



Universidad de las Ciencias Informáticas

Facultad 7

**Título: Implementación del módulo Registro
de Métodos Depuradores.**

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

Autores: Yocasta Rodríguez Reimon

Dunior Socarras Benitez.

Tutores: Ing. Renier Ricardo Figueredo

Ing. Yanersy Díaz Colomé.

Asesor: Jorge Carlos Yero.

Ciudad de La Habana, Junio del 2008

“Año 50 de la Revolución”

Declaración de Autoría

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

Para que así conste firmamos la presente a los 26 días del mes de junio del año 2008

Yocasta Rodríguez Reimon

Ing. Renier Ricardo Figueredo

Firma de la Autora

Firma del Tutor

Dunior Socarras Benitez.

Ing. Yanersy Díaz Colomé.

Firma del Autor

Firma de la Tutora

"no hay más goces reales en el mundo que el amor fiel de la casa, y la amistad en los pocos hombres buenos."

José Martí

Agradecimientos

A la Revolución, ese proceso enorme que ha engendrado tantos sueños y ha hecho el nuestro realidad.

Agradecemos a todas las personas que nos han brindado su apoyo incondicional, a nuestros amigos en especial a Liudmila, que tanto nos ayudaste, no te olvidaremos nunca.

A nuestros tutores, Reinier y Yanerci, por apoyarnos y haber depositado toda su confianza en nosotros, para poder realizar esta tarea.

A nuestros profesores, por contribuir a nuestra formación profesional a lo largo de estos años.

A la profesora Lourdes por su comprensión, ayuda, y por todo el tiempo dedicado.

A nuestras familias por su cariño, dedicación y preocuparse por nosotros siempre.

A la profesora Pura, por su gran ayuda.

A los desconocidos sin nombres que nos hicieron todo más fácil

A todos ¡Muchas Gracias!

De Yocasta

A mis amigas, Nayla, Yeiki, Marilenis, Danaytis, y Odelkis por todos los momentos buenos y malos, y sobre todo por haber estado conmigo cuando más lo necesitaba. Gracias nunca las olvidare.

A Serguey por haber compartido conmigo, y apoyado en estos últimos años. Gracias por todo cielo.

A Juan Carlos por ayudarme siempre y haber sido tan especial.

De Dunior

A mis amigos Omar, Jorge Carlos, Daybert, Serguey, Runer, Enrique, Josue, Oslie, Maylin, Malena y Niurka por haber compartido conmigo estos cinco maravillosos años. Tengan por seguro que siempre los recordaré.

A mis compañeros de aula por hacerme mucho más amenas las clases.

A mi primo Rubiel y mi primita Rosana, por brindarme siempre su apoyo.

A todos ¡Muchas Gracias!

Dedicatoria

De Yocasta

A mis padres, no solo por traerme a la vida que en ocasiones suele ser sencillo, sino por ayudarme a transitar por ella, por brindarme sus manos cuando el camino era inseguro, por darme la luz cuando la oscuridad era intensa, a ellos antes que nadie por hacerme el ser humano que hoy soy, a mi mama especialmente por ser mi otro yo.

A mi hermana, que segura estoy se sentiría contenta de ver este nuevo paso que doy en la vida.

A mi amiga, Liudmila, por aguantarme todo este tiempo, y por estar conmigo cuando lo necesitaba.

A mi compañero de tesis, y amigo, Duniar, por apoyarme, y ayudarme a convertir mi sueño en realidad.

Te quiero Duni

A Fidel ese gigante que hoy ve su sueño convertido en realidad.

A los que ya no están conmigo, o no existen en mi vida pero en su momento, me hicieron feliz, me apoyaron y me convirtieron en la persona que ahora soy.

De Duniar

A mis padres, por todo el amor y la comprensión que me han dado, por estar a mi lado en todo momento y darme lo mejor de ellos, por haber sido mi guía en la vida; porque sin ellos no sería quien soy.

A mi abuelita y mis abuelitos, por haberme hecho el hombre que hoy soy, por su amor y dedicación hacia mí.

A mi hermanito, por todo el cariño que siento por él.

A mi novia y compañera Yaumara, por todo el amor y la confianza que me ha dado. Gracias mi amor.

A Yocasta, mi compañera de tesis y amiga, por toda la ayuda y el trabajo realizado, por aguantarme y darme apoyo en todo momento.

A mi amiga y compañera Liudmila, por ser tan especial y por brindarme siempre todo su apoyo.

A la Revolución y al Comandante en Jefe Fidel Castro cuyo sueño materializado permitió hacer realidad el mío y estudiar en esta Universidad tan valiosa.

Resumen

Cuba realiza un esfuerzo en la atención a la Enfermedad Renal Crónica terminal con requerimientos de métodos depuradores. El presente trabajo se refiere a la implementación del módulo de registro de métodos depuradores, este se encarga de gestionar la información de los pacientes en terapia renal de remplazo que requieren de métodos depuradores.

Para el desarrollo del trabajo se tuvo en cuenta la arquitectura definida por el grupo de arquitectura MINSAP-MIC para el desarrollo de todas las aplicaciones que van a ser desplegadas en Infomed. Esta se basa en Componentes, sigue el patrón de aplicaciones en tres capas y está orientada a servicios. Además, utiliza el patrón Modelo-Vista-Controlador y el servidor Web Apache 2.0. Define como lenguaje de programación PHP en su versión 5; como gestor de bases de datos MySQL en la versión 5. Para la elaboración de la aplicación, se utiliza el IDE de programación Zend Studio y los framework de clases de PHP CodeIgniter y de Javascript YUI.

El aporte del presente trabajo es que facilita el proceso de gestión de la información relacionada con los pacientes en terapia renal de remplazo, con enfermedades renales en Hemodiálisis y Diálisis Peritoneal. Además, aumenta la capacidad organizativa de los Servicios de Diálisis del país y ayuda a mejorar el trabajo de los médicos; logrando que sea más eficiente y rápido, incrementando de esta forma la calidad de la asistencia médica.

Palabras Claves: Aplicación, Diálisis Peritoneal, Gestión de información, Hemodiálisis, Métodos Depuradores, Nefrología, Sistema informático, Terapia Renal de Remplazo.

TABLA DE CONTENIDOS.

Introducción	1
Capítulo I: Fundamentación Teórica	5
1.1 Objeto de estudio. Descripción general	5
1.2 Objetivos que se propone el trabajo.	6
1.3 Tendencias y tecnologías actuales.....	6
1.3.1 ¿Qué es Internet?.....	6
1.3.2 Aplicaciones Web. Ventajas y desventajas.	7
1.3.3 Lenguajes de programación Web.	8
1.3.4 Sistemas Gestores de Bases de Datos (SGBD).	12
1.3.5 IDE de programación.....	17
1.3.6 XML / WebServices.....	19
1.3.7 Framework.....	22
1.4. Tecnología y herramientas a utilizar.....	24
Capítulo 2. Elementos de arquitectura.	26
2.1 Requisitos no funcionales del sistema propuesto.....	26
2.2 Estrategia de codificación, estándares y estilos de código a utilizar.	28
2.3 Patrones o estilos arquitectónicos presentes en la solución. Justificación de su uso y características.	33
2.4 Vista de despliegue.	40
2.5 Vista de implementación.....	41
Capítulo 3. Descripción y análisis de la solución propuesta.	44
3.1 Estrategia de integración de la solución con otros módulos o sistemas.	44
3.2 Descripción de las clases u operaciones necesarias.....	45
3.3.1 Diagrama Entidad Relación de la BD	67
3.3.2 Descripción de las tablas de la BD	67
Conclusiones	71
Recomendaciones	72
Referencias Bibliográficas	73
Bibliografía	79
Anexos	88

LISTA DE TABLAS

Tabla 2.1 Estrategia y estándares de codificación.	33
Tabla 3.1 Descripción de la clase E_PacienteRMD, CU Gestionar Movimiento.	45
Tabla 3.2 Descripción de la clase M_PacienteRMD, CU Gestionar Movimiento.....	47
Tabla 3.3 Descripción de la clase C_PacienteRMD, CU Gestionar Movimiento.....	48
Tabla 3.4 Descripción de la clase E_Paciente_Metodo, CU Gestionar Movimiento.	49
Tabla 3.5 Descripción de la clase M_Paciente_Metodo, CU Gestionar Movimiento.	50
Tabla 3.6 Descripción de la clase E_PacienteRMDHistorico, CU Gestionar Movimiento.	51
Tabla 3.7 Descripción de la clase M_PacienteAccesoVascular, CU Gestionar Movimiento.	53
Tabla 3.8 Descripción de la clase E_Datos_Iniciales, CU Gestionar Movimiento.	54
Tabla 3.9 Descripción de la clase E_PacienteAccesoVascular, CU Gestionar Acceso Vascular.....	56
Tabla 3.10 Descripción de la clase C_PacienteAccesoVascular, CU Gestionar Acceso Vascular.....	58
Tabla 3.11 Descripción de la clase M_PacienteAccesoVascular, CU Gestionar Acceso Vascular.	60
Tabla 3.12 Descripción de la clase E_PacienteAccesoHist, CU Gestionar Acceso Vascular.	61
Tabla 3.13 Descripción de la clase C_Metodo, CU Método.....	62
Tabla 3.14 Descripción de la clase M_Metodo, CU Método.	62
Tabla 3.15 Descripción de la clase E_Metodo, CU Método.....	63
Tabla 3.16 Descripción de la clase E_Entidad, CU Método.....	64
Tabla 3.17 Descripción de la clase C_Acceso_Vascular, CU Acceso Vascular.....	65
Tabla 3.18 Descripción de la clase E_Acceso_Vascular, CU Acceso Vascular.....	65
Tabla 3.19 Descripción de la clase M_Acceso_Vascular, CU Acceso Vascular	66
Tabla 3.20 Descripción de la clase E_Entidad, CU Acceso Vascular	67
Tabla 3.21 Descripción de la Tabla e_pacientermd.....	68
Tabla 3.22 Descripción de la Tabla e_metodocfg.....	68
Tabla 3.23 Descripción de la Tabla er_paciente_metodormd.....	68
Tabla 3.24 Descripción de la Tabla er_paciente_metodo_historicormd.....	69
Tabla 3.25 Descripción de la Tabla e_accesovascularcfg	69
Tabla 3.26 Descripción de la Tabla er_paciente_accesovascularrmd	70

LISTA DE FIGURAS

Fig. 2.1 Diagrama de la Vista de Despliegue.	40
Fig. 2.2 Diagrama de paquetes.	41
Fig.2.3 Diagrama de Componentes de las controladoras.	41
Fig.2.4 Diagrama de Componente de las Vista	42
Fig. 2.5 Diagrama de componentes de las Modelos.....	43
Fig. A3. 1 – Diagrama Entidad Relación del CU Gestionar Movimiento.	95
Fig. A3. 2 – Diagrama Entidad Relación del CU Método Dialítico.	96
Fig. A3. 3 – Diagrama Entidad Relación del CU Gestionar Acceso Vascular.	97
Fig. A3. 4 – Diagrama Entidad Relación del CU Gestionar Indicaciones.....	98
Fig. A3. 5 – Diagrama Entidad Relación del CU Gestionar Datos de la Diálisis.....	99

Introducción

En Cuba, se han alcanzado importantes logros y beneficios sociales, entre ellos: un alto nivel de desarrollo en la educación, la salud, la cultura y el empleo. Desde el triunfo de la Revolución, se han dado grandes pasos para mejorar la calidad de vida de la población cubana. Se hacen grandes esfuerzos para el desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC), dirigidas principalmente a los sectores de educación y salud. Las TIC se han convertido en un elemento fundamental de las estrategias cubanas, para el avance socioeconómico del país. La estrategia de informatización, es expresión del proceso revolucionario cubano, tiene al ciudadano en el centro de sus objetivos, buscando elevar su calidad de vida.

Esta tarea es preocupación de la dirección del país, y se lleva a cabo, casi desde el inicio de la Revolución, cuando se utilizaron algunas técnicas de computación en el área de la salud, pero no fue hasta finales de los 60 y principios del 1970 que se amplía su uso, y comienza a generalizarse por todo el país.

Uno de los mayores retos en la salud cubana es lograr equidad y calidad en los servicios que se prestan y para ello, es importante, alcanzar un equilibrio entre la medicina social, la atención primaria de la salud y el uso de las tecnologías médicas.

Actualmente, a través de la red INFOMED están informatizados los centros de información de atención primaria, se ha estado trabajando en el establecimiento de las redes nacionales de Bancos de Sangre, Nefrología, e Imágenes Médicas. Los avances en esta tecnología y el desarrollo de redes de información permiten una reducción de costos y aumentan la calidad del cuidado de los pacientes.

Las TIC influyen significativamente en la enseñanza de las ciencias médicas. Lo que permite tener en la computadora, una importante herramienta de trabajo. Además de, posibilitar cambios en la forma de actuación médica, aumentando la base de los conocimientos, el uso de los procesos analíticos y una mayor eficacia en la toma de decisiones.

Con el uso de la informática, el médico podrá ser liberado de tareas repetitivas y tediosas, pasará a ser el eje sobre el que se apoye todo el soporte informático, al actuar como selector y controlador de los datos de entrada, creador de los protocolos y algoritmos que éste seguirá, interpretando, además, sus resultados y por lo tanto siendo el máximo responsable del sistema.

Cuba posee adelantos en el terreno de la Nefrología. La atención a los pacientes con Enfermedad Renal Crónicas que poseen requerimientos dialíticos se hace cada vez más importantes, se han

creados hasta estos momentos, nuevas unidades de diálisis, y se han adquiridos modernas tecnologías, que pueden ayudar a mejorar la vida de estos pacientes [1]

En 1961, se realiza la primera hemodiálisis en un enfermo portador de un fallo renal agudo. En 1966, se crea el Instituto de Nefrología, cuna de la especialidad, centro de referencia, coordinador de la actividad y asesor para el MINSAP. En este se integró, desde sus inicios, numerosas especialidades en un equipo multidisciplinario, dedicado a la atención integral del enfermo renal. [2]

En 1968, se inició el tratamiento por diálisis, un año después se extendía a 4 territorios del país. En 1970, se realizó el primer Programa Nacional de Nefrología para integrar el manejo de los pacientes, con los objetivos de diagnosticar, tratar científicamente, prevenir en lo posible el desarrollo de enfermedades renales, desarrollar las investigaciones, la docencia y preparar a los pacientes para un posible trasplante renal, con vistas a una completa rehabilitación y reintegración a la vida social. En Cuba, se realizó el primer trasplante renal exitoso el 24 de febrero 1970. En 1970, se realizaría el primer trabajo de Informática aplicado a la Medicina en Cuba y fue en el campo de la Nefrología, relacionado con el estudio de la computación en el campo del trasplante renal. [3]

La práctica de la diálisis, el trasplante renal y el papel rector del Instituto, requirió desde sus inicios, el desarrollo de una comunicación permanente de los Centros que se fueron creando. Comenzando en los principios de los 80, de modo esporádico, la distribución nacional de los órganos para el trasplante renal.

Estos trasplantes están basados en los principios de la compatibilidad biológica a partir del Sistema Histocompatibilidad Humano (HLA); teniendo como presupuesto “el mejor riñón donado para el mejor receptor del país”. En 1989 se crea el Centro Nacional Coordinador de Trasplante con un programa de selección automatizado para la selección de la pareja donante-receptor, cuyo fundamento se aplica actualmente. Ha permitido hasta hoy el intercambio de órganos entre todos los territorios y la posibilidad de acceder a la diálisis y el trasplante renal con máxima justicia y equidad a los miles de enfermos dializados y trasplantados. [4]

Durante estos años, se ha extendido y fortalecido el trabajo de las unidades nefrológicas abriendo nuevos Servicios con actividades territoriales, vinculados al Instituto. Luego del impacto, causado por el derrumbe del campo socialista y aún en los duros momentos del período de crisis subsiguiente; se trata de lograr el rescate y perfeccionamiento de la asistencia médica, por lo que en el 2003 se adopta por el Estado, la decisión de priorizar esta especialidad. Para lo cual se construyen nuevos servicios

de Nefrología, se rehabilitan totalmente otros centros, gran reto que se acompañó de un plan intensivo para la capacitación de los recursos humanos que los atenderían.

Cuba realiza un esfuerzo en la atención del problema de salud que representa la Enfermedad Renal Crónica terminal con requerimientos de métodos depuradores. La atención al paciente en Terapia Renal de Reemplazo (TRR) se realiza a nivel nacional, coordinada por la Dirección Nacional de Atención al Programa de Enfermedad Renal Crónica, Diálisis y Trasplante. Esta radica en el Instituto de Nefrología (INEF) donde existe un archivo de datos nacional. Este, se actualiza con la información que recibe mensualmente de los centros bases.

Diariamente se recogen de forma manual, los resultados de las diálisis realizadas a los pacientes, estas estadísticas son enviadas al INEF mensualmente por medio de correo electrónico o vía telefónica. Lo que provoca errores humanos o algunos inconvenientes al no obtenerse la información de estos pacientes en tiempo real, ni con la exactitud requerida. Además, la calidad de la gestión de la información aún no es efectiva, ni se puede realizar una comparación a nivel municipal, provincial y nacional.

Por todo esto, se ha hecho necesario mejorar el trabajo de los médicos, y que este se realice de una forma más eficiente y rápida, dirigida fundamentalmente a elevar el nivel de la atención y la calidad de vida de los pacientes.

De la situación anterior se tiene el siguiente **problema**: ¿Cómo facilitar el proceso de gestión de la información relacionada con los pacientes en terapia renal de reemplazo que requieren de métodos depuradores?

El **objeto de estudio** consiste en el proceso de gestión de la información de los pacientes en terapia renal de reemplazo en Cuba, de aquí se deriva que el **campo de acción** sea el proceso de gestión de la información de los pacientes en terapia renal de reemplazo que necesitan de la realización de métodos depuradores.

Para dar solución a la problemática antes mencionada se ha definido como **objetivo general** implementar un sistema informático que mejore el proceso de gestión de la información relacionada con los pacientes en terapia renal de reemplazo que requieren de métodos depuradores.

Las **tareas de la investigación**, para dar solución al objetivo planteado son:

- Realizar un análisis de las técnicas de programación, lenguaje, plataforma y librerías propuestas por el cliente y la universidad, así como de otras tecnologías más utilizadas en el desarrollo de aplicaciones.
- Obtener Modelo de Implementación y los artefactos necesarios que describan la Base de Datos.
- Realizar el diseño de la base de datos y la implementación del módulo.
- Brindar una interfaz gráfica orientada al usuario, agradable y fácil de operar.

Como **posible resultado** se espera obtener la implementación de las funcionalidades y diseño de la BD del módulo de Registro de Métodos Depuradores del sistema de la Red Nacional de Nefrología.

Lo que permitirá una mayor organización y eficiencia a la hora de atender a los pacientes en los centros de diálisis del país. Además, que los datos e informaciones necesarias para cada consulta, la descripción de cada hemodiálisis y diálisis peritoneal, se obtengan de forma más rápida y ordenada.

El presente trabajo consta de tres capítulos:

Capítulo I.- Fundamentos teóricos: Se realiza un análisis de las distintas tendencias, tecnologías y metodologías relacionadas con distintas técnicas de programación, así como de las plataformas de desarrollo que la soportan. Se realiza un estudio crítico y valorativo de dichas técnicas de programación, plataformas y librerías usadas para el desarrollo del módulo Registro de Métodos Depuradores.

Capítulo II.- Elementos de la arquitectura: Se realiza una argumentación de los requisitos no funcionales, un análisis de los Patrones o estilos arquitectónicos presentes en la propuesta de solución y se muestran las vistas de despliegue e implementación.

Capítulo III.- Descripción y análisis de la solución propuesta: Se realiza un análisis de la estrategia de integración con otros módulos, las descripciones de de la clases y métodos más importantes, así como la descripción de las tablas de la Base de Datos y el Diagrama de Entidad Relación correspondiente a dichas clases.

En los anexos, se encuentran tablas sobre las descripciones de las clases, de los Casos de Usos del módulo, el diagrama Entidad Relación de la Base de Datos y pantallas del sistema.

Capítulo I: Fundamentación Teórica

Este capítulo está dedicado a realizar un análisis detallado de las distintas técnicas de programación que existen a nivel internacional y nacional. Incluye las tendencias, tecnologías y metodologías relacionadas con dichas técnicas, así como de las plataformas de desarrollo que la soportan. Se hace estudio del lenguaje de programación, plataforma y librerías usadas para el desarrollo del módulo.

Además se puede observar el objetivo que se persigue con el trabajo, así como los objetivos de los métodos depuradores a los cuales está dirigido el módulo en específico.

1.1 Objeto de estudio. Descripción general

En el año 1966 se fundó el Instituto de Nefrología "Dr. Abelardo Buch López", centro para el desarrollo de la especialidad dedicado integralmente a la investigación, a la docencia y a la asistencia médica calificada de las enfermedades.

El centro está adjunto al Hospital General Universitario "Joaquín Albarrán". En el Instituto de Nefrología se encuentra el Centro Coordinador Nacional de Enfermedad Renal Crónica, Diálisis y Trasplante Renal.

En el país, existen actualmente 47 unidades de diálisis de las cuales 9 son además Centros de Trasplantes, en Santa Clara, Camagüey, Holguín, Santiago de Cuba y 5 en C. Habana, estas últimas incluyen Laboratorios de Tipaje Tisular, así como otros de apoyo a esta actividad.

La Insuficiencia Renal Crónica (IRC) corresponde a la situación clínica derivada de la pérdida de función renal permanente y con carácter progresivo a la que puede llegarse por múltiples etiologías, tanto de carácter congénito, como de carácter hereditario, como adquiridas. En su etapa terminal requiere tratamiento de sustitución renal por diálisis o trasplante renal.

El trasplante renal puede ser una de las mejores soluciones, pero no siempre sus resultados son positivos, los pacientes que inician el tratamiento para realizarse el trasplante en muchos casos pueden rechazar el trasplante, estando obligados a volver al programa de diálisis, o pueden morir. Para los especialistas en Nefrología, es muy importante evitar que el paciente llegue a la Insuficiencia Renal Crónica.

Actualmente se brinda atención a 2 363 pacientes en Métodos Dialíticos Sustitutivos, de ellos 92 están en Diálisis Peritoneal Intermitente (DPI), 7 en Diálisis Peritoneal Continua Ambulatoria (DPCA), los restantes 2264 en Hemodiálisis (HD).

