

**Universidad de las Ciencias Informáticas**  
**Facultad 7**



**Título: Implementación del Módulo Diálisis del Sistema de Información Hospitalaria alas HIS**

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

**Autora: Angélica de la Caridad Vázquez Rúa**

**Tutores: Ing. Lisandro Calvis Pérez  
Ing. Alejandro Martínez Brito**

**La Habana, junio 2011**

**“ Año 53 de la Revolución”**

## DATOS DE CONTACTO

**Ing. Lisandro Calvis Pérez.** Graduado de Ingeniería Industrial en la Universidad Oscar Lucero de Holguín en el 2008. Posee la categoría docente de Instructor. Durante su trabajo como profesor ha impartido las asignaturas Matemática 1 y Matemática 2. Se ha desempeñado como jefe de la asignatura Matemática 1. Actualmente imparte la asignatura Matemática 1.

En la vinculación con la producción pertenece al Departamento de Sistemas Especializados en Salud (SES) del Centro de Informática Médica (CESIM), donde se desempeña como Gestor de la Configuración y Planificador.

E-mail: [lisandroc@uci.cu](mailto:lisandroc@uci.cu)

**Ing. Alejandro Martínez Brito.** Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2009. Instructor recién graduado en adiestramiento. Durante su trabajo como profesor ha impartido cursos pertenecientes a la asignatura Práctica Profesional. Actualmente se desempeña como tutor de estudiantes en la asignatura Práctica Profesional.

En la vinculación con la producción pertenece al Departamento de Sistemas Especializados en Salud (SES) del Centro de Informática Médica (CESIM) y específicamente trabaja en el desarrollo del proyecto NefroRed donde se desempeña como Líder de Proyecto.

E-mail: [ambrito@uci.cu](mailto:ambrito@uci.cu)

## RESUMEN

Cuba, con un proyecto de desarrollo que tiene como pilares la justicia social, la participación popular, la equidad y la solidaridad. Ha diseñado e iniciado la aplicación de estrategias que permiten convertir los conocimientos y las tecnologías de la información y las comunicaciones en instrumentos a disposición del avance y las profundas transformaciones revolucionarias, entre estas evoluciones se ha centrado en el sector de la salud.

Este trabajo tiene como objetivo implementar el Módulo de Diálisis del Sistema de Información Hospitalaria alas HIS, con funcionalidades como el seguimiento de los datos durante la diálisis, la gestión de la indicación que realiza el médico nefrólogo, la gestión de los riñones artificiales, entre otras.

Para el desarrollo de la aplicación se utilizó como metodología de desarrollo de software: el Proceso Unificado de Desarrollo (RUP), Visual Paradigm como herramienta CASE y el Lenguaje Unificado de Modelado (UML). Java Server Faces (JSF) como framework para desarrollo de aplicaciones web, PostgreSQL como sistema gestor de Base de Datos y JBoss Server como servidor de aplicación. Se utilizó además, Eclipse como entorno de desarrollo y el lenguaje de programación Java, el cual es robusto, seguro y multiplataforma.

Con la culminación de este módulo se podrá atender mayor cantidad de pacientes diariamente, facilitará una implementación más rápida del programa médico de atención en todos los hospitales nefrológicos de Cuba, con la consiguiente mejora en los procesos a los que se refiere.

**Palabras clave:** Diálisis, Nefrología, Consulta Externa

# TABLA DE CONTENIDOS

---

<b>INTRODUCCIÓN:</b> .....	<b>1</b>
<b>CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>4</b>
1.1 DESCRIPCIÓN DEL SISTEMA DE INFORMACIÓN HOSPITALARIA ALAS HIS. ....	4
1.2 DESCRIPCIÓN DEL MÓDULO DE DIÁLISIS. ....	4
1.3 METODOLOGÍAS DE DESARROLLO DE SOFTWARE Y HERRAMIENTAS PARA EL MODELADO.....	4
1.4 TECNOLOGÍAS Y HERRAMIENTAS A UTILIZAR. ....	11
1.4.1 LENGUAJE DE PROGRAMACIÓN. ....	14
1.4.2 HERRAMIENTAS. ....	14
1.2.3 SISTEMAS DE GESTIÓN DE BASE DE DATOS. ....	16
<b>CAPÍTULO II: DESCRIPCIÓN DE LA ARQUITECTURA</b> .....	<b>18</b>
2.1 REQUISITOS NO FUNCIONALES. ....	18
2.2 DESCRIPCIÓN DE LA ARQUITECTURA, FUNDAMENTACIÓN.....	21
2.3 ANÁLISIS DE POSIBLES IMPLEMENTACIONES, COMPONENTES O MÓDULOS YA EXISTENTES Y QUE PUEDAN SER REHUSADOS. ESTRATEGIAS DE INTEGRACIÓN.....	22
2.4 SEGURIDAD. ....	23
2.5 VISTA DE DESPLIEGUE. ....	25
2.6 ESTRATEGIAS DE CODIFICACIÓN. ESTÁNDARES Y ESTILOS A UTILIZAR.....	25
<b>CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA</b> .....	<b>30</b>
3.1 VALORACIÓN CRÍTICA DEL DISEÑO PROPUESTO POR EL ANALISTA. ....	30
3.2 DESCRIPCIÓN DE LAS NUEVAS CLASES U OPERACIONES NECESARIAS. ....	30
3.3 MODELO DE DATOS. ....	56
3.3.1 DESCRIPCIÓN DE LAS TABLAS DE LA BASE DE DATOS. ....	58
3.4 DIAGRAMA DE COMPONENTE. ....	66
<b>CONCLUSIONES</b> .....	<b>73</b>
<b>RECOMENDACIONES</b> .....	<b>74</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	<b>75</b>
<b>GLOSARIO DE TÉRMINOS</b> .....	<b>80</b>

## INTRODUCCIÓN:

Cuba está consciente de que para mejorar su economía, lograr una sociedad más eficaz, eficiente y competitiva debe aplicar la informatización en todas sus esferas y procesos, convencida de que para los países subdesarrollados resulta imprescindible el logro de este propósito.

En este sentido, se ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las Tecnologías de la Información y las Comunicaciones (TIC) y lograr una cultura informática, como una de las características imprescindibles del hombre nuevo, lo que facilitaría a la sociedad acercarse más hacia un desarrollo sostenible.

Para suplir estas necesidades surge la Universidad de las Ciencias Informáticas, teniendo dos objetivos principales: informatizar el país y desarrollar la industria del Software para contribuir al desarrollo económico del mismo.

La Nefrología es una especialidad médica, una rama de la medicina interna, que estudia la organización y la función renal. Así como las enfermedades renales (nefropatías) y sus repercusiones, las cuales son reconocidas como un problema de salud y un reto internacionalmente, por su complejidad, frecuencia y crecimiento epidémico.

La diálisis es un proceso por medio del cual se produce un filtrado artificial de la sangre. En este, se retiran los elementos tóxicos del torrente sanguíneo cuando los riñones han perdido su capacidad. Esta puede ser usada para pacientes muy enfermos que han perdido repentinamente su función renal (fallo renal agudo) o para pacientes absolutamente estables que han perdido permanentemente su función renal (enfermedad renal en estado terminal).

El presupuesto dedicado a estos enfermos es muy alto y existen enormes disparidades en su atención entre los países pobres y ricos. Un tratamiento de diálisis cuesta entre 100 y 200 dólares por sesión y en el año el enfermo debe hacerse entre 156 y 160 sesiones, sin contar el costo de la tecnología, que incluye una planta para el proceso del agua, riñones artificiales o generadores de diálisis y medicamentos de última generación, entre otros recursos.

Cuba realiza un esfuerzo mantenido en el terreno de la Nefrología y la atención al problema de salud internacional que representa la Enfermedad Renal Crónica terminal con requerimientos dialíticos, pese a

# INTRODUCCIÓN

---

su condición de pequeño país en vías de desarrollo. La prioridad dada a la construcción de nuevas unidades, para acercar la atención a donde viven los pacientes, la adquisición de modernas tecnologías y la formación de nuevos especialistas, es una demostración de la importancia concedida por el Estado cubano a la misma, en términos de equidad y justicia. Con la particularidad de que los medicamentos y las tecnologías necesarias son comprados por terceros países a causa del bloqueo impuesto por los Estados Unidos de América, como consecuencia los costos son mayores. A pesar de esto, son servicios que se brindan gratuitamente a la población.

Diariamente en cada uno de los servicios nefrológicos del país se recogen los resultados de las diálisis que se le realizan a los pacientes, con esta información se elabora mensualmente un informe estadístico. Es enviado al Instituto Nacional de Nefrología (INEF) mediante el correo electrónico o vía telefónica, para la actualización del archivo de datos nacional, que se encuentra en dicho centro. Todo este proceso se realiza de forma manual, lo que trae como consecuencia que se puedan producir errores humanos, que no se obtenga la información de estos pacientes enfermos renales en tiempo real, ni con la exactitud requerida, ni se pueda establecer una comparación a nivel municipal, provincial y nacional.

En el curso 2007-2008 en la Universidad de las Ciencias Informáticas se desarrolló una primera versión de un sistema (alas NEFRORED v1.0). Ésta gestiona la información relacionada con los procesos de diálisis, específicamente la asociada con la indicación que realiza el médico nefrólogo, el seguimiento durante la diálisis y la gestión de reportes relacionados con dicho proceso. Luego del desarrollo de esta versión el cliente identificó nuevas funcionalidades con las que debería contar el sistema, como son: la gestión de los riñones artificiales y las camas en las salas de hemodiálisis, la planificación de los equipos para este tratamiento y la generación de nuevos reportes.

Actualmente se está desarrollando en la Universidad de las Ciencias Informáticas el Sistema de Información Hospitalaria alas HIS. El cual tiene como atributo fundamental una historia clínica electrónica (HCE), que incluye toda la documentación, imágenes e información que se genera en torno a un paciente: datos de enfermería, información sobre pruebas, diagnósticos, sistemas de peticiones, resultados de exámenes, citas, entre otros. Estos datos aparecen totalmente digitalizados, descentralizados y almacenados con seguridad, cumpliendo con los principios de ética médica.

Este sistema no cuenta con un componente para el proceso de diálisis. No se puede integrar con la aplicación alas NEFRORED (donde se gestiona una parte considerable de dicho proceso) ya que las

# INTRODUCCIÓN

---

arquitecturas son incompatibles. Por lo cual se hace necesario desarrollar un nuevo módulo para el sistema alas HIS, que incluya las funcionalidades que están implementadas en alas NEFRORED para el proceso de diálisis y las nuevas necesidades determinadas por el cliente. En el curso 2009-2010 se realizó el Análisis y Diseño del mismo.

Por lo antes planteado se tiene como **problema a resolver**: ¿Cómo hacer funcional el diseño del proceso de gestión de la información de diálisis en los servicios nefrológicos de Cuba?

Este problema se enmarca en el **objeto de estudio**: el proceso de desarrollo de software de la especialidad de Nefrología del Sistema de Información Hospitalaria alas HIS. Se tiene como **campo de acción**: el proceso de implementación y pruebas del Módulo de Diálisis del Sistema de Información Hospitalaria alas HIS.

Para la solución del problema se plantea como **objetivo general**: Implementar el Módulo de Diálisis del Sistema de Información Hospitalaria alas HIS a partir del diseño realizado.

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

1. Describir los procesos de diálisis propuestos por el analista.
2. Describir la arquitectura definida por el Departamento Gestión Hospitalaria.
3. Refinar el diseño de la base de datos correspondiente al Módulo de Diálisis del sistema alas HIS.
4. Realizar la implementación del Módulo de Diálisis del sistema alas HIS, utilizando los patrones establecidos en el diseño.
5. Realizar las pruebas necesarias para garantizar el correcto funcionamiento de la aplicación implementada.

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

---

## Capítulo I: Fundamentación Teórica

En el presente capítulo se realiza una descripción de las herramientas a usar en la implementación del Módulo de Diálisis del Sistema de Información Hospitalaria alas HIS.

### **1.1 Descripción del Sistema de Información Hospitalaria alas HIS.**

El Sistema de Información Hospitalaria alas HIS, está orientado a satisfacer las necesidades de generación de información, para almacenar, procesar y reinterpretar datos médico-administrativos de una institución hospitalaria. Tiene como objetivo agilizar el proceso de atención al paciente y elevar la calidad de los servicios médicos, minimizando los inconvenientes burocráticos que enfrentan los pacientes. Está compuesto por una gran variedad de subsistemas que se encargan de gestionar la información en las diferentes áreas del hospital, lo que permite la optimización de los recursos humanos y materiales. Una de estas áreas es Nefrología, donde se realizan varios procesos, dentro de los cuales se encuentra el proceso de diálisis.

### **1.2 Descripción del Módulo de Diálisis.**

El Módulo de Diálisis perteneciente al sistema alas HIS, se encuentra actualmente en desarrollo, gestiona la información que se genera en los procesos de diálisis. Cuenta con funcionalidades como el seguimiento de los datos durante la diálisis, la gestión de la indicación que realiza el médico nefrólogo, la gestión de los riñones artificiales y de las camas en las salas de hemodiálisis, la planificación de los equipos para el tratamiento de diálisis y la generación de reportes relacionados con dicho tratamiento.

### **1.3 Metodologías de desarrollo de software y herramientas para el modelado.**

Una Metodología de Desarrollo de Software es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software. Una metodología puede seguir uno o varios modelos del ciclo de vida, es decir, indica lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo.

La metodología a utilizar es el Proceso Unificado de Desarrollo (RUP), fue propuesta por el Departamento de Gestión Hospitalaria debido a que integra todos los aspectos a tener en cuenta durante todo el ciclo de vida del software. Con el objetivo de asegurar la producción de software de calidad, dentro de plazos y presupuestos predecibles, define quién debe hacer qué, cuándo y cómo debe hacerlo.



# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

---

Además se propone utilizar como lenguaje de modelado el Lenguaje Unificado de Modelado (UML) y como notación para la descripción de los procesos del negocio a informatizar, la Notación para el Modelado de Procesos de Negocio (BPMN). RUP es el resultado de varios años de trabajo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. En este se han agrupado las actividades en grupos lógicos en los que se definen nueve flujos de trabajo principales. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como flujos de apoyo. Las principales características de esta metodología son:

- Dirigido por casos de uso: Los casos de uso reflejan lo que el cliente necesita, lo cual se capta al modelar el negocio y se representa a través de los requerimientos. Luego los casos de uso guían el proceso de desarrollo, ya que como resultado de los diferentes flujos de trabajo, los modelos que se obtienen representan la realización de los casos de uso, es decir, estos se van haciendo por cada caso de uso.
- Centrado en la arquitectura: La arquitectura muestra la visión común del sistema, RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.
- Iterativo e Incremental: Cada fase se desarrolla en iteraciones, una iteración implica actividades de todos los flujos de trabajo, aunque de algunos más que de otros. RUP propone dividir el trabajo en partes más pequeñas o mini proyectos, donde cada uno de ellos es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

RUP agrupa las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales, basándose en el UML.

