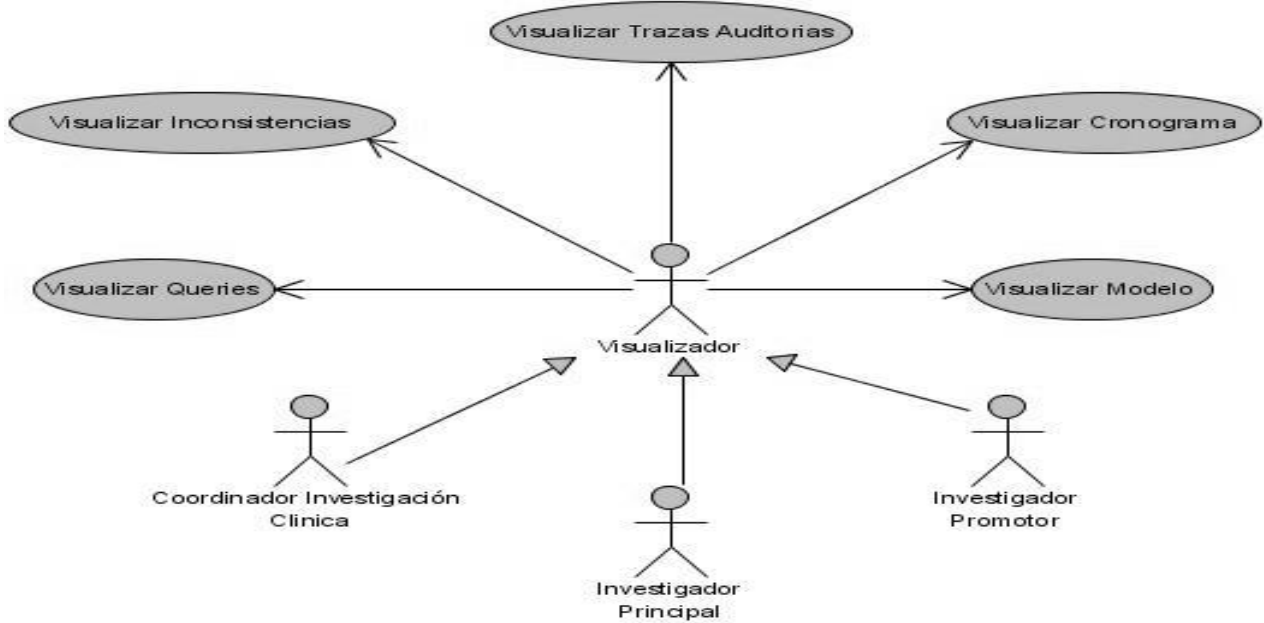


Figura 41 Diagrama de Casos de Uso del Sistema: Generalización_Actor_Visualizador



Figura 42 Diagrama de Casos de Uso del Sistema: Gerente de Datos

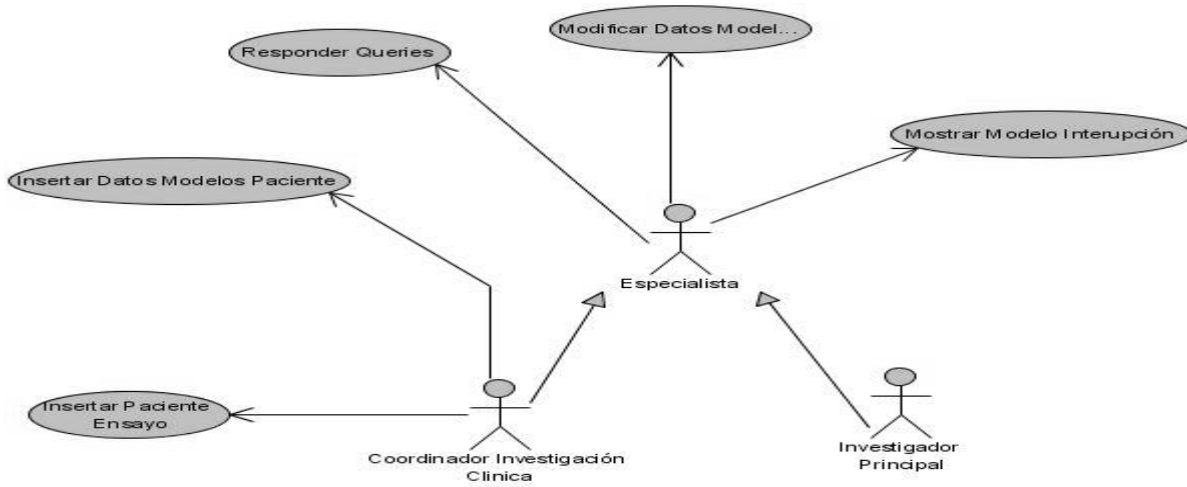


Figura 43 Diagrama de Casos de Uso del Sistema: Generalización_Actor_Especialista