Objetivos de los Métodos Depuradores.

Los Métodos depuradores son Hemodiálisis, y Diálisis Peritoneal. En las consultas cuando un paciente entra al Registro de Métodos Depurados, es cuando sus riñones están muy deteriorados y ya prácticamente no cumplen su función.

Los métodos depuradores o métodos dialíticos, es un procedimiento que se realiza para retirar los elementos tóxicos (impurezas o desechos) de la sangre cuando los riñones no pueden hacerlo.

Los pacientes que necesitan estos métodos, comienzan a dializarse con Hemodiálisis o Diálisis Peritoneal, el comportamiento será diferente a dependencia del tipo de tratamiento y método dialítico que se le asigne.

Para comenzar con estos métodos debe haberse realizado una vía de acceso, a través de la cual, se llevará la sangre al dializador para limpiarla de las sustancias que esta no puede eliminar por si sola.

1.2 Objetivos que se propone el trabajo.

El objetivo de este trabajo es la implementación completa del módulo de registro de métodos depuradores. Este se encarga de gestionar la información referente a los métodos depuradores, de Hemodiálisis y Diálisis Peritoneal. Con esto se debe mejorar el servicio que se brinda en los centros nefrológicos, de una forma más eficiente, rápida y que permita de igual manera mejorar y cuidar la vida de los pacientes con ERC.

1.3 Tendencias y tecnologías actuales.

1.3.1 ¿Qué es Internet?

Internet es una red de redes a escala mundial de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP. Permite comunicar, buscar y transferir información sin grandes requerimientos tecnológicos ni económicos relativos para el individuo. La World Wide Web ha influenciado mucho a la popularidad de internet, esta se dice que es la Telaraña Mundial o WWW. Esta permite desplegar gráficos y usar el mouse para "navegar" (visitar) los lugares en Internet. La WWW es solo uno de los muchos servicios que se brindan en la red internet, es decir la web emplea a esta como medio de transmisión. [5]

Internet funciona con la estrategia "Cliente/Servidor", lo que significa que en la Red hay ordenadores Servidores que dan una información concreta en el momento que se solicite, y por otro lado están los ordenadores que piden dicha información, los llamados Clientes. Hoy en día, los servicios más usados

en Internet son: Correo Electrónico, World Wide Web, FTP, Grupos de Noticias, IRC y Servicios de Telefonía.[6]

1.3.2 Aplicaciones Web. Ventajas y desventajas.

Una aplicación web es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero.

La facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Aunque muchas variaciones son posibles, una aplicación web está comúnmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web es la primera capa, un motor usando alguna tecnología web dinámica es la capa del medio, y una base de datos como última capa. El navegador web manda peticiones a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos generando una interfaz de usuario. [7]

Las aplicaciones Web ofrecen grandes **ventajas** que pueden ser aprovechadas. Entre las ventajas que se pueden mencionar están: [8]

- No requieren instalación, pues usan tecnología Web, lo cual permite el aprovechamiento de todas las características del Internet.
- Son fáciles de usar (No se requieren complicadas combinaciones de Hardware/Software para utilizar estas aplicaciones. Solo un computador con un buen navegador web.[9]
- Alta disponibilidad, ya que puede realizar consultas en cualquier parte del mundo donde tenga acceso a Internet y a cualquier hora.
- Una empresa puede migrar de sistema operativo o cambiar el Hardware libremente sin afectar el funcionamiento de las aplicaciones de servidor.
- Se facilita el trabajo a distancia. Se puede trabajar desde cualquier PC o computador portátil con conexión Internet.
- Actualizar o hacer cambios en el Software es sencillo y sin riesgos de incompatibilidades. Existe solo una versión en el servidor lo que implica que no hay que distribuirla entre los demás computadores. El proceso es rápido y limpio.
- Al funcionar en un navegador, se requiere un conocimiento básico de informática para utilizar una aplicación web.

Desventajas: [10]

- Acceso limitado, la necesidad de conexión permanente y rápida a Internet hacen que el acceso a estas aplicaciones no esté al alcance de todos.
- La interactividad no se produce en tiempo real, en las aplicaciones web cada acción del usuario conlleva un tiempo de espera excesivo hasta que se obtiene la reacción del sistema.
- Elementos de interacción muy limitados. En comparación con el software de escritorio, las posibilidades de interacción con el usuario que ofrecen las aplicaciones web (mediante formularios principalmente) son muy escasas.
- Diferencias de presentación entre plataformas y navegadores. La falta de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones.

1.3.3 Lenguajes de programación Web.

PHP.

PHP es un lenguaje de programación usado normalmente para la creación de páginas web dinámicas. Es un acrónimo recursivo que significa "PHP Hypertext Pre-processor" (inicialmente PHP Tools o Personal Home Page Tools). Es conocido como una tecnología de código abierto que resulta muy útil para diseñar de forma rápida y eficaz aplicaciones Web dirigidas a bases de datos. Es un potente lenguaje de secuencia de comandos diseñado específicamente para permitir a los programadores crear aplicaciones en la Web con distintas prestaciones de forma rápida. El mismo se integra a la perfección con MySQL y que resulta muy adecuada para aplicaciones dinámicas basadas en Internet.

[11]

Su interpretación y ejecución se da en el servidor en el cual se encuentra almacenada la página y el cliente solo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página web, enriquecida con código PHP, el servidor interpretará las instrucciones mezcladas en el cuerpo de la página y las sustituirá con el resultado de la ejecución antes de enviar el resultado a la computadora del cliente. Además es posible utilizarlo para generar archivos PDF, Flash o JPG, entre otros. Permite la conexión a numerosas bases de datos de forma nativa tales como MySQL, Postgres, Oracle, ODBC, IBM DB2, Microsoft SQL Server y SQLite, lo cual permite la creación de Aplicaciones web muy robustas. [12]

PHP tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX, Linux, Windows y Mac OS X, y puede interactuar con los servidores de web más populares. El modelo PHP puede ser visto como una alternativa al sistema de Microsoft que utiliza ASP.NET/C#/VB.NET, a

ColdFusion de la compañía Macromedia, a JSP/Java de Sun Microsystems, y al famoso CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un compilador comercial denominado Zend Optimizer. [13]

Ventajas: [14]

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones.

Desventajas: [15]

- No posee una abstracción de base de datos estándar, sino bibliotecas especializadas.
- Por sus características promueve la creación de código desordenado y complejo de mantener.
- Todo el trabajo lo realiza el servidor. Por tanto puede ser más ineficiente a medida que aumenten las solicitudes.
- La orientación a objetos es aún muy deficiente para aplicaciones grandes.

ASP

Active Server Pages (ASP, Active Server Pages) es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). Esta tecnología está estrechamente relacionada con el modelo tecnológico de su fabricante. Intenta ser una solución para el modelo de programación rápida ya que programar en

ASP es como programar en VisualBasic, por supuesto con muchas limitaciones ya que es una plataforma que no se ha desarrollado como lo esperaba Microsoft. [16]

Traduciendo la definición de Microsoft: "Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para la Web". [17]

Lo interesante de este modelo tecnológico, es poder utilizar diversos componentes ya desarrollados como algunos controles ActiveX. Otros problemas que han hecho evolucionar esta tecnología es el no disponer de información "que oriente a quienes desean aprenderla y resulta muy costosa en tiempo descubrir aquí y allá toda la información para volverla altamente útil". ASP ha pasado por cuatro iteraciones mayores, ASP 1.0 (distribuido con IIS 3.0), ASP 2.0 (distribuido con IIS 4.0), ASP 3.0 (distribuido con IIS 5.0) y ASP.NET (parte de la plataforma .NET de Microsoft). Las versiones pre-.NET se denominan actualmente (desde 2002) como ASP clásico. [18]

En el último ASP clásico, ASP 3.0, hay seis objetos integrados disponibles para el programador, Application, ASPError, Request, Response, Server y Session. Cada objeto tiene un grupo de funcionalidades frecuentemente usadas y útiles para crear páginas web dinámicas. [19]

PERL

Perl es un lenguaje de propósito general originalmente desarrollado para la manipulación de texto y que ahora es utilizado para un amplio rango de tareas incluyendo administración de sistemas, desarrollo web, programación en red, desarrollo de GUI y más.[20]

Es un lenguaje interpretado que tiene varias utilidades, pero está principalmente orientado a la búsqueda, extracción y formateado de ficheros de tipo texto. También es muy usado para manejo y gestión de procesos (estado de procesos, conteo y extracción de parámetros característicos, entre otros) [21]

Sus principales características son que es fácil de usar, soporta tanto la programación estructurada como la programación orientada a objetos y la programación funcional, tiene incorporado un poderoso sistema de procesamiento de texto y una enorme colección de módulos disponibles. La estructura completa de Perl deriva ampliamente del lenguaje C. Es un lenguaje imperativo, con variables, expresiones, asignaciones, bloques de código delimitados por llaves, estructuras de control y subrutinas. Es gratuito, libre y aunque se desarrollo originalmente en un entorno UNIX. Actualmente existen en casi todos los sistemas operativos: Windows, MacOS, de manera que es un lenguaje Multiplataforma. [22]

Algunas de las **ventajas** del uso del lenguaje PERL son las siguientes: [23]

- Construcción de pequeños programas que pueden ser usados como filtros para obtener información de ficheros y realizar búsquedas.
- Se puede utilizar en varios entornos, como puede ser Windows 95, OS/2, sin realizar cambios de código, siendo únicamente necesario la introducción del interprete PERL correspondiente a cada sistema operativo.
- También es uno de los lenguajes mas utilizados en la programación de CGI scripts, que son guiones o scripts que utilizan el interface CGI (Common Gateway Interface), para intercambio de información entre aplicaciones externas y servicios de información.
- El mantenimiento y depuración de un programa en PERL es mucho más sencillo que la de cualquier programa en C.

JSP.

JavaServer Pages (JSP) es la tecnología para generar páginas web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje Java. La tecnología JSP, o de JavaServer Pages, es una tecnología Java que permite a los programadores generar dinámicamente HTML, XML o algún otro tipo de página web. Esta tecnología permite al código Java y a algunas acciones predefinidas ser embebidas en el contenido estático. La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento HTML en el archivo jsp. [24]

Además, se caracteriza por ser un lenguaje que puede ejecutarse en cualquier sistema, lo que sumado a JSP le da mucha versatilidad. Sin embargo, no se puede considerar un script al 100% ya que antes de ejecutarse el servidor web compila el script y genera un servlet, por lo tanto se puede decir que aunque este proceso sea transparente para el programador no deja de ser una aplicación compilada. La ventaja de esto es algo más de rapidez y disponer del API de Java en su totalidad. [25]

Javascript.

Javascript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al contrario que Java, este no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia,

es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.[26]

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Todos los navegadores interpretan el código Javascript integrado dentro de las páginas web. La ventaja que presenta sobre el HTML es que permite crear páginas Web más dinámicas, lo que las hace más atractivas para el usuario [27]

Lenguaje de programación escogido para desarrollar el trabajo

Luego de hacer el análisis entre las distintas técnicas de programación, se decide utilizar el PHP por las siguientes características:

- Está soportado en la mayoría de las plataformas de Sistemas Operativos, mientras que por ejemplo ASP por ser propiedad de Microsoft no es multiplataforma.
- El PHP no tiene costo oculto, o sea que cuando se adquiere incluye un sin número de bibliotecas que proporcionan el soporte para la mayoría de las aplicaciones Web, por ejemplo e-mail, generación de ficheros PDF y otros. En caso de que no se tengan las bibliotecas estas se pueden encontrar gratis en Internet. En el caso de ASP forma parte del Internet Information Server que viene integrado en Windows NT-2000 Server con su elevado costo de adquisición.
- Algunos son parecidos en cuanto a la forma de utilización, pero PHP es más rápido, gratuito y multiplataforma.

Por todas estas características y ventajas previamente descritas, por políticas del Área Temática donde el trabajo está inmerso, y además porque el sistema se desplegará en los servidores de INFOMED, para la cual existe una arquitectura definida que incluye la utilización del lenguaje programación PHP para el desarrollo de las aplicaciones web, es por lo que se escoge dicho lenguaje.

1.3.4 Sistemas Gestores de Bases de Datos (SGBD).

Un SGBD debe proporcionar a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos, esta es la función fundamental y por supuesto, debe ocultar al usuario la estructura física interna (la organización de los ficheros y las estructuras de almacenamiento).

Además debe proporcionar un catálogo en el que se almacenen las descripciones de los datos y que sea accesible por los usuarios. Debe proporcionar un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción, proporcionar un mecanismo que asegure que la base de datos se actualice correctamente cuando varios usuarios la están actualizando

concurrentemente. Uno de los principales objetivos es permitir que varios usuarios tengan acceso concurrente a los datos que comparten. Además debe proporcionar un mecanismo capaz de recuperar la base de datos en caso de que ocurra algún suceso que la dañe y un mecanismo que garantice que sólo los usuarios autorizados pueden acceder a la base de datos. [28]

Resumiendo todo SGBD debe cumplir con los siguientes objetivos: [29]

- Abstracción de la información.
- Redundancia mínima.
- Consistencia.
- Seguridad.
- Integridad.
- Tiempo de respuesta mínimo.

PostgreSQL.

PostgreSQL es un servidor de base de datos relacional orientada a objetos de software libre. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. [30]

Algunas de sus principales características son la alta concurrencia. Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último que se actualizo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos. [31]

PostgreSQL provee nativamente soporte para: [32]

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.

- Direcciones MAC.
- Arrays.
- Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL.

Otras características: [33]

- Claves ajenas también denominadas Llaves ajenas o Foráneas (foreign keys).
- Disparadores (triggers).

Un disparador o trigger se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características, el nombre del trigger o disparador, el momento en que el disparador debe arrancar, el evento del disparador que deberá activarse, la tabla donde el disparador se activara, la frecuencia de la ejecución y la función que podría ser llamada. Combinando estas seis características, PostgreSQL le permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores (triggers). [34]

- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.

SQL (Structure Query Language)

Es mucho mejor manejar datos desde bases de datos que escribir y leer datos para o desde archivos de texto. SQL es un lenguaje de consulta estructurado. [35]

Algunas de las características del SQL son: [36]

- Es una forma estándar de consulta de datos específicos.
- Es una forma de extraer y manipular datos de una base de datos.
- Usado para todas las funciones de bases de datos, incluyendo administración.
- Creación de esquemas y datos recuperables.
- Puede ser usado de forma implícita dentro de una aplicación.

- Existen SGDB que utilizan el SQL para realizar el tratamiento de los datos almacenado como son MySQL y SQL Server.

MySQL.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL, es también uno de los sistemas gestores de bases de datos más populares desarrollados bajo la filosofía de código abierto.[37]

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre.

Entre las características disponibles en las últimas versiones se puede destacar: [38]

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.
- MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo
- Por último, MySQL es rápido, estable, gratuito, y soporta múltiples lenguajes de programación.

Características de la versión 5.0: [39]

- Un amplio subconjunto de ANSI SQL 99, y varias extensiones.
- Soporte a multiplataforma
- Procedimientos almacenados
- Triggers
- Cursores

- Vistas actualizables
- Soporte a VARCHAR
- Esquema de Información
- Modo Strict
- Soporte X/Open XA de transacciones distribuidas; transacción en dos fases como parte de esto, utilizando el motor InnoDB de Oracle
- Motores de almacenamiento independientes (MyISAM para lecturas rápidas, InnoDB para transacciones e integridad referencial)
- Transacciones con los motores de almacenamiento InnoDB, BDB Y Cluster; puntos de recuperación(savepoints) con InnoDB
- Soporte para SSL
- Query caching
- Sub-SELECTs (o SELECTs anidados)
- Réplica con un maestro por esclavo, varios esclavos por maestro, sin soporte automático para múltiples maestros por esclavo.
- Indexing y buscando campos de texto completos usando el motor de almacenamiento MyISAM
- Librería de Base de Datos Embebida
- Soporte completo para Unicode
- Conforme a las reglas ACID usando los motores InnoDB, BDB y Cluster

Sistema Gestor de Base de Datos escogido para desarrollar el trabajo

Luego de analizadas las características y facilidades de los SGBD presentados, y las de la herramienta a desarrollar se decide usar el MySQL en su versión 5.0 como SGBD, por todas las características mostradas anteriormente; además por las siguientes razones: PHP lo maneja más fácil que a cualquier otro sistema debido a la gran cantidad de funciones que tiene explícitas. Es multiplataforma y hasta su versión 5.0 no tiene precio en el mercado, se adquiere libremente. [40]

Además de todas las ventajas que trae implícita como gestor de Base de Datos en sí, se definió por políticas del Área Temática, el uso de MySQL para el desarrollo de todas las aplicaciones médicas, debido que el sistema será desplegado en los servidores de INFOMED para la cual existe una arquitectura definida que rige la utilización de este Gestor.

1.3.5 IDE de programación

Un entorno de desarrollo integrado o en inglés Integrated Development Environment ('IDE'), está compuesto por un conjunto de herramientas para agilizar el proceso de desarrollo del programador. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Estos pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes. [41]

Además proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación. En algunos lenguajes, puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación. [42]

Los componentes que debe tener los IDEs son: [43]

- Un editor de texto.
- Un compilador.
- Un intérprete.
- Herramientas de automatización.
- Un depurador.
- Posibilidad de ofrecer un sistema de control de versiones.
- Factibilidad para ayudar en la construcción de interfaces gráficas de usuarios.

Zend Studio como IDE de programación

Editor web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código. Los expertos en PHP consideran a Zend Studio como el entorno IDE más maduro y con más características útiles. La última versión ofrece manipulación avanzada de bases de datos y otras mejoras. Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones web, en dicho lenguaje. [44]

El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más. [45]

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o en caso de que estén instalados, los configura para trabajar juntos en depuración. [46]

Notepad++

Notepad++ es un poderoso editor de código fuente que soporta diversos lenguajes de programación. Este interesante editor permite imprimir el código escrito con los diferentes colores de la sintaxis, posee la opción de auto completado con la posibilidad de definir una API personalizada, permite la edición de varios documentos al mismo tiempo, con sus correspondientes vistas separadas o integradas a la misma interfaz. Notepad++ posee soporte para búsqueda y reemplazo de expresiones regulares, detección automática del estado del texto, incluye una herramienta de Zoom, trabaja con puntos de marca para editar y desplazarse más dinámicamente y permite la creación de marcos con su correspondiente secuencia de teclas para acceso rápido. [47]

Con Notepad++ se puede programar en los siguientes lenguajes: C, C++, Java, C#, XML, HTML, PHP, CSS, makefile, ASCII art (.nfo), doxygen, ini file, batch file, Javascript, ASP, VB/VBS, SQL, Objective-C, RC resource file, Pascal, Perl, pitón, Lua, TeX, TCL, Assembler, Ruby, Lisp, Scheme, Properties, Diff, Smalltalk, Postscript, VHDL, Ada, Caml, Autolt, KiXtart, Matlab, Verilog, Haskell, InnoSetup y CMake. Notepad++ es un excelente editor para tener en cuenta. [48].

NuSphere

PHpEd es un editor para programadores con soporte para múltiples formatos, similar a otras aplicaciones como PHP. Facilita el trabajo de programación con numerosas características de gran utilidad entre las que destacan: [49]

- Completo sistema de ayuda
- Plantillas de documento y de fragmentos de código frecuentes
- Código de colores para comandos en PHP, Perl, Javascript, SQL, HTML y más.

Además, esta herramienta incluye un cliente de FTP y un servidor Web integrados, totalmente configurables según las necesidades de trabajo. NuSphere PhpED IDE es un ambiente profesional de desarrollo integrado mayormente usado para el desarrollo de aplicaciones web usando el lenguaje de

scripting PHP, aunque soporta otros lenguajes populares tales como HTML, XML, CSS, Perl, Javascript y Python. PhpED es un editor robusto de código con un poderoso depurador de código PHP. También incluye clientes de base de datos y CVS integrados, servicios SOAP, herramientas para Validar y formatear el código HTML, ayuda y mucho más. [50]

Las características principales de PhpED 4 incluyen:

- Un editor avanzado, eficiente y altamente configurable con soporte para la programación orientada a objetos. Resaltador de código, atajos definidos por el usuario, análisis instantáneo de sintaxis, plantillas de código y mucho más.
- Sofisticado depurador de PHP que puede funcionar localmente o de forma remota. El módulo de depuración para la última versión 5.0.4 de PHP se incluye en el paquete. [51]
- Amplio explorador de código que permite mostrar todas las clases, métodos, características, funciones y variables del php en detalle.
- Mejoras para la gestión y despliegue de los proyectos. Soporte para los protocolos FTPS (TLS/SSL), SFTP y WebDAV/HTTPS (SSL) haciendo el despliegue y la transferencia de los datos segura.
- Subsistema único de ayuda. Incluye manuales de PHP, HTML, MySQL, PostgreSQL y CSS muy bien diseñados.
- Clientes integrados de MySQL, MSSQL, Oracle y UltraSQL/PostgreSQL, permitiéndole al desarrollador conectarse con una base de datos directamente desde IDE, explorar las bases de datos, hacer consultas y trabajar con los datos de estas sin abandonar el IDE[52]
- Cliente integrado de CVS, permitiendo la revisión de cambios en viejas versiones de los archivos de código fuente para buscar errores mientras se trabaja en el mismo proyecto en un equipo de desarrolladores.
- Asistente de NuSOAP, el cual facilita la construcción de servicios web profesionales usando la librería de clases NuSoap.[53]

1.3.6 XML / WebServices.

Web Services

Un servicio web (en inglés Web Services) es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de

programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. Los Servicios Web necesitan diversas especificaciones y tecnologías, ahí la necesidad de apoyarse en los 4 principales estándares: XML, SOAP, UDDI y WSDL. [54]

Web Services Protocol Stack, se denomina así al conjunto de servicios y protocolos, de los servicios Web: [55]

XML (Extensible Markup Language): Es el formato estándar para los datos que se vayan a intercambiar.

SOAP (Simple Object Access Protocol) o XML-RPC (XML Remote Producer Call): Protocolos sobre los que se establece el intercambio.

WSDL (Web Services Description Languages): Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.

UDDI (Universal Description, Discovery and Integration): Protocolo para publicar la información de los servicios Web. Permite comprobar qué servicios web están disponibles.

WS-Security (Web Services Security): Protocolo de seguridad aceptado como estándar por OASIS (Organization for the Advancement of Structured Information Standards). Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados

Otros protocolos: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), o SMTP (Simple Mail Transfer Protocol).

