

**Universidad Central “Marta Abreu” de Las Villas  
Facultad de Matemática, Física y Computación  
Departamento de Ciencia de la Computación**



**“Implementación de las reglas de negocio en la arquitectura de  
los sistemas de información”**

**Trabajo de Diploma**

**AUTOR**

José Luis Díaz Vellón

**TUTORES**

M.Sc. María Elena Martínez Busto

M.Sc. Jorge Alberto González Mena

**CONSULTANTE**

Dr. Paulino Hernández Hernández

SANTA CLARA – 2012

**“Año 54 de la Revolución”**

## Dictamen con derechos de autor para MFC



Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Ingeniería Informática, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del tutor

Firma del jefe del Seminario

*A mis padres*

*A mi abuela*

*A mi hermana y mi cuñado*

*A mi sobrina*

*A todos mis amigos*

*A mis tutores M.Sc. María Elena Martínez Busto y  
M.Sc. Jorge Alberto González Mena por su asesoría y orientación en este trabajo.*

*A mi consultante Dr. Paulino Hernández Hernández.*

*A mi hermana por ser mi guía*

*A mis compañeros del grupo de investigación, en especial a Asiel y Félix*

*A todos los profesores que me formaron y enseñaron.*

*A todos los que de una u otra forma me han ayudado y han contribuido a  
la terminación de esta tesis.*

## **SÍNTESIS**

Se tiene la necesidad de obtener Sistemas de Información (SI) capaces de reflejar los cambios que se llevan a cabo en los negocios. En los últimos años el manejo de las Reglas de Negocio (RN) ha llegado a ser muy popular en la comunidad de los SI, debido a que permite obtener aplicaciones flexibles y modificables.

En el presente trabajo se obtiene un SI para el control de pacientes en el Área de Nefrología (AN) del Hospital Universitario “Arnaldo Milián Castro” siguiendo el enfoque de RN. Las RN son identificadas e implementadas siguiendo lo recomendado en trabajos precedentes (Mena, 2011). Se logran implementar un significativo número de reglas y se realiza un análisis comparativo con sistemas que le anteceden, evidenciando su adaptabilidad. Dicho sistema se complementa con un portal informativo para la publicación de contenidos referentes al área nefrológica, tales como: artículos, imágenes, foros, blog, libros, etc. Se facilita el trabajo con la información guardada en el sistema gracias a las posibilidades que brindan los nuevos generadores automáticos de reportes. Como resultado adicional se conforma un video que sirve como presentación del producto.

## **ABSTRACT**

There is the need for Information Systems (IS) to reflect those changes taking place in business. In recent years the management of Business Rules (BR) has become very popular in the community of IS, because it allows applications to obtain flexible and changeable.

In this paper we obtain an IS to control patients in the Area of Nephrology (AN), University Hospital "Arnaldo Milian Castro" following the approach of BR. Neural networks are identified and implemented following the recommendations in previous work (Mena, 2011). He succeeded in implementing a significant number of rules and performed a comparative analysis with systems that predate, demonstrating its adaptability. This system is complemented by an information portal for the publication of content relating to the nephrology area, such as articles, images, forums, blog, books, etc. It is easy to work with information stored in the system thanks to the possibilities offered by new automatic report generators. As a further result is content that serves as a video presentation of the product.

## TABLA DE CONTENIDOS

<i>INTRODUCCIÓN</i> .....	1
<i>CAPÍTULO I. CONCEPTOS BÁSICOS Y CONTROL DE LOS PACIENTES DEL ÁREA DE NEFROLOGÍA</i> .....	4
1.1.    Enfoque de reglas de negocio .....	4
1.1.1.    Reglas de negocio .....	4
1.1.2.    Definición .....	5
1.1.3.    Clasificación .....	6
1.1.4.    Práctica de desarrollo .....	9
1.1.4.1.    Lineamientos para el descubrimiento .....	9
1.1.4.2.    Pasos para el análisis.....	11
1.1.4.3.    Implementación de las reglas en el sistema sobre el MVC .....	14
1.2.    Caso de estudio: área de nefrología .....	16
1.2.1.    Control de los pacientes .....	16
1.2.2.    Caracterización del caso de estudio .....	19
1.3.    Calidad de software .....	21
1.4.    Conclusiones parciales .....	23
<i>CAPÍTULO II. ADAPTABILIDAD DEL SISTEMA DE INFORMACIÓN</i> .....	24
2.1.    Implementación de las RN en el SI usando el patrón MVC .....	24
2.1.1.    Reglas implementadas para el caso de estudio .....	25
2.2.    Generadores automáticos de reportes .....	36
2.2.1.    Guardado de la búsqueda mediante la serialización .....	37
2.3.    Comparación del sistema actual con versiones anteriores .....	37
2.3.1.    Modificaciones en el sistema de información.....	38
2.3.2.    Modificaciones respecto a las reglas de negocio .....	40
2.4.    Conclusiones parciales .....	45
<i>CAPÍTULO III. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE NEFROLOGÍA</i> .....	47
3.1.    Diseño del sistema .....	47
3.1.1.    Modelo de casos de uso del sistema .....	47
3.1.2.    Diagrama de actividad .....	52
3.1.3.    Diagrama de navegación.....	52
3.1.4.    Cambio en el diseño de la base de datos.....	54
3.2.    Estructura de la aplicación .....	56
3.2.1.    Front-end.....	56