2 Descripción de los Casos de Uso del Sistema

Tabla 8 Descripción del CU Visualizar_Queries

Caso de Uso:	Visualizar_Queries
Actores:	Visualizador (Inicia)
Propósito:	Ver Listado de Queries y su estado.
Resumen:	El caso de uso comienza cuando el Coordinador de la investigación clínica, Investigador promotor, Investigador principal deciden ver las queries de un paciente o de un modelo en particular.
Referencia:	R4, R5, R10, R11, R12
Precondiciones:	Que exista la query.
Poscondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

1.- El Visualizador selecciona ver queries de un paciente o de un modelo determinado.		1.1.- El sistema muestra la "Lista de Queries" paciente o modelo seleccionado.
2.- El visualizador selecciona la query que desea ver.		2.1.- El sistema muestra la query seleccionada. Finalizando el caso de uso.
Prioridad:	Secundario	
Prototipo:	Expediente de Proyecto (Tabla 2.6 Caso de Uso del Sistema: Visualizar_Queries).	

Tabla 9 Descripción del CU Responder_Queries

Caso de Uso:	Responder_Queries	
Actores:	Especialista (Inicia)	
Propósito:		
Resumen:	El caso de uso comienza cuando el Coordinador de la investigación clínica o Investigador Principal decide responder query de un modelo, paciente. El caso de uso finaliza cuando el coordinador o investigador principal responde una query.	
Referencia:	R4, R5, R10, R11, R12, R13, R34	
Precondiciones:	Que existan queries asignadas.	
Poscondiciones:	Se obtengan las queries actualizadas.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1.- El especialista decide responder una query de un modelo, paciente.	1.1.- Se dispara el caso de Uso visualizar queries. 1.2.- El sistema le muestra la Zona de Comentario.	
2.- El especialista revisa y responde la query.	2.1.- El sistema guarda los datos de la query y se habilita la opción de Imprimir.	
3.- El especialista decide imprimir o cerrar la query. Finalizando el caso de uso.		
Prioridad:	Crítico	

Prototipo:	Expediente de Proyecto (Tabla 2.7 Caso de Uso del Sistema: Responder_Queries).
-------------------	--

Tabla 10 Descripción del CU Visualizar_Inconsistencias

Caso de Uso:	Visualizar_Inconsistencias	
Actores:	Visualizador (Inicia)	
Propósito:	Ver listado de inconsistencias de un modelo o paciente.	
Resumen:	El caso de uso comienza cuando el Coordinador de la investigación clínica, Investigador promotor, Investigador principal, deciden ver las inconsistencias de un modelo o paciente.	
Referencia:	R3, R4, R14, R15, R33	
Precondiciones:	Que exista el paciente.	
Poscondiciones:		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1.- El Visualizador decide ver la lista de Inconsistencias” de un paciente o un modelo.	1.1.- El sistema muestra la “Lista de Inconsistencias” del modelo o paciente seleccionado. 1.2.- Se habilita la opción de Imprimir.	
2.- El Visualizador decide imprimir o cerrar la inconsistencia. Finalizando el caso de uso.		
Prioridad:	Crítico	
Prototipo:	Expediente de Proyecto (Tabla 2.8 Caso de Uso del Sistema: Visualizar_Inconsistencias).	

Tabla 11 Descripción del CU Visualizar_Trazas

Caso de Uso:	Visualizar_Trazas_Auditorias
Actores:	Visualizador (Inicia)
Propósito:	Ver trazas de auditorías de un ensayo.
Resumen:	El caso de uso comienza cuando el Coordinador de la investigación clínica, Investigador promotor, Investigador principal, deciden ver las trazas de auditoría de un ensayo, por

	determinado criterio de búsqueda.
Referencia:	R4, R5, R19, R20, R36
Precondiciones:	Que existan las trazas de auditorías.
Poscondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Visualizador decide “Ver las trazas de auditorías” de un ensayo determinado.	1.1.- El sistema muestra los criterios por los cuales el Visualizador puede realizar su búsqueda.
2. El Visualizador introduce los datos por los cuales desea realizar su búsqueda, según los criterios seleccionados.	2.1 El sistema muestra las trazas atendiendo a los criterios escogidos por el Visualizador.
	3.1- Se habilita la opción de Imprimir.
4.- El visualizador decide imprimir o cerrar la traza. Finalizando el caso de uso.	
Prioridad:	Crítico
Prototipo:	Expediente de Proyecto (Tabla 2.11 Caso de Uso del Sistema: Visualizar_Trazas_Auditorias).