XML

XML, sigla en inglés de Extensible Markup Language («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Este lenguaje es la base de los servicios Web. Por lo tanto no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Este no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar por ejemplo en bases de datos, editores de texto y hojas de cálculo. [56]

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. Se emplea principalmente para separar el contenido de la presentación, esta usa sus etiquetas sólo para delimitar piezas de datos, y la interpretación de los datos, la hace la aplicación que los lee. Es gratis, independiente de la plataforma y ampliamente distribuido. [57]

SOAP

SOAP (siglas de Simple Object Access Protocol/protocolo de acceso simple) es un protocolo estándar creado por Microsoft, IBM y otros, está actualmente bajo el auspicio de la W3C que define como dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML, por lo cual se dice que se utiliza para el intercambio de datos.[58]

WSDL

WSDL son las siglas de Web Services Description Language, describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. [59]

UDDI

UDDI son las siglas del catálogo de negocios de Internet denominado Universal Description, Discovery and Integration. El registro en el catálogo se hace en XML. Es una iniciativa industrial abierta entroncada en el contexto de los servicios Web. Además es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL. [60]

Apache

Apache es un potente y gratuito servidor web, que ofrece un servicio estable, sencillo de mantener y configurar. Es indiscutiblemente uno de los mayores logros del Software Libre. Este se demuestra más rápido que otros webserver free. Se desarrolla dentro del proyecto HTTP Server (http) de la Apache Software Foundation. [61]

Se destacan las siguientes características: [62]

- Es multiplataforma, aunque idealmente está preparado para funcionar bajo Linux.
- Muy sencillo de configurar.
- Es Open-source.

- Muy útil para proveedores de Servicios de Internet que requieran miles de sitios pequeños con páginas estáticas.
- Amplias librerías de PHP y Perl a disposición de los programadores.
- Posee diversos módulos que permiten incorporarle nuevas funcionalidades, muy simples de cargar.
- Es capaz de utilizar lenguajes como PHP, TCL y Python.

Apache presenta además entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. Apache tiene amplia aceptación en la red, desde 1996, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en los últimos años. La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. [63]

1.3.7 Framework

En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, librerías y un lenguaje de scripting, además de otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Este representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. [64]

Los frameworks simplifican el desarrollo de las aplicaciones mediante la automatización de muchas de las tareas comunes. Además proporciona estructura al código fuente, forzando al programador a crear código más legible y más fácil de mantener. [65]

Framework CodeIgniter

CodeIgniter es un Framework para desarrollo de Aplicaciones- un toolkit- para personas que quieran construir sitios usando PHP. Su objetivo es poder hacer los proyectos más rápidos de lo que usted puede hacerlos escribiendo código desde el principio, suministrando un rico conjunto de librerías para tareas comunes, con una simple interface y estructura lógica para acceder a estas librerías. CodeIgniter deja que su creatividad se centre en el proyecto minimizando la cantidad de código necesitado para ejecutar una tarea. [66]

Entre sus características se puede encontrar su compatibilidad con PHP 4 y PHP 5, está basado en el Modelo Vista Controlador, posee soporte para múltiples bases de datos entre las que se pueden mencionar PostgreSQL, MySQL, MSSQL, además de que se lo pueden incluir drivers para otros gestores de bases de datos que usted programe, incluye también plantillas, validaciones, no requiere instalación, es una librería con un gran número de clases. Una de las características más interesantes de CodeIgniter es el elevado número de clases que incluye para trabajar con distintos objetos: calendario, bases de datos, correo electrónico, manipulación de imágenes, FTP, lenguaje, tablas, sesiones, compresión ZIP, etcétera. [67]

A diferencia de otros frameworks, CodeIgniter cuenta con una documentación excelente que permite conocer todos los secretos de este entorno de trabajo. Está liberado bajo la licencia de código abierto Apache-BSD, lo que significa que es totalmente libre y puede ser usado. [68]

Framework Symfony

Symfony es un framework para construir aplicaciones web con PHP. En otras palabras, Symfony es un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones web. Este emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación web. El modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos. [69]

La vista transforma la información obtenida por el modelo en las páginas web a las que acceden los usuarios. El controlador es el encargado de coordinar todos los demás elementos y transformar las peticiones del usuario en operaciones sobre el modelo y la vista. 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. [70]

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. [71]

Yahoo user Interface

Yahoo user Interface (YUI) Library es un conjunto de utilidades y controles, escrito en JavaScript para desarrollar aplicaciones Web interactivas (RIA - Rich Internet Application) usando técnicas DOM, DHTML y AJAX. Este incluye también un conjunto de fuentes CSS. Todos los componentes YUI han sido desarrollados bajo Licencia BSD Open Source y están disponibles para todo tipo de uso. [72]

Características:

Están disponibles dos tipos de componentes diferentes: Utilidades y Controles: [73]

Las YUI Utilidades simplifican el desarrollo para la compatibilidad entre Navegadores basados en técnicas DOM, DHTML y AJAX. Los controles de YUI proporcionan elementos visuales altamente interactivos del diseño para sus aplicaciones Web. Estos elementos se crean y se manejan íntegramente del lado del cliente (usuario) y nunca requieren de una recarga de página.

Prototype JavaScript Framework

Prototype es un framework basado en JavaScript que se orienta al desarrollo sencillo y dinámico de aplicaciones Web. Presenta un conjunto único de herramientas fáciles de utilizar para el desarrollo orientado a clases y con la mejor librería de Ajax que se pueda encontrar, Prototype se está convirtiendo en la elección base para los desarrolladores de aplicaciones Web de todas partes. [74]

Características: [75]

- Implementa aplicaciones Ajax fácilmente.
- DOM Extendido: Agrega métodos de utilidad para los elementos devueltos con la función `$()`, por ejemplo puedes escribir `$('#comments').addClassName('active').show()` para obtener el elemento con ID 'comments' además de agregarle el estilo 'active' y mostrar el elemento con show (si el elemento estaba oculto previamente).
- Utiliza JSON (JavaScript Object Notation): Json es una alternativa rápida y más ligera que XML para las solicitudes de Ajax.

1.4. Tecnología y herramientas a utilizar.

El sistema basa su arquitectura en una ya elaborada por el grupo de arquitectura MINSAP-MIC para el desarrollo de todas las aplicaciones para infomed. Esta arquitectura está basada en componentes siguiendo el patrón de aplicaciones en tres capas y orientada a servicios. De la misma forma se definen varias de las herramientas que se deben utilizar, servidor Web Apache 2.0, el cual es libre de código abierto y bajo la licencia Apache/GPL cuya descripción y aplicaciones ya fue dada anteriormente.

Como lenguaje de programación esta arquitectura define PHP en su versión 5, como gestor de bases de datos se usará el ya conocido y libre MySQL en la versión 5. Para la elaboración de la aplicación, se usará el IDE de programación Zend Studio por las grandes prestaciones que este brinda. El sistema además está implementado usando el framework de clases de PHP CodeIgniter, y de Javascript YUI, se hace rehúso de todas las librerías y clases que estos dos potentes frameworks ofrecen.

Conclusiones

En este capítulo se han abordado los temas relacionados con el objeto de estudio y los objetivos del trabajo. Se profundiza sobre las tendencias y tecnologías actuales así como las herramientas a utilizar en el desarrollo de dicha aplicación, logrando así establecer las más adecuadas para el desarrollo de la aplicación.

Todo esto con el objetivo de profundizar en el conocimiento de estos temas y fundamentar la necesidad de dar una solución informática a la situación existente en los centros de diálisis del país para gestionar la información relacionada con los pacientes en terapia Renal de Reemplazo.

Capítulo 2. Elementos de arquitectura.

El siguiente capítulo pretende realizar una selección y argumentación de los requisitos no funcionales propuestos, se hace un análisis de los patrones o estilos arquitectónicos presentes en la propuesta de solución, justificando así su uso y características que presentan. Además se puede encontrar la vista de despliegue y de implementación.

2.1 Requisitos no funcionales del sistema propuesto.

Interfaz externa.

La interfaz de usuario será sencilla, amigable, intuitiva y de fácil navegación por el usuario, con el objetivo de evitar la resistencia humana al uso del nuevo sistema. Se seleccionará un esquema de colores a la vez atractivo. Existirá una paginación de reportes de búsqueda y listados. Presentará un diseño perfectamente encuadrado para resoluciones de 1024 x 768, pero preparado para verse en otras resoluciones.

Usabilidad.

La aplicación Web será flexible y de fácil aprendizaje, pues se trata en todo lo posible de mantener un estándar de operabilidad que logre que las interacciones del usuario con el sistema sean predecibles y familiares. La misma podrá ser usada por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general. Deberá visualizarse bien en los principales navegadores que existen en el mundo y estará disponible en todo momento.

Rendimiento.

Las pantallas estarán poco cargadas de imágenes para garantizar que el tiempo de ejecución de los hipervínculos, las adiciones, modificaciones y eliminaciones no excedan los 4 segundos y garantizar de esta manera una respuesta rápida del sistema.

Soporte.

El sistema contará con una ayuda para el usuario con la cual podrá aprender rápidamente a utilizar la aplicación Web y estará bien documentado para garantizar futuros mantenimientos.

Portabilidad.

El sistema podrá ejecutarse sobre plataforma Linux, Windows 98 o superior.

Seguridad.

La protección del sistema contra el acceso desautorizado y las modificaciones de información está garantizada por el Sistema de Autenticación, Autorización y Auditoría (SAAA).

Confiabilidad.

Todas las partes del diseño del sistema serán realmente aplicadas y se hará una transformación correcta del diseño en un lenguaje de programación. Además deberá prevenir los posibles fallos y/o errores que pudieran presentarse así como posibilitar una rápida recuperación en dichos casos.

Software

Para el cliente:

- Un navegador Web, recomendados: Mozilla 1.5, Internet Explorer 4.0 o superior.
- Sistema operativo Linux o Windows 98 ó Superior

Para el servidor:

- Sistema operativo Linux.
- Servidor Web Apache 2.0 y PHP 5.
- Framework CodeIgniter.
- Servidor de Base de Datos MySQL 5.1.

Hardware

PC clientes:

- Procesador Pentium III o superior.
- 128 de memoria RAM o superior.
- Monitor VGA o superior.
- Tarjeta de red.

Servidor:

- Procesador Pentium IV o superior.
- 512 de memoria RAM o superior.
- Disco Duro de 80 GB.

Diseño e implementación

Para el análisis y el diseño del sistema debe ser utilizado la metodología RUP, usando el lenguaje de modelación UML y como herramienta para llevarlo a cabo el Visual Paradigm, utilizando los patrones de diseño establecidos. La implementación debe ser en el lenguaje de programación php 5, utilizando Zend Studio.

2.2 Estrategia de codificación, estándares y estilos de código a utilizar.

Estándar de codificación para PHP

Idioma:

Se debe utilizar como idioma el español, las palabras no se acentuarán.

Palabras Reservadas:

Las palabras reservadas van en minúsculas sin excepción alguna.

Identación		
Objetivo: Lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento		
0 espacios en blanco desde la izquierda en	require include class	No se empleará ningún espacio en blanco desde la izquierda para las instrucciones antes mencionadas. Se tomará como inicio de la página el tag PHP <?
2 espacio en blanco desde la izquierda en	function define	Se dejarán dos espacios en blanco desde la izquierda en las instrucciones antes mencionadas.
2 espacio en blanco desde la referencia en	Inicio y fin de bloque	Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones If, else, For, While, Do While, Switch, Foreach.
Niveles de anidación	Hasta 5 niveles	Se recomienda emplear hasta 5 niveles de anidación en instrucciones If, For, While.

<p>Aspectos Generales</p>	<p>El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la PC o la configuración de dicha tecla.</p> <p>Los inicios ({) y cierre (}) de ámbito deber estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción.</p> <p>Nunca colocar { en la línea de un código cualquiera, esto requiere una línea propia.</p>
<p>Ejemplo de indentación</p> <pre> <? require ('class/Interface.php'); class MiClase { function BuscarUnidad(\$sTabla, \$sCampos, \$lIndice) { if (\$sTabla) { ... } for (...) { ... } } } ?> </pre>	
<p align="center">Comentarios, separadores, líneas, espacios en blanco y márgenes.</p>	
<p>Objetivo: Establecer un modo común para comentar el código de forma tal que sea comprensible con</p>	

sólo leerlo una vez.		
Ubicación de comentarios	Al inicio de cada clase o función y al final de cada bloque de código, si existe.	Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma.
Separador de instrucciones	Se emplea el punto y coma.	Se recomienda usar el separador al final de cada instrucción y no en la línea de abajo. Ejemplo: define ("CONSTANT", "value1");
Líneas en blanco	Se emplean antes y después de métodos, clases y estructuras.	Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura, y de la implementación de una función
Espacios en blanco	Entre operadores lógicos y aritméticos.	Se recomienda usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: \$sAcceso = \$sAccesoH if ((\$sAcceso) && (\$sLocalizacion))
Márgenes y líneas de continuidad	Sobre márgenes y líneas de continuación	Los márgenes de cada línea de código no deben exceder los 80 c 80 caracteres, pero puede exceptuarse si es para terminar la escritura de una palabra. Las líneas de continuación deben estar alineadas entre sí e indentadas respecto al paréntesis abierto. Las líneas de continuación nunca deben comenzar con un Operador binario.
Aspectos generales	Sobre el comentario	Se debe evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se

		escribe a continuación de la instrucción
	Sobre los espacios en blanco	No se debe usar espacio en blanco: Después del corchete abierto y antes del cerrado de un arreglo. Después del paréntesis abierto y antes del cerrado.
Variables y constantes		
Apariencia de variables	Las variables tendrán un prefijo para el tipo de datos en minúscula.	El nombre que se le da a las variables debe comenzar con la primera letra en minúscula, la cual identificara el tipo de datos al que se refiere en caso de que sea un nombre compuesto se pone junto y con mayúscula inicial el segundo nombre, y los restantes también. Ejemplo: \$sPacienteAcceso
Apariencia de constantes	Todas sus letras en mayúscula	Se recomienda declarar una constante por cada línea y con las asignaciones a las variables sucede lo mismo. Ejemplo: <pre>define("CONSTANT1","value1"); define("CONSTANT2","value2"); \$sAcceso = \$sAccesoH</pre>
Declaración de constantes y asignación a variables	Nombres de las variables y constantes	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.
Clases y Objeto		
Objetivo: Nombrar las clases e instancias de forma estándar para todas las aplicaciones		
Para nombrar clases	La primera letra con la inicial del tipo, y luego seguida de un (_)	Se usa C_ para las controladoras, M_ para los modelos V_ para los ficheros de las vistas.

Apariencia de clases y objetos	Primera letra en mayúscula	Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*. Para el caso de las instancias se comenzara con un prefijo que identificara el tipo de dato, este se escribirá en minúscula.
Apariencia de atributos	Primera letra en minúscula	El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, la cual estará en correspondencia al tipo de dato al que se refiere, en caso de que sea un nombre compuesto se empleará notación CamellCasing.
Apariencia de las funciones	Primera letra en mayúscula	Para nombrar las funciones se debe tratar de utilizar verbos que denoten la acción que hace la función. Se empleará notación PascalCasing. Ejemplo: function BuscarUnidad(). Si son funciones que obtienen un dato se emplea el prefijo get y si fijan algún valor se emplea el prefijo set
Declaración de parámetro en funciones	Agrupados por tipos Poner los string numéricos 2, además, agrupar según valores por defecto.	Los parámetros que se le pasan a las funciones se recomienda sean declarados de forma tal que estén agrupados por el tipo de dato que contienen, especificando el tipo de datos.
Aspectos generales	Sobre las clases, los objetos, los atributos y las funciones.	El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con sólo leerlo se conozca el propósito de los mismos.
Bases de Datos, Tablas, esquemas y Campos		
Apariencia de la BD	Las 2 primeras letras representan el tipo.	Los nombres de las Bases de Datos deben comenzar con el prefijo bd a continuación

		underscoard y luego el nombre comienza con mayúscula y el resto en minúscula.
Apariencia de los campos	Todas las letras en minúscula.	El nombre a emplear para los campos debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor.
Nombre de los campos	En caso de identificadores.	Todos los campos identificadores van a comenzar con el identificador id seguido de underscoard y posteriormente el nombre del campo Ejemplo: id _ paciente
Controles		
Apariencia de los controles.	Los controles tendrán un prefijo para el tipo de datos en minúscula.	El nombre que se le da a los controles deben comenzar con las primeras letras en minúscula, las cuales identificarán el tipo de datos al que se refiere en caso de que sea un nombre compuesto se empleará notación CamellCasing. Ejemplo: btnAceptar

Tabla 2.1 Estrategia y estándares de codificación.

2.3 Patrones o estilos arquitectónicos presentes en la solución. Justificación de su uso y características.

Arquitectura Modelo Cliente- Servidor.

Esta arquitectura consiste básicamente en que un programa (Cliente) realiza peticiones a otro programa (servidor) el cual le da respuesta. Con respecto a la definición de arquitectura cliente/servidor se encuentran las siguientes definiciones: [76]

Cualquier combinación de sistemas que pueden colaborar entre si para dar a los usuarios toda la información que ellos necesiten sin que tengan que saber donde está ubicada. Es una arquitectura de procesamientos cooperativa donde uno de los componentes pide servicios a otro. Es un procesamiento de datos de índole colaborativo entre dos o más computadoras conectadas a una red. El término

cliente/servidor es originalmente aplicado a la arquitectura de software que describe el procesamiento entre dos o más programas: una aplicación y un servicio soportante. [77]

Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores. La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico. [78]

Ventajas de la arquitectura cliente-servidor [79]

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Se reduce el tráfico de red considerablemente.
- En esta arquitectura el cliente se conecta al servidor cuando es necesario nada más, obtiene los datos que necesita y cierra la conexión dejando libre la red.

Se puede decir que todas las aplicaciones tienen la misma arquitectura básica y se pueden subdividir en tres partes:

- Interfaz del Usuario: La presentación al usuario, con las entradas de datos y las pantallas de consulta.
- Reglas de negocio: Sería el procesamiento de la información.
- Accesos a Datos: El control del almacén de datos.

Aplicaciones mono-capa

Se entiende por aplicaciones mono-capa, aquellas que tanto la propia aplicación como los datos que maneja se encuentran en la misma máquina y son administradas por la misma herramienta, funciona como una sola entidad. [80]

Aplicaciones con Arquitectura en dos capas (Two-Tier)

Estas aplicaciones son más conocidas como aplicaciones Cliente/Servidor y lo más característico es que dividen una aplicación entre un cliente y un servidor estableciendo un middleware que controla las comunicaciones entre ambos. Un programa Visual FoxPro que interroga a una Base de Datos SQLServer es un ejemplo de aplicación en dos capas. En la raíz de las aplicaciones cliente/servidor está la separación de la aplicación en componentes encapsulados u objetos. La ventaja de romper una aplicación en trozos es que cualquier cambio de uno de esos componentes no tiene un impacto directo sobre los otros o en el resto de la aplicación. [81]

En las arquitecturas two-tier, la aplicación se divide en dos entidades separadas. La arquitectura two-tier está dividida en dos entidades la interfaz por un lado y las reglas de negocio junto con el Acceso a Bases de Datos por otro. Encapsular las reglas de negocio junto con los datos tiene la ventaja de que se pueden cambiar sin tener que tocar los interfaces de los clientes que seguramente estarán muy distribuidos. El inconveniente es que normalmente los servidores de datos no son muy moldeables y es bastante complicado implementar reglas de negocio en los servidores. En estas aplicaciones el servidor de datos procesa las consultas y realiza todas las actividades relacionadas con la base de datos. Cada cliente inicia y deja abierta una conexión al servidor para poder enviar las peticiones y poder procesar las respuestas. [82]

Este modelo suele ser costoso de mantener, difícil de escalar y pesado de depurar. Aspectos a tener en cuenta a la hora de pasar de una aplicación de una sola capa a otra en dos Capas: usa vistas locales en vez de acceder a tablas directamente, encapsula las reglas de negocio fuera de contenedores visuales, y las reglas de negocio deberían ir en clases no visuales tipo Custom. [83]

Aplicaciones de la arquitectura en tres Capas (Three – Tier)

Con la arquitectura en tres capas (three-tier) se añade una nueva capa entre el cliente y el servidor donde se implementa la lógica de la aplicación. De esta forma el cliente es básicamente una interface, que no tiene por qué cambiar si cambian las especificaciones de la base de datos o de la aplicación, el acceso a los datos queda aislado completamente. En este caso se tiene total libertad para escoger dónde se coloca la lógica de la aplicación, en el cliente, en el servidor de base de datos, o en otro(s) servidor(es). También se tiene total libertad para la elección del lenguaje a utilizar. [84]

Puede determinarse en qué servidor(es) se quiere hacer funcionar estos procedimientos. En aplicaciones críticas se pueden agregar tantos servidores de aplicación como sean necesarios, de forma simple, y sin comprometer en absoluto la integridad de la base de datos, obteniéndose una escalabilidad muy grande sin necesidad de tocar el servidor de dicha base de datos. En teoría, los

sistemas en 3 capas son de más fácil ampliación y más robustos y flexibles. Además, pueden integrar datos de múltiples fuentes. La programación por capas es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario. [85]

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además, permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles.

Diseño en tres niveles (o en tres capas).

Capas o niveles: [86]

1.- Capa de presentación: es la que ve el usuario (hay quien la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario.

2.- Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

3.- Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores, si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores.

En una arquitectura de tres niveles, los términos "capas" y "niveles" no significan lo mismo ni son similares. El término "capa" hace referencia a la forma como una solución es segmentada desde el punto de vista lógico analizando sería: [87]

Lógica de Presentación/ Lógica de Negocio/ Lógica de Datos.

En cambio, el término "nivel" corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física. Por ejemplo:

Una solución de tres capas (presentación, lógica, datos) que residen en un solo ordenador (Presentación+lógica+datos). Se dice que la arquitectura de la solución es de tres capas y un nivel.

Una solución de tres capas que residen en dos ordenadores (presentación+lógica, lógica+datos). Se dice que la arquitectura de la solución es de tres capas y dos niveles.

Una solución de tres capas que residen en tres ordenadores (presentación, lógica, datos). La arquitectura que la define es solución de tres capas y tres niveles.