3.2.1.1. Página principal .....	56
3.2.1.2. Tipos de contenidos .....	57
3.2.2. Back-end .....	58
3.2.2.1. Inicio de sesión .....	58
3.2.2.2. Página principal .....	59
3.2.2.3. Manejo de historias clínicas.....	60
3.2.2.4. Tipos de estudios .....	62
3.2.2.5. Generadores automáticos de reportes .....	64
3.2.2.6. Trabajo con trasplantes .....	66
3.2.2.7. Configuración y nomencladores .....	67
3.3. Conclusiones parciales .....	69
<i>CONCLUSIONES</i> .....	70
<i>RECOMENDACIONES</i> .....	71
<i>REFERENCIAS BIBLIOGRÁFICAS</i> .....	72
<i>ANEXOS</i> .....	74
Anexo1 .....	74



## **INTRODUCCIÓN**

La práctica de la medicina de trasplante ha sido adoptada, como primera opción terapéutica, para un número creciente de enfermedades orgánicas durante los últimos 50 años. En la última década del siglo pasado, la mejoría en las técnicas quirúrgicas, cuidados médicos, supervivencia a mediano y largo plazo (90% a los 3 años), calidad de vida y costo beneficio, establecieron la superioridad del Trasplante Renal (TR) (Hernández et al., 2003). El TR es una alternativa que permite a los pacientes que presentan la Enfermedad Renal Crónica (ERC) salir de las listas de espera que se crean para los casos que necesitan tratamiento urgente. El TR no solo es mejor en el orden biológico, sino que también económicamente es más rentable. Por lo que constituye la mejor opción terapéutica para los pacientes con insuficiencia renal crónica terminal en diálisis, ya que les ofrece el mayor potencial para restaurar una vida sana y productiva, mejorando su calidad de vida.

En la actualidad, con la aparición de nuevos fármacos inmunosupresores, se está alcanzando una mortalidad mínima, aunque si bien es cierto, el TR es un procedimiento quirúrgico electivo o semi-electivo que se realiza a pacientes que han sido sometidos a una evaluación cuidadosa (Redín et al., 2006).

La atención a estos pacientes supone un gran reto para la enfermería, ya que son pacientes sometidos a una cirugía mayor que requieren múltiples cuidados y que además tienen asociado un componente de educación sanitaria muy importante.

El proceso de TR no solo supone un reto desde el punto de vista médico y material, sino que precisa un respaldo informativo que en su fase primaria puede ser bastante estándar pero que en la medida que los casos pasan a una atención más especializada, requieren un tratamiento diferenciado y difícilmente se puedan abordar soluciones generales. La organización y coordinación de trasplantes, es una tarea compleja que requiere varias actividades clínicas que involucran numerosas personas y equipos de trabajos; así como un proceso administrativo paralelo a los procesos clínicos.

Para la solución del respaldo informativo necesario se requiere el uso de tecnologías que respondan a diferentes paradigmas computacionales para la captación de las políticas,

regulaciones o leyes que deben hacerse cumplir en la entidad, lo que nos lleva al uso del enfoque de RN.

Las RN representan el enlace necesario entre el negocio y el modelado de la información, estas emergen de los objetivos de la empresa (*Rosca et al., 1995*) y regulan cómo opera un determinado negocio y cómo el mismo es estructurado (Ross, 2005).