- Modelamiento del negocio: describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- Requerimientos: define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- Análisis y diseño: describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

---

- Implementación: define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán, la ubicación de los componentes en ellos y la estructura de capas de la aplicación.
- Prueba (Testeo): busca los defectos a lo largo del ciclo de vida del software.
- Instalación: produce el despliegue del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- Administración del proyecto: involucra actividades con las que se busca elaborar un producto que satisfaga las necesidades de los clientes.
- Administración de configuración y cambios: describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- Ambiente: contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización. (1)

RUP divide el proceso de desarrollo en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

Durante la fase de inicio las iteraciones ponen mayor énfasis en las actividades de modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis y diseño y una parte de la implementación orientado a la línea de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones. Para cada iteración se seleccionan algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se ejecutan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se posee un producto preparado para su entrega a la comunidad de usuarios. (2)

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

De forma general, el objetivo fundamental de RUP es producir software de alta calidad, que cumpla con los requerimientos de los usuarios dentro de la planificación y presupuesto establecidos.

Como RUP es un proceso, en su modelación define como sus principales elementos:

Los roles, que responden a la pregunta ¿quién?; las actividades que responden a la pregunta ¿cómo?; los artefactos, que responden a la pregunta ¿qué? y los flujos de trabajo, que responde a la pregunta ¿cuándo?

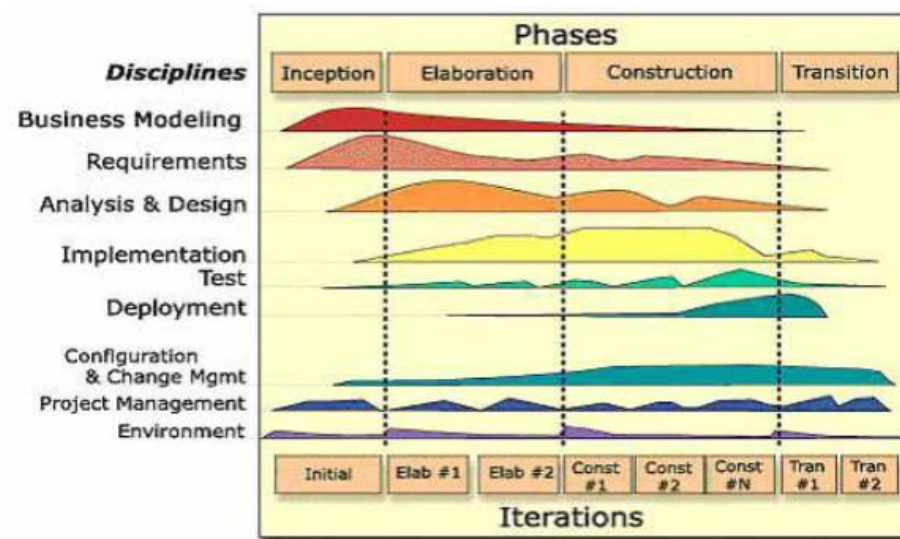


Figura 1. Fases y Flujos de trabajo del RUP.

## Descripción del Flujo de Trabajo de Implementación.

En este flujo de trabajo se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y otros. Se deben realizar los tests de unidad: cada implementador es responsable de testear las unidades que produzca. El resultado final de este flujo de trabajo es un sistema ejecutable.

En cada iteración se ejecutarán las siguientes actividades:

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

---

- Planear qué subsistemas deben ser implementados y en qué orden deben ser integrados, formando el Plan de Integración.
- Cada implementador decide en qué orden implementa los elementos del subsistema. Si encuentra errores de diseño, los notifica.
- Se testean los subsistemas individualmente.
- Se integra el sistema siguiendo el plan.

La estructura de todos los elementos implementados forma el Modelo de Implementación. (3)

Los propósitos fundamentales de este flujo son:

- Planificar las integraciones de sistema necesarias en cada iteración.
- Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue.
- Implementar las clases y subsistemas encontrados durante el diseño.
- Probar los componentes individualmente, y a continuación integrarlos, compilándolos y enlazándolos en uno o más ejecutables.

Intervienen los siguientes trabajadores:

- Arquitecto de Software (genera el Modelo de Implementación).
- Implementador (crea los artefactos: Elementos de Implementación, Subsistema de Implementación y Elementos de Pruebas).
- Integrador (compone el artefacto: Plan de Integración).
- Revisor Técnico (revisa los artefactos generados en este flujo). (4)

En este flujo se crea un artefacto de gran importancia en el ciclo de vida del software, el Modelo de Implementación, el mismo consiste en una visión general de lo que tiene que ser implementado, y un apartado para cada iteración con los componentes y subsistemas a implementar durante esa iteración, así como los resultados software (builds) que se han de obtener y el testeo que se ha de realizar sobre ellos.

También se elaboran:

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

---

- Diagrama de Componentes: Incluye a los componentes y archivos que se utilizan para ensamblar y hacer disponible el sistema físico.
- Diagrama de Despliegue: Contiene los nodos que forman la topología hardware, sobre la que se ejecuta el sistema y la distribución de las partes del sistema en ellos.

Esta es una de las fases más técnicas del sistema, en la cual hay una interacción y comunicación permanente entre el equipo de analistas, diseñadores y el equipo de programadores. La participación del usuario es mínima, estando limitada a coadyuvar en las pruebas de programas o módulos. Un aspecto muy importante a considerar es la definición de un estándar de codificación de los programas, esto comprende: la estructuración del programa, funciones, definición de clases, métodos, definición de variables globales, definición de variables locales, tablas internas, tablas temporales, uso de encabezados en los programas, documentación interna de programas, etc. En este flujo de trabajo se desempeña el rol de implementador.

## **Descripción del Flujo de Trabajo Prueba.**

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de existencia de fallos humanos aumentan. Los errores pueden aparecer desde el primer momento del proceso en el que los objetivos pueden estar especificados de forma errónea e imperfecta; así en los posteriores pasos del diseño y desarrollo. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad.

Las pruebas constituyen una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente. (5) Constituye un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

La prueba está enfocada principalmente en la evaluación y determinación de la calidad del producto.

## **Tipos de prueba**

Cualquier producto de ingeniería se puede probar de dos formas:

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

---

Pruebas de caja negra: encargadas de comprobar que cada función del software es operativa. Las pruebas de caja negra se llevan a cabo sobre la interfaz del software. Tratan de demostrar que las entradas se manejan de forma adecuada y que se produce el resultado esperado.

Buscan encontrar errores en cinco categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos.
- Errores en accesos a bases de datos externas.
- Errores de rendimiento. (6)

Pruebas de caja blanca: se centran en la estructura lógica interna del software. Se basan en un examen detallado de los procedimientos y caminos lógicos del sistema. Son pruebas que aseguran que la operación interna se ajusta a las especificaciones, y que todos los componentes internos se han probado de forma adecuada. (7)

En el desarrollo de este flujo se desempeña el rol de probador y se aplicarán las pruebas de caja negra para comprobar la operatividad de las funcionalidades. No se realizará la liberación del producto, se efectuarán solamente las pruebas unitarias.

La **prueba unitaria** es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Se deben escribir casos de prueba para cada función no trivial o método en el módulo de forma que cada caso sea independiente del resto. Para que una prueba unitaria sea *buena* se deben cumplir los siguientes requisitos:

- **Automatizable:** no debería requerirse una intervención manual. Esto es especialmente útil para integración continúa.
- **Completas:** deben cubrir la mayor cantidad de código.
- **Repetibles o Reutilizables:** no se deben crear pruebas que sólo puedan ser ejecutadas una sola vez. También es útil para integración continua.

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

---

- **Independientes:** la ejecución de una prueba no debe afectar a la ejecución de otra.
- **Profesionales:** las pruebas deben ser consideradas igual que el código, con la misma profesionalidad, documentación, etc.

Aunque estos requisitos no tienen que ser cumplidos al pie de la letra, se recomienda seguirlos o de lo contrario las pruebas pierden parte de su función.

## Lenguaje Unificado de Modelado.

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Permite la modelación de sistemas con tecnología orientada a objetos. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como RUP), pero no especifica en sí mismo qué metodología o proceso utilizar. Este lenguaje de modelado formal permite tener un mayor rigor en la especificación, realizar una verificación y validación del modelo desarrollado, automatizar determinados procesos y generar código a partir de los modelos y a la inversa. Esto último permite que el modelo y el código estén actualizados.

## 1.4 Tecnologías y herramientas a utilizar.

A continuación se expondrán diferentes tecnologías y herramientas que serán utilizadas.

### Java Server Faces (JSF)

Es un framework para desarrollo de aplicaciones web que utiliza el patrón de arquitectura Modelo-Vista-Controlador (MVC). Es muy flexible, ofrece una clara separación entre el comportamiento y la presentación, separar la lógica de negocio de la presentación, permite que cada miembro del equipo de desarrollo de la aplicación web se centre en su parte asignada del proceso de diseño, y proporciona un modelo sencillo de programación para enlazar todas las piezas. (8)

### RichFaces

RichFaces es una librería de componentes visuales para JSF, posee un framework avanzado de código abierto para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF. RichFaces incluye ciclo de vida, validaciones, conversores y la gestión de recursos estáticos y dinámicos. (9)

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

---

## **AJAX**

Asynchronous JavaScript And XML (AJAX). No es un lenguaje de programación sino un conjunto de tecnologías (HTML-JavaScript-CSS-DHTML-PHP/ASP.NET/JSP-XML) que permiten hacer páginas de web más interactivas. Su característica fundamental es permitir actualizar parte de una página con información que se encuentra en el servidor sin tener que refrescar completamente la página. De modo similar se puede enviar información al servidor. (10)

## **Ajax4jsf**

Ajax4jsf es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Javascript. Mediante este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, control de cualquier evento de usuario, etc. (11)

## **Facelets**

JavaServer Facelets es un framework para plantillas (templates) centrado en la tecnología JSF (JavaServer Faces), por lo cual se integran de manera muy fácil. JSP (JavaServer Pages) y JSF no se complementan naturalmente, cuando se usan juntos ambos escriben output al response, pero lo hacen de una manera diferente: JSP crea output ni bien encuentra código JSP (es decir procesa los elementos de la página de arriba a abajo), mientras que JSF dicta su propio re-rendering (ya que su ciclo de vida está dividido en fases marcadas). Facelets llena este vacío entre JSP y JSF, siendo una tecnología centrada en crear árboles de componentes y estar relacionado con el complejo ciclo de vida JSF. (12)

## **XHTML**

XHTML (Lenguaje de Marcado de Hipertexto Extensible) es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. Surge como el lenguaje cuyo etiquetado, más estricto que HTML, permitir una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella. (13)



# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

---

## **CSS**

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. (14)

## **JBoss Seam**

JBoss Seam es un framework que integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB). Con SEAM basta agregar anotaciones propias de este a los objetos Entidad y Session de EJB, logrando con esto escribir menos código Java y XML. Ha sido diseñado intentado simplificar al máximo el desarrollo de aplicaciones, basando el diseño en Plain Old Java Objects (POJOs) con anotaciones. Estos componentes se usan desde la capa de persistencia hasta la de presentación, poniendo todas las capas en comunicación directa. (15)

## **Hibernate**

Hibernate es una herramienta que realiza el mapping entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado es object/relational mapping (ORM) y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa. Hibernate no solo realiza esta transformación sino que proporciona capacidades para la obtención y almacenamiento de datos de la base de datos que nos reducen el tiempo de desarrollo. (16)

## **EJB**

Enterprise Java Beans (EJB) es una plataforma para construir aplicaciones de negocio portables, reusables y escalables usando el lenguaje de programación Java. Desde el punto de vista del desarrollador, un EJB es una porción de código que se ejecuta en un contenedor EJB, que no es más que un ambiente especializado (runtime) que provee determinados componentes de servicio.

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

---

Los EJBs pueden ser vistos como componentes, desde el punto de vista que encapsulan comportamiento y permite reutilizar porciones de código, pero también pueden ser vistos como un framework, ya que, desplegados en un contenedor, proveen servicios para el desarrollo de aplicaciones enterprise, servicios que son invisibles para el programador y no ensucian la lógica de negocio con funcionalidades transversales al modelo de dominio (a menudo requerimientos no funcionales o aspectos). En la especificación 3.0, los EJB no son más que POJOs (clases planas comunes y corrientes de Java) con algunos poderes especiales implícitos, que se activan en runtime cuando son ejecutados en un contenedor de EJBs. (17)

## **JPA**

Java Persistence API (JPA) proporciona un modelo de persistencia basado en POJOs para mapear bases de datos relacionales en Java. El JPA fue desarrollado por el grupo de expertos de EJB 3.0 como parte de JSR 220, aunque su uso no se limita a los componentes software EJB. También puede utilizarse directamente en aplicaciones web y aplicaciones clientes; incluso fuera de la plataforma Java EE, por ejemplo, en aplicaciones Java SE. (18)

### **1.4.1 Lenguaje de Programación.**

El lenguaje de programación seleccionado fue Java, este es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principio de la década de los 90. La programación en Java permite el desarrollo de aplicaciones bajo el esquema de Cliente Servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras u ordenadores, ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar.

### **1.4.2 Herramientas.**

#### **Java Runtime Environment (JRE)**

Es la máquina virtual de Java más básica. Sin ella, ningún programa Java lograría arrancar, y lo mismo se aplica a los applets de las páginas web. Es un componente tan imprescindible como .NET o las librerías DirectX. Su instalador se encarga de integrar Java con el sistema y los principales navegadores (como Internet Explorer o Firefox). (19)

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

---

## **Eclipse**

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados, como el IDE de Java llamado Java Development Toolkit y el compilador que se entrega como parte de Eclipse. Emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de los llamados clientes ricos. Esto lo diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. (20)

## **JBoss Tools**

Es un conjunto de plug-in para el Eclipse que permite el manejo de diferentes frameworks que facilitan el desarrollo de aplicaciones. Está constituido por varios módulos:

- Editor visual JSF y Facelets
- Herramientas de generación, refactorización y completado de código Seam
- Herramientas para Hibernate como ficheros de mapeo, anotaciones, ingeniería reversa con JPA
- Herramienta para inicializar y parar el servidor de aplicaciones JBoss
- Editor de workflows jBPM
- Herramientas basadas en SoapUI para el desarrollo de web services. (21)

## **JBoss Server**

JBoss Application Server es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Por ser una plataforma certificada J2EE, soporta todas las funcionalidades de J2EE 1.4 e incluye servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta EJB 3.0, lo que hace el desarrollo de las aplicaciones mucho más simple. Además, al ser desarrollado con tecnología Java, es multiplataforma. (22)

## **Visual Paradigm6.4 para UML**

Aunque el Rational Rose es la herramienta case más utilizada mundialmente en la actualidad, se ha seleccionado para la realización de los diagramas del presente trabajo de diploma el Visual Paradigm, debido a que es una herramienta multiplataforma, que utiliza UML como lenguaje de modelado, que da

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

---

soporte completo al ciclo de vida de un software, desde el análisis, el diseño, la implementación y las pruebas hasta el despliegue.