Tabla 12 Descripción del CU Generar_Reporte

Caso de Uso:	Generar_Reporte
Actores:	Gerente de datos (Inicia)
Propósito:	
Resumen:	El caso de uso comienza cuando el Gerente de datos decide realizar algún reporte para conocer el comportamiento de diferentes variables en determinados pacientes del ensayo, el mismo se puede realizar de dos formas por modelos o por variables independientes. Obteniéndose una tabla con el reporte generado. Que se puede imprimir o exportar a diferentes formatos.
Referencia:	R23, R24, R25, R26, R35

Precondiciones:	
Poscondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1.- El gerente de datos escoge la acción a realizar.	1.1.- El sistema ejecuta alguna de las siguientes acciones: a) Si escoge realizar reporte por modelo, ir a la sección: Reporte por Modelo. b) Si escoge realizar reporte por variables independientes, ir a la sección: Reporte por Variables independientes.
Sección: Reporte por modelo	
2.- El gerente de datos entra la cantidad de variables a recoger para iniciar el reporte por pacientes.	2.1.- El sistema muestra a partir de la cantidad de variables entradas una tabla para asignar las variables que se utilizarán en el reporte.
3.- El gerente de datos le asigna las variables por separado.	3.1.- El sistema guarda los valores entrados por el gerente y los agrega en la tabla de reportes.
4.- El gerente de datos selecciona la opción de filtrar los pacientes.	4.1.- El sistema muestra los criterios por los cuales se puede realizar el filtrado.
5.- El gerente de datos selecciona los criterios por los cuales realizará el filtrado.	5.1.- El sistema muestra el reporte generado para los pacientes y las variables seleccionadas. 5.2.- Se habilitan las opciones de imprimir y exportar el reporte.
6.- El gerente de datos decide imprimir, exportar o cerrar el reporte. Finalizando el caso de uso.	
Sección: Reporte por variables independientes	
2.- El gerente de datos escoge la visita y el modelo para iniciar el reporte por pacientes.	2.1.- El sistema muestra a partir de la visita y el modelo escogido una tabla con las variables asociadas al mismo, que se utilizarán en el reporte.

3.- El gerente de datos selecciona la opción de filtrar los pacientes.	3.1.- El sistema muestra los criterios por los cuales se puede realizar el filtrado.
4.- El gerente de datos selecciona los criterios por los cuales realizará el filtrado.	4.1.- El sistema muestra el reporte generado para los pacientes y las variables seleccionadas. 4.2.- Se habilitan las opciones de imprimir y exportar el reporte.
5.- El gerente de datos decide imprimir, exportar o cerrar el reporte. Finalizando el caso de uso.	
Prioridad:	Crítico
Prototipo:	Expediente de Proyecto (Tabla 2.14 Caso de Uso del Sistema: Generar_Reporte).

3 Trabajadores del Sistema

Tabla 13 Trabajadores del Sistema

Actor	Descripción
Coordinador_Investigación_Clínica (CIC)	Es el responsable de introducir en un sitio dado, en un ensayo dado, la información clínica al Cuaderno de recogida de datos electrónico.
Investigador_Principal (IP)	Es el responsable máximo de toda la información clínica recogida en su sitio de investigación y además es el encargado en el sistema de insertar y retirar su firma de los modelos terminados, puede modificar los datos de los modelos y responder las queries existe en el ensayo.
Investigador_Promotor (IProm)	Es un especialista responsable del ensayo clínico en el centro promotor y orienta la recogida de datos de un paciente, en el CRD, tiene acceso a visualizar, imprimir y salvar cualquier información relacionada con el ensayo.
Gerente_Datos (GD)	Es el responsable de generar reportes por pacientes sobre el comportamiento de diferentes variables, exportar los mismos para su utilización en análisis estadísticos.