En las aplicaciones clásicas de BD las reglas suelen estar dispersas dentro de la lógica de las aplicaciones, lo que hace difícil su modelación y mantenimiento. Dado los continuos avances en la medicina y los constantes cambios en el Área de Nefrología (AN) un sistema de esta índole no resulta una solución al problema. Con el enfoque de RN los usuarios tienen la posibilidad de definir las reglas previo al desarrollo de las aplicaciones, logrando de esta forma una independencia de las aplicaciones que las usan y facilitándose así su mantenimiento; o sea si cambia una regulación, solo hay que reflejarlo en la definición de la regla y no en los múltiples controles que hacen uso de la misma.

Con el propósito de lograr la automatización de las actividades necesarias en el Hospital Universitario “Arnaldo Milián Castro”, de Villa Clara, el cual es el objeto de estudio de este trabajo, se han realizado múltiples investigaciones y proyectos, entre los que se destaca la realización de un SI con arquitectura de tres capas y un ambiente web (Mena et al., 2008; Velazco et al., 2010; Mena, 2011) que cumple con las RN propias de la integridad de los datos para automatizar los procesos relacionados con el AN.

Para dar continuidad a este trabajo se impone la necesidad de identificar, ubicar e implementar las RN que se obtienen desde los objetivos de los procesos del AN.

Lo anteriormente expuesto justifica el planteamiento del siguiente ***problema de investigación*** para la presente tesis: Obtener una nueva versión del sistema para el control de pacientes en nefrología, aplicando los resultados de investigaciones relativas al ciclo de vida de las RN, analizando la adaptabilidad del SI respecto a sistemas precedentes.

Para el cumplimiento del ***problema de investigación*** planteado se proponen los objetivos siguientes:

El **objetivo general** de la investigación consiste en: Implementar las reglas dentro de la arquitectura del sistema para el control de pacientes en el AN usando el patrón de diseño Modelo Vista Controlador (MVC), para solucionar el problema de adaptabilidad utilizando el enfoque de RN.

Este se desglosa en los siguientes **objetivos específicos**:

1. Estudiar el diseño de la base de datos existente.
2. Aplicar criterios que determinan la ubicación de las RN dentro de los SI e insertarlas en su capa correspondiente para lograr la fácil actualización a las nuevas condiciones del negocio.
3. Obtener una nueva versión de la aplicación del sistema para el control de historias clínicas de pacientes de la sala de nefrología.
4. Validar el software obtenido en al menos un centro hospitalario.

Como resultado de la revisión bibliográfica y de la factibilidad del trabajo se define la **hipótesis de investigación** siguiente: la aplicación de criterios para la ubicación de RN en los SI implementadas utilizando el MVC soluciona el problema de adaptabilidad.

El trabajo se ha estructurado en tres capítulos de la siguiente forma:

En el Capítulo I se resume el enfoque de RN, la clasificación cercana a la implementación, la práctica de desarrollo, el caso de estudio del AN y se abordarán criterios para la valoración de la calidad de software.

Se continúa con un Capítulo II donde se describe las nuevas condiciones para el SI del control hospitalario, se hace el análisis de las nuevas RN identificadas, clasificadas y se determina su ubicación en la arquitectura del SI según las pautas propuestas por otros autores. Además se expondrán las diferencias entre el sistema anterior y el actual.

En el Capítulo III del documento se valora la efectividad de los criterios empleados para lograr la adaptabilidad del SI. Finalmente se documentan las principales funcionalidades del sistema para el AN.

Este documento culmina con las conclusiones, recomendaciones y referencias bibliográficas.

## **CAPÍTULO I. CONCEPTOS BÁSICOS Y CONTROL DE LOS PACIENTES DEL ÁREA DE NEFROLOGÍA**

### **1.1. *Enfoque de reglas de negocio***

Al abordar el *enfoque de RN* se deben tener claros los conceptos acerca de las RN y su modelación. Este enfoque permite el manejo de las RN independiente de las aplicaciones que las hacen cumplir, dicho término ha sido utilizado por diferentes autores (Layzell et al., 1988; Date, 2000; Bajec et al., 2000; Weiden, 2000; Bajec et al., 2001; Ross, 2003a). Bajec (2001) asegura que tal enfoque “es la combinación de técnicas y tecnologías, nuevas y otras ya existentes”, desde el que se intenta identificar el conocimiento necesario para administrar una empresa, documentarlo, razonar sobre el mismo, hacerlo operacional de una manera consistente, adaptarlo sistemáticamente a cualquier vaivén del mercado y tratar, en la medida de lo posible, de automatizarlo.