Es muy fácil de usar y presenta un ambiente gráfico agradable para el usuario. Permite configurar las líneas de redacción, el modelado de base de datos, requerimientos y del proceso del negocio, además de la interoperabilidad, la generación de documentación y la generación de código base para diferentes lenguajes de programación como Java, C# y PHP, permite la integración con herramientas de desarrollo; corrige sintaxis en tiempo de edición, además de contar con plugin para el soporte de las extensiones UML 2.0, aunque no es una herramienta libre la UCI tiene una licencia para utilizarla. (23)

## 1.2.3 Sistemas de Gestión de Base de Datos.

Un Sistema Gestión de Bases de Datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El Objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos. El SGBD es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma.

**PostgreSQL Server 8.3** PostgreSQL es un SGBD Objeto-Relacionales (Object-Relational Database Management System (ORDBMS)) libre, no tiene costo asociado por lo que cualquiera puede disponer de su código fuente, modificarlo a voluntad y redistribuirlo libremente. PostgreSQL presenta alta concurrencia, para esto utiliza la tecnología de Control de Concurrencia Multi-Versión (Multiversion concurrency control (MVCC)), con lo que se logra que ningún lector sea bloqueado por un escritor. Es altamente extensible, soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario. Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/PGSQL.

Este lenguaje es comparable al lenguaje procedural del sistema de gestión de base de datos relacional Oracle, PL/SQL. En cuanto a sus funciones, poseen bloques de código que se ejecutan en el servidor los cuales pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos brinda. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que se intente conectar a la base de datos. Tiene una excelente documentación y está bien organizada, además de contar con una comunidad de usuarios y desarrolladores a los que acudir en caso de tener problemas. (24)

# **CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.**

---

En este capítulo se abordaron las tecnologías y herramientas que se van a usar a lo largo del desarrollo del sistema, las cuales fueron asimiladas debido a la propuesta del grupo de arquitectura de la facultad y el Área Temática de Gestión Hospitalaria para el desarrollo de sus aplicaciones.

## Capítulo II: Descripción de la Arquitectura.

En el presente capítulo se abordará, de manera general, los requisitos no funcionales que permitirá la viabilidad de los usuarios al acceso de las funcionalidades del proceso de diálisis, además se describe la arquitectura, las estrategias de codificación, los estándares y estilos a utilizar en el desarrollo de la aplicación.

### 2.1 Requisitos No Funcionales.

#### 2.1.1 Usabilidad.

El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido:

Usuarios normales: 30 días

Usuarios avanzados: 20 días

#### 2.1.2 Fiabilidad.

Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.

Con el objetivo de elaborar reportes estadísticos por parte del hospital o entidad, ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, independientemente de que para el sistema, este elemento ya no exista. También es configurable el tiempo que los datos perduran en la BD. A estos datos se pueden acceder mediante Visor portable de la Historia Clínica que es capaz de extraer todos los datos de un paciente determinado y grabarlos en un CD.

#### 2.1.3 Eficiencia.

El sistema respetará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos. Se deberá usar siempre que sea posible el patrón Singleton, este se implementa mediante el framework SEAM, el cual proporciona un conjunto de anotaciones, de las cuales se usa @Scope, el cual crea una instancia diferente de la clase por cada

## CAPÍTULO II: DESCRIPCIÓN DE LA ARQUITECTURA

---

sesión Http. Además se debe destruir referencias que ya no estén siendo usadas, optimizar el trabajo con cadenas, entre otras buenas prácticas que ayudan a mejorar el rendimiento.

### 2.1.4 Soporte.

Abarcan las acciones que se llevarán a cabo una vez que se ha terminado el desarrollo del software, con motivos de asistir a los clientes de este, logrando así su mejora progresiva y evolución en el tiempo. Pueden incluir: pruebas, extensibilidad, adaptabilidad, mantenimiento, configuración, compatibilidad, servicios, instalación.

Las notificaciones de las deficiencias detectadas en la aplicación desplegada deberán realizarse por escrito.

Una vez notificada por la entidad médica, la deficiencia detectada en la aplicación desplegada, el equipo de desarrollo deberá solucionarla en un período de 7 días.

La capacitación y entrenamiento del profesional de salud para el uso del sistema, se realizará en un período de 6 meses.

### Configuración de parámetros.

Se permitirá establecer parámetros de configuración del sistema y actualización de nomencladores.

### 2.1.5 Restricciones de diseño.

#### 2.1.5.1 Capas físicas.

**Cliente:** Computadora con sistema operativo Windows XP o GNU/Linux que cuente con un navegador actualizado y que siga los estándares web (se recomienda IE 7 o superior o Firefox 3.x).

**Servidor de Aplicaciones:** Servidor con sistema operativo GNU/Linux que soporte el Java Runtime Environment (JRE) 1.5 o superior y al JBoss AS 4.2 o superior.

**Servidor de Base de Datos:** Servidor con sistema operativo Windows XP o GNU/Linux que soporte a PostgreSQL Server 8.3 o superior en los servidores de base de datos de cada hospital.

#### Capas lógicas.

# CAPÍTULO II: DESCRIPCIÓN DE LA ARQUITECTURA

---

**Presentación:** Contiene todas las vistas y la lógica de la presentación. El flujo web se maneja de forma declarativa y basándose en definiciones de procesos del negocio.

**Negocio:** Mantiene el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario. En los casos de que algún objeto del negocio tenga una interfaz externa, siendo accesible la misma desde sistemas legados o directamente del cliente, se garantiza la seguridad a nivel de objeto y métodos.

**Acceso a Datos:** Contiene las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate.

## 2.1.6 Requisitos para la documentación de usuarios en línea y ayuda del sistema.

Se posibilitará el uso de ayudas dinámicas y tutoriales en línea sobre el funcionamiento del sistema.

## 2.1.7 Interfaz.

### 2.1.7.1 Interfaces de usuario.

Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información.

La interfaz contará con teclas de función y menús desplegados que faciliten y aceleren su utilización.

La entrada de datos incorrecta será detectada claramente e informada al usuario.

El sistema permitirá al usuario configurar el idioma de los mensajes que serán mostrados.

El diseño de la interfaz del sistema responderá a la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso. El sistema incluirá reportes estándares y parametrizables que permitirán al usuario configurar la información de salida y el orden en que aparecen los datos. Las salidas se podrán generar en el formato de fichero PDF.



# CAPÍTULO II: DESCRIPCIÓN DE LA ARQUITECTURA

---

## 2.2 Descripción de la Arquitectura, fundamentación.

El uso de patrones hace posible que el desarrollo del sistema sea más rápido y confiable, además de que facilitan la construcción de software con propiedades definidas. La arquitectura que se utiliza en el desarrollo del sistema fue definida por el grupo de arquitectura de la facultad y el Departamento de Gestión Hospitalaria, las cuales son: el patrón de diseño Modelo-Vista-Controlador y el Patrón en Capas.

**Modelo Vista Controlador (MVC):** Es un patrón de diseño de software en el cual todo el proceso está dividido en 3 capas, típicamente estas capas son el Modelo, la Vista y el Controlador.

**El modelo de datos (Acceso a Datos)** se encarga del almacenamiento de los datos, su estructura y las relaciones entre los mismos. Para la administración de los datos se utiliza el framework Hibernate. Lo cual tiene como ventajas el control de la accesibilidad a la información almacenada, así como la realización de rápidas consultas. Hibernate implementa una serie de patrones para la administración del modelo de datos, entre ellos están los siguientes:

- Active record: Permite la asociación de filas únicas de la base de datos con objetos del lenguaje de programación usado.
- Identity map: Mantiene referencias a los objetos cargados y devuelve las mismas cada vez que se solicita un objeto ya cargado. De esta forma actúa como un cache dentro de la transacción, lo cual también contribuye a reducir el número de accesos a la base de datos.
- Identity field: Permite mapear las relaciones entre referencias entre objetos y claves entre tablas.
- Foreign key mapping: Permite mapear las relaciones de asociación de clave foránea.
- Association Table mapping: Permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, lo cual crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).
- Lazy load: Carga los distintos componentes de una clase a medida que son solicitados.

# CAPÍTULO II: DESCRIPCIÓN DE LA ARQUITECTURA

---

- Query object: Permite manipular las consultas en tiempo de ejecución. Dado que las consultas son los objetos, se pueden cambiar para agregar o quitar condiciones. Esto proporciona la flexibilidad necesaria para crearlas o modificarlas dinámicamente.

En esta capa se encuentran las tecnologías:

Hibernate, EJB, JPA

**El controlador (Lógica de Negocio)** está compuesto por las clases controladoras de los procesos identificados y modelados, además de las controladoras personalizadas y auto generadas. Es en estas clases que se manejan las reglas que regulan el flujo de la información. Acceden al modelo de datos a medida que se producen los eventos en las vistas que requieren de una determinada información.

En esta capa se encuentra la tecnología:

JBoss Seam

**Las vistas (Presentación)** se identifican con las interfaces con las que el usuario interactúa que muestran la información manejada por el modelo. Es a través de ellas que se introducen los datos requeridos para la ejecución de una u otra funcionalidad, además de que también se visualiza el resultado de las operaciones realizadas que son de interés para el usuario. Para consolidar la integración entre los diferentes elementos existentes en la aplicación se utiliza el framework Jboss Seam que se encarga de abstraer al desarrollador del funcionamiento interno, permite la utilización de anotaciones con las que es posible acceder al modelo de datos directamente desde la vista, característica que permite un mejor funcionamiento del sistema.

En esta capa se encuentran las siguientes tecnologías:

Java Server Faces (JSF), RichFaces, AJAX, Ajax4jsf, Facelets, XHTML, CSS

## **2.3 Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados. Estrategias de integración.**

Como estrategia de integración se trazó que todos los viernes los desarrolladores del Módulo de Diálisis deben realizar la integración con el arquitecto principal del área temática Sistema Especializado en Salud

## CAPÍTULO II: DESCRIPCIÓN DE LA ARQUITECTURA

---

(SES), y este a su vez cada 15 días se dirige hacia el Departamento de Gestión Hospitalaria, donde realiza la integración con el arquitecto principal del mismo.

**Admisión** es un módulo del sistema alas HIS, que entre sus funciones se encuentran gestionar la información que se genera en el ingreso de los pacientes en el hospital, recibir referencias de los mismos desde otros hospitales, y crear la hoja frontal, que es la HCE del paciente. En el Módulo de Diálisis se utiliza dicha hoja para obtener los pacientes que se encuentran ingresados en el hospital cuando se van a dar los turnos y crear las hojas de diálisis, que son las que gestionan la información que se genera cuando se le va a realizar el proceso de diálisis al paciente.

**Consulta Externa de Nefrología** es un módulo del sistema alas HIS que actualmente se está desarrollando. Se encarga de gestionar la información que se genera en las consultas de nefrología. Cuenta con funcionalidades como registrar los datos de los trasplantes de riñón del paciente, seguimiento previo al trasplante de los posibles donantes, seguimiento posterior al trasplante y propuestas de donante idóneo. Es en este módulo donde se crea la Hoja Nefrológica, en la misma se realizan gran parte de las funcionalidades antes mencionadas, además de que es la encargada de crear la indicación de diálisis, mediante la cual se le asignan los turnos al paciente, y se crean las hojas de diálisis en el Módulo de Diálisis.

### 2.4 Seguridad.

La seguridad es un elemento primordial en todo tipo de aplicación, pues aquí se define como se va a gestionar el acceso de los usuarios al sistema, así como los permisos efectivos para cada uno de ellos. Específicamente en el sistema propuesto en lo que se refiere a la autorización a los distintos recursos y elementos del sistema será basada en reglas y estará presente en todas las capas de la aplicación, esto posibilita que si hay algún cambio con respecto a esto solamente requiera la modificación de un único fichero.

El acceso al sistema será a nivel de usuarios y contraseñas. Estas contraseñas solo pueden ser alteradas por el usuario en cuestión o por el administrador en caso excepcional. Un usuario podrá tener más de un rol en el sistema de acuerdo con las acciones que realiza, mediante el cual se le otorgan determinados permisos para el acceso a la información y cada vez que el usuario realice una acción sobre el sistema, se registrará una traza que contiene la información gestionada mediante su estancia en el módulo.

## CAPÍTULO II: DESCRIPCIÓN DE LA ARQUITECTURA

---

En las estaciones de trabajo sólo se podrán ejecutar las aplicaciones definidas para la estación en la que se vaya a desplegar el sistema. Esto dota al sistema de un segundo nivel de seguridad que lo hace más fiable. La fidelidad de los datos está respaldada, pues el intercambio de los datos con sistemas externos se hará de forma cifrada, imposibilitando el acceso y modificación de los mismos. Además se hace uso del estándar HL7, que define las especificaciones para el intercambio de información en el ámbito de la salud, lo que posibilita que los datos manejados por el sistema sean más fidedignos.

A continuación se describen detalladamente las funcionalidades que ofrece el módulo de Configuración del Sistema de Información Hospitalaria alas HIS, el cual se encargará de garantizar la seguridad en el sistema a desarrollar.

### **Iniciar/Cerrar sesión de trabajo**

Cuando el usuario necesita acceder al sistema, este solicita: nombre de usuario y contraseña. El usuario introduce los datos solicitados, el sistema verifica que los datos introducidos sean válidos, si es así, el usuario accede al módulo Diálisis. El sistema muestra como opciones del menú las funcionalidades a las que tiene permiso de acceder el usuario en el módulo, lo que garantiza el acceso de los mismos solo a los niveles establecidos de acuerdo con la función que realizan. El sistema permite: cerrar sesión y salir del módulo.

### **Registrar trazas**

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

### **Administrar seguridad**

El sistema brinda la posibilidad de asignar o denegar permiso a roles y usuarios en las funcionalidades del módulo.

### **Configurar funcionalidades**

El sistema brinda la posibilidad de configurar las funcionalidades del módulo.

# CAPÍTULO II: DESCRIPCIÓN DE LA ARQUITECTURA

## 2.5 Vista de Despliegue.

El diagrama de despliegue muestra la configuración de los elementos de hardware (nodos) y cómo los elementos y artefactos del software se trazan en esos nodos. Permite describir la vista de despliegue estática de un sistema.