4 Requisitos no Funcionales de la Aplicación

Los Requerimientos no Funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

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 iconos para indicar el estado de los elementos utilizados en el diseño. Además, los iconos contendrán funcionalidades específicas. También se hará uso de colores para especificar el estado de los modelos en el publicador.

RNF de Usabilidad Las personas que interactuarán con el software serán médicos, clínicos y especialistas de la salud ubicados en el CIM, CIGB, Instituto Finlay, CENCEC y todos los hospitales del país. La aplicación tendrá un ambiente sencillo y será fácil de manejar para los usuarios incluso aquellos que no han tenido mucha experiencia en el trabajo con computadoras o con sistemas informáticos. Se impartirá una preparación a los usuarios explicando como se realizará el trabajo con el software. El sistema contendrá un manual de usuario, que será usado como ayuda para el trabajo con la aplicación.

RNF de Soporte Una vez terminada la aplicación se instalará en el CIM para realizar pruebas piloto del software y pruebas de despliegue. Una vez aprobada la aplicación y lista para comenzar su ejecución se instalará en los centros del polo científico y hospitales donde se realice la conducción de Ensayos Clínicos. Cada cierto tiempo previsto por los administradores del sistema se realizará el mantenimiento del software. La capacidad de mantenimiento deberá ser adecuada, documentando cuidadosamente todas las actividades realizadas en el desarrollo de la aplicación informática. Se debe facilitar la posibilidad de actualización y cambios sobre la base de un diseño escalable y robusto.

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. Las contraseñas deberán tener más de 7 caracteres de longitud y tener una fortaleza media. Los usuarios estarán obligados a cambiar la contraseña cada 60 días como máximo. Cada usuario tendrá asignado uno o varios roles en el sistema. Cada rol definido tendrá niveles de acceso al Software. Solo podrán acceder a la aplicación, clientes a través de direcciones IP específicas bien controladas. Todo cambio o modificación en el sistema debe ser atribuible a un usuario particular según su autenticación. Paralelo a la base de datos primaria se debe mantener una base de datos que registre todas las modificaciones hechas a la base de datos original, ordenadas cronológicamente y con la especificación del usuario responsable de dicha modificación, de manera que siempre se realicen trazas a la información manejada. Se debe garantizar comunicaciones seguras entre los clientes y el servidor, encriptando todo el tráfico de información usando llaves negociadas, algoritmos y protocolos.

RNF de Software Para la instalación de la aplicación se debe disponer del sistema operativo Windows o GNU Linux. En las computadoras de los clientes también deberán existir las mismas restricciones de los Sistemas Operativos incluyendo un navegador asociado al sistema operativo finalmente escogido para la visualización de las interfaces Web.

RNF de Hardware Para el funcionamiento de la aplicación son imprescindibles un navegador y conectividad. El servidor web debe tener alta disponibilidad y un rendimiento adecuado, garantizado por al menos un procesador Dual Intel Xeon 3 GHz o similar y RAM suficiente (4 GB a 8 GB). Los servidores de almacenamiento de datos deben tener de 1 a 3 TB disponibles pues el volumen de información es bastante grande y perdura en el tiempo hasta 15 años.

RNF de Restricciones en el Diseño y la Implementación El análisis y diseño de la aplicación será basado en la Metodología RUP haciendo uso del lenguaje de modelado UML. Se usará como herramienta CASE Rational Rose de la Suite 2003 para el modelado de los artefactos que se generan en cada uno de los flujos de trabajo definidos por RUP. Para el diseño de las interfaces se utilizará Dreamweaver 8 del paquete Macromedia. Se

usará como lenguaje de programación PHP. Se usará como Gestor de Base de Datos Postgre-SQL. Podrán ser utilizados varios estándares como HTTP, HTML, XML, SOAP, UDDI.

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

RNF de Disponibilidad Se garantizará que la aplicación se mantenga funcionando las 24 horas del día y los siete días de la semana con el menor tiempo posible de recuperación de fallos. El servidor de aplicación debe soportar un aumento de usuarios concurrentes por minuto de 1 a 400.

Glosario

B

Biomoléculas: son las moléculas constituyentes de los seres vivos formadas por sólo cuatro elementos que son el hidrógeno, oxígeno, carbono, y nitrógeno, que representan el 99 % de los átomos de los seres vivos.

C

CASE: (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computación).

Centro Promotor: Centro que promueve la realización de un ensayo clínico.

CIM: Centro de Inmunología Molecular.

Conectado: Muestra si el usuario esta logueado en el sistema.