Es importante discernir, para orientarse en el empeño de la modelación de RN vinculadas a su soporte final por un SI, qué tipos de reglas son relevantes a ambos entornos: universo de discurso y SI. En principio, pudiera afirmarse que todas las reglas que gobiernan una aplicación son RN, sin embargo esto no siempre es cierto de acuerdo al *enfoque de RN*. En la práctica se pueden encontrar restricciones computacionales que no tienen correlación con el negocio (Morgan, 2002); este tipo de regla no surge del propio negocio y no resulta de interés para el propósito de esta investigación.

Los aspectos de los procesos del negocio y la modelación de los SI están estrechamente relacionados con los esfuerzos por lograr la evolución de los negocios. Las RN representan el enlace necesario entre el negocio y el modelado de la información; recibiendo especial atención tanto desde el punto de vista académico como desde el punto de vista de la industria, pero no se tiene una definición aceptada y común. A continuación se presentan varios criterios de definición y clasificación para las RN, así como las herramientas que sostienen el enfoque de RN.

#### **1.1.1. Reglas de negocio**

En un artículo escrito por Daniel Appleton (1984) aparece por primera vez en el ámbito computacional la frase “*regla de negocio*”. El autor plantea que si los usuarios

utilizaban términos que variaban en significado de un departamento a otro dentro de una misma organización, los analistas del negocio no podían proporcionar soluciones integradoras. Más tarde comienza a manejarse la idea de captar las RN mediante herramientas computacionales tanto por profesionales de negocios como de sistemas. Estas herramientas garantizan el cumplimiento de las reglas dentro del SI y aseguran que los procesos de negocio sean controlados y manejados de acuerdo a estándares, políticas y procedimientos de negocio (Struck, 1999).

“Los propietarios del negocio tratan las RN de una manera diferente a los desarrolladores”, según Ross<sup>1</sup> (1997). Para los propietarios las RN no son más que directivas que tienden a influenciar o guiar el comportamiento del negocio. Sin embargo para los desarrolladores, las RN no siempre surgen desde las políticas del negocio y son vistas como estructuras dispersas en la lógica del programa.

Desde el surgimiento del término RN los autores abordan su definición de forma diferente (Appelton, 1984; Rosca et al., 1995; Youdeowei, 1997; Hay, 1997; Ceri, 1997; BRG, 2000; Bajec et al., 2000; Morgan, 2002; Ross, 2003b, 2010). Aun cuando ya se logra identificar en la literatura un consenso respecto a concepto.

### 1.1.2. Definición

Varios autores de reconocimiento internacional, como Ross (2003b, 2010), Bajec<sup>2</sup> (2000), Morgan<sup>3</sup> (2002) y Ceri (1997), se acogen a la definición expuesta en (Hay, 1997) al considerar que:

---

<sup>1</sup> Reconocido internacionalmente como “el padre de las reglas de negocio”. Ha organizado la Conferencia Fórum de Reglas de Negocio desde 1997, fundador del Grupo de Reglas de Negocio (BRG) en los años 1980, y autor-editor de importantes artículos del BRG, tales como: *“The Business Motivación Model: Business Governance in a VolatileWorld”* (2007) y *“Business Rules Manifesto”* (2003). Es también miembro activo de la OMG, donde dirige el desarrollo de estándares para RN y modelos del negocio, incluido SBVR.

<sup>2</sup> Profesor asistente de la Universidad de Liubliana, Facultad de Computación y Ciencia de la Información, imparte cursos de: Sistemas de información, Desarrollo de sistemas de información, Bases de datos, entre otros. Sus principales intereses de investigación incluyen metodologías para el desarrollo de los SI y renovación de los procesos de negocio, en los últimos años se ha dedicado fundamentalmente al

*“Una RN es:*

- *Una sentencia que define o restringe algunos aspectos del negocio.*
- *Establece restricciones a la estructura del negocio, controlando o influyendo en el comportamiento del mismo.*
- *No podrá ser fraccionada o descompuesta en RN más detalladas.*
- *En caso de ser reducida perdería información importante sobre el negocio.”*

Esta forma de definir una RN responde a las preguntas: ¿qué hace una regla?, mostrado en los dos primeros puntos; y ¿qué la caracteriza?, expuesto en los dos últimos -donde se destacan propiedades que deben observarse para las reglas: compactación y atomicidad.

El autor de este informe considera como válida esta definición durante el desarrollo del trabajo.