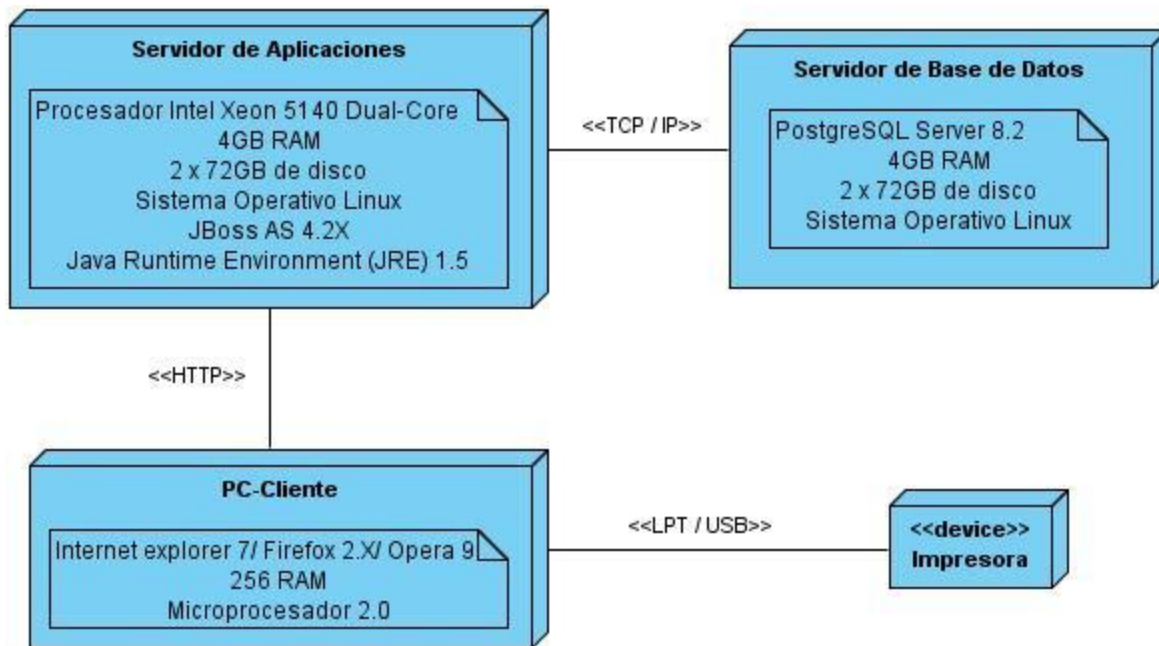


Figura 2.1 Diagrama de Despliegue

## 2.6 Estrategias de codificación. Estándares y estilos a utilizar.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación, lo que trae como posibilidad de que el proyecto de software se convierta en un sistema de software fácil de comprender (legibilidad) y de mantener (mantenibilidad). La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. La mantenibilidad del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento.

## CAPÍTULO II: DESCRIPCIÓN DE LA ARQUITECTURA

A continuación se muestran los elementos de los estándares de codificación de los cuales se hacen para el desarrollo del sistema en cuestión.

**Notación Camello** Se emplea para denotar variables y parámetros. Especifica que la palabra de inicio del identificador comienza con minúscula. Si el identificador está compuesto por más de una palabra entonces éstas deben comenzar con mayúsculas.

**Notación Pascal** Se emplea para denotar los métodos (funciones) y las clases. Especifica que la palabra de inicio del identificador comienza con mayúscula. Si el identificador está compuesto por más de una palabra entonces éstas deben comenzar con mayúsculas.

A continuación se especifican algunas restricciones para la nomenclatura, al igual que se expone el uso de los estándares así como el uso de los estándares mencionados anteriormente.

<b>Identación</b>	
<b>Inicio y fin de bloque</b>	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.
<b>Aspectos Generales</b>	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. Los inicios ( { ) y cierre ( } ) de ámbito deben estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción.  Nunca colocar { en la línea de un código cualquiera, esto requiere una línea propia.
<b>Comentarios, separadores, líneas, espacios en blanco y márgenes.</b>	
<b>Ubicación de comentarios</b>	Al inicio de cada clase o función y al final de cada bloque de código. Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas.
<b>Líneas en</b>	Se emplean antes y Se recomienda dejar una línea en blanco antes y después

## CAPÍTULO II: DESCRIPCIÓN DE LA ARQUITECTURA

<b>blanco</b>	después de métodos, clases y estructuras.	de la declaración de una clase o de una estructura y de la implementación de una función
<b>Espacios en blanco</b>	Entre operadores lógicos y aritméticos.	Se recomienda usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: objeto = objeto1
<b>Aspectos generales</b>	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. Después del paréntesis abierto y antes del cerrado. Antes de un punto y coma.
<b>Variables y constantes.</b>		
<b>Apariencia de variables</b>		El nombre que se le da a las variables debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación Camello Ejemplo: volumenIncial.
<b>Apariencia de constantes</b>	Todas sus letras en mayúscula	Se deben declarar las constantes con todas sus letras en mayúscula.
<b>Aspectos generales</b>	Nombres de las variables y constantes	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.
<b>Clases y objetos.</b>		

## CAPÍTULO II: DESCRIPCIÓN DE LA ARQUITECTURA

<b>Apariencia de clases y objetos</b>	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 Pascal Ejemplo: NomCausaSalidaDializador(). Para el caso de las instancias se comenzara en minúscula.
<b>Apariencia de atributos</b>	Primera letra en minúscula	El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación Camello.
<b>Apariencia de las funciones</b>	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 Pascal. Ejemplo: ModificarNomCausaSalidaDializador(). Si son funciones que obtienen un dato se emplea el prefijo get y si fijan algún valor se emplea el prefijo set.
<b>Declaración de parámetro en funciones</b>	Agrupados por tipos Poner los string 1 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.
<b>Aspectos generales.</b>	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.

**Tabla 1. Restricciones del código**

En este capítulo se analizaron los requerimientos no funcionales del sistema, siendo estos de vital importancia. También se hizo un estudio de cómo lograr un sistema con buena seguridad, parámetro imprescindible para todo producto para lograr la confidencialidad, autenticidad e integridad de los datos e



## **CAPÍTULO II: DESCRIPCIÓN DE LA ARQUITECTURA**

---

información que en él se gestiona. Se elaboró el diagrama de despliegue como propuesta para el establecimiento del sistema, identificando los nodos físicos que lo componen. Además para un mejor entendimiento del código, se realizó un estudio de los estándares de codificación y estilos de programación a utilizar.

# CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

## Capítulo III: Descripción y análisis de la solución propuesta.

En este capítulo se efectúa una valoración crítica del diseño propuesto por el analista. Se refinan los diagramas de clases del diseño y diagramas de interacción. Se describen las nuevas clases, u operaciones necesarias.

### 3.1 Valoración crítica del diseño propuesto por el analista.

#### (Refinar el diseño: Diagrama de clases del diseño y diagramas de interacción).

La implementación del Módulo de Diálisis no tiene una completa correspondencia con el diseño propuesto, ya que el mismo no cumple con las pautas de diseño definidas por el Departamento de Gestión Hospitalaria, en las clases controladoras los métodos propuestos no eran los indicados y hubo que realizar modificaciones en la base de datos (BD), además de agregar nuevas clases. Debido a esto se han realizado cambios en los diagramas de clases del diseño y en los diagramas de interacción, donde se muestran las modificaciones realizadas en las vistas, controladoras y entidades.

### 3.2 Descripción de las nuevas clases u operaciones necesarias.

#### Vistas:

<b>Nombre: NomCausaSalidaDializador</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página se encarga de listar los datos del nomenclador. Mediante ella se puede acceder a las otras opciones como son: buscar, adicionar, ver detalles, modificar, eliminar.
<b>Nombre: NomMarca</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página se encarga de listar los datos del nomenclador. Mediante ella se puede acceder a las otras opciones como son: buscar, adicionar, ver detalles, modificar, eliminar.

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Nombre: NomModelo</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página se encarga de listar los datos del nomenclador. Mediante ella se puede acceder a las otras opciones como son: buscar, adicionar, ver detalles, modificar, eliminar.
<b>Nombre: NomTipoAccesoVascular</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página se encarga de listar los datos del nomenclador. Mediante ella se puede acceder a las otras opciones como son: buscar, adicionar, ver detalles, modificar, eliminar.
<b>Nombre: NomTipoDializador</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página se encarga de listar los datos del nomenclador. Mediante ella se puede acceder a las otras opciones como son: buscar, adicionar, ver detalles, modificar, eliminar.
<b>Nombre: NomTipoSolucionDialisisPeritoneal</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página se encarga de listar los datos del nomenclador. Mediante ella se puede acceder a las otras opciones como son: buscar, adicionar, ver detalles, modificar, eliminar.
<b>Nombre: NomTipoSolucionHemodialisis</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página se encarga de listar los datos del nomenclador. Mediante ella se puede acceder a las otras opciones como son: buscar, adicionar, ver detalles,

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

	modificar, eliminar.
<b>Nombre: ServicioDialisis</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página se encarga de listar los datos del objeto. Mediante ella se puede acceder a las otras opciones como son:, ver detalles y modificar.
<b>Nombre: Cama</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página se encarga de listar los datos del objeto. Mediante ella se puede acceder a las otras opciones como son: buscar, adicionar, ver detalles, modificar, eliminar.
<b>Nombre: RinnonArtificial</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página se encarga de listar los datos del objeto. Mediante ella se puede acceder a las otras opciones como son: buscar, adicionar, ver detalles, modificar, eliminar.
<b>Nombre: Reuso</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página se encarga de listar los datos del objeto. Mediante ella se puede acceder a las otras opciones como son: buscar, adicionar, ver detalles, modificar.
<b>Nombre: CrearNomCausaSalidaDializador</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite crear el nomenclador una vez introducidos los datos referentes al mismo.
<b>Nombre: CrearNomMarca</b>	

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite crear el nomenclador una vez introducidos los datos referentes al mismo.
<b>Nombre: CrearNomModelo</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite crear el nomenclador una vez introducidos los datos referentes al mismo.
<b>Nombre: CrearNomTipoAccesoVascular</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite crear el nomenclador una vez introducidos los datos referentes al mismo.
<b>Nombre: CrearNomTipoDesinfectante</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite crear el nomenclador una vez introducidos los datos referentes al mismo.
<b>Nombre: CrearNomTipoDializador</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite crear el nomenclador una vez introducidos los datos referentes al mismo.
<b>Nombre: CrearNomTipoSolucionDialisisPeritoneal</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite crear el nomenclador una vez introducidos los datos referentes al mismo.

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Nombre: CrearNomTipoSolucionHemodialisis</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite crear el nomenclador una vez introducidos los datos referentes al mismo.
<b>Nombre: CrearCama</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite crear el objeto una vez introducidos los datos referentes al mismo.
<b>Nombre: CrearRinnonArtificial</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite crear el objeto una vez introducidos los datos referentes al mismo.
<b>Nombre: CrearReuso</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite crear el objeto una vez introducidos los datos referentes al mismo.
<b>Nombre: DatosNomCausaSalidaDializador</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página muestra los datos del nomenclador una vez insertados.
<b>Nombre: DatosNomMarca</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página muestra los datos del nomenclador una vez insertados.
<b>Nombre: DatosNomModelo</b>	

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página muestra los datos del nomenclador una vez insertados.
<b>Nombre: DatosNomTipoAccesoVascular</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página muestra los datos del nomenclador una vez insertados.
<b>Nombre: DatosNomTipoDesinfectante</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página muestra los datos del nomenclador una vez insertados.
<b>Nombre: DatosNomTipoDializador</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página muestra los datos del nomenclador una vez insertados.
<b>Nombre: DatosNomTipoSolucionDialisisPeritoneal</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página muestra los datos del nomenclador una vez insertados.
<b>Nombre: DatosNomTipoSolucionHemodialisis</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página muestra los datos del nomenclador una vez insertados.
<b>Nombre: DatosCama</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página muestra los datos del objeto una vez insertados.
<b>Nombre: DatosRinnonArtificial</b>	

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página muestra los datos del objeto una vez insertados.
<b>Nombre: DatosReuso</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página muestra los datos del objeto una vez insertados.
<b>Nombre: ModificarNomCausaSalidaDializador</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite modificar los datos del nomenclador.
<b>Nombre: ModificarNomMarca</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite modificar los datos del nomenclador.
<b>Nombre: ModificarNomModelo</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite modificar los datos del nomenclador.
<b>Nombre: ModificarNomTipoAccesoVascular</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite modificar los datos del nomenclador.
<b>Nombre: ModificarNomTipoDesinfectante</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite modificar los datos del nomenclador.
<b>Nombre: ModificarNomTipoDializador</b>	



## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite modificar los datos del nomenclador.
<b>Nombre: ModificarNomTipoSolucionDialisisPeritoneal</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite modificar los datos del nomenclador.
<b>Nombre: ModificarNomTipoSolucionHemodialisis</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite modificar los datos del nomenclador.
<b>Nombre: ModificarCama</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite modificar los datos del objeto.
<b>Nombre: ModificarRinnonArtificial</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite modificar los datos del objeto.
<b>Nombre: ModificarReuso</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite modificar los datos del objeto.
<b>Nombre: ModificarServicioDialisis</b>	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite modificar los datos del objeto.
<b>Nombre: VerNomCausaSalidaDializador</b>	

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite ver los detalles del nomenclador.
<b>Nombre:</b> VerNomMarca	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite ver los detalles del nomenclador.
<b>Nombre:</b> VerNomModelo	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite ver los detalles del nomenclador.
<b>Nombre:</b> VerNomTipoAccesoVascular	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite ver los detalles del nomenclador.
<b>Nombre:</b> VerNomTipoDesinfectante	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite ver los detalles del nomenclador.
<b>Nombre:</b> VerNomTipoDializador	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite ver los detalles del nomenclador.
<b>Nombre:</b> VerNomTipoSolucionDialisisPeritoneal	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite ver los detalles del nomenclador.
<b>Nombre:</b> VerNomTipoSolucionHemodialisis	

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite ver los detalles del nomenclador.
<b>Nombre:</b> VerCama	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite ver los detalles del objeto.
<b>Nombre:</b> VerRinnonArtificial	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite ver los detalles del objeto.
<b>Nombre:</b> VerReuso	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite ver los detalles del objeto.
<b>Nombre:</b> VerServicioDialisis	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Descripción:</b>	Esta página permite ver los detalles del objeto.