Ventajas de la arquitectura en 3 capas:

Separación clara de la interfaz de usuario de la lógica de la aplicación. Esta separación permite tener diferentes presentaciones accediendo a la misma lógica. La redefinición del almacenamiento de información no tiene influencia sobre la presentación. En contraste con una arquitectura en 2 capas, donde solamente los datos están accesibles al público, los objetos de negocios pueden brindar servicios (lógica de la aplicación) por la red. [88]

Se pueden realizar cambios en las diferentes capas sin necesidad de modificarlas todas.

Arquitectura Basada en Componentes

El objetivo de la tecnología de componentes software es construir aplicaciones complejas mediante ensamblado de módulos (componentes) que han sido previamente diseñados por otras personas a fin de ser reusados en múltiples aplicaciones. La arquitectura software de una aplicación basada en componentes consiste en uno o un número pequeño de componentes específicos de la aplicación (que se diseñan específicamente para ella), que hacen uso de otros muchos componentes prefabricados que se ensamblan entre sí para proporcionar los servicios que se necesitan en la aplicación.[89]

Es decir consiste en mapear los componentes funcionales en la arquitectura lógica de la aplicación.

Para que una arquitectura de componentes pueda operar es necesario disponer de un entorno normalizado que proporcione soporte a los mecanismos con que se comunican las interfaces.

Reusabilidad de Servicios: Reducción considerable de tiempos y costos de desarrollo de aplicaciones al utilizar servicios disponibles ya desarrollados, para resolver problemáticas comunes a otras aplicaciones. Aumentado por esta razón la robustez del nuevo sistema, al utilizarse software ya probado.

Interoperabilidad de aplicaciones: Disminución de la complejidad en el proceso de integración, pues se interactúa con elementos que se abstraen de la tecnología y ubicación de los servicios.

Entonces se puede decir que las características principales son la modularidad, la reusabilidad y compatibilidad, en la tecnología basada en componentes también se requiere robustez ya que los componentes han de operar en entornos mucho más heterogéneos y diversos. Su premisa es que los componentes cumplan con alta cohesión y bajo acoplamiento.

Patrones de diseños.

Los patrones del diseño tratan los problemas del diseño que se repiten y se presentan en situaciones particulares del diseño, con el fin de proponer soluciones a ellas. Por tanto estos son soluciones exitosas a problemas comunes. Existen muchas ideas para implementarlos, los detalles son llamados estrategias. Hay patrones que abarcan las distintas etapas del desarrollo, desde el análisis hasta el diseño y desde la arquitectura hasta la implementación. [90].

Modelo Vista Controlador (MVC).

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde se separa la vista, de la modelo, de las controladoras. La modelo se encarga de la lógica del negocio y el controlador se encarga de controlar los eventos del usuario o también se puede decir que se encarga de la lógica de la aplicación. [91]

Para construir una aplicación utilizando el patrón MVC hay que definir tres clases de módulos: [92]

El modelo:

Accede a la capa de almacenamiento de datos o acceso a datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento. Lleva un registro de las vistas y controladores del sistema. Ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo.

El controlador:

Responde a eventos y modifica la vista y el modelo. Recibe los eventos de entrada. Contiene reglas de gestión de eventos. Estas acciones pueden suponer peticiones al modelo o a las vistas.

Las vistas:

Existen distintas formas de implementar el patrón, en la vista se reciben los datos del controlador y se los muestra al usuario. Define la interfaz de usuario.

La principal ventaja de esta separación reside en la facilidad para realizar cambios en la aplicación puesto que: [93]

- Cuando se realiza un cambio de bases de datos, programación o interfaz de usuario solo trabaja en uno de los componentes.
- Modificar uno de los componentes sin conocer como funcionan los otros.

Aunque se pueden encontrar diferentes implementaciones del patrón MVC, el flujo que sigue, el control generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace)
- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario. Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. La vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos de este a la vista.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Arquitectura Orientada a Servicios

La Arquitectura Orientada a Servicios (en inglés Service Oriented Architecture o SOA), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos de software del usuario. SOA proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación. En este ambiente, los nodos de la red hacen disponibles sus recursos a otros participantes en la red como servicios independientes a los que tienen acceso de un modo estandarizado. La mayoría de las definiciones de SOA identifican la utilización de Servicios Web (empleando SOAP y WSDL) en su implementación, no obstante se puede implementar utilizando cualquier tecnología basada en servicios. [94]

Al contrario de las arquitecturas orientadas a objetos, las SOAs están formadas por servicios de aplicación débilmente acoplados y altamente interoperables. Para comunicarse entre sí, estos servicios se basan en una definición formal independiente de la plataforma subyacente y del lenguaje de programación. La definición de la interfaz encapsula (oculta) las particularidades de una implementación, lo que la hace independiente del fabricante, del lenguaje de programación o de la tecnología de desarrollo (como Plataforma Java o Microsoft.NET). Con esta arquitectura, se pretende que los componentes software desarrollados sean muy reusables, ya que la interfaz se define siguiendo un estándar; así, un servicio C Sharp podría ser usado por una aplicación Java. [95]

2.4 Vista de despliegue.

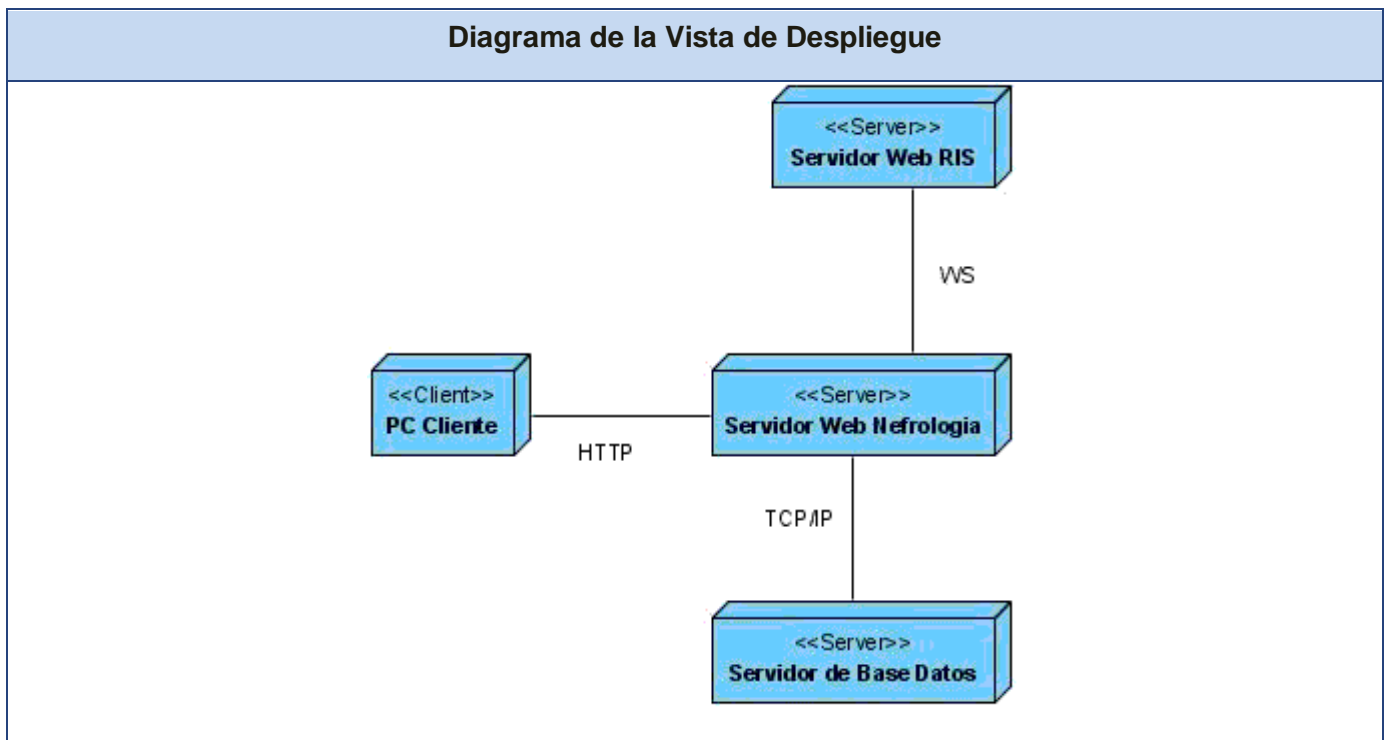


Fig. 2.1 Diagrama de la Vista de Despliegue.

Servidor Web RIS: En este nodo se encuentran desplegados los componentes del RIS con los que debe interactuar la Red Cubana de Nefrología.

Servidor Web Nefrología: En este nodo se encuentra la aplicación Red Cubana de Nefrología.

Servidor de Base de Datos: En este nodo se encuentra el servidor de la BD del sistema Red Cubana de Nefrología.

PC cliente: Desde ella se accede a través de un navegador al sistema Red Cubana de Nefrología.

2.5 Vista de implementación.

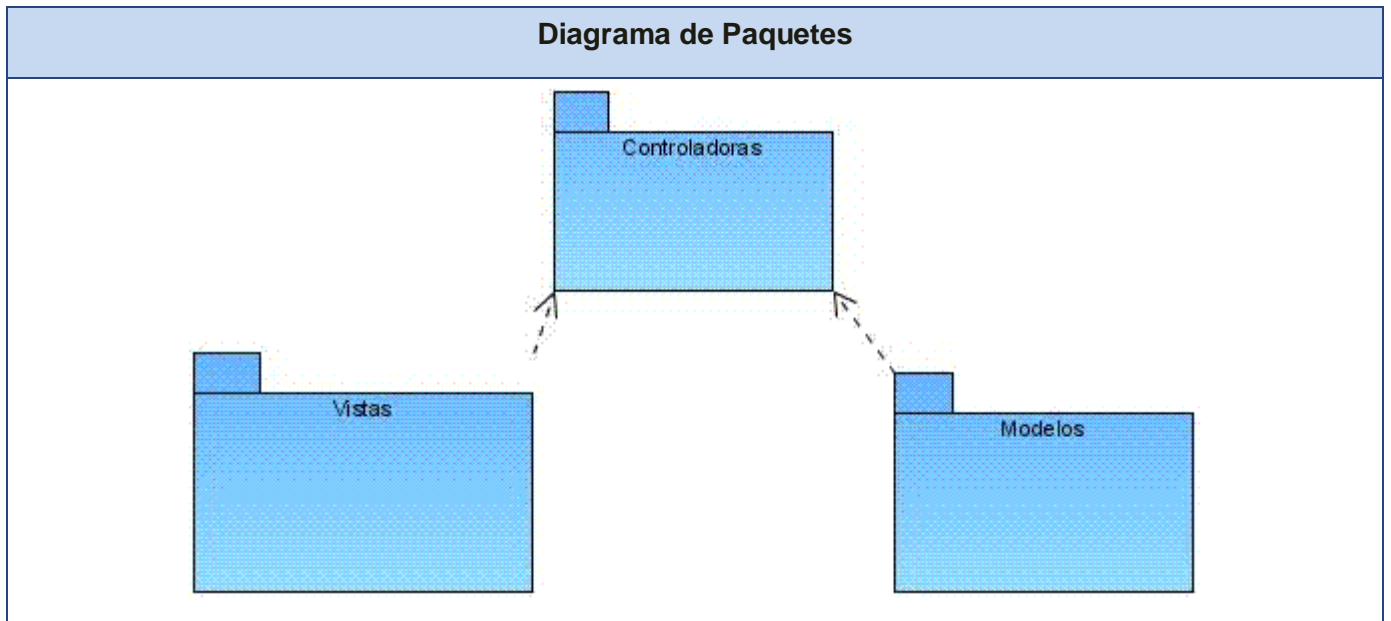


Fig. 2.2 Diagrama de paquetes.

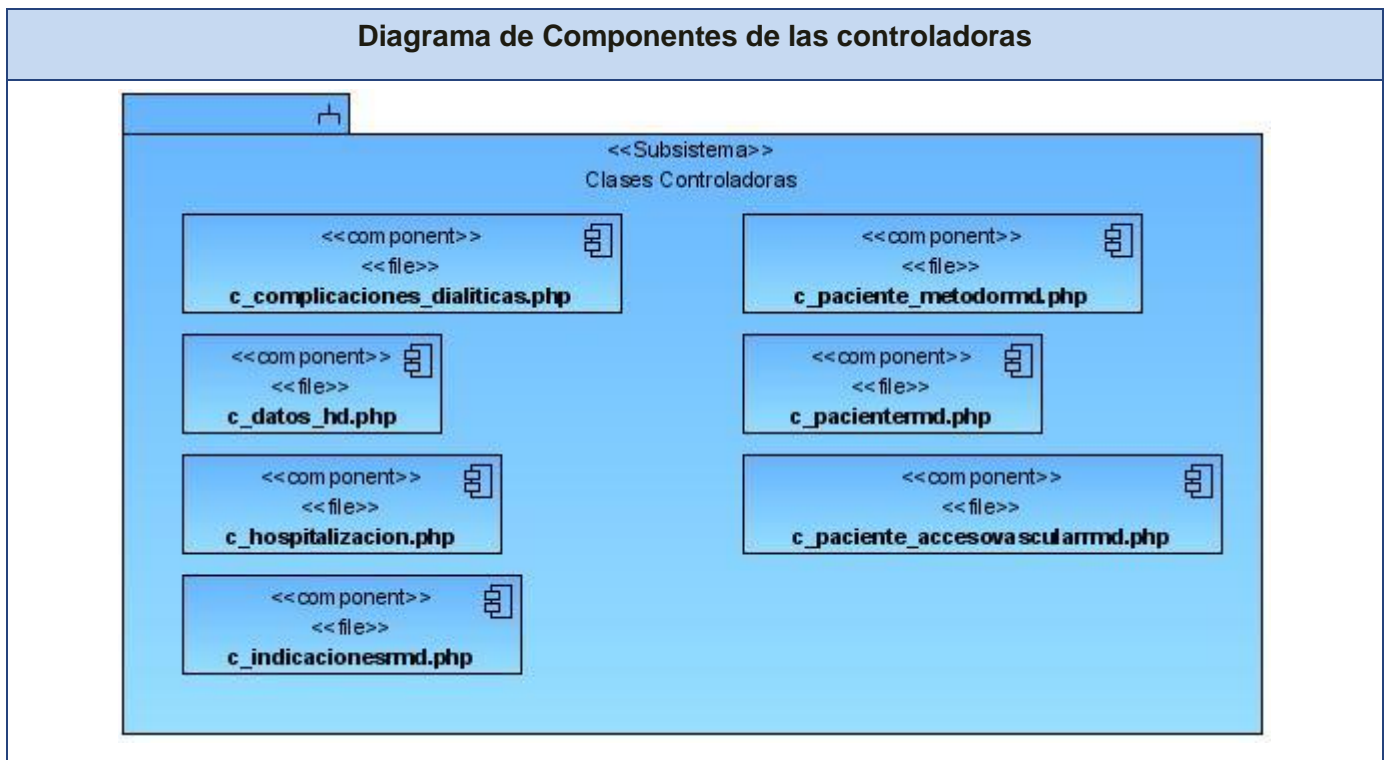


Fig.2.3 Diagrama de Componentes de las controladoras.

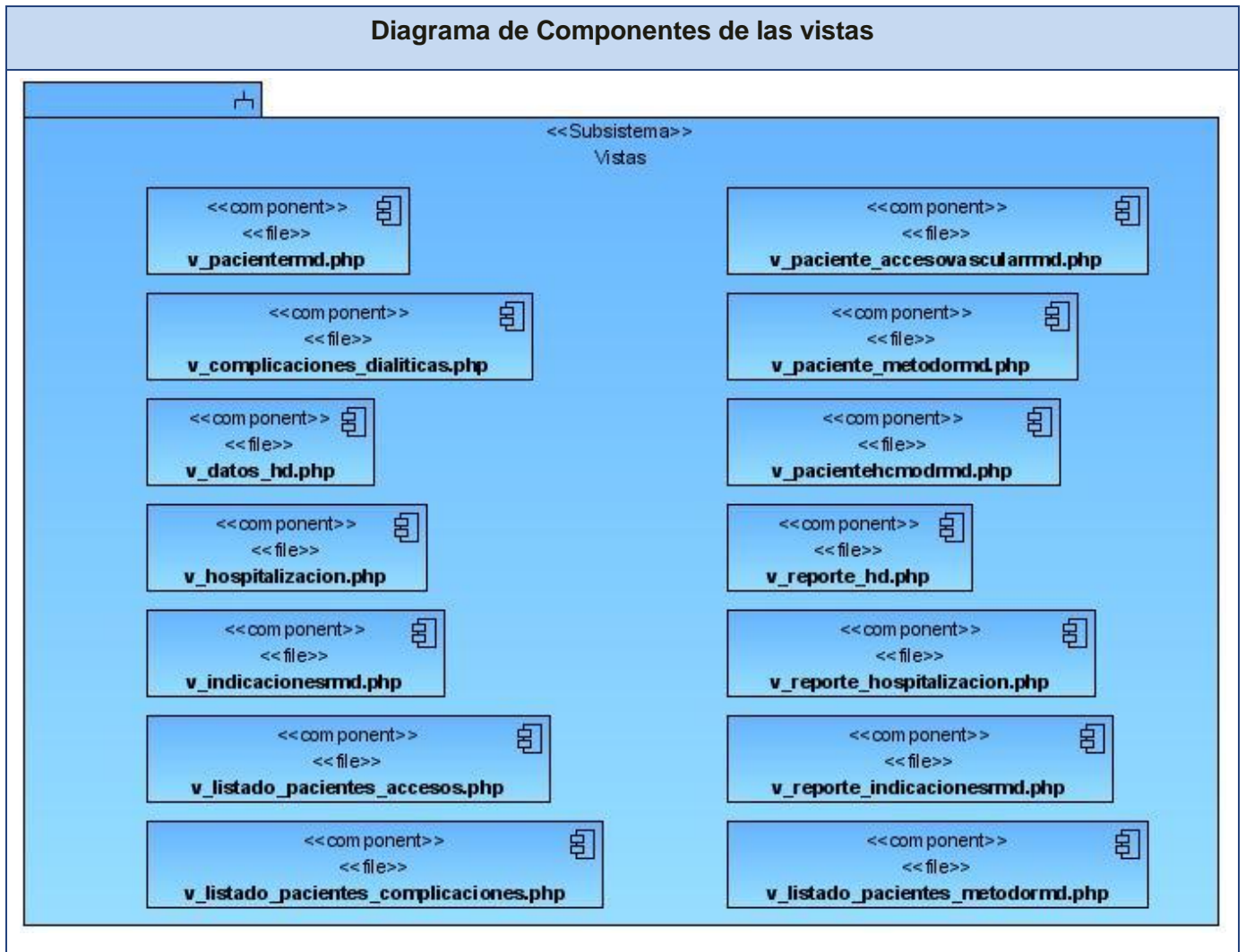


Fig.2.4 Diagrama de Componente de las Vista

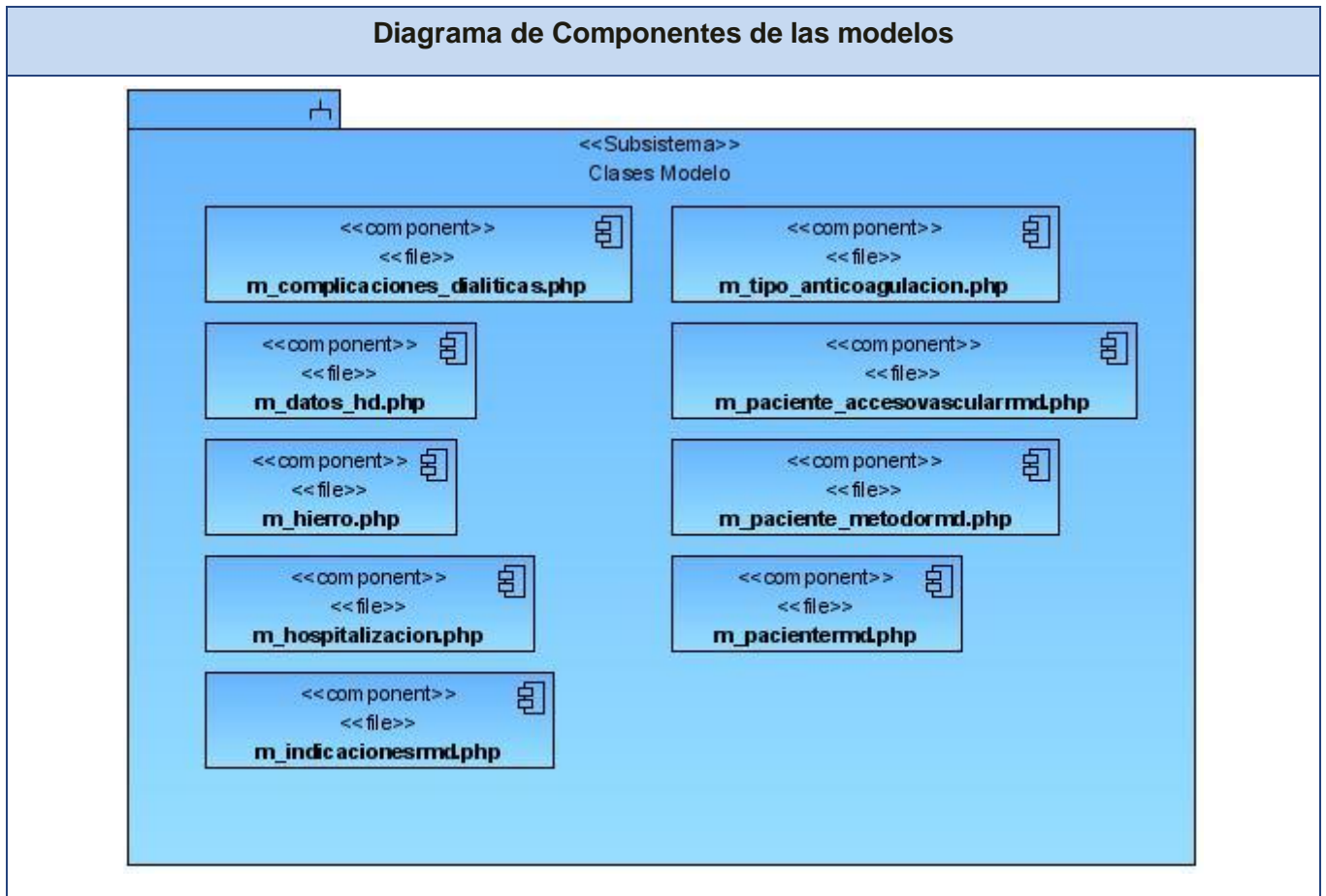


Fig. 2.5 Diagrama de componentes de las Modelos.