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

Cronograma: Constituye la distribución de los modelos que serán llenados al paciente en las diferentes visitas programadas.

D

Datos Primarios: Son todos aquellos documentos de los cuales se obtiene información que se registra en la Historia Clínica del paciente. Ej. Rayos X, pruebas de laboratorio, entre otros.

E

EC: Ensayo Clínico.

Especificación del Sitio: Comentario que caracteriza al Sitio.

Estudio Multicéntrico: Estudio que se efectúa en varios centros de investigación y que emplea un mismo protocolo. Los estudios multicéntricos permiten obtener acceso a un mayor número de pacientes y, por consiguiente, ofrecen la ventaja de poder efectuarse más rápidamente. Sin embargo, el tamaño de la muestra (número de sujetos) requerida para garantizar una misma potencia de la prueba es mayor que cuando el estudio se realiza en un solo lugar puesto que en un estudio multicéntrico existen fuentes adicionales de variación (las posibilidades de sesgo son mayores debido, por ejemplo, a posibles diferencias en las siguientes áreas:

GLOSARIO

población de pacientes que se evalúa en cada centro, formación de investigadores y su cultura médica, valores reconocidos como normales en cada centro para las pruebas de laboratorio, etc.

Estudios pre-clínicos: Investigación que se realiza utilizando animales para descubrir si un medicamento, un procedimiento o un tratamiento tiene posibilidades de ser útil. Los estudios pre-clínicos se llevan a cabo antes de realizar cualquier prueba en seres humanos.

F

Fármaco: Producto químico empleado en el tratamiento o prevención de enfermedades. Pueden elaborarse a partir de plantas minerales, animales o mediante síntesis.

Farmacología: Ciencia que estudia la acción y distribución de los fármacos en el cuerpo humano.

G

GTK: Es un grupo importante de bibliotecas o rutinas para desarrollar interfaces gráficas de usuario (GUI) para principalmente los entornos gráficos GNOME, XFCE y ROX de sistemas Linux. GTK+ es la abreviatura de GIMP toolkit (conjunto de rutinas para GIMP).

Grupo IP: Diferentes números de IP que poseen las maquinas en las que puede desempeñar un usuario la asignación hecha.

H

Herramientas CASE: 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.

HTML: Lenguaje de marcado de hipertexto.

HTTP: Protocolo de transferencia de hipertexto.

I

Inconsistencias: Calificación que se le atribuye a una variable que no cumple con las reglas de validación.

IP: protocolo de internet

L

GLOSARIO

Librería: Son un conjunto de subprogramas utilizados para desarrollar software. Contienen código "ayuda" y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de estos

Licencia LGPL: Es una licencia de software creada por la Free Software Foundation

M

Modelo: Grupo de variables que responden a un determinado examen o evaluación.

P

Polo Científico: Conjunto de centros científicos cubanos que conforman una red investigativa. Su papel fundamental es contribuir al desarrollo social, particularmente en esferas como la salud pública y la agricultura, así como lograr la competitividad necesaria que le permita consolidarse como fuente de ingresos para el país.

Protocolo: Documento rector de un Ensayo Clínico, que contiene toda la documentación referida a dicho ensayo (reglas, pasos a seguir) y adjunto el cuaderno de recogida de dato.

Q

Queries: Constituyen una especie de consulta entre el Monitor y el Coordinador de la investigación clínica o el Investigador principal, su objetivo es comunicar y establecer un debate sobre alguna deficiencia encontrada relacionada con los pacientes, los modelos o el ensayo en general.

R

Rol: Privilegio que posee un usuario dentro de un Ensayo Clínico.

S

Sistema inmune: Red de defensa del cuerpo contra la infección y las partículas extrañas.

Sitio: Lugar específico o departamento dentro del Hospital donde se realiza el Ensayo Clínico. En un sitio solo se realiza un Ensayo Clínico. Un Ensayo Clínico puede realizarse en varios sitios.

Sitio de investigación clínica: Sitio donde será llevada a cabo la conducción de un ensayo clínico.

Sub-modelo: Constituye un subconjunto del modelo.

GLOSARIO

T

Trazas de auditorias: La trazabilidad del sistema constituye un registro que permite durante la manipulación de los datos conocer cada una de las acciones que se realizan sobre un ensayo clínico.

V

Variable: Cada uno de los datos que se recogen en el CRD.

X

XML: Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Permite definir la gramática de lenguajes específicos. No es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.