**Tabla 2 Vistas**

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

Controladoras:

<b>Nombre: ConfigurarCausaSalidaDializador</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarNomenclador()
<b>Descripción:</b>	Permite insertar el nomenclador
<b>Nombre:</b>	modificarNomenclador()
<b>Descripción:</b>	Permite modificar el nomenclador
<b>Nombre:</b>	seleccionarNomenclador()
<b>Descripción:</b>	Permite seleccionar el nomenclador
<b>Nombre:</b>	eliminarNomenclador()
<b>Descripción:</b>	Permite eliminar el nomenclador
<b>Nombre: ConfigurarTbnMarca</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarNomenclador()
<b>Descripción:</b>	Permite insertar el nomenclador
<b>Nombre:</b>	modificarNomenclador()
<b>Descripción:</b>	Permite modificar el nomenclador
<b>Nombre:</b>	seleccionarNomenclador()
<b>Descripción:</b>	Permite seleccionar el nomenclador
<b>Nombre:</b>	eliminarNomenclador()
<b>Descripción:</b>	Permite eliminar el nomenclador
<b>Nombre: ConfigurarTbnModelo</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarNomenclador()

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Descripción:</b>	Permite insertar el nomenclador
<b>Nombre:</b>	modificarNomenclador()
<b>Descripción:</b>	Permite modificar el nomenclador
<b>Nombre:</b>	seleccionarNomenclador()
<b>Descripción:</b>	Permite seleccionar el nomenclador
<b>Nombre:</b>	eliminarNomenclador()
<b>Descripción:</b>	Permite eliminar el nomenclador
<b>Nombre: ConfigurarTbnTipoDialisisPeritoneal</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarNomenclador()
<b>Descripción:</b>	Permite insertar el nomenclador
<b>Nombre:</b>	modificarNomenclador()
<b>Descripción:</b>	Permite modificar el nomenclador
<b>Nombre:</b>	seleccionarNomenclador()
<b>Descripción:</b>	Permite seleccionar el nomenclador
<b>Nombre:</b>	eliminarNomenclador()
<b>Descripción:</b>	Permite eliminar el nomenclador
<b>Nombre: ConfigurarTbnTipoDializador</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarNomenclador()
<b>Descripción:</b>	Permite insertar el nomenclador
<b>Nombre:</b>	modificarNomenclador()
<b>Descripción:</b>	Permite modificar el nomenclador
<b>Nombre:</b>	seleccionarNomenclador()
<b>Descripción:</b>	Permite seleccionar el nomenclador
<b>Nombre:</b>	eliminarNomenclador()

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Descripción:</b>	Permite eliminar el nomenclador
<b>Nombre: ConfigurarTbnVirologia</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarNomenclador()
<b>Descripción:</b>	Permite insertar el nomenclador
<b>Nombre:</b>	modificarNomenclador()
<b>Descripción:</b>	Permite modificar el nomenclador
<b>Nombre:</b>	seleccionarNomenclador()
<b>Descripción:</b>	Permite seleccionar el nomenclador
<b>Nombre:</b>	eliminarNomenclador()
<b>Descripción:</b>	Permite eliminar el nomenclador
<b>Nombre: ConfigurarTipoAccesoVascular</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarNomenclador()
<b>Descripción:</b>	Permite insertar el nomenclador
<b>Nombre:</b>	modificarNomenclador()
<b>Descripción:</b>	Permite modificar el nomenclador
<b>Nombre:</b>	seleccionarNomenclador()
<b>Descripción:</b>	Permite seleccionar el nomenclador
<b>Nombre:</b>	eliminarNomenclador()
<b>Descripción:</b>	Permite eliminar el nomenclador
<b>Nombre: ConfigurarTipoDesinfectante</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarNomenclador()
<b>Descripción:</b>	Permite insertar el nomenclador

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Nombre:</b>	modificarNomenclador()
<b>Descripción:</b>	Permite modificar el nomenclador
<b>Nombre:</b>	seleccionarNomenclador()
<b>Descripción:</b>	Permite seleccionar el nomenclador
<b>Nombre:</b>	eliminarNomenclador()
<b>Descripción:</b>	Permite eliminar el nomenclador
<b>Nombre: ConfigurarTipoSolucionDialisisPeritoneal</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarNomenclador()
<b>Descripción:</b>	Permite insertar el nomenclador
<b>Nombre:</b>	modificarNomenclador()
<b>Descripción:</b>	Permite modificar el nomenclador
<b>Nombre:</b>	seleccionarNomenclador()
<b>Descripción:</b>	Permite seleccionar el nomenclador
<b>Nombre:</b>	eliminarNomenclador()
<b>Descripción:</b>	Permite eliminar el nomenclador
<b>Nombre: ConfigurarTipoSolucionHemodialisis</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarNomenclador()
<b>Descripción:</b>	Permite insertar el nomenclador
<b>Nombre:</b>	modificarNomenclador()
<b>Descripción:</b>	Permite modificar el nomenclador
<b>Nombre:</b>	seleccionarNomenclador()
<b>Descripción:</b>	Permite seleccionar el nomenclador
<b>Nombre:</b>	eliminarNomenclador()
<b>Descripción:</b>	Permite eliminar el nomenclador

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Nombre: ConfigurarTipoSolucionHemodialisis</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarNomenclador()
<b>Descripción:</b>	Permite insertar el nomenclador
<b>Nombre:</b>	modificarNomenclador()
<b>Descripción:</b>	Permite modificar el nomenclador
<b>Nombre:</b>	seleccionarNomenclador()
<b>Descripción:</b>	Permite seleccionar el nomenclador
<b>Nombre:</b>	eliminarNomenclador()
<b>Descripción:</b>	Permite eliminar el nomenclador
<b>Nombre: CCama</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarObjeto()
<b>Descripción:</b>	Permite insertar el objeto
<b>Nombre:</b>	modificarObjeto ()
<b>Descripción:</b>	Permite modificar el objeto
<b>Nombre:</b>	seleccionarObjeto ()
<b>Descripción:</b>	Permite seleccionar el objeto
<b>Nombre:</b>	eliminarObjeto ()
<b>Descripción:</b>	Permite eliminar el objeto
<b>Nombre: CReuso</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarObjeto()
<b>Descripción:</b>	Permite insertar el objeto
<b>Nombre:</b>	modificarObjeto ()



## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Descripción:</b>	Permite modificar el objeto
<b>Nombre:</b>	seleccionarObjeto ()
<b>Descripción:</b>	Permite seleccionar el objeto
<b>Nombre:</b>	eliminarObjeto ()
<b>Descripción:</b>	Permite eliminar el objeto
<b>Nombre: CRinnonArtificial</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarObjeto()
<b>Descripción:</b>	Permite insertar el objeto
<b>Nombre:</b>	modificarObjeto ()
<b>Descripción:</b>	Permite modificar el objeto
<b>Nombre:</b>	seleccionarObjeto ()
<b>Descripción:</b>	Permite seleccionar el objeto
<b>Nombre:</b>	eliminarObjeto ()
<b>Descripción:</b>	Permite eliminar el objeto
<b>Nombre: CServicioDialisis</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarObjeto()
<b>Descripción:</b>	Permite insertar el objeto
<b>Nombre:</b>	modificarObjeto ()
<b>Descripción:</b>	Permite modificar el objeto
<b>Nombre:</b>	seleccionarObjeto ()
<b>Descripción:</b>	Permite seleccionar el objeto
<b>Nombre:</b>	eliminarObjeto ()
<b>Descripción:</b>	Permite eliminar el objeto
<b>Nombre: CIndicacionDialisisPeritoneal</b>	

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarObjeto()
<b>Descripción:</b>	Permite insertar el objeto
<b>Nombre:</b>	modificarObjeto ()
<b>Descripción:</b>	Permite modificar el objeto
<b>Nombre:</b>	seleccionarObjeto ()
<b>Descripción:</b>	Permite seleccionar el objeto
<b>Nombre:</b>	eliminarObjeto ()
<b>Descripción:</b>	Permite eliminar el objeto
<b>Nombre: CIndicacionHemodialisis</b>	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	insertarObjeto()
<b>Descripción:</b>	Permite insertar el objeto
<b>Nombre:</b>	modificarObjeto ()
<b>Descripción:</b>	Permite modificar el objeto
<b>Nombre:</b>	seleccionarObjeto ()
<b>Descripción:</b>	Permite seleccionar el objeto
<b>Nombre:</b>	eliminarObjeto ()
<b>Descripción:</b>	Permite eliminar el objeto

**Tabla 3 Controladoras**

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

### Entidades:

<b>Nombre de la Clase: tb_acceso_historico</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idAccesoVascular	int
version	int
tbAccesoVascular	TbAccesoVascular
fechaSalida	Date
causaSalida	causaSalida
<b>Nombre de la Clase: tb_acceso_vascular</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
hojaFrontal	HojaFrontal
localizacion	String
fecha_realizacion	Date
tbAccesoHistorico	TbAccesoHistorico
<b>Nombre de la Clase: tb_anticoagulacion</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
id_indicacion	int
dosisInicial	float
tbAnticoagulacionContinua	TbAnticoagulacionContinua
tbAnticoagulacionIntermitente	TbAnticoagulacionIntermitente
<b>Nombre de la Clase: tb_anticoagulacion_continua</b>	
<b>Tipo de clase:</b> Entidad	

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

Atributo	Tipo
idAnticuagulacion	int
dosisInfusion	int
velocidadBomba	int
tbAnticoagulacion	TbAnticoagulacion
<b>Nombre de la Clase: tb_anticoagulacion_intermitente</b>	
<b>Tipo de clase:</b> Entidad	
Atributo	Tipo
idAnticuagulacion	int
dosisIntermedia	int
frecuencia	Date
tbAnticoagulacion	TbAnticoagulacion
<b>Nombre de la Clase: tb_cama</b>	
<b>Tipo de clase:</b> Entidad	
Atributo	Tipo
id	int
valor	String
<b>Nombre de la Clase: tb_complicacion_dialisis</b>	
<b>Tipo de clase:</b> Entidad	
Atributo	Tipo
id	int
tbHojaDialisis	TbHojaDialisis
fecha	Date
tipoComplicacion	String
tiempoDialisis	Date
otrosProcedimientos	String
observaciones	String
<b>Nombre de la Clase: tb_datos_dialisis</b>	

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
tbHojaDialisis	TbHojaDialisis
frecuenciaCardiaca	int
temperatura	folat
sistolica	int
diastolica	int
<b>Nombre de la Clase:</b> tb_datos_intra_dialisis	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idDatosDialisis	int
tiempoDialisis	int
tbDatosDialisis	TbDatosDialisis
<b>Nombre de la Clase:</b> tb_datos_post_dialisis	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idDatosDialisis	int
horaFinal	Date
tbDatosDialisis	TbDatosDialisis
pesoFinal	float
ultrafiltracionObtenida	float
observaciones	String
<b>Nombre de la Clase:</b> tb_datos_pre_dialisis	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idDatosDialisis	int
horalnicio	Date

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

tbDatosDialisis	TbDatosDialisis
<b>Nombre de la Clase: tb_dializador</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
tbnTipoDializador	TbnTipoDializador
fechaAsignacion	date
tbDializadorHistorico	TbDializadorHistorico
<b>Nombre de la Clase: tb_dializador_historico</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idDializador	int
tbnCausaSalidaDializador	TbnCausaSalidaDializador
fechaDesecho	Date
tbDializador	TbDializador
<b>Nombre de la Clase: tb_hoja_dialisis</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
tbnMetodoDialitico	TbnMetodoDialitico
frecuencia	int
fecha	Date
tbIndicacion	TbIndicacion
hojaFrontal	HojaFrontal
<b>Nombre de la Clase: tb_indicacion</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

fecha	Date
pesoSeco	float
pesoActual	float
<b>Nombre de la Clase: tb_indicacion_dialisis_peritoneal</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idIndicacion	int
tbnTipoSolucionDialisisPeritoneal	TbnTipoSolucionDialisisPeritoneal
tbnTipoDialisisPeritoneal	TbnTipoDialisisPeritoneal
volumen	float
tbIndicacion	TbIndicacion
<b>Nombre de la Clase: tb_indicacion_hemodialisis</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idIndicacion	int
tbnTipoDializador	TbnTipoDializador
tbnTipoSolucionHemodialisis	TbnTipoSolucionHemodialisis
tbnTipoHeparina	TbnTipoHeparina
flujoSangre	int
tbIndicacion	TbIndicacion
<b>Nombre de la Clase: tb_medicamento</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idMedicamentoProducto	int
frecuencia	String
dosis	String
via	String

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

medicamentoProducto	MedicamentoProducto
<b>Nombre de la Clase: tbn_causa_salida_dializador</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
valor	String
codigo	String
<b>Nombre de la Clase: tbn_causa_salida_paciente_dialisis</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
valor	String
codigo	String
<b>Nombre de la Clase: tbn_marca</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
valor	String
codigo	String
<b>Nombre de la Clase: tbn_metodo_dialitico</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
valor	String
codigo	String
<b>Nombre de la Clase: tbn_modelo</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>



## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

id	int
valor	String
codigo	String
tbnMarca	TbnMarca
<b>Nombre de la Clase: tbn_tipo_acceso_vascular</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
valor	String
codigo	String
clasificacion	String
<b>Nombre de la Clase: tbn_tipo_desinfectante</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
valor	String
codigo	String
<b>Nombre de la Clase: tbn_tipo_dialisis_peritoneal</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
valor	String
codigo	String
<b>Nombre de la Clase: tbn_tipo_dializador</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
valor	String

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

codigo	String
volumenInicial	float
<b>Nombre de la Clase: tbn_tipo_heparina</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
valor	String
codigo	String
<b>Nombre de la Clase: tbn_tipo_solucion_dialisis_peritoneal</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
valor	String
codigo	String
<b>Nombre de la Clase: tbn_tipo_solucion_hemodialisis</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
valor	String
codigo	String
<b>Nombre de la Clase: tbn_virologia</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id	int
valor	String
codigo	String
<b>Nombre de la Clase: tb_reuso</b>	
<b>Tipo de clase:</b> Entidad	

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

Atributo	Tipo
id	int
tbDializador	TbDializador
tbnTipoDesinfectante	TbnTipoDesinfectante
volumenResidual	int
metodoReuso	String
<b>Nombre de la Clase: tb_rinnon_artificial</b>	
<b>Tipo de clase:</b> Entidad	
Atributo	Tipo
id	int
tbnModelo	TbnModelo
tbnVirologia	TbnVirologia
tiempoDesinfeccionInterDialisis	Date
tiempoDesinfeccionFinal	Date
<b>Nombre de la Clase: tb_servicio_dialisis</b>	
<b>Tipo de clase:</b> Entidad	
Atributo	Tipo
id	int
horalInicioServicio	Date
horaFinServicio	Date
<b>Nombre de la Clase: tb_turno</b>	
<b>Tipo de clase:</b> Entidad	
Atributo	Tipo
id	int
hojaFrontal	HojaFrontal
tbTurnoPeritoneal	TbTurnoPeritoneal
tbTurnoHemodialisis	TbTurnoHemodialisis
horalInicio	Date

# CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

horaFin	Date
<b>Nombre de la Clase: tb_turno_hemodialisis</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idTurno	int
tbRinnonArtificial	TbRinnonArtificial
tbTurno	TbTurno
<b>Nombre de la Clase: tb_turno_peritoneal</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idTurno	int
tbCama	TbCama
tbTurno	TbTurno

**Tabla 4 Entidades**

### 3.3 Modelo de Datos.

El modelo de datos ayuda al ingeniero de software a representar lo siguiente:

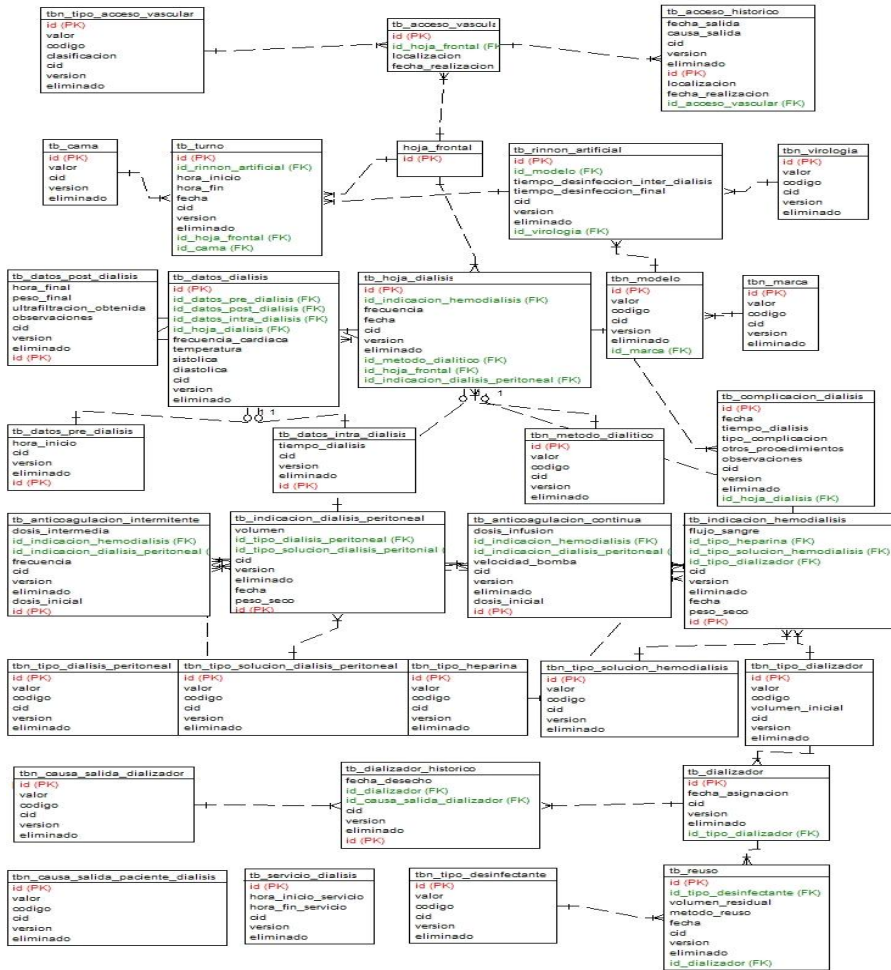
- Los objetos (entidades) de datos que el sistema debe entender
- Los atributos que describen a cada objeto
- Las relaciones entre los objetos

El modelado de los datos emplea el Diagrama Entidad Relación (DER) como base. El DER se centra solo en los datos, esto hace que proporcione el entendimiento del dominio de la información del problema. El DER servirá a los diseñadores del software como base para producir el diseño de la base de datos. (25)

# CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

[1,1]

[2,1]



[2,2]

Fig 3.1 Modelo de Datos.

# CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

## 3.3.1 Descripción de las tablas de la Base de Datos.

<b>tb_acceso_historico</b>	
<b>Atributos</b>	<b>Descripción</b>
id	Identificador del acceso histórico
id_acceso_vascular	Identificador del acceso vascular
fecha_salida	Fecha de salida del acceso vascular
causa_salida	Causa de salida del acceso vascular.
<b>tb_acceso_vascular</b>	
<b>Atributos</b>	<b>Descripción</b>
id	Identificador del acceso vascular.
id_tipo_acceso	Identificador del tipo de acceso vascular.
localizacion	Lugar donde se encuentra el acceso vascular.
fecha_realizacion	Día en que se le puso el acceso vascular.
id_hoja_frontal	Identificador de la hoja frontal ( del paciente)
<b>tb_anticoagulacion</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de la anticoagulación.
id_indicacion	Identificador de la indicación de la diálisis.
dosis_inicial	Cantidad inicial de la anticoagulación.
<b>tb_anticoagulacion_continua</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de la anticoagulación continua
id_indicacion	Identificador de la indicación de la diálisis.
dosis_inicial	Cantidad inicial de la anticoagulación.
id_anticoagulacion	Identificador de la anticoagulación.
dosis_infusion	Cantidad de infusión.
<b>tb_anticoagulacion_intermitente</b>	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Descripción</b>

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

id	Identificador de la anticoagulación intermitente.
id_indicacion	Identificador de la indicación de la diálisis.
dosis_inicial	Cantidad inicial de la anticoagulación.
id_anticoagulacion	Identificador de la anticoagulación.
dosis_intermedia	Cantidad de anticoagulación intermedia.
frecuencia	Frecuencia con la que se le aplica al paciente la anticoagulación intermitente.
<b>tb_cama</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de la cama
valor	Valor de la cama(identificador de la cama dentro de la sala)
<b>tb_complicacion_dialisis</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de la complicación de la diálisis
id_hoja_dialisis	Identificador de la hoja de diálisis.
fecha	Día en que ocurrió la complicación.
tipo_complicacion	Tipo de complicación.
tiempo_dialisis	Duración de la diálisis.
otros_procedimientos	Descripción de otras cosas que pueden ocurrir durante la diálisis.
observaciones	Observaciones del médico durante la diálisis.
<b>tb_datos_dialisis</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de los datos de la diálisis.
temperatura	Temperatura del paciente antes, durante y luego de realizada la diálisis.
frecuencia_cardiaca	Cantidad de latidos por minutos
id_hoja_dialisis	Identificador de la hoja de diálisis.

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

sistólica	Presión arterial máxima.
diastólica	Presión arterial mínima.
id_datos_pre_dialisis	Identificador de los datos de pre-diálisis.
id_datos_post_dialisis	Identificador de los datos de post-diálisis.
id_datos_intra_dialisis	Identificador de los datos de intra-diálisis.
<b>tb_datos_intra_dialisis</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de los datos intra-diálisis.
id_datos_dialisis	Identificador de los datos de la diálisis.
tiempo_dialisis	Tiempo de duración del proceso de diálisis.
<b>tb_datos_post_dialisis</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de los datos post-diálisis.
id_datos_dialisis	Identificador de los datos de la diálisis.
hora_final	Hora en que se acabó el proceso de diálisis
peso_final	Peso final
ultrafiltracion_obtenida	Ultrafiltración obtenida
observaciones	Observaciones
<b>tb_datos_pre_dialisis</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de los datos pre-diálisis.
id_datos_dialisis	Identificador de los datos de la diálisis.
hora_inicio	Hora de inicio de la diálisis
<b>tb_dializador</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del dializador.
id_tipo_dializador	Identificador del tipo de dializador.
fecha_asignacion	Fecha en la que se le asignó un dializador al



## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

	paciente.
<b>tb_dializador_historico</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del dializador histórico.
id_dializador	Identificador del dializador.
id_causa_salida_dializador	Identificador de la causa de salida del dializador.
fecha_desecho	Fecha en la que se eliminó el dializador.
<b>tb_hoja_dialisis</b>	
<b>Atributo</b>	<b>Descripción</b>
Id	Identificador de la hoja de diálisis.
id_metodo_dialitico	Identificador del método dialítico.
id_hoja_frontal	Identificador de la hoja frontal (del paciente)
id_indicacion	Identificador de la indicación de la diálisis.
frecuencia	La cantidad de veces que se realiza una diálisis.
fecha	Día de la diálisis.
<b>tb_indicacion_dialisis_peritoneal</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de la indicación de la diálisis peritoneal.
fecha	Fecha de la indicación de diálisis.
peso_seco	Peso seco del paciente.
peso_actual	Peso que posee el paciente en el momento de indicarle la indicación.
id_indicacion	Identificador de la indicación de la diálisis.
id_tipo_solucion_dialisis_peritoneal	Identificador del tipo de solución de diálisis peritoneal.
id_tipo_dialisis_peritoneal	Identificador del tipo de diálisis peritoneal.
volumen	Volumen de líquido a enfundir

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

<b>tb_indicacion_hemodialisis</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de la indicación de la hemodiálisis.
fecha	Fecha de la indicación de diálisis.
peso_seco	Peso seco del paciente.
peso_actual	Peso que posee el paciente en el momento de indicarle la indicación.
id_indicacion	Identificador de la indicación de la diálisis.
id_tipo_dializador	Identificador del tipo de dializador.
id_tipo_solucion_homodialisis	Identificador del tipo de solución hemodiálisis.
id_tipo_heparina	Identificador del tipo de heparina.
flujo_sangre	Flujo de sangre del paciente.
<b>tb_rinnon_artificial</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del riñón artificial
id_virologia	Identificador de la virología del paciente.
tiempo_desinfeccion_inter_dialisis	Tiempo que estuvo desinfectándose el riñón durante la diálisis.
tiempo_desinfeccion_final	Tiempo de desinfección final del riñón.
id_modelo	Identificador del modelo del riñón artificial.
<b>tb_reuso</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del reuso.
id_dializador	Identificador del dializador.
id_tipo_desinfectante	Identificador del tipo de desinfectante.
volumen_residual	Volumen residual del dializador.

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

metodo_reuso	La forma de realizar el reuso.
fecha	Fecha en la que se realizó el reuso.
<b>tb_servicio_dialisis</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del servicio de diálisis.
hora_fin_servicio	Hora en que se termina la jornada de diálisis.
hora_inicio_servicio	Hora en que comienza la jornada de diálisis.
<b>tb_turno</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del turno.
id_hoja_frontal	Identificador de la hoja frontal (del paciente).
hora_inicio	Hora en que se comienza a dializar al paciente.
hora_fin	Hora en la que se termina de dializar al paciente.
fecha	Fecha en la que se le va a realizar la diálisis al paciente.
id_rinnon_artificial	Identificador del riñón artificial
id_cama	Identificador de la cama.
<b>tbn_causa_salida_dializador</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de la causa de salida del dializador.
valor	Descripción de la causa, nombre
codigo	Código
<b>tbn_causa_salida_paciente_dialisis</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de la causa de la salida del paciente de la diálisis.

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

valor	Valor.
codigo	Código.
<b>tbn_marca</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador de la marca.
valor	Valor.
codigo	Código.
<b>tbn_metodo_dialitico</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del método dialítico.
valor	Valor.
codigo	Código.
<b>tbn_modelo</b>	
<b>Atributos</b>	<b>Descripción</b>
id	Identificador del modelo.
valor	Valor.
codigo	Código.
id_marca	Identificador de la marca
<b>tbn_tipo_acceso_vascular</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del tipo de acceso vascular.
valor	Valor.
codigo	Código.
clasificacion	Clasificación del tipo de acceso vascular.
<b>tbn_tipo_desinfectante</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del tipo de desinfectante.
valor	Valor.

## CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

codigo	Código.
<b>tbn_tipo_dialisis_peritoneal</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del tipo de diálisis peritoneal.
valor	Valor.
codigo	Código.
<b>tbn_tipo_dializador</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del tipo de dializador.
valor	Valor.
codigo	Código.
volumen_inicial	Volumen inicial.
<b>tbn_tipo_heparina</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del tipo de heparina.
valor	Valor.
codigo	Código.
<b>tbn_tipo_solucion_dialisis_peritoneal</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del tipo de solución de diálisis peritoneal.
valor	Valor.
codigo	Código.
<b>tbn_tipo_solucion_hemodialisis</b>	
<b>Atributo</b>	<b>Descripción</b>
id	Identificador del tipo de solución hemodiálisis.
valor	Valor.
codigo	Código.

# CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

tbn_virologia	
Atributo	Descripción
id	Identificador de la virología.
valor	Valor.
codigo	Código.

**Tabla 5. Descripción de las tablas de la BD.**

### 3.4 Diagrama de Componente.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

Los distintos componentes pueden agruparse en paquetes según un criterio lógico y con vistas a simplificar la implementación.

# CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

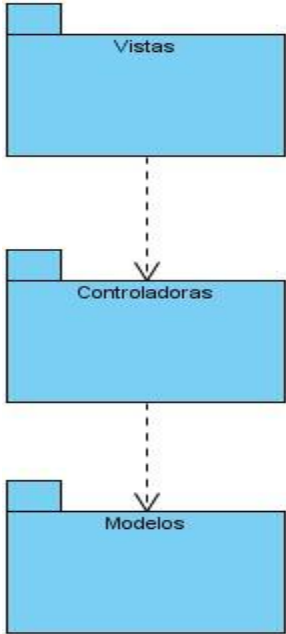


Figura 3.2 Diagrama de Componentes Diálisis (Paquetes).

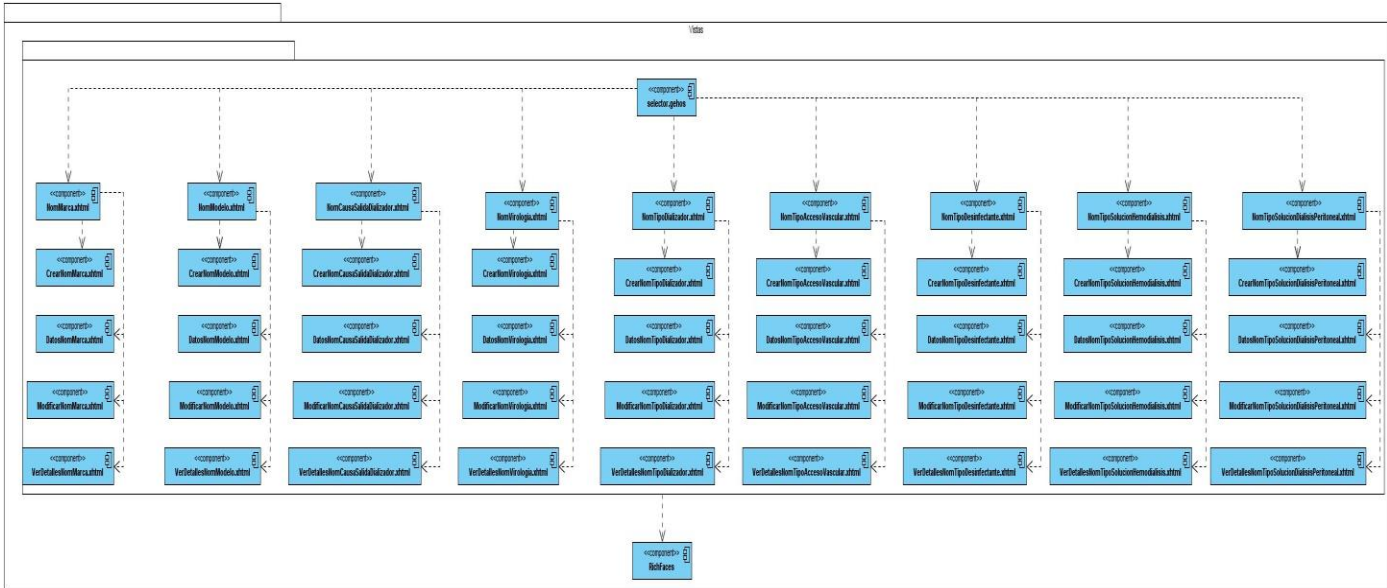


Figura 3.3 Diagrama de Componentes Paquete Vista de Configuración.

# CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

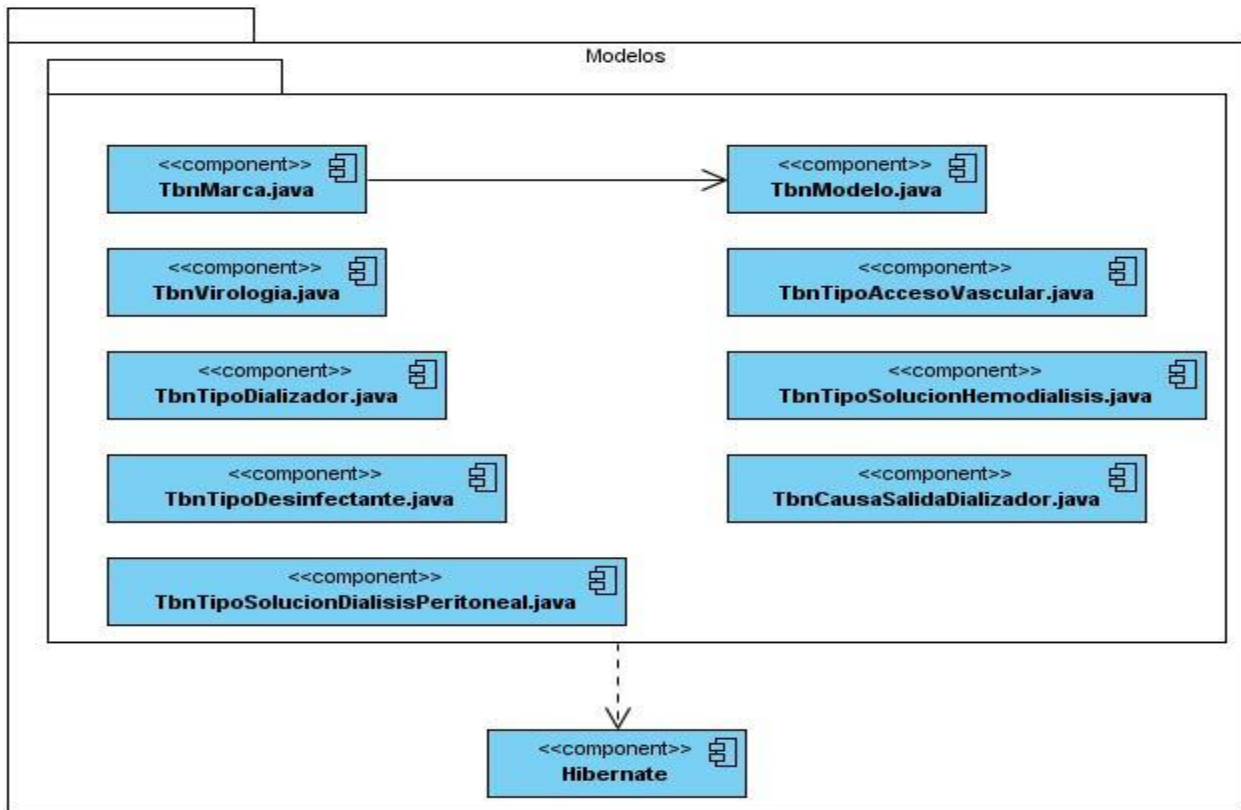


Figura 3.4 Diagrama de Componentes Paquete Modelos de Configuración.



# CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

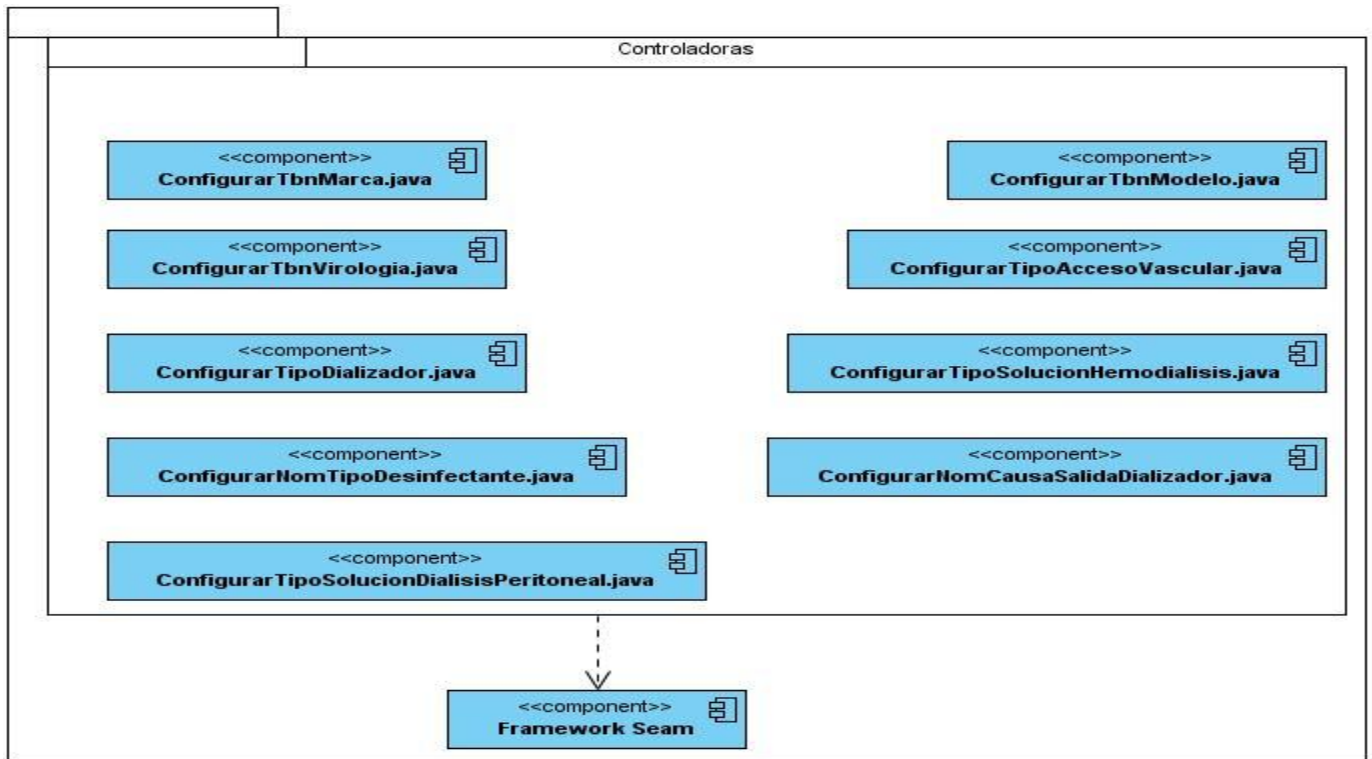


Figura 3.5 Diagrama de Componentes Paquete Controladoras de Configuración.

# CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

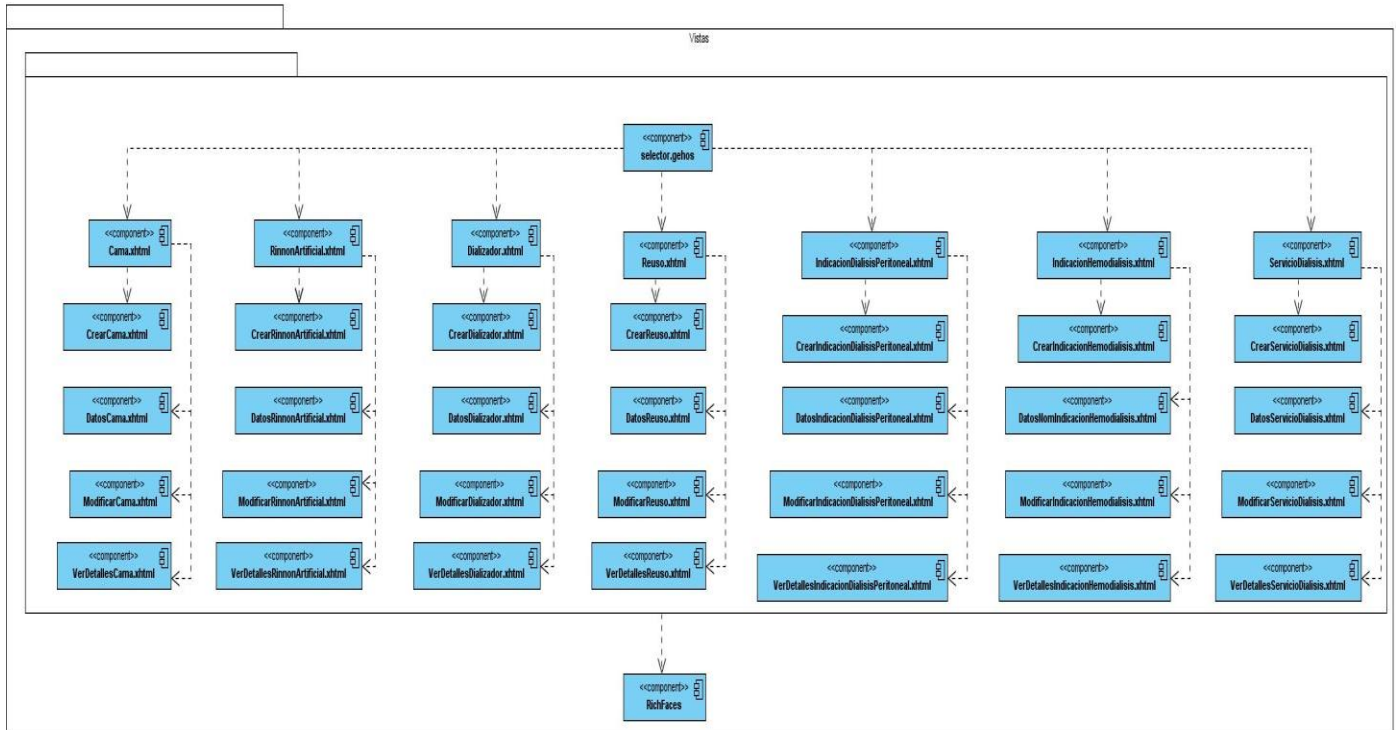


Figura 3.6 Diagrama de Componentes Paquete Vista de Diálisis.

# CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

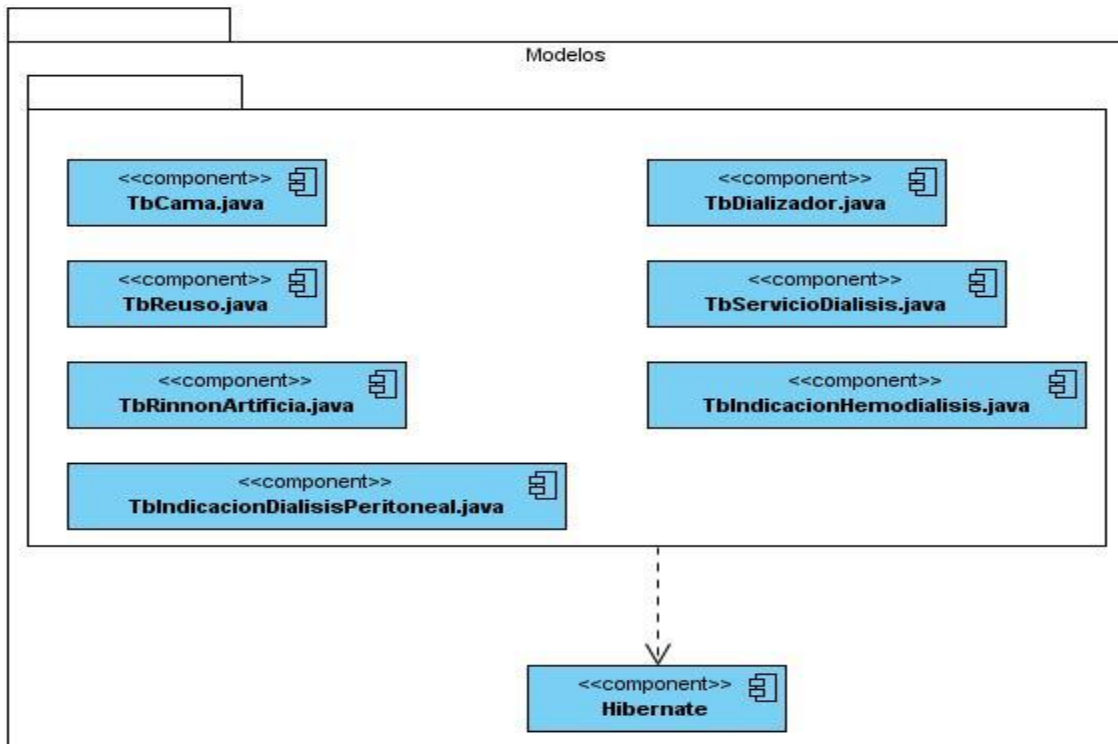
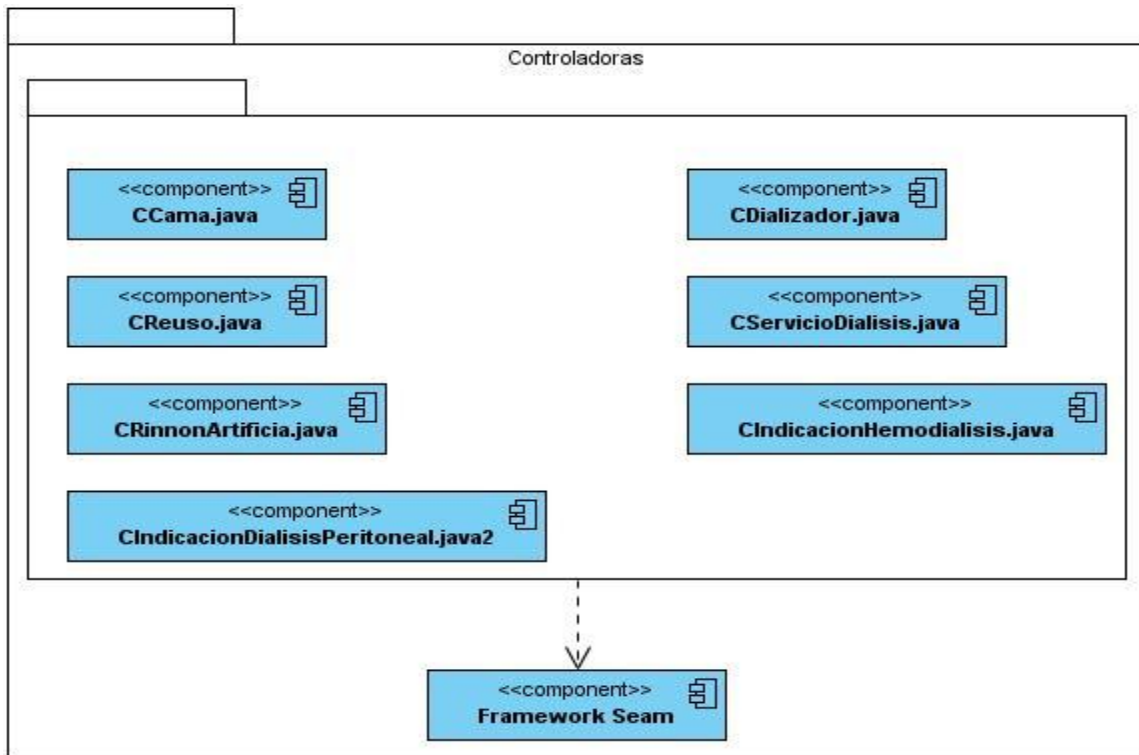


Figura 3.7 Diagrama de Componentes Paquete Modelos de Diálisis.