Conclusiones

En el capítulo se describieron los requisitos no funcionales del sistema, los que se tendrán en cuenta para realizar la aplicación. Fueron mostrados los estándares y las estrategias de codificación que están presentes en el sistema, para que el código esté más organizado y fácil de entender. Además, se hace un estudio de los patrones y estilos arquitectónicos necesarios. Se ha establecido una arquitectura centralizada, en este caso en los servidores de infomed, basada en componentes, en tres capas, y basada en el patrón modelo-vista- controlador.

Capítulo 3. Descripción y análisis de la solución propuesta.

El capítulo pretende realizar un análisis de la estrategia de integración con otros módulos. Además se podrán observar descripciones de algunas clases, datos, y métodos importantes para el módulo, incluyendo el Diagrama Entidad Relación de la BD correspondiente a las clases a dichas clases.

3.1 Estrategia de integración de la solución con otros módulos o sistemas.

El Sistema Red Cubana de Nefrología pertenece al Sistema Informatizado de Atención Especializada (SIAE), perteneciente al SISalud, pues es aquí, donde se agruparán los módulos que pertenecen al nivel de atención terciario o especializado.

Siguiendo esta filosofía, la Red Cubana de Nefrología necesita para su buen funcionamiento integrarse a una serie de componentes ya desarrollados para el SISalud.

Del Registro Informatizado de la Salud (RIS) para el control de la seguridad se integra con el:

SAAA (Sistema de Autenticación, Autorización y Auditoría)

Módulo de seguridad y Administración, está basado en un modelo de autenticación, autorización y auditoría (SAAA). Es el sistema que gestiona el nivel de acceso a los distintos componentes del SISalud, permitiendo que los usuarios tengan un solo usuario y una contraseña para acceder a todos los sistemas. Este brinda a través de Web Services una serie de métodos que facilitan todos los servicios necesarios para asegurar el sistema.

Para su uso se crearon diferentes clases que encapsulan el manejo de la seguridad así como la comunicación con este módulo, de manera que esta queda de forma global para todo el sistema sin necesidad de ser implementada en cada subsistema pero si utilizada por todos y de manera particular por cada uno.

RC (Registro de Ciudadanos)

Brinda la información primaria de los pacientes como es: su nombre, apellidos, CI, dirección y particular.

Este módulo se utilizo para el registro de los pacientes, del cual se obtienen los datos iniciales de los pacientes que van, ser registrados en el módulo. Estos datos no son pedidos directamente al RC por parte del módulo RMD, se creó un módulo (RPN), encargado de la comunicación con el RC y de la obtención de aquellos ciudadanos que fueran solicitados por los demás módulos, a este es al cual RMD le solicita los pacientes a ser insertados y así se asegura la obtención de las personas que

realmente puedan ser registradas, pues RPN se encarga de una serie de validaciones necesarias para que el sistema marche de manera correcta.

RU (Registro de Ubicación Geográfica): Se encarga de gestionar la información de las provincias, municipios, localidades, calles y manzanas del país.

RUS (Registro de Unidades de Salud): Se utiliza para obtener la información que identifica a las unidades donde está ubicada los centros de diálisis.

Además el sistema está implementado usando el framework de clases de PHP CodeIgniter, y el framework de clases de JavaScript YUI, se hace rehúso de todas las librerías y clases que estos dos potentes frameworks ofrecen.

3.2 Descripción de las clases u operaciones necesarias.

Se presentará la descripción de las clases y operaciones necesarias de algunos de los casos de uso críticos del sistema. Las otras descripciones se podrán encontrar en el **Anexo I**.

Nombre: E_PacienteRMD	
Tipo de clase: Entidad	
Atributo	Tipo
ID_EntradaRMD : int	int
Para cada responsabilidad:	
Nombre:	E_PacienteRMD()
Descripción:	Constructor de la clase.
Nombre:	Set_Entrada(identrada)
Descripción:	Cambia el id de entrada del paciente.
Nombre:	Get_Entrada()
Descripción:	Devuelve id de entrada del paciente.

Tabla 3.1 Descripción de la clase E_PacienteRMD, CU Gestionar Movimiento.

Nombre: M_PacienteRMD
Tipo de clase: Controladora

Para cada responsabilidad:	
Nombre:	ListarRegistrarPaciente(EstructuraEntrada, E_PacienteRMD) : EstructuraSalida
Descripción:	Método que lista los pacientes a ser registrados, teniendo en cuenta la estructura de entrada que se entra.
Nombre:	RegistrarPaciente(E_PacienteRMD)
Descripción:	Registra paciente que necesitan servicios en el registro de métodos depuradores.
Nombre:	CambiarPaciente(E_PacienteRMD)
Descripción:	Cambia un paciente.
Nombre:	RegistrarSalida(E_PacienteRMD, CS, obs)
Descripción:	Registra la salida de un paciente.
Nombre:	ModificarDatosIniciales(E_DatosIniciales)
Descripción:	Modifica los datos iniciales que fueron introducidos a un paciente determinado.
Nombre:	EliminarPaciente(E_PacienteRMD)
Descripción:	Elimina el paciente que se desee.
Nombre:	EstaPacienteMódulo(E_PacienteRMD)
Descripción:	Método que informa si el paciente está en el módulo.
Nombre:	PerteneceOtroCD(E_PacienteRMD)
Descripción:	Método que informa si el paciente pertenece o no a otro centro de diálisis diferente al del usuario que esta logueado.
Nombre:	DarSalida(módulo, E_PacienteRMD, fecha, CS)
Descripción:	Método que da salida a un paciente del módulo.
Nombre:	ListarPacientes(EstructuraEntrada, E_PacienteRMD)
Descripción:	Método que lista los pacientes registrados, teniendo en cuenta la estructura de

	entrada que se entra.
--	-----------------------

Tabla 3.2 Descripción de la clase M_PacienteRMD, CU Gestionar Movimiento.

Nombre: C_PacienteRMD	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	ListarRegistrarPaciente(offset, Cant, OrdenadoPor, direccion, nomb, apell1, apell2)
Descripción:	Método que recibe diferentes parámetros del paciente, accede a la clase moduladora y devuelve un listado de los pacientes a registrar.
Nombre:	RegistrarPaciente()
Descripción:	Método que recibe diferentes parámetros del registro de pacientes y devuelve un objeto al método correspondiente en la moduladora
Nombre:	CambiarPaciente()
Descripción:	Método que recibe diferentes parámetros del paciente que se desea cambiar y devuelve un objeto al método correspondiente en la moduladora
Nombre:	ModificarDatosIniciales()
Descripción:	Método que recibe diferentes parámetros del método que modifica los datos iniciales y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	RegistrarSalida()
Descripción:	Método que recibe diferentes parámetros para registrar la salida de un paciente y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	EliminarPaciente(idciudadano)
Descripción:	Método que recibe diferentes parámetros para eliminar un paciente y devuelve un objeto al método correspondiente en la moduladora

Nombre:	CrearOpciones(EstructuraSalida)
Descripción:	Método auxiliar para crear las opciones de los dropdown
Nombre:	EstaPacienteMódulo(idpaciente)
Descripción:	Método que recibe diferentes parámetros para saber si un paciente está o no en el módulo y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	ComprobarPacienteInsertar(idpaciente)
Descripción:	Método que recibe diferentes parámetros para comprobar si un paciente se puede insertar y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	ListarPaciente(offset, Cant, OrdenadoPor, direccion, nomb, apell1, apell2)
Descripción:	Método que recibe diferentes parámetros listar pacientes y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	GetDatosPaciente(idciudadano)
Descripción:	Método que recibe diferentes parámetros para devolver datos de un paciente y devuelve un objeto al método correspondiente en la moduladora.

Tabla 3.3 Descripción de la clase C_PacienteRMD, CU Gestionar Movimiento.

Nombre: E_Paciente_Metodo	
Tipo de clase: entidad	
Atributo	Tipo
Paciente	E_PacienteRMD
Método	E_Metodo
Fecha_Inicio	DateTime
Para cada responsabilidad:	

Nombre:	E_Paciente_Metodo()
Descripción:	Constructor de la clase, por defecto.
Nombre:	E_Paciente_Metodo(Paciente, Método)
Descripción:	Constructor de la clase, con parámetros.
Nombre:	TiempoDelMetodo()
Descripción:	Tiempo que ha estado con el método.
Nombre:	Get_Paciente() : E_PacienteRMD
Descripción:	Método que devuelve los datos del paciente.
Nombre:	Get_Metodo()
Descripción:	Devuelve el método depurador del paciente.
Nombre:	Get_Fecha_Inicio()
Descripción:	Método que devuelve la fecha en que inició el paciente con el método.
Nombre:	Set_Paciente(E_PacienteRMD)
Descripción:	Método que cambia el paciente.
Nombre:	Set_Metodo(E_Metodo)
Descripción:	Método que cambia el método que se le asigno antes.
Nombre:	Set_Fecha_Inicio(fecha_inicio)
Descripción:	Método que cambia la fecha en que inició el paciente con el método.

Tabla 3.4 Descripción de la clase E_Paciente_Metodo, CU Gestionar Movimiento.

Nombre: M_Paciente_Metodo	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Asignar_MetodoDialitico(E_Paciente_Metodo)

Descripción:	Método que asigna, al paciente un método dialítico.
Nombre:	Cambiar_MetodoDialitico(E_Paciente_Metodo, E_Paciente_Metodo_Historico)
Descripción:	Método que cambia el método dialítico que tiene el paciente.
Nombre:	ListarPacientesMetodo(EstructuraEntrada, E_Metodo)
Descripción:	Método que devuelve una lista de los pacientes organizados por métodos, teniendo en cuenta estructuras de entrada.
Nombre:	Metodo_Activo(E_PacienteRMD)
Descripción:	Método que devuelve el método que tiene indicado el paciente actualmente.
Nombre:	ListarMetodoHist(EstructuraEntrada, E_PacienteRMD)
Descripción:	Método que listas los métodos históricos, es decir los métodos que fueron asignados a un determinado paciente.

Tabla 3.5 Descripción de la clase M_Paciente_Metodo, CU Gestionar Movimiento.

Nombre: E_PacienteRMDHistorico	
Tipo de clase: Entidad	
Atributo	Tipo
Fecha_Salida	DateTime
Causa_Salida	E_CausaSalida
Observaciones	string
Para cada responsabilidad:	
Nombre:	E_PacienteRMDHistorico(Fecha_Salida)
Descripción:	Constructor de la clase
Nombre:	Get_FechaSalida()
Descripción:	Método que devuelve la fecha de salida del paciente que se encuentra en los históricos del hospital.

Nombre:	Get_CausaSalida()
Descripción:	Método que devuelve la causa de salida del paciente que se encuentra en los históricos del hospital.
Nombre:	Get_Observaciones()
Descripción:	Método que devuelve las observaciones del paciente que se encuentra en los históricos del hospital.
Nombre:	Set_FechaSalida(fechasalida)
Descripción:	Método que cambia la fecha de salida del paciente que se encuentra en los históricos del hospital.
Nombre:	Set_CausaSalida(CS)
Descripción:	Método que cambia la causa de salida del paciente que se encuentra en los históricos del hospital.
Nombre:	Set_Observaciones(obs)
Descripción:	Método que cambia las observaciones del paciente que se encuentra en los históricos del hospital.

Tabla 3.6 Descripción de la clase E_PacienteRMDHistorico, CU Gestionar Movimiento.

Nombre: M_PacienteAccesoVascular	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Registrar_AccesoVascular(E_PacienteAccesoVascular)
Descripción:	Método que registra el acceso vascular a un paciente.
Nombre:	Realizar_Acceso(E_PacienteAccesoVascular)
Descripción:	Método que realiza el acceso vascular a un paciente.
Nombre:	EliminarAV_Realizado(E_PacienteAccesoRealizado)

Descripción:	Método que elimina el acceso vascular realizado a un paciente.
Nombre:	ListarAV_Realizado(EstructuraEntrada, E_PacienteRMD)
Descripción:	Método que lista los accesos vasculares realizados de acuerdo a la estructura de entrada que se entren.
Nombre:	ListarAccesosNoActivos(EstructuraEntrada, E_PacienteRMD)
Descripción:	Método que lista los accesos vasculares que no están activos.
Nombre:	CambiarAcceso(E_PacienteAccesoVascular, fechaC, E_MotivoCambioAcceso, obs)
Descripción:	Método que cambia el acceso el vascular que tiene un paciente, ya sea sus datos.
Nombre:	Acceso_Activo(E_PacienteRMD)
Descripción:	Método que devuelve el acceso que tiene actualmente el paciente.
Nombre:	ListarCambiosAccesos(EstructuraEntrada, E_PacienteRMD)
Descripción:	Método que lista los accesos vasculares cambiados de algún paciente, según la estructura de entrada.
Nombre:	ListarPacientesAcceso(EstructuraEntrada, E_Acceso_Vascular)
Descripción:	Método que lista los pacientes según un acceso vascular, teniendo en cuenta la estructura de entrada.
Nombre:	ListarAccesoHistorico(EstructuraEntrada, E_PacienteRMD)
Descripción:	Método que lista los accesos vasculares históricos de algún paciente, según la estructura de entrada.
Nombre:	ListarTodosAV_Realizados(E_PacienteRMD)
Descripción:	Método que lista todos los accesos vasculares realizados a un paciente
Nombre:	EliminarAcceso(E_PacienteAccesoVascular)
Descripción:	Método que elimina un acceso sino está siendo utilizado por ninguna persona

Nombre:	EliminarAV_Realizado(E_PacienteAccesoRealizado)
Descripción:	Método que elimina el acceso vascular que ha sido realizado a algún paciente.
Nombre:	Dar_SalidaAcceso(E_PacienteAccesoHist)
Descripción:	Método que da salida de un acceso que tiene un paciente, poniendo este en histórico.
Nombre:	ListarPacienteAcceso(E_PacienteAccesoVascular)
Descripción:	Método que lista los pacientes y los accesos vasculares realizados.

Tabla 3.7 Descripción de la clase M_PacienteAccesoVascular, CU Gestionar Movimiento.

Nombre: E_Datos_Iniciales	
Tipo de clase: Entidad	
Atributo	Tipo
Peso_Inicial	float
Fecha_entrada	DateTime
ANef_antes_díalisis	int
Eritropoyectina	bool
Fecha_Eritropoyectina	DataTime
Talla	float
Para cada responsabilidad:	
Nombre:	E_Datos_Iniciales()
Descripción:	Constructor.
Nombre:	Get_Fecha_entrada() : DateTime
Descripción:	Método que devuelve la fecha de entrada.

Nombre:	Get_PesoInicial()
Descripción:	Método que devuelve el peso inicial.
Nombre:	Get_Talla()
Descripción:	Método que devuelve la talla.
Nombre:	Get_Eritropoyectina()
Descripción:	Método que devuelve la si tiene o no Eritropoyetina.
Nombre:	Get_Fecha_Eritroproyectina()
Descripción:	Método que devuelve la fecha que inicio la Eritropoyetina.
Nombre:	Get_ANefAntesDiálisis()
Descripción:	Método que devuelve atención nefrológica antes de la diálisis.
Nombre:	Set_Fecha_entrada(fechaentrada)
Descripción:	Método que cambia la fecha de entrada.
Nombre:	Set_PesoInicial(peso inicial)
Descripción:	Método que cambia el peso inicial.
Nombre:	Set_Eritroproyectina(eritrop)
Descripción:	Método que cambia la Eritropoyetina.
Nombre:	Set_Fecha_Eritroproyectina(fechaeritrop)
Descripción:	Método que cambia la fecha de Eritropoyetina.
Nombre:	Set_Talla(talla)
Descripción:	Método que cambia la talla.
Nombre:	Set_ANefAntesDiálisis(a_nef_antes_diálisis)
Descripción:	Método que cambia la atención nefrológica antes de la diálisis.

Tabla 3.8 Descripción de la clase E_Datos_Iniciales, CU Gestionar Movimiento.

Casos de uso Gestionar Acceso

Nombre: E_PacienteAccesoVascular	
Tipo de clase: entidad	
Atributo	Tipo
Paciente	E_PacienteRMD
Acceso_Vascular	E_Acceso_Vascular
FechaRealizacion	DateTime
Localización	E_Localizacion_Acceso
Activo	bool
Fecha_Inicio	DateTime
Para cada responsabilidad:	
Nombre:	E_PacienteAccesoVascular()
Descripción:	Constructor de la clase, por defecto.
Nombre:	E_PacienteAccesoVascular(Paciente, Acceso_Vascular)
Descripción:	Constructor de la clase, con parámetros.
Nombre:	TiempodeUso() : int
Descripción:	Tiempo que lleva el paciente con el acceso Vascular.
Nombre:	Get_Paciente()
Descripción:	Método que devuelve el paciente.
Nombre:	Get_Acceso_Vascular()
Descripción:	Devuelve el acceso vascular que tiene el paciente.

Nombre:	Get_Fecharealizacion()
Descripción:	Método que devuelve la fecha de realización del acceso.
Nombre:	Get_Localizacion() : E_Localizacion_Acceso
Descripción:	Método que devuelve la localización del acceso.
Nombre:	Get_Activo()
Descripción:	Método que devuelve si el método está en uso o no.
Nombre:	Get_Inicio()
Descripción:	Método que devuelve la fecha de inicio de uso del acceso vascular.
Nombre:	Set_Paciente(E_PacienteRMD)
Descripción:	Método que cambia el paciente.
Nombre:	Set_Fecharealizacion(fecharealizacion)
Descripción:	Método que cambia la fecha de realización.
Nombre:	Set_Acceso_Vascular(E_Acceso_Vascular)
Descripción:	Método que cambia el acceso vascular que tiene el paciente.
Nombre:	Set_Localizacion(E_Localizacion_Acceso) : void
Descripción:	Método que cambia la localización
Nombre:	Set_Activo(activo) : void
Descripción:	Método que cambia si está activo o no.
Nombre:	Set_Inicio(fechaini) : void
Descripción:	Método que cambia la fecha de inicio de uso del acceso vascular.

Tabla 3.9 Descripción de la clase E_PacienteAccesoVascular, CU Gestionar Acceso Vascular.

Nombre: C_PacienteAccesoVascular

Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	C_PacienteAccesoVascular()
Descripción:	Constructor de la clase, por defecto.
Nombre:	RealizarAcceso()
Descripción:	Método que recibe diferentes parámetros para realizar el acceso y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	ListarAV_Realizado(offset, Cant, OrdenadoPor, direccion, idciudadano)
Descripción:	Método que recibe diferentes parámetros para listar los accesos ya realizados y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	Acceso_Activo()
Descripción:	Método que recibe diferentes parámetros para comprobar si el acceso está Activo o no y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	ListarAccesosNoActivos(offset, Cant, OrdenadoPor, direccion)
Descripción:	Método que recibe diferentes parámetros para listar los accesos no activo y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	CambiarAcceso()
Descripción:	Método que recibe diferentes parámetros para cambiar los accesos y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	EliminarAcceso(id)
Descripción:	Método que recibe diferentes parámetros para eliminar los accesos y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	EliminarAVRealizado(id)

Descripción:	Método que recibe diferentes parámetros para eliminar los accesos ya realizados y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	Dar_SalidaAcceso()
Descripción:	Método que recibe diferentes parámetros para dar salida a un acceso vascular y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	ListarCambiosAcceso(offset, Cant, OrdenadoPor, direccion)
Descripción:	Método que recibe diferentes parámetros para listar algún cambio de un acceso y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	ListarAccesoHistorico(offset, Cant, OrdenadoPor, direccion)
Descripción:	Método que recibe diferentes parámetros para listar los accesos históricos y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	ListarPacientesAcceso(offset, Cant, OrdenadoPor, direccion, acceso, activo)
Descripción:	Método que recibe diferentes parámetros para listar los pacientes que se encuentran realizando el mismo acceso y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	CrearOpciones(EstructuraSalida)
Descripción:	Método auxiliar para crear las opciones de los componentes List de las vistas, con el objetivo de que se creen dinámicos a partir de la información obtenida de la Base de Datos.

Tabla 3.10 Descripción de la clase C_PacienteAccesoVascular, CU Gestionar Acceso Vascular.

Nombre: M_PacienteAccesoVascular
Tipo de clase: Controladora
Para cada responsabilidad:

Nombre:	M_PacienteAccesoVascular()
Descripción:	Constructor de la clase, por defecto.
Nombre:	Realizar_Acceso(E_PacienteAccesoVascular)
Descripción:	Método que realiza el acceso Vascular
Nombre:	Registrar_AccesoVascular(E_PacienteAccesoVascular)
Descripción:	Método que registra el Acceso Vascular.
Nombre:	ListarAV_Realizado(EstructuraEntrada, E_PacienteRMD)
Descripción:	Método que devuelve un listado de los Accesos realizados.
Nombre:	ListarTodosAV_Realizados(E_PacienteRMD)
Descripción:	Método que lista todos los Accesos realizados.
Nombre:	ListarAccesosNoActivos(EstructuraEntrada, E_PacienteRMD)
Descripción:	Método que lista los Acceso no activos.
Nombre:	Acceso_Activo(E_PacienteRMD)
Descripción:	Método que devuelve cual es el acceso que está activo.
Nombre:	CambiarAcceso(E_PacienteAccesoVascular, fechaC,E_MotivoCambioAcceso, obs)
Descripción:	Método que le cambia el acceso vascular y sus datos a un paciente.
Nombre:	EliminarAcceso(E_PacienteAccesoVascular)
Descripción:	Método que elimina el acceso Vascular.
Nombre:	EliminarAV_Realizado(E_PacienteAccesoRealizado)
Descripción:	Método que elimina los accesos Vasculares que han sido realizados.
Nombre:	Dar_SalidaAcceso(E_PacienteAccesoHist)
Descripción:	Método que da salida a un acceso.

Nombre:	ListarCambiosAccesos(EstructuraEntrada, E_PacienteRMD)
Descripción:	Método que lista los cambios de los accesos.
Nombre:	ListarAccesoHistorico(EstructuraEntrada, E_PacienteRMD)
Descripción:	Método que lista los accesos vasculares que son históricos.
Nombre:	ListarPacientesAcceso(EstructuraEntrada, E_Acceso_Vascular)
Descripción:	Método que lista los pacientes por accesos.
Nombre:	ListarPacienteAcceso(E_PacienteAccesoVascular)
Descripción:	Método que lista los pacientes por accesos.

Tabla 3.11 Descripción de la clase M_PacienteAccesoVascular, CU Gestionar Acceso Vascular.

Nombre: E_PacienteAccesoHist	
Tipo de clase: entidad	
Atributo	Tipo
Fecha Salida	DateTime
Observaciones	string
Para cada responsabilidad:	
Nombre:	E_PacienteAccesoHist()
Descripción:	Constructor de la clase, por defecto.
Nombre:	Get_FechaSalida()
Descripción:	Método que devuelve la fecha de salida.
Nombre:	Get_Observaciones() : string
Descripción:	Método que devuelve las observaciones.
Nombre:	Set_FechaSalida(fecha) : void

Descripción:	Método que cambia la fecha de salida del acceso.
Nombre:	Set_Observaciones(obs) : void
Descripción:	Método que cambia las observaciones del acceso.

Tabla 3.12 Descripción de la clase E_PacienteAccesoHist, CU Gestionar Acceso Vascular.

Clases y operaciones necesarias del módulo de Configuración, caso de uso **Método**.

Nombre: C_Metodo	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Adicionar_Metodo()
Descripción:	Método que recibe diferentes parámetros del método adicionar y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	Eliminar()
Descripción:	Método que recibe diferentes parámetros del método eliminar y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	Actualizar()
Descripción:	Método que recibe diferentes parámetros del método actualizar y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	Listar_Metodo(offset, cant, campo, orden)
Descripción:	Método que recibe diferentes parámetros del método listar y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	Listar_MetodoEliminado(offset, cant, campo, orden)
Descripción:	Método que recibe diferentes parámetros del método listar métodos eliminar y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	Activar(id : int [])

Descripción:	Método que recibe diferentes parámetros del método activar y devuelve un objeto al método correspondiente en la moduladora.
--------------	---

Tabla 3.13 Descripción de la clase C_Metodo, CU Método.

Nombre: M_Metodo	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Adicionar(E_Metodo)
Descripción:	Adiciona el método que se le da como parámetro.
Nombre:	Editar(E_Metodo)
Descripción:	Registra el método que se le pasa como parámetro.
Nombre:	Eliminar(E_Metodo : E_Metodo [])
Descripción:	Elimina el método que se desee sino lo utiliza ningún paciente.
Nombre:	Listar(offset, cant, campo, orden)
Descripción:	Brinda una lista de los métodos que se encuentran registrados.
Nombre:	Listar_Eliminados(offset, cant, campo, orden)
Descripción:	Lista los métodos que han sido eliminados, teniendo en cuenta algunas estructuras de entradas.
Nombre:	Activar(E_Metodo : E_Metodo [])
Descripción:	Activa el método que se desee.

Tabla 3.14 Descripción de la clase M_Metodo, CU Método.

Nombre: E_Metodo	
Tipo de clase: Entidad	
Atributo	Tipo
Descripción	string

Eliminado	bool
Para cada responsabilidad:	
Nombre:	E_Metodo()
Descripción:	Constructor de la clase, por defecto.
Nombre:	E_Metodo(Descripcion)
Descripción:	Constructor de la clase, con parámetros.
Nombre:	Get_Descripcion()
Descripción:	Devuelve la descripción que tiene el método.
Nombre:	Set_Descripcion(Descripcion)
Descripción:	Cambia la descripción del método.
Nombre:	Get_Eliminado()
Descripción:	Devuelve si o no el método ha sido eliminado.
Nombre:	Set_Eliminado(Eliminado)
Descripción:	Cambia el si el método ha sido eliminado o no.

Tabla 3.15 Descripción de la clase E_Metodo, CU Método.

Nombre: E_Entidad	
Tipo de clase: Entidad	
Atributo	Tipo
ID	int
Para cada responsabilidad:	
Nombre:	Get_ID()
Descripción:	Devuelve el id del método
Nombre:	Set_ID(ID)

Descripción:	Cambia el id del método.
--------------	--------------------------

Tabla 3.16 Descripción de la clase E_Entidad, CU Método.

Clases del Módulo de configuración del caso de uso **Acceso Vascular**.

Nombre: C_Acceso_Vascular	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Adicionar_Acceso_Vascular()
Descripción:	Método que recibe diferentes parámetros del método adicionar acceso vascular y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	Eliminar()
Descripción:	Método que recibe diferentes parámetros del método eliminar acceso vascular y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	Actualizar()
Descripción:	Método que recibe diferentes parámetros del método actualizar acceso vascular y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	Listar_Acceso_Vascular(offset, cant, campo, orden)
Descripción:	Método que recibe diferentes parámetros del método listar acceso vascular y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	Listar_AccesoVascularEliminado(offset, cant, campo, orden)
Descripción:	Método que recibe diferentes parámetros del método listar accesos vasculares eliminados y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	Activar(id)
Descripción:	Método que recibe diferentes parámetros del método activar y devuelve un

	objeto al método correspondiente en la moduladora.
--	--

Tabla 3.17 Descripción de la clase C_Acceso_Vascular, CU Acceso Vascular

Nombre: E_Acceso_Vascular	
Tipo de clase: Entidad	
Atributo	Tipo
Descripción	string
Eliminado	bool
Para cada responsabilidad:	
Nombre:	E_Acceso_Vascular()
Descripción:	Constructor de la clase, por defecto.
Nombre:	E_Acceso_Vascular(Descripción)
Descripción:	Constructor de la clase, con parámetros.
Nombre:	Get_Descripcion()
Descripción:	Devuelve la descripción del acceso vascular.
Nombre:	Set_Descripcion(Descripción)
Descripción:	Método que cambia la descripción del acceso vascular.
Nombre:	Get_Eliminado()
Descripción:	Devuelve el estado del método si está eliminado o no.
Nombre:	Set_Eliminado()
Descripción:	Cambia el estado del método, en eliminado o no.

Tabla 3.18 Descripción de la clase E_Acceso_Vascular, CU Acceso Vascular

Nombre: M_Acceso_Vascular

Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Adicionar(E_Acceso_Vascular) : void
Descripción:	Método para adicionar el Acceso Vascular.
Nombre:	Editar(E_Acceso_Vascular)
Descripción:	Método para editar el acceso vascular.
Nombre:	Eliminar(E_Acceso_Vascular : E_Acceso_Vascular [])
Descripción:	Método para eliminar el acceso vascular.
Nombre:	Listar(offset, cant, campo, orden) : E_Acceso_Vascular []
Descripción:	Método para listar el acceso vascular, según algunos criterios.
Nombre:	Listar_Eliminados(offset, cant, campo, orden) : E_Acceso_Vascular []
Descripción:	Método para listar el los acceso eliminados.
Nombre:	Activar(E_Acceso_Vascular : E_Acceso_Vascular [])
Descripción:	Método para activar el acceso vascular.

Tabla 3.19 Descripción de la clase M_Acceso_Vascular, CU Acceso Vascular

Nombre: E_Entidad	
Tipo de clase: Entidad	
Atributo	Tipo
ID	int
Para cada responsabilidad:	
Nombre:	Get_ID()
Descripción:	Devuelve el identificador.
Nombre:	Set_ID(ID)

Descripción:	Cambia el ID del acceso vascular.
--------------	-----------------------------------

Tabla 3.20 Descripción de la clase E_Entidad, CU Acceso Vascular

3.3 Diseño de la BD

3.3.1 Diagrama Entidad Relación de la BD

El diagrama Entidad Relación, posee gran extensión en número de tablas, por tal motivo solo serán expuestas las tablas de los casos de uso más significativos. **Ver anexo III**

3.3.2 Descripción de las tablas de la BD

Descripción de las tablas relacionadas con las clases del caso de uso **Gestionar Movimiento**.

Nombre: e_pacientermd		
Descripción: Se encarga de guardar los datos iniciales del paciente que llega a la consulta por primera vez.		
Atributo	Tipo	Descripción
id_entrada	integer	Guarda el identificador de la entrada de cada paciente
id_paciente	integer	Guarda un identificador de cada paciente.
fecha_entrada	date	Guarda la fecha de entrada del paciente.
id_motivo_entrada	integer	Guarda el identificador de cada motivo de entrada que tiene el paciente.
a_nef_antes_analisis	integer	Guarda los días que hace que el paciente se atendió antes de la diálisis
peso_inicial	float	Guarda el peso inicial del paciente.
talla	float	Guarda la talla de cada paciente.
eritroproyectina	TINYINT	Guarda la eritropoyetina de cada paciente.
fecha_eritroproyectina	date	Guarda la fecha de eritropoyetina de cada paciente.

Tabla 3.21 Descripción de la Tabla e_pacientermd

Nombre: e_metodocfg		
Descripción: Tabla que guarda los métodos depuradores que pueden tener los pacientes.		
Atributo	Tipo	Descripción
id	integer	Identificador del método dialítico.
descripción	string	Guarda la descripción del método dialítico
eliminado	bool	Guarda si el método está eliminado o no
tipo	int	Identifica el tipo del método dialítico

Tabla 3.22 Descripción de la Tabla e_metodocfg

Nombre: er_paciente_metodormd		
Descripción: Entidad que relaciona la tabla paciente con la tabla método, es decir cada paciente lo relaciona con un método.		
Atributo	Tipo	Descripción
identrada	integer	Guarda el identificador del paciente con el método dialítico
idmetodo	integer	Guarda el identificador del método del paciente.
fechainicio	date	Guarda la fecha de inicio del método que se le indica.

Tabla 3.23 Descripción de la Tabla er_paciente_metodormd

Nombre: er_paciente_metodo_historicormd		
Descripción: Entidad que relaciona la tabla paciente con la tabla método histórico, a cada paciente se relaciona con un método histórico es decir un método que ya termino de usar.		

Atributo	Tipo	Descripción
id	integer	Identificador del método dialítico.
Id_entrada	integer	Guarda el identificador del paciente con el método dialítico.
idmetodo	integer	Guarda el identificador del método del paciente.
fechainicio	date	Guarda la fecha de inicio del método que ya no usa.
fecha_salida	date	Guarda la fecha de salida del paciente.
id_motivocambiometodo	integer	Guarda el id del motivo por que se cambio el método.
observaciones	varchar	Guarda las observaciones que tiene el método.

Tabla 3.24 Descripción de la Tabla er_paciente_metodo_historicormd

Nombre: e_accesovascularcfg		
Descripción: Tabla que guarda los accesos vasculares que pueden tener los paciente.		
Atributo	Tipo	Descripción
id	integer	Identificador del Acceso Vascular.
descripción	string	Guarda la descripción del Acceso Vascular
eliminado	bool	Guarda si el Acceso Vascular está eliminado o no.

Tabla 3.25 Descripción de la Tabla e_accesovascularcfg

Nombre: er_paciente_accesovascularrmd		
Descripción: Entidad que relaciona la tabla paciente con la tabla de acceso vascular, es decir cada paciente lo relaciona con un acceso vascular.		

Atributo	Tipo	Descripción
id	integer	Identificador del Acceso Vascular del paciente.
identrada	integer	Identificador del paciente con el Acceso Vascular.
idaccesovascular	integer	Guarda el identificador del acceso vascular del paciente.
idlocalizacion	integer	Guarda el identificador de la localización del acceso vascular.
fecharealizacion	date	Guarda la fecha de realización de dicho acceso vascular.
fechainicio	date	Guarda la fecha de inicio del acceso vascular que se le indicó.
activo	tinyint	Guarda si el acceso vascular es el que está actualmente en uso por el paciente.

Tabla 3.26 Descripción de la Tabla *er_paciente_accesovascularrmd*

Conclusiones

Este capítulo es de gran importancia, describe como está implementado el módulo. Además se muestra la descripción de las clases y operaciones necesarias, así como las de la Base de Datos y el Diagrama Entidad Relación para generar el modelo físico de esta. El sistema cumple con los requisitos funcionales preestablecido por los analistas del sistema y las descripciones de las tablas presentes en este.

También, se describió la integración del módulo con otros ya existentes, así como sus características, para mejorar la comprensión del trabajo realizado, y del sistema en general.

Conclusiones

Una vez concluida la investigación realizada, se dio cumplimiento al objetivo y las tareas planteadas, obteniéndose los siguientes resultados:

- Se realizó un estudio de las técnicas de programación, lenguajes y librerías para el desarrollo de la aplicación.
- Se implementó el módulo Registro de Métodos depuradores utilizando los patrones de diseños establecidos.
- Se obtuvo una interfaz gráfica del sistema orientada al usuario, agradable y fácil de operar.

Con la implementación del módulo, y el funcionamiento de este sistema se logra el incremento de la capacidad organizativa de los centros de diálisis del país, lo cual contribuye al aumento de la calidad de la asistencia médica a este tipo de Pacientes.

Se logra además una mejor atención a los Pacientes que se encuentran en Terapia Renal de Reemplazo en Diálisis y con ello agilizar el proceso y mejorar este servicio, solucionando las deficiencias existentes actualmente.

Recomendaciones

Luego de la presentación del estudio realizado que culmina con la implementación del Sistema Registro de Métodos Depuradores, perteneciente al producto Alas NefroRed, se listan a continuación una serie de recomendaciones para la ampliación, modificación, mejora y construcción de nuevas versiones de este sistema:

- Continuar el desarrollo del módulo Registro de Métodos Depuradores, incrementando en próximas iteraciones el número de reportes que el sistema puede generar, dado que en la presente investigación se tuvieron en cuenta solo los reportes fundamentales.
- Mantener sobre el sistema un estricto cumplimiento del proceso de mantenimiento y actualización periódica, en un periodo de 6 meses como máximo logrando así que se mantenga la fiabilidad y funcionamiento óptimo del sistema y de la información que se gestiona a través de el.
- Impartir cursos de capacitación a las personas que laboran en los servicios nefrológicos y trabajarán con la aplicación.

Referencias Bibliográficas

1. RED DE INFORMATIZACION EN NEFROLOGIA.
http://www.ucmh.sld.cu/rhab/vol5_num2/rhcm07206.htm , (11/3/2008)
2. Idém a la Referencia 1
3. Idém a la Referencia 1
4. Idém a la Referencia 1
5. Internet- Wikipedia
<http://es.wikipedia.org/wiki/Internet>, (11/3/06).
6. Idém a la Referencia 5
7. Aplicación Web- Wikipedia
http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web, (11/3/2008)
8. ¿Qué es una aplicación Web?
http://www.dimagin.net/es/contenido.php?t_id=6, (15/3/2008)
9. Aplicaciones Web.
http://www.intellia.com.mx/esp/servicios/aplicaciones_web_a_la_medida.php, (15/3/2008)
10. Aplicaciones Web, (Ventajas y Desventajas)
<http://www.avidos.net/blogold/aplicaciones-web/> , (15/3/2008)
11. Lenguaje de programación PHP-Monografías
<http://www.monografias.com/trabajos38/programacion-php/programacion-php.shtml?monosearch>,
(16/3/2008)
12. Idém a la Referencia 11
13. Lenguaje de programación PHP, (Ventajas y Desventajas)
<http://es.wikipedia.org/wiki/Php>, (16/3/2008)
14. Idém a la Referencia 13
15. Idém a la Referencia 13
16. Active Server Pages (ASP)- Wikipedia
http://es.wikipedia.org/wiki/Active_Server_Pages, (16/3/2008)
17. ¿Qué es ASP?
<http://www.maestrosdelweb.com/editorial/aspintro/>, (16/3/2008)

18. Idém a la Referencia 16
19. Idém a la Referencia 16
20. Perl-Wikipedia
<http://es.wikipedia.org/wiki/Perl>, (20/3/2008)
21. Introducción al Lenguaje PERL
<http://kataix.umag.cl/~mmarin/topinf/perl.html>, (20/3/2008)
22. Idém a la Referencia 21
23. Idém a la Referencia 23
24. JSP -Wikipedia
http://es.wikipedia.org/wiki/Java_Server_Pages(21/3/2008)
25. Ventajas del JSP
http://www.programacion.net/java/tutorial/servlets_jsp/1/ , (21/3/2208)
26. Java Script- Wikipedia
<http://es.wikipedia.org/wiki/JavaScript>, (23/3/2008)
27. ¿Que es Java Script?
<http://www.desarrolloweb.com/articulos/25.php>, (23/3/2008)
28. Funciones de los sistemas de gestión de bases de datos
<http://www3.uji.es/~mmarques/f47/apun/node39.html>, (23/3/2008)
29. SGBD - Wikipedia
<http://es.wikipedia.org/wiki/DBMS>, (23/3/2008)
30. PostgreSQL – Wikipedia
<http://es.wikipedia.org/wiki/PostgreSQL>, (24/3/2008)
31. Idém a la Referencia 30
32. Idém a la Referencia 30
33. Idém a la Referencia 30
34. Idém a la Referencia 30
35. SQL - Wikipedia
http://es.wikipedia.org/wiki/Celda_activa, (24/3/2008)
36. Idém a la Referencia 35

37. MySQL- Wikipedia

http://es.wikipedia.org/wiki/MySQL#Caracter.C3.ADsticas_adicionales, (27/3/2008)

38. Definición MySQL

<http://www.naturasoftware.com/main.php?f=tecnologia>, (27/3/2008)

39. Idém a la Referencia 37

40. Idém a la Referencia 38

41. IDE de programación – Wikipedia

http://es.wikipedia.org/wiki/Ambiente_integrado_de_desarrollo, (29/3/2008)

42. ¿Que son los IDE de programación?

<http://elcodigok.blogspot.com/2007/09/que-son-los-ide-de-programacin.html>, (4/4/2008)

43. Idém a la Referencia 41

44. Zend Studio

<http://www.desarrolloweb.com/articulos/1178.php>, (4/4/2008)

45. Idém a la Referencia 44

46. Idém a la Referencia 44

47. Sobre Notepad++

<http://notepad-plus.sourceforge.net/es/site.htm>, (4/4/2008)

48. Notepad++

<http://descargar.mp3.es/lv/group/view/kl36301/Notepad%2B%2B.htm>, (5/4/2008)

49. Nusphere PHPEd

<http://nusphere-phped.softonic.com/>, Nusphere PHPEd, (5/4/2008)

50. Nusphere PHPEd

http://www.filehungry.com/spanish/product/windows_software/programming/code_management_tools/nusphere_phpedit, (7/4/2008)

51. Idém a la Referencia 51

52. Idém a la Referencia 51

53. Idém a la Referencia 51

54. . Servicios Web - Wikipedia

http://es.wikipedia.org/wiki/Servicios_Web, (7/4/2008)

55. Servicios Web

<http://www.scribd.com/doc/1955854/Pregunta-4>, (7/4/2008)

56. XML,(Ventajas)- Wikipedia

http://es.wikipedia.org/wiki/XML#Ventajas_del_XML, (9/4/2008)

57. Idém a la Referencia 56

58. SOAP- Wikipedia

<http://es.wikipedia.org/wiki/SOAP>, (9/4/2008)

59. WSDL- Wikipedia

<http://es.wikipedia.org/wiki/WSDL>, (9/4/2008)

60. UDDI- Wikipedia

<http://es.wikipedia.org/wiki/UDDI>, (11/4/2008)

61. Servidor HTTP Apache- Wikipedia

http://es.wikipedia.org/wiki/Apache_http_server, (11/4/2008)

62. Idém a la Referencia 61

63. Idém a la Referencia 61

64. Framework- Wikipedia

<http://es.wikipedia.org/wiki/Framework>, (12/4/2008)

65. Idém a la Referencia 64

66. CodeIgniter, Framework PHP

<http://www.tufuncion.com/codeigniter>, (13/4/2008)

67. Idém a la Referencia 66

68. Idém a la Referencia 66

69. El framework Symfony, una introducción práctica.

<http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>, (16/4/2008)

70. Symfony Framework.

<http://geekymty.blogspot.com/2007/03/symfony-framework.html>, (16/4/2008)

71. Idém a la Referencia 69
72. Yahoo User Interface Library
<http://blog.unijimpe.net/yahoo-user-interface-library/>, (18/4/2008)
73. Idém a la Referencia 72
74. Prototype
<http://es.wikipedia.org/wiki/Prototype>, (18/4/2008)
75. Idém a la Referencia 74
76. Arquitectura cliente-servidor, Monografías
<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml#intro>, (20/4/2008)
77. Idém a la Referencia 76
78. Idém a la Referencia 76
79. Cliente-servidor- Wikipedia
<http://es.wikipedia.org/wiki/Cliente-servidor>, (20/4/2008)
80. Diseño de Aplicaciones Three Tier
<http://www.fpress.com/revista/Num9711/Nov97.htm>, (20/4/2008)
81. Idém a la Referencia 80
82. Arquitectura de tres capas con Access.
<http://dianakids.tripod.com/disen/bd01a.htm>, (23/4/2008)
83. Idém a la Referencia 80
84. ¿Que es una Arquitectura?
http://docente.ucol.mx/sadanary/public_html/bd/cs.htm#_¿Que_es_una_1, (25/4/2008)
85. Idém a la Referencia 84
86. Arquitectura de tres Niveles.
http://es.wikipedia.org/wiki/Arquitectura_de_tres_niveles, (26/4/2008)
87. ARQUITECTURAS Y TECNOLOGÍAS PARA EL DESARROLLO DE APLICACIONES WEB
<http://www.fing.edu.uy/inco/grupos/coal/investigacion/publicaciones/vp01.pdf>, (26/4/2008)
88. . Diseñando Aplicaciones Distribuidas, Monografías
<http://www.monografias.com/trabajos14/aplicacion-distrib/aplicacion-distrib.shtml>, (26/4/2008)

89. Idém a la Referencia 88

90. Patrones de Diseño en aplicaciones Web

http://java.ciberaula.com/articulo/disenio_patrones_j2ee/, (27/4/2008)

91. Modelo Vista Controlador.

http://es.wikipedia.org/wiki/Modelo_Vista_Controlador, (28/4/2008)

92. Idém a la Referencia 91

93. Idém a la Referencia 91

94. Arquitectura orientada a servicios.

http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios , (28/4/2008)