# CAPÍTULO III: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.



**Figura 3.8 Diagrama de Componentes Paquete Controladoras de Diálisis.**

En el capítulo concluido se realizó una valoración crítica del diseño propuesto por el analista donde se determinó que el mismo estaba incompleto. También, se describieron las nuevas clases u operaciones, para poder llevar a cabo la implementación. Se realizó el modelo de datos, donde se mostró la totalidad de las tablas, así como sus atributos y relaciones. Además se realizó el diagrama de componentes para conocer la distribución física del sistema.

## CONCLUSIONES

Con la realización del Módulo Diálisis del Sistema de Información Hospitalaria alas HIS se concluye que:

La utilización de RUP como metodología de desarrollo de software permitió dar continuidad al diseño propuesto por el analista, comprobándose que el uso de esta metodología es eficiente para guiar el proceso de desarrollo de un software.

Las herramientas y tecnologías para el desarrollo del sistema definidas por el Departamento de Gestión Hospitalaria posibilitó que se obtuviera una aplicación sencilla, flexible y robusta.

La descripción de la arquitectura propuesta para el desarrollo del sistema, demostró que la utilización del patrón MVC perfecciona la ejecución de las aplicaciones. Ya que al separar los datos, la interfaz de usuario y la lógica de control, viabiliza que las modificaciones se ejecuten solo en nivel requerido impactando en menor orden en el resto de las capas.

Al realizar una valoración crítica del diseño propuesto por el analista, se confirmó que este no resultó ser óptimo, pues tenía errores en su contenido, por lo que se hizo necesario realizar un refinamiento del diseño.

Se implementaron las funcionalidades requeridas, siguiendo las pautas de diseño definidas por el Departamento de Gestión Hospitalaria, obteniéndose una aplicación que facilita una implementación más rápida del programa médico de atención en todos los hospitales nefrológicos de Cuba, con la consiguiente mejora de los procesos.

## RECOMENDACIONES

Para llevar un control estadístico actualizado del estado actual de los pacientes dentro del proceso de Nefrología, se recomienda desarrollar funcionalidades que permitan generar los reportes:

- Prevalentes en diálisis.
- Incidentes en diálisis.
- Accesos vasculares.
- Reuso de dializador.

## REFERENCIAS BIBLIOGRÁFICAS

1. Conferencia Introducción al Proceso Unificado de Desarrollo de Software (RUP) y al Lenguaje Unificado de Modelado (UML). s.l. : Entorno Virtual de Aprendizaje.
2. **IDEM** Conferencia Introducción al Proceso Unificado de Desarrollo de Software (RUP) y al Lenguaje Unificado de Modelado (UML). s.l. : Entorno Virtual de Aprendizaje.
3. Conferencia resumen del Flujo de Implementación. s.l. : Entorno Virtual de Aprendizaje, 2010.
4. **Perez, Yasmin Caridad.** Conferencia : Fase de Construcción.Flujo de Trabajo de Implementación. 2009.
5. EcuRed. [En línea] [Citado el: 6 de Febrero de 2011.]  
[http://www.ecured.cu/index.php/Pruebas\\_de\\_Calidad\\_de\\_Software#Pruebas\\_de\\_Software](http://www.ecured.cu/index.php/Pruebas_de_Calidad_de_Software#Pruebas_de_Software).
6. **Pressman, Roger.** Ingeniería de Software. Un enfoque práctico. 2002.
7. INFORMATION SYSTEM GROUP TIC-194. [En línea] [Citado el: 9 de Febrero de 2011.]  
<http://indalog.ual.es/mtorres/LP/Prueba.pdf>.
8. **Almirón González, Cristóbal.** [www.adictosaltrabajo.com](http://www.adictosaltrabajo.com). [En línea] 24 de 11 de 2006.  
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=EclipseJSF>.
9. **Suárez Sánchez, Jose Manuel.** [adictosaltrabajo](http://www.adictosaltrabajo.com). [En línea] 01 de 02 de 2010. [Citado el: 11 de 11 de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsfIntro>.
10. ajaxya. [En línea] [Citado el: 11 de 11 de 2010.]  
<http://www.ajaxya.com.ar/temarios/descripcion.php?cod=8&punto=1>.
11. **Ramos, Juan Alonso.** [adictosaltrabajo](http://www.adictosaltrabajo.com). [En línea] 09 de 04 de 2007. [Citado el: 11 de 11 de 2010.]  
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
12. **Theo, Otto.** antartec. [En línea] 17 de 12 de 2008. [Citado el: 02 de 12 de 2010.]  
<http://blogs.antartec.com/desarrolloweb/2008/12/facelets-y-jsf-uso-de-templates/>.
13. w3c. [En línea] 07 de 02 de 2008. [Citado el: 02 de 12 de 2010.]  
<http://www.w3c.es/divulgacion/guiasbreves/XHTML>.
14. w3c. [En línea] 09 de 01 de 2008. [Citado el: 02 de 12 de 2010.]  
<http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>.

## REFERENCIAS BIBLIOGRÁFICAS

---

15. debugmodeon. [En línea] 2009. [Citado el: 16 de 11 de 2010.]  
<http://es.debugmodeon.com/articulo/que-es-jboss-seam>.
16. **Mazza Gómez, Ramiro**. dosideas. [En línea] 26 de 05 de 2009. [Citado el: 09 de 11 de 2010.]  
<http://www.dosideas.com/noticias/java/592-primeros-pasos-con-drools.html>.
17. **Crespo, Cesar**. adictosaltrabajo. [En línea] 14 de 08 de 2004. [Citado el: 16 de 11 de 2010.]  
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=hibernate>.
18. **Paredes, Paola A. Grajeda y Adrián M.** tecnologiasjava. [En línea] 27 de 08 de 2008. [Citado el: 2 de 11 de 2010.] <http://tecnologiasjava.blogspot.com/2008/08/introduccion-ejb-30-parte-1.html>.
19. juntadeandalucia. [En línea] 02 de 06 de 2008. [Citado el: 2 de 11 de 2010.]  
[http://www.juntadeandalucia.es/xwiki/bin/export/MADEJA/JPA\\_Introduccion?format=pdf&](http://www.juntadeandalucia.es/xwiki/bin/export/MADEJA/JPA_Introduccion?format=pdf&).
20. freeclipse. [En línea] 11 de 10 de 2009. [Citado el: 04 de 11 de 2010.]  
<http://freeclipse.foroactivo.com/t5-introduccion-lo-basico-de-eclipse>.
21. cantabria. [En línea] 10 de 10 de 2008. [Citado el: 02 de 02 de 2011.]  
<http://amap.cantabria.es/confluence/display/DEV/JBoss+Tools> .
22. **Ferri, Fabrizio**. java. [En línea] [Citado el: 02 de 02 de 2011.] <http://java.softonic.com/> .
23. **Alfonso, Lianes**. Diseño del proceso de Diálisis del Sistema de Información Hospitalaria: alas HIS. 2010.
24. **IDEM Alfonso, Lianes**. Diseño del proceso de Diálisis del Sistema de Información Hospitalaria: alas HIS. 2010.
25. Gobierno Federal SEP. [En línea] [Citado el: 13 de Febrero de 2011.]  
<http://www.itlalaguna.edu.mx/academico/carreras/sistemas/ingsoftware1/UNIDAD5.pdf>.



# BIBLIOGRAFÍA

---

- ajaxya. [En línea] [Citado el: 11 de 11 de 2010.]  
<http://www.ajaxya.com.ar/temarios/descripcion.php?cod=8&punto=1>.
- **Alfonso, Daniel Isaac y Lianes.** *Diseño del proceso de Diálisis del Sistema de Información Hospitalaria: alas HIS.* 2010.
- **Almirón González, Cristóbal.** [www.adictosaltrabajo.com](http://www.adictosaltrabajo.com). [En línea] 24 de 11 de 2006.  
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=EclipseJSF>.
- cantabria. [En línea] 10 de 10 de 2008. [Citado el: 02 de 02 de 2011.]  
<http://amap.cantabria.es/confluence/display/DEV/JBoss+Tools> .
- *Conferencia Introducción al Proceso Unificado de Desarrollo de Software (RUP) y al Lenguaje Unificado de Modelado (UML).* s.l. : Entorno Virtual de Aprendizaje.
- *Conferencia resumen del Flujo de Implementación.* s.l. : Entorno Virtual de Aprendizaje, 2010.
- **Crespo, Cesar.** [adictosaltrabajo](http://www.adictosaltrabajo.com). [En línea] 14 de 08 de 2004. [Citado el: 16 de 11 de 2010.]  
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=hibernate>.
- debugmodeon. [En línea] 2009. [Citado el: 16 de 11 de 2010.]  
<http://es.debugmodeon.com/articulo/que-es-jboss-seam>.
- EcuRed. [En línea] [Citado el: 6 de Febrero de 2011.]  
[http://www.ecured.cu/index.php/Pruebas\\_de\\_Calidad\\_de\\_Software#Pruebas\\_de\\_Software](http://www.ecured.cu/index.php/Pruebas_de_Calidad_de_Software#Pruebas_de_Software).
- **Ferri, Fabrizio.** java. [En línea] [Citado el: 02 de 02 de 2011.] <http://java.softonic.com/> .
- freeclipse. [En línea] 11 de 10 de 2009. [Citado el: 04 de 11 de 2010.]  
<http://freeclipse.foroactivo.com/t5-introduccion-lo-basico-de-eclipse>.
- Gobierno Federal SEP. [En línea] [Citado el: 13 de Febrero de 2011.]  
<http://www.itlalaguna.edu.mx/academico/carreras/sistemas/ingsoftware1/UNIDAD5.pdf>.
- INFORMATION SYSTEM GROUP TIC-194. [En línea] [Citado el: 9 de Febrero de 2011.]  
<http://indalog.ual.es/mtorres/LP/Prueba.pdf>.
- juntadeandalucia. [En línea] 02 de 06 de 2008. [Citado el: 2 de 11 de 2010.]  
[http://www.juntadeandalucia.es/xwiki/bin/export/MADEJA/JPA\\_Introduccion?format=pdf&](http://www.juntadeandalucia.es/xwiki/bin/export/MADEJA/JPA_Introduccion?format=pdf&).
- **Mazza Gómez, Ramiro.** dosideas. [En línea] 26 de 05 de 2009. [Citado el: 09 de 11 de 2010.]  
<http://www.dosideas.com/noticias/java/592-primeros-pasos-con-drools.html>.

# BIBLIOGRAFÍA

---

- **Paredes, Paola A. Grajeda y Adrián M.** tecnologiasjava. [En línea] 27 de 08 de 2008. [Citado el: 2 de 11 de 2010.] <http://tecnologiasjava.blogspot.com/2008/08/introduccion-ejb-30-parte-1.html>.
- **Perez, Yasmin Caridad.** *Conferencia : Fase de Construccion.Flujo de Trabajo de Implementación.* 2009.
- **Pressman, Roger.** *Ingeniería de Software. Un enfoque práctico.* 2002.
- **Ramos, Juan Alonso.** adictosaltrabajo. [En línea] 09 de 04 de 2007. [Citado el: 11 de 11 de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
- **Suárez Sánchez, Jose Manuel.** adictosaltrabajo. [En línea] 01 de 02 de 2010. [Citado el: 11 de 11 de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsflntro>.
- **Theo, Otto.** antartec. [En línea] 17 de 12 de 2008. [Citado el: 02 de 12 de 2010.] <http://blogs.antartec.com/desarrolloweb/2008/12/facelets-y-jsf-uso-de-templates/>.
- w3c. [En línea] 07 de 02 de 2008. [Citado el: 02 de 12 de 2010.] <http://www.w3c.es/divulgacion/guiasbreves/XHTML>.
- w3c. [En línea] 09 de 01 de 2008. [Citado el: 02 de 12 de 2010.] <http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>.
- EcuRed. [En línea] [Citado el: 5 de diciembre de 2010.] [http://www.ecured.cu/index.php/Proceso\\_Unificado\\_de\\_Desarrollo](http://www.ecured.cu/index.php/Proceso_Unificado_de_Desarrollo).
- González. javahispano. [En línea] 22 de abril de 2003. [Citado el: 12 de diciembre de 2010.] <http://www.javahispano.org/contenidos/archivo/77/ManualHibernate.pdf>.
- Grupo Soluciones Innova. [En línea] 2007. [Citado el: 14 de diciembre de 2010.] <http://www.rational.com.ar/herramientas/rup.html>.
- Martínez, Rafael. postgresSQL-es. [En línea] 2 de octubre de 2010. [Citado el: 21 de enero de 2011.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
- Richfaces Developer Guide. [En línea] 2008. [Citado el: 12 de diciembre de 2010.] <http://www.jboss.org/jbossrichfaces/docs/>.
- Rivera, Javier Fernández. aurea. [En línea] [Citado el: 12 de abril de 2011.] <http://aurea.es/wp-content/uploads/modelodedatos.pdf>.

# BIBLIOGRAFÍA

---

- Scribd. [En línea] [Citado el: 12 de diciembre de 2010.]  
<http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
- seamframework.org. [En línea] 2009. [Citado el: 14 de diciembre de 2010.]  
<http://www.seamframework.org/Documentation>.
- slideshare. [En línea] 2010. [Citado el: 11 de enero de 2011.]  
<http://www.slideshare.net/inspireunaula/mapeo-de-objeto-relacional>.

## GLOSARIO DE TÉRMINOS

**Base de datos:** es un “almacén” que permite guardar grandes cantidades de información de forma organizada y que permita las posibles consultas de acuerdo a los derechos de acceso.

**Diálisis peritoneal:** es un método de depuración sanguínea extrarrenal de solutos y toxinas.

**Dializador:** filtra la sangre, actuando como un riñón artificial, para retirar de la sangre el exceso de agua y los productos de desecho.

**Framework:** es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

**Hardware:** son los dispositivos físicos como la placa base, la CPU o el monitor.

**Hemodiálisis:** consiste en filtrar el exceso de líquidos y las sustancias tóxicas del organismo mediante el paso de la sangre del paciente por un filtro periódicamente.

**Insuficiencia renal:** cuando los riñones pierden la capacidad de filtrar y limpiar la sangre, el agua y los productos de desecho pueden acumularse y ser tóxicos para el organismo.

**Interfaz:** es un conjunto de protocolos y técnicas para el intercambio de información entre una aplicación computacional y el usuario.

**MVC (Modelo Vista Controlador):** Es un patrón de arquitectura de software compuesto de tres componentes distintos: datos, interfaz de usuario y lógica del negocio.

**Nefrólogo:** es el **médico** que diagnostica y trata las enfermedades renales, **de los riñones**, como la insuficiencia renal (con diálisis).

**RUP (Rational Unified Process):** Proceso unificado de desarrollo de software.

**Software:** es todo el conjunto intangible de datos y programas de la computadora.

**SQL (Lenguaje de Consulta Estructurado):** es un lenguaje declarativo de acceso a bases de datos relacionales, que permite especificar diversos tipos de operaciones en éstas.

**UML (Lenguaje Unificado de Modelado):** Es un lenguaje de modelado visual para especificar, visualizar, construir y documentar artefactos de un sistema de software.