95. Idém a la Referencia 94.

Bibliografía

- ❖ Active Server Pages (ASP)- Wikipedia
http://es.wikipedia.org/wiki/Active_Server_Pages, (16/3/2008)
- ❖ Aplicaciones Web.
http://www.intellia.com.mx/esp/servicios/aplicaciones_web_a_la_medida.php, (15/3/2008)
- ❖ Aplicaciones Web, (Ventajas y Desventajas)
<http://www.avidos.net/blogold/aplicaciones-web/> , (15/3/2008)
- ❖ Aplicación Web- Wikipedia
- ❖ http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web, (11/3/2008)
- ❖ Arquitectura de tres capas con Access.
- ❖ <http://dianakids.tripod.com/disenio/bd01a.htm>, (23/4/2008)
- ❖ Arquitectura de tres Niveles.
http://es.wikipedia.org/wiki/Arquitectura_de_tres_niveles, (26/4/2008)
- ❖ Arquitectura cliente-servidor, Monografías
<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml#intro>, (20/4/2008)
- ❖ ARQUITECTURAS Y TECNOLOGÍAS PARA EL DESARROLLO DE APLICACIONES WEB
<http://www.fing.edu.uy/inco/grupos/coal/investigacion/publicaciones/vp01.pdf>, (26/4/2008)
- ❖ Arquitectura orientada a servicios.
http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios, (28/4/2008)
- ❖ Bradenbaugh, Jerry. Aplicaciones JavaScript. EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.), 2000
- ❖ Cliente-servidor- Wikipedia
<http://es.wikipedia.org/wiki/Cliente-servidor>, (20/4/2008)
- ❖ CodeIgniter, Framework PHP
<http://www.tufuncion.com/codeigniter>, (13/4/2008)
- ❖ Definición MySQL
- ❖ <http://www.naturasoftware.com/main.php?f=tecnologia>, (27/3/2008)

- ❖ Diseñado Aplicaciones Distribuidas, Monografías
<http://www.monografias.com/trabajos14/aplicacion-distrib/aplicacion-distrib.shtml>, (26/4/2008)
- ❖ Diseño de Aplicaciones Three Tier
<http://www.fpress.com/revista/Num9711/Nov97.htm>, (20/4/2008)
- ❖ DOQI Clinical Practice Guidelines for Hemodiálisis. s.l. : National Kidney Foundation, 1997
- ❖ El framework Symfony, una introducción práctica.
<http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>, (16/4/2008)
- ❖ Framework- Wikipedia
<http://es.wikipedia.org/wiki/Framework>, (12/4/2008)
- ❖ Funciones de los sistemas de gestión de bases de datos
<http://www3.uji.es/~mmarques/f47/apun/node39.html>, (23/3/2008)
- ❖ Health, National Institutes of. Annual Data Report. 1998.
- ❖ IDE de programación – Wikipedia
http://es.wikipedia.org/wiki/Ambiente_integrado_de_desarrollo, (29/3/2008)
- ❖ Introducción al Lenguaje PERL
<http://kataix.umag.cl/~mmarin/topinf/perl.html>, (20/3/2008)
- ❖ Introducción al Lenguaje PERL
<http://kataix.umag.cl/~mmarin/topinf/perl.html>, (20/3/2008)
- ❖ Java Script- Wikipedia
<http://es.wikipedia.org/wiki/JavaScript>, (23/3/2008)
- ❖ Jacobson, Ivar y Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software Volumen I. s.l. Pearson Educación, S.A., 2000.
- ❖ Jacobson, Ivar y Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software.
- ❖ Jeremy Levy, Julie Morgan, Edwina Brown. *OXFORD HANDBOOK OF DIALYSIS*. s.l. : OXFORD University Press, 2001 Volumen II. s.l. Pearson Educación, S.A., 2000.
- ❖ JSP -Wikipedia
http://es.wikipedia.org/wiki/Java_Server_Pages(21/3/2008)

- ❖ Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a Objetos Volumen I. Prentice Hall Hispanoamericana, S.A. 1999.
- ❖ Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a Objetos Volumen II. Prentice Hall Hispanoamericana, S.A. 1999.
- ❖ Lago, Ramiro. Patrones. Modelo-Vista-Controlador. [En línea] [Citado el: 1 de Marzo de 2008.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>
- ❖ Lenguaje de programación PHP-Monografías <http://www.monografias.com/trabajos38/programacion-php/programacion-php.shtml?monosearch,> (16/3/2008)
- ❖ Lenguaje de programación PHP, (Ventajas y Desventajas)
- ❖ <http://es.wikipedia.org/wiki/Php>, (16/3/2008)
- ❖ Modelo en Capas. [En línea] [Citado el: 1 de Marzo de 2008.] <http://www.cescaarequipa.edu.pe/jornada%20II.html>.
- ❖ Modelo Vista Controlador. http://es.wikipedia.org/wiki/Modelo_Vista_Controlador, (28/4/2008)
- ❖ MySQL- Wikipedia http://es.wikipedia.org/wiki/MySQL#Caracter.C3.ADsticas_adicionales, (27/3/2008)
- ❖ MySQL. [En línea] [Citado el: 2 de Marzo de 2008.] <http://www.lsi.us.es/cursos/cursophp/apuntes/tema4.pdf>
- ❖ MySQL. [En línea] [Citado el: 2 de Marzo de 2008.] <http://www.e-ghost.deusto.es/docs/TutorialMySQL.html>
- ❖ Notepad++ <http://descargar.mp3.es/lv/group/view/kl36301/Notepad%2B%2B.htm>, (5/4/2008)
- ❖ Nusphere PHPEd <http://nusphere-phped.softonic.com/>, Nusphere PHPEd, (5/4/2008)
- ❖ Nusphere PHPEd http://www.filehungry.com/spanish/product/windows_software/programming/code_management_tools/nusphere_phped, (7/4/2008)
- ❖ Patrones de Diseño en aplicaciones Web http://java.ciberaula.com/articulo/disenio_patrones_j2ee/, (27/4/2008)

- ❖ Perl-Wikipedia
<http://es.wikipedia.org/wiki/Perl>, (20/3/2008)
- ❖ PostgreSQL – Wikipedia
<http://es.wikipedia.org/wiki/PostgreSQL>, (24/3/2008)
- ❖ Prototype
<http://es.wikipedia.org/wiki/Prototype>, (18/4/2008)
- ❖ Pressman, R. Ingeniería de software. Un enfoque práctico Parte 1. Interamericana de España, S.A., 2002.
- ❖ Pressman, R. Ingeniería de software. Un enfoque práctico Parte 2. Interamericana de España, S.A., 2002
- ❖ ¿Qué es ASP?
- ❖ <http://www.maestrosdelweb.com/editorial/aspintro/>, (16/3/2008)
- ❖ ¿Que son los IDE de programación?
<http://elcodigok.blogspot.com/2007/09/que-son-los-ide-de-programacin.html>, (4/4/2008)
- ❖ ¿Que es Java Script?
<http://www.desarrolloweb.com/articulos/25.php>, (23/3/2008)
- ❖ ¿Qué es una aplicación Web?
http://www.dimagin.net/es/contenido.php?t_id=6, (15/3/2008)
- ❖ ¿Que es una Arquitectura?
http://docente.ucol.mx/sadany/public_html/bd/cs.htm#_¿Que_es_una_1, (25/4/2008)
- ❖ Servicios Web - Wikipedia
http://es.wikipedia.org/wiki/Servicios_Web, (7/4/2008)
- ❖ Servicios Web
<http://www.scribd.com/doc/1955854/Pregunta-4>, (7/4/2008)
- ❖ Servidor HTTP Apache- Wikipedia
http://es.wikipedia.org/wiki/Apache_http_server, (11/4/2008)
- ❖ SOAP- Wikipedia
<http://es.wikipedia.org/wiki/SOAP>, (9/4/2008)
- ❖ Sobre Notepad++
- ❖ <http://notepad-plus.sourceforge.net/es/site.htm>, (4/4/2008)

- ❖ SGBD - Wikipedia
<http://es.wikipedia.org/wiki/DBMS>, (23/3/2008)
- ❖ Symfony Framework.
<http://geekymty.blogspot.com/2007/03/symfony-framework.html>, (16/4/2008)
- ❖ SQL - Wikipedia
http://es.wikipedia.org/wiki/Celda_activa, (24/3/2008)
- ❖ UDDI- Wikipedia
<http://es.wikipedia.org/wiki/UDDI>, (11/4/2008)
- ❖ Ventajas del JSP
http://www.programacion.net/java/tutorial/servlets_jsp/1/ , (21/3/2208)
- ❖ WSDL- Wikipedia
<http://es.wikipedia.org/wiki/WSDL>, (9/4/2008)
- ❖ XML,(Ventajas)- Wikipedia
http://es.wikipedia.org/wiki/XML#Ventajas_del_XML, (9/4/2008)
- ❖ Yahoo User Interface Library
<http://blog.unijimpe.net/yahoo-user-interface-library/>, (18/4/2008)
- ❖ ¿Zend Studio
<http://www.desarrolloweb.com/articulos/1178.php>, (4/4/2008)

Glosario de Términos

D

Diálisis

Es un procedimiento que se realiza para retirar los elementos tóxicos (impurezas o desechos) de la sangre cuando los riñones no pueden hacerlo. La diálisis se puede llevar a cabo usando diferentes métodos: diálisis peritoneal y hemodiálisis.

Diálisis peritoneal

Es método dialítico que se realiza al utilizar la membrana peritoneal del cuerpo que se encuentra dentro del abdomen como membrana semipermeable. Se infunden soluciones especiales que ayudan a eliminar las toxinas, permanecen en el abdomen por un lapso de tiempo y luego se drenan.

Diálisis peritoneal Continua Ambulatoria (DPCA)

Se realiza manualmente mediante intercambios de líquidos. El tratamiento consiste en eliminar el líquido de diálisis usado (10-20 min) e infundir el líquido nuevo (10 min) permitiendo la movilidad del enfermo durante todo el día.

Diálisis Peritoneal Intermitente (DPI)

Técnica que consiste en la realización de múltiples cambios automatizados de corta duración durante periodos de ocho a diez horas; habitualmente es nocturna y la frecuencia varía de acuerdo a las necesidades metabólicas del paciente, aunque suele aplicarse de tres a cuatro veces por semana.

E

Enfermedad Renal Crónica (ERC)

Es una enfermedad que se presenta cuando los riñones ya no pueden funcionar al nivel necesario para la vida diaria. Este padecimiento se presenta a medida que la insuficiencia renal crónica progresa a tal punto en que la capacidad de los riñones para excretar los desechos, concentrar la orina y regular los electrolitos es menos del 10% de su capacidad normal.

Eritropoyetina

Hormona que es producida en los riñones que juega un papel importante en el metabolismo de la hemoglobina.

H

Hemodiálisis

Es un método dialítico que se realiza al hacer circular la sangre a través de filtros especiales por fuera del cuerpo. La sangre fluye a través de una membrana semipermeable (dializador o filtro), junto con soluciones que ayudan a eliminar las toxinas. La hemodiálisis requiere un flujo de sangre de 400 a 500 mililitros por minuto, por lo que utiliza formas especiales para llevar la sangre a los vasos sanguíneos (vías de accesos)

Hemodiafiltración

La hemodiafiltración es un método de diálisis renal que combina la hemodiálisis y la hemofiltración. La hemofiltración conjuntamente con la hemodiálisis es usada a veces en algunos centros como tratamiento crónico, esto es llamado hemodiafiltración. Al usar grandes cantidades de líquidos de sustitución (60 a 90 litros por tratamiento), las toxinas y los solutos pueden ser removidos del paciente.

Heparina

La heparina es una sustancia natural de la sangre que interfiere con el proceso de la coagulación sanguínea. Actúa sobre una sustancia llamada trombina, que juega un importante papel en la formación del coágulo en la sangre. Es una sustancia natural de la sangre que interfiere.

Hierro

Mineral que forma parte de la estructura de la hemoglobina.

Historia clínica

Documento que es utilizado por los profesionales de la salud para recoger información general de un paciente y anotar todas las acciones realizadas por los profesionales. Este consta de varias partes para su confección que puede variar en dependencia de la especialidad.

I

INFOMED

Nombre que identifica a la primera red electrónica cubana de información para la salud y surgió como parte de un proyecto del Centro Nacional de Información de Ciencias Médicas de Cuba. INFOMED es el Portal de Salud Cubano y la red de personas e instituciones que comparten el propósito de facilitar el acceso a la información de salud en Cuba.

Insuficiencia renal

Es una pérdida súbita de la capacidad del riñón para excretar los residuos, concentrar la orina y conservar los electrolitos.

L

Laboratorio de Tipaje Tisular

Local donde se realizan exámenes que determina los principales antígenos de histocompatibilidad de una persona.

M

Método dialítico o Método Depurador

Es un procedimiento que se realiza para retirar los elementos tóxicos (impurezas o desechos) de la sangre cuando los riñones no pueden hacerlo.

N

Nefrólogo

Especialista en el tratamiento de enfermedades de los riñones.

T

Terapia Renal de Reemplazo

Es un término usado para abarcar los tratamientos de soporte de la vida para la falta renal. Incluye: Hemodiálisis, Diálisis Peritoneal, Hemofiltración, Hemodiafiltración y Trasplante Renal.

U

Ultrafiltración

Ocurre en la barrera entre la sangre y el filtrado en el corpúsculo renal o la cápsula de Bowman en los riñones. La cápsula de Bowman contiene una densa red capilar llamada glomérulo. La sangre fluye en estos tubos capilares a través de una ancha arteriola aferente y se va a través de una más estrecha arteriola eferente.

V

Vía de acceso

Puede ser temporal o permanente. El primero toma la forma de catéteres para diálisis, que son catéteres de gran tamaño (tubos huecos de uso médico), colocados en las venas grandes, que pueden soportar flujos de sangre considerables. La mayoría de los catéteres se usan en situaciones de emergencia durante cortos períodos de tiempo. Sin embargo, los catéteres llamados catéteres en forma de túnel se pueden usar durante períodos prolongados, a menudo de semanas a meses. El acceso permanente es creado uniéndolos quirúrgicamente una arteria a una vena. Esto permite que la vena reciba sangre a alta presión. La conexión entre una arteria y una vena se puede hacer utilizando vasos sanguíneos (una fístula arteriovenosa o AVF por su sigla en inglés) o un puente sintético (injerto arteriovenoso o AVG, por sus siglas en inglés).

Anexos

Anexo I: Descripción de las clases.

Casos de uso Método Dialítico

Nombre: C_Paciente_Metodo	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	C_Paciente_Metodo()
Descripción:	Constructor de la Clase.
Nombre:	Cambiar_MetodoDialitico
Descripción:	Recibe diferentes parámetros del método para cambiar el método dialítico y devuelve un objeto al método correspondiente en la moduladora.
Nombre:	ListarMetodoHist(offset, Cant, OrdenadoPor, direccion)
Descripción:	Método que recibe diferentes parámetros para listar método históricos y devuelve un objeto al método correspondiente en la moduladora
Nombre:	ListarPacientesMetodo(offset, Cant, OrdenadoPor, direccion, idmetodo)
Descripción:	Método que recibe diferentes parámetros para listar pacientes por métodos dialíticos y devuelve un objeto al método correspondiente en la moduladora
Nombre:	Metodo_Activo()
Descripción:	Método que recibe diferentes parámetros para ver si el método está activo o no y devuelve un objeto al método correspondiente en la moduladora
Nombre:	CrearOpciones(EstructuraSalida)
Descripción:	Método auxiliar para crear las opciones de los dropdown

Tabla. A1.1 –. Descripción de la clase C_Paciente_Metodo, CU Método Dialítico.

Nombre: M_Paciente_Metodo	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Asignar_MetodoDialitico(E_Paciente_Metodo)
Descripción:	Método que asigna, al paciente un método dialítico.
Nombre:	Cambiar_MetodoDialitico(E_Paciente_Metodo, E_Paciente_Metodo_Historico)
Descripción:	Método que cambia el método dialítico que tiene el paciente.
Nombre:	ListarPacientesMetodo(EstructuraEntrada, E_Metodo)
Descripción:	Método que devuelve una lista de los pacientes organizados pro métodos, teniendo en cuenta estructuras de entradas.
Nombre:	Metodo_Activo(E_PacienteRMD)
Descripción:	Método que devuelve el método que actualmente tiene el paciente indicado.
Nombre:	ListarMetodoHist(EstructuraEntrada, E_PacienteRMD)
Descripción:	Método que listas los métodos históricos, es decir los método que ya se han realizados un determinado paciente.

Tabla. A1.2 –. Descripción de la clase M_Paciente_Metodo, CU Método Dialítico.

Nombre: E_Paciente_Metodo	
Tipo de clase: entidad	
Atributo	Tipo
Paciente	E_PacienteRMD
Método	E_Metodo
Fecha_Inicio	DateTime
Para cada responsabilidad:	

Nombre:	E_Paciente_Metodo()
Descripción:	Constructor de la clase, por defecto.
Nombre:	E_Paciente_Metodo(Paciente, Método)
Descripción:	Constructor de la clase, con parámetros.
Nombre:	TiempoDelMetodo()
Descripción:	Tiempo que se demora con el método.
Nombre:	Get_Paciente() : E_PacienteRMD
Descripción:	Método que devuelve el paciente.
Nombre:	Get_Metodo()
Descripción:	Devuelve el método depurador del paciente.
Nombre:	Get_Fecha_Inicio)
Descripción:	Método que devuelve la fecha de entrada del paciente.
Nombre:	Set_Paciente(E_PacienteRMD)
Descripción:	Método que cambia el paciente.
Nombre:	Set_Metodo(E_Metodo)
Descripción:	Método que cambia el método que se le asigno antes.
Nombre:	Set_Fecha_Inicio(fecha_inicio)
Descripción:	Método que cambia la fecha de inicio.

Tabla. A1.3 –. Descripción de la clase M_Paciente_Metodo, CU Método Dialítico.

Anexo II. Pantallas del Registro Nacional de Pacientes con ERC en el Programa de Diálisis.

1. Pantalla para agregar un paciente al Registro de Métodos Depuradores

The screenshot shows the 'Agregar - Paciente al registro de Métodos Depuradores' form in the NefroRed system. The interface includes a navigation menu on the left and a main form area with several sections:

- Datos Generales:** Includes fields for 'Nombre', 'Primer Apellido', 'Segundo Apellido', 'C.I.', and 'Sexo', each with a red asterisk indicating it is required. A 'Buscar Paciente' button is located to the right.
- Datos del Paciente en Métodos Depuradores:** Includes 'Fecha de Entrada' (set to 10/06/2008), 'Motivo de Entrada' (dropdown menu), 'At. Nef. A. Dialisis', 'Eritroproyectina' (radio buttons for 'Si' and 'No'), 'Talla', and 'Peso Inicial', all with red asterisks.
- Datos del Método:** Includes 'Método Dialítico' (dropdown menu) and 'Fecha Inicio' (set to 10/06/2008), both with red asterisks.
- Datos del Acceso Vascular:** Includes 'Acceso Vascular', 'Localizacion', and 'Fecha de Realizac.', all with red asterisks.

A note at the bottom states: '(*) Los campos señalados son de entrada obligatoria.' The form concludes with 'Aceptar' and 'Cancelar' buttons.

2. Pantalla para listar los pacientes que han sido ingresados al Registro de Métodos Depuradores.

Red Nacional de Nefrología - Mozilla Firefox

http://localhost/Nefrologia/web/nefrologia.php/rmd/c_hospitalizacion/MostrarPacientesIngresados

NefroRed
Red Nacional de Nefrología

Usuario: **Valdomero** Privilegio: **Médico Nefrólogo**

Inicio Módulos Ayuda Salir

Opciones de Menú:

- RERC**
 - Gestionar Pacientes
 - Reportes
- RAD**
 - Insumos
 - Riñón Artificial
 - Máquina de Reuso
 - Sillón de Hemodiálisis
 - Planta Agua
- RMD**
 - Gestionar Pacientes
 - Realizar Acceso
 - Accesos Realizados
 - Reportes
 - Listar Pacientes
- RPN**
 - Abrir HC

Listar pacientes ingresados:

Fecha Inicial: 2/4/2008 Fecha Final: 10/06/2008

Orden: Junio 2008 **Buscar**

Escoja los criterios de búsqueda y presione **"Buscar"**.

CI	Nombre	Ter Apellido	Sexo
62083010470	Magdalena	González	F

1 - 1 de 1

3. Pantalla para ver los reportes de las indicaciones.

The screenshot shows the NefroRed web application interface. The browser title is "Red Nacional de Nefrologia - Mozilla Firefox". The address bar shows the URL: `http://localhost/Nefrologia/web/nefrologia.php/rmd/e_indicacionesmd/ReportesIndicaciones`. The page header includes the NefroRed logo and navigation links: "Inicio", "Módulos", "Ayuda", and "Salir". The user is logged in as "Valdomero" with the privilege of "Médico Nefrólogo".

The main content area is titled "Reporte de Indicaciones". The "Reporte:" dropdown menu is set to "Flujo de Sangre". Below this, there are buttons for "Listar Pacientes" and "Buscar". A message prompts the user to "Escoja los criterios de búsqueda y presione 'Buscar'".

A table displays the search results:

Categoría	Cantidad Pacientes	Por ciento
<200	2	100
200-249	0	0
250-299	0	0
300-349	0	0
350-399	0	0
>=399	0	0

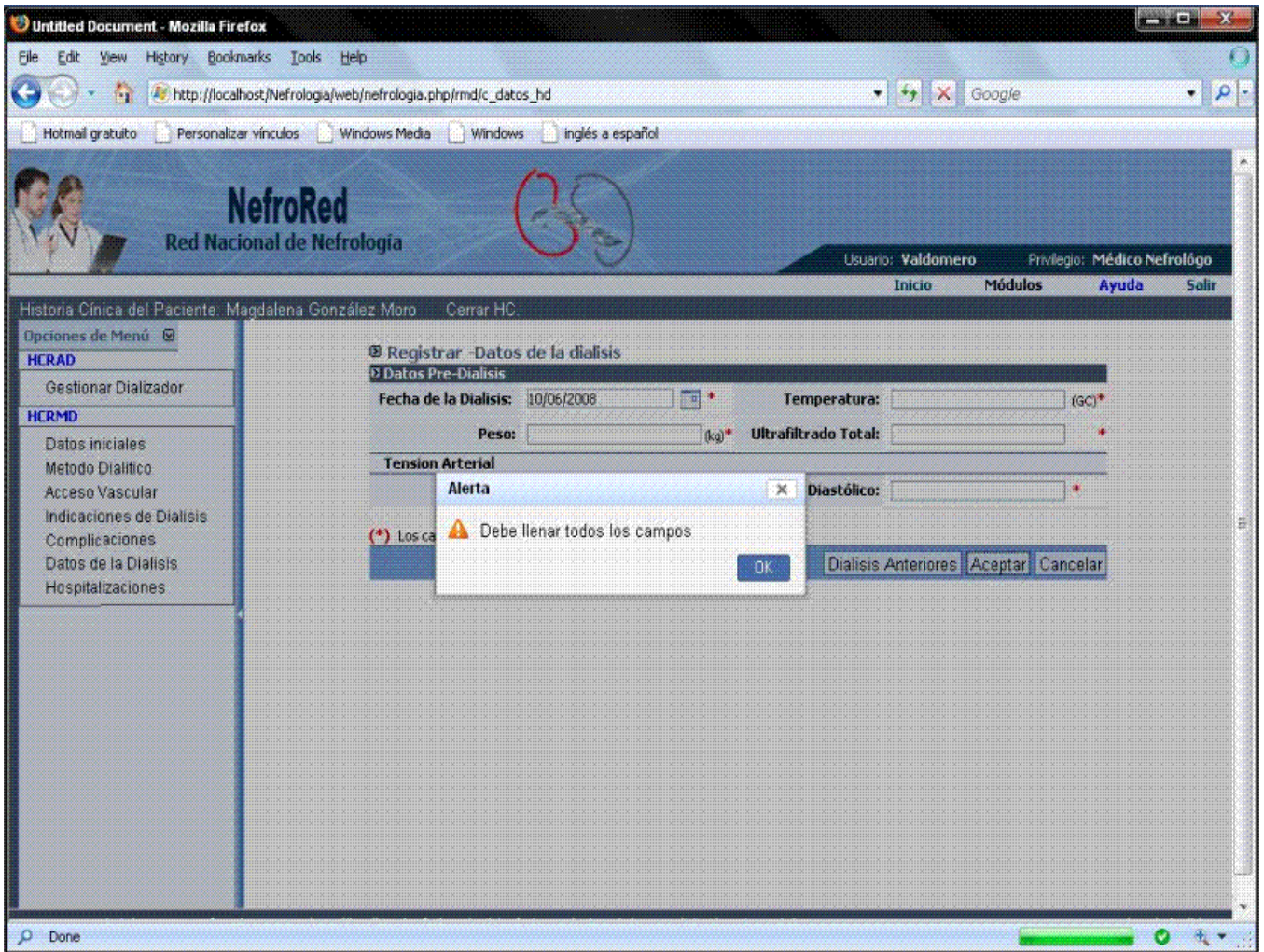
Below the table, the average value is shown: "Promedio: 7.5000".

A modal window titled "Pacientes" is open, displaying a table of patient details:

Nombre	Primer Apellido	Segundo Apellido
márcos	Frade	Frade
Magdalena	González	Moro

The modal window also shows a pagination control: "1 - 2 de 2" and the text "Pacientes en esos criterios".

4. Pantalla para Registrar datos de la diálisis.



Anexo III: Diagrama Entidad Relación de los casos de uso críticos

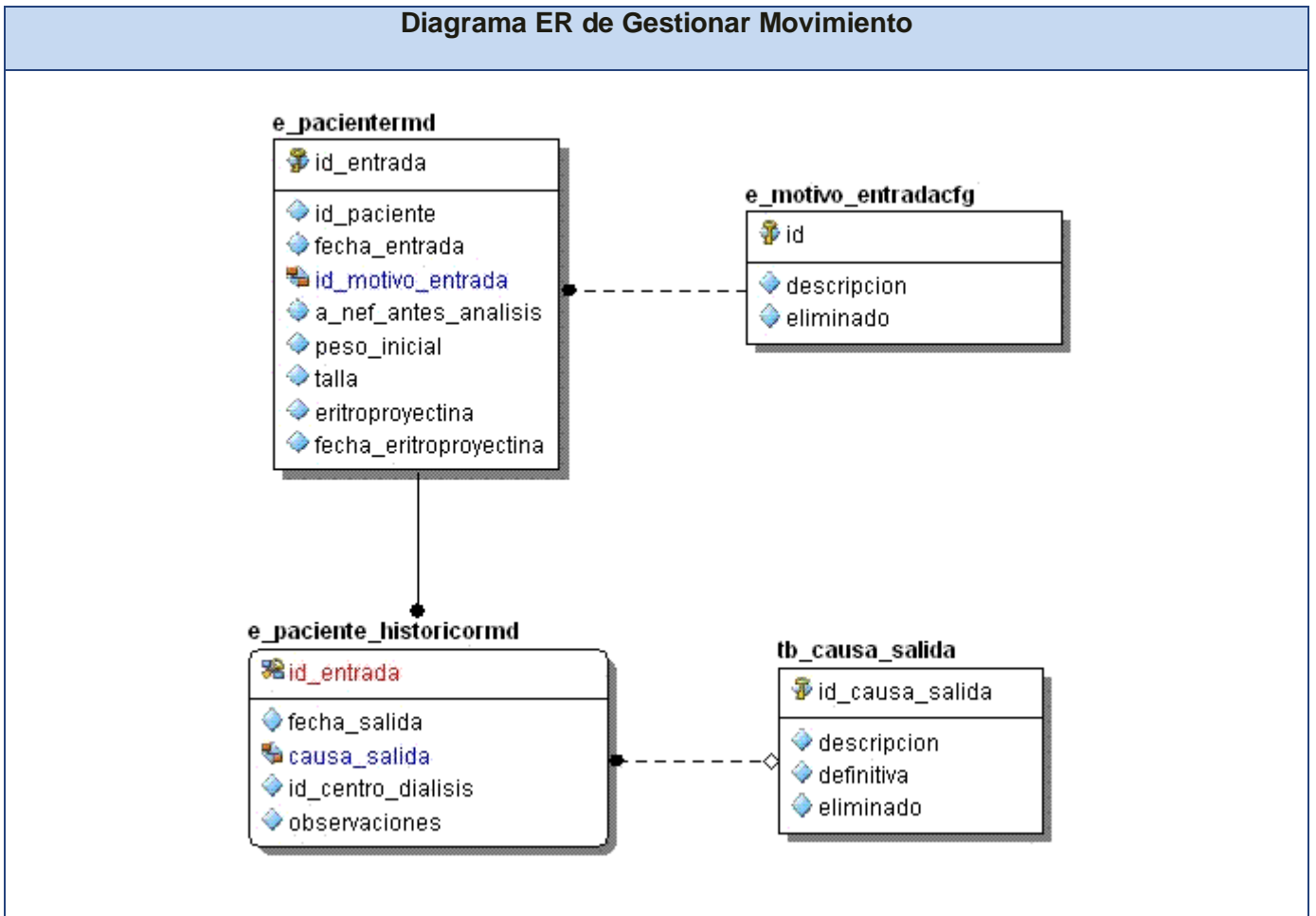


Fig. A3. 1 – Diagrama Entidad Relación del CU Gestionar Movimiento.

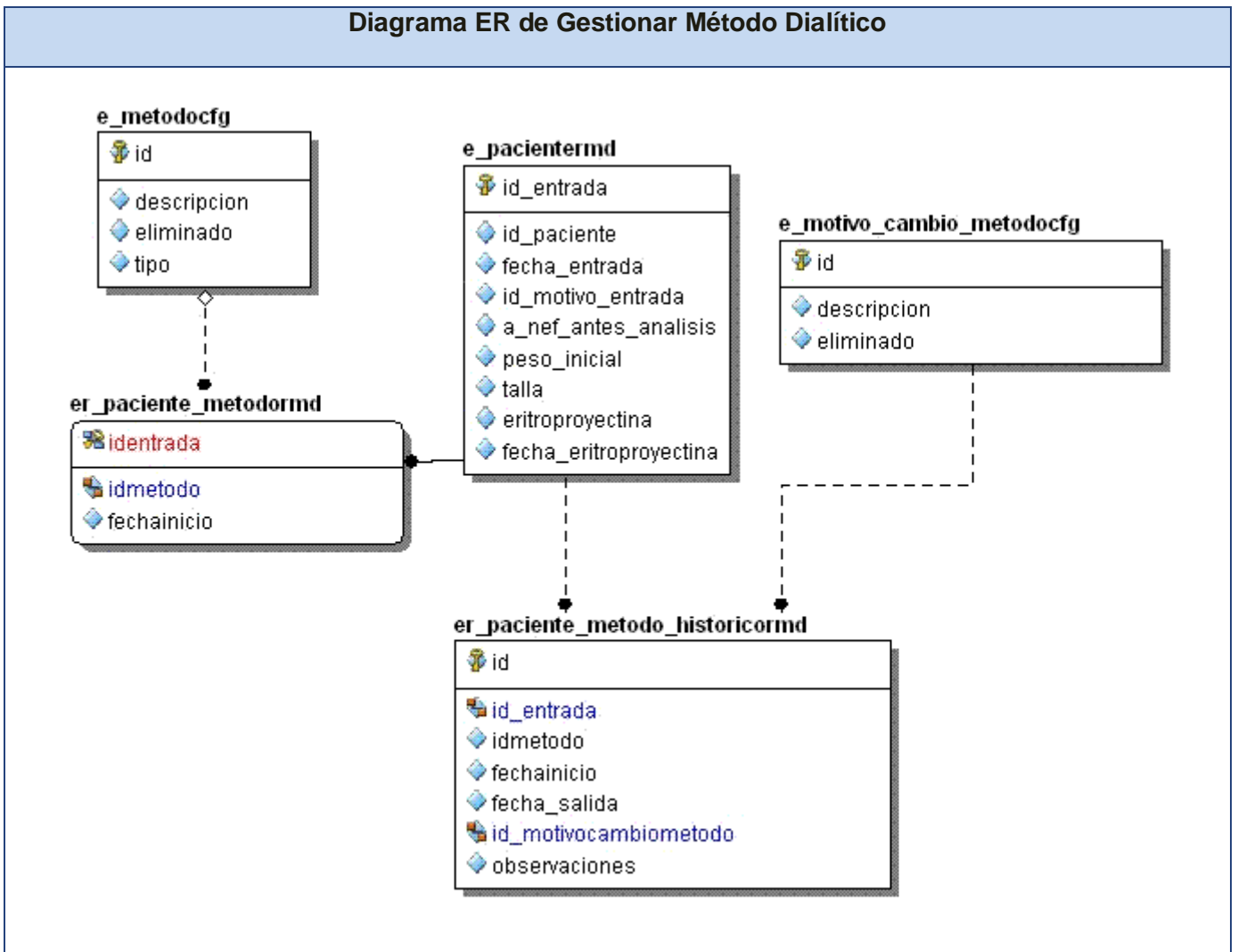


Fig. A3. 2 – Diagrama Entidad Relación del CU Método Dialítico.

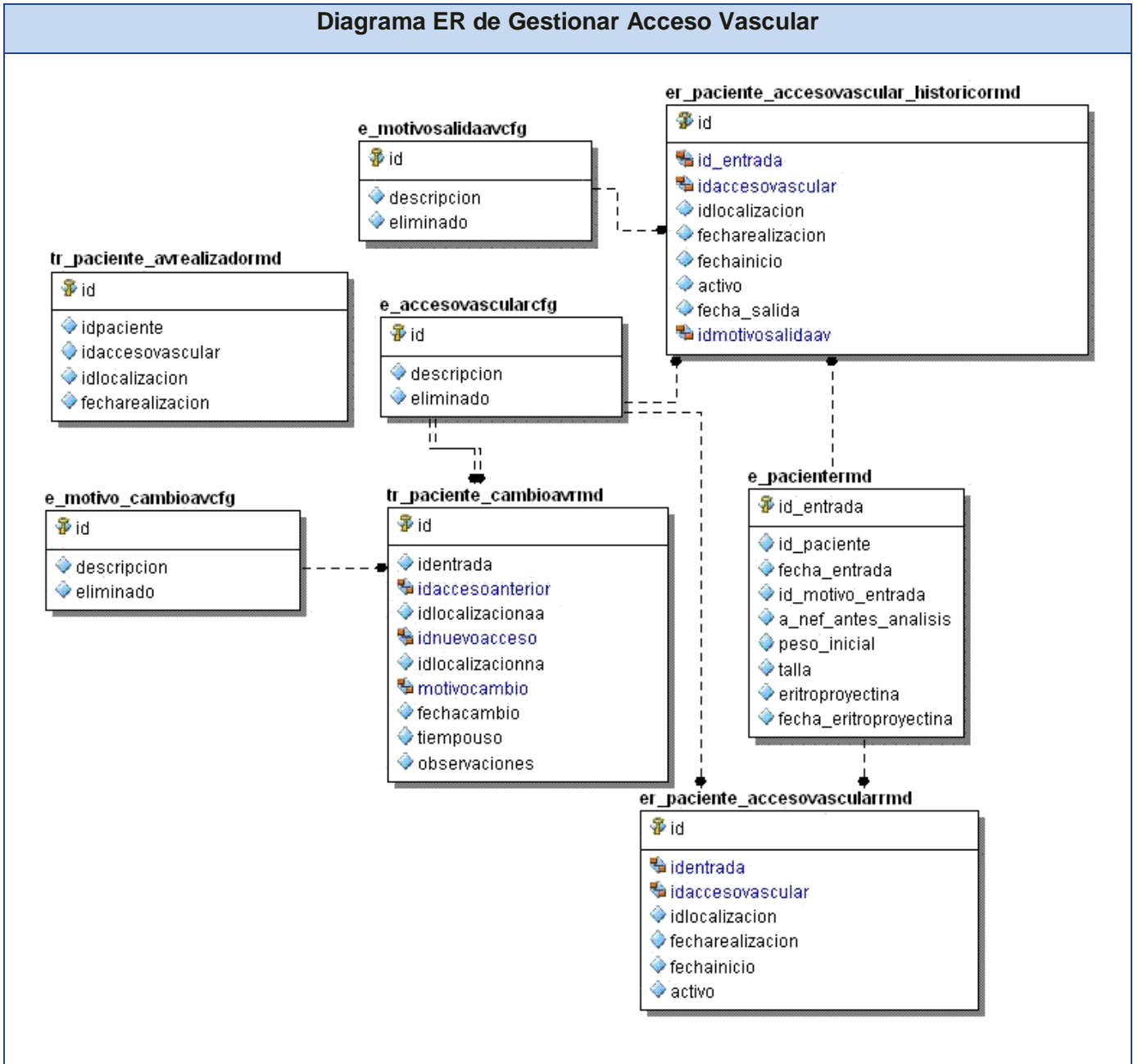


Fig. A3. 3 – Diagrama Entidad Relación del CU Gestionar Acceso Vascular.

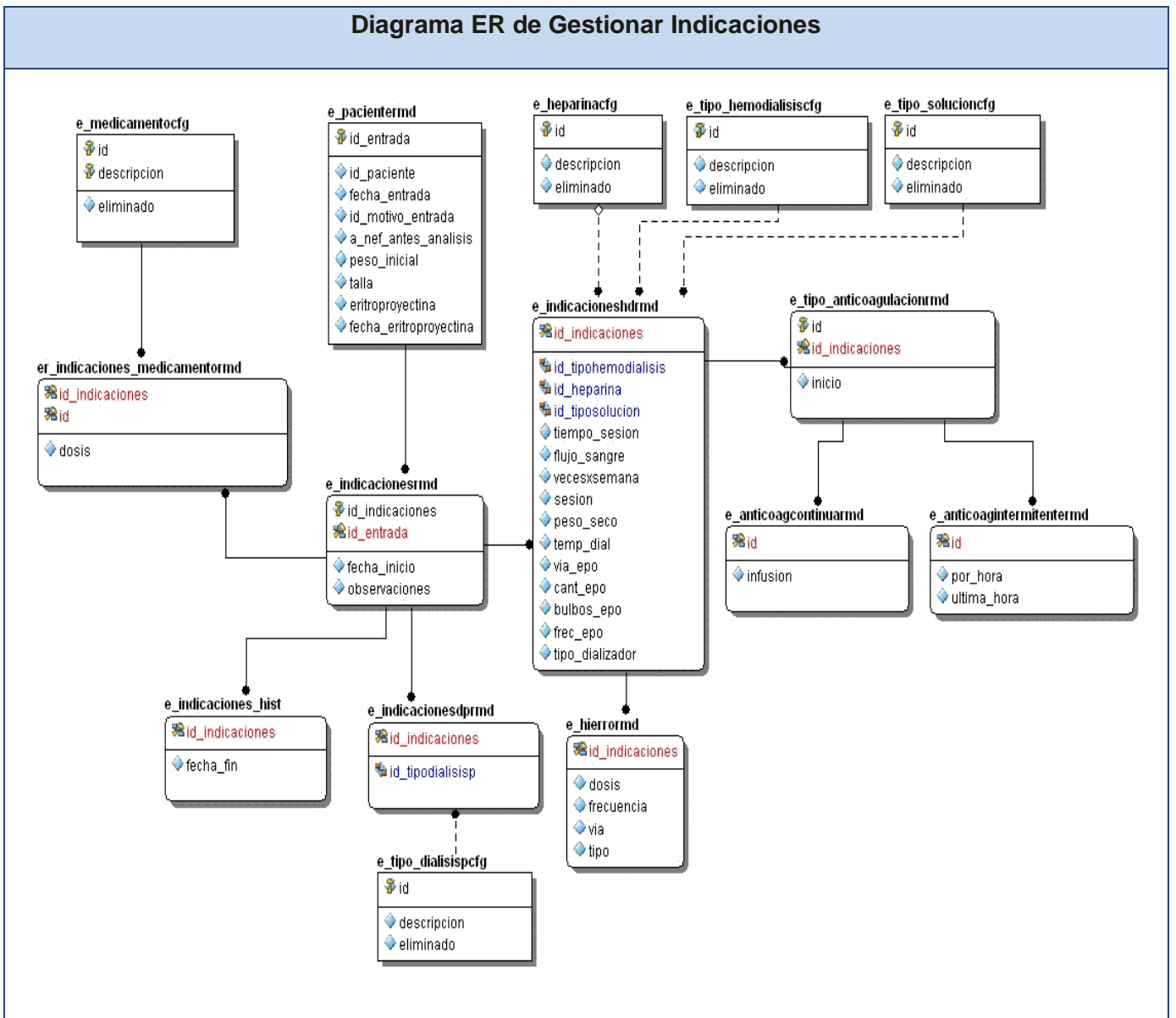


Fig. A3. 4 – Diagrama Entidad Relación del CU Gestionar Indicaciones.

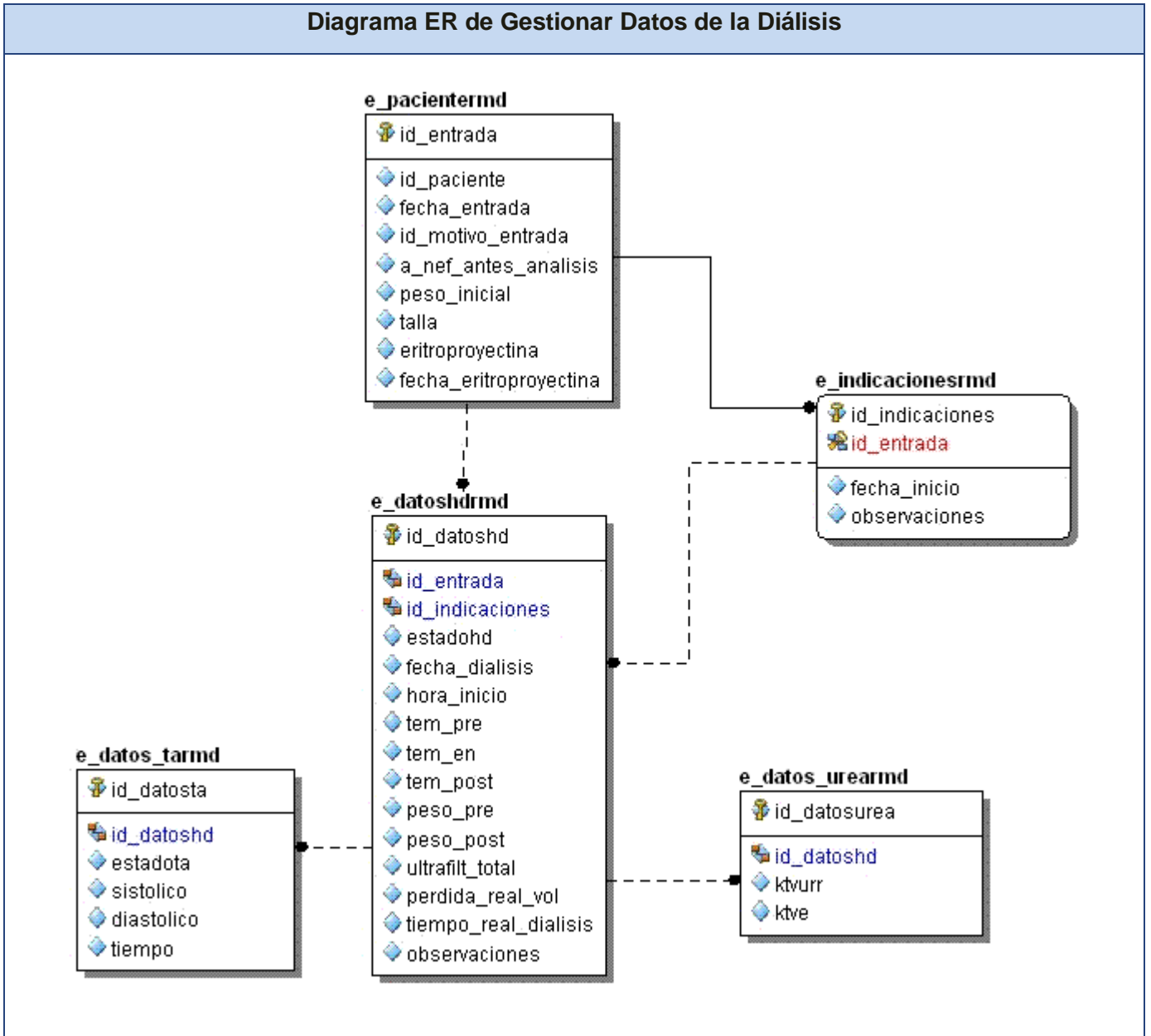


Fig. A3. 5 – Diagrama Entidad Relación del CU Gestionar Datos de la Diálisis.