### **1.1.3. Clasificación**

La clasificación de las reglas facilita el descubrimiento, análisis y finalmente su modelación. Esta razón hace que, sin constituir una tarea obligatoria, se recomiende hacerlo para simplificar la formalización de las RN. Además, clasificar las reglas garantiza la consistencia de la descripción y eleva su claridad. El propósito de dicha tarea, como etapa dentro del análisis, es facilitar la identificación y lograr su atomicidad.

Varios autores proponen taxonomías de clasificación según el nivel de abstracción, funcionalidad, relación con el SI, entre otros criterios (Schreiber et al., 1999; Kovacic et al., 2001; Weiden et al., 2002; Morgan, 2002; Horrocks et al., 2004). Este epígrafe aborda dos de estas taxonomías por considerarlas relevantes para la presente investigación.

---

desarrollo de metodologías para el desarrollo ágil y basado en RN de software. Tiene una activa participación en numerosas investigaciones y desarrollo de proyectos.

<sup>3</sup> *Profesor Distinguido* en Ciencia de la Computación en la Universidad de Neumont, donde ha impartido docencia en niveles de pregrado y posgrado, además de investigar en el futuro de los sistemas de información. Tiene una vasta experiencia práctica en relación a la creación y desarrollo de SI. Ha trabajado en compañías tales como EDS y Unisys. Su interés técnico principal está en el uso de manejadores de modelos para construir SI que evidencien el valor de negocios reales. Autor de numerosos artículos y libros que abarcan la temática de las RN.

### Clasificación semántica

El esquema propuesto por Weiden y colaboradores (2002) puede ser usado para clasificar las RN de acuerdo a sus propiedades semánticas. Dicho autor define tipos de RN agrupados en tres categorías (Weiden, 2000; Weiden et al., 2002): *estructurales*, *de comportamiento o conducta* y *de administración*. Estas categorías representan diferentes visiones del negocio. Las reglas *estructurales* son las concernientes a la descripción de aspectos estáticos de un negocio. Las de *comportamiento* definen las condiciones sobre la ejecución de tareas en el negocio. La categoría de reglas de *administración* define las restricciones de alto nivel sobre el negocio.

La distinción entre los puntos de vista: *estructural* y *de comportamiento*, es bien conocida y sirve como base a muchos métodos de análisis de sistema, incluyendo UML. Ambos puntos de vista definen la visión interna del proceso de negocio. El punto de vista *administrativo* agrega una vista externa e introduce nociones tales como: objetivo, valor, recursos necesarios de tareas, actores y procesos del negocio (Weiden, 2000). Este esquema de clasificación asume que se ha creado el modelo de procesos inicial del negocio en términos de los procesos y tareas principales. Cada una de las tres categorías es subdividida en varios tipos de RN:

*Estructural*: estructura de conceptos, persistencia e historia.

*De Conducta*: flujo de información, pre-condición, post-condición, frecuencia, duración, flujo de control y conocimiento de la tarea.

*Administrativa*: organización, objetivo y valor, actitud del actor, responsabilidad del actor y recursos.

Este esquema se apoya en la semántica de las reglas, proporciona categorías, que guardan un estrecho vínculo con los elementos del modelo del negocio, por lo que se considera cercano a los propietarios. A diferencia de la taxonomía propuesta por Goedertier<sup>4</sup> (2007b), que también se apoya en la semántica de las reglas desde la óptica

---

<sup>4</sup> Es Dr. en Ciencias Económicas Aplicadas en 2008, el título de su tesis fue: "Declarative Techniques for Modeling and Mining Business Processes" (Goedertier, 2008). Investigador en administración de procesos de negocio. Profesor de numerosos cursos doctorales, tales como: introducción a la

de las personas del negocio y proporciona muchas sub-categorías, alejándose del negocio y dando una visión más cercana al programador.

#### Clasificación cercana a la implementación

Reportada por Soliveres (1997), este autor se acerca a la visión del implementador y propone las siguientes categorías de RN:

*Reglas del modelo de datos:* engloba todas aquellas reglas que se encargan de validar la información básica almacenada para cada atributo o propiedad de una entidad u objeto.

*Reglas de relación:* incluye todas aquellas reglas que controlan las relaciones entre los datos.

*Reglas de restricción:* restringen los datos que el sistema puede contener. Nótese que este grupo de reglas se solapa, en cierto modo, con las *reglas del modelo de datos*, dado que aquellas también impiden la introducción de datos erróneos. La diferencia estriba en que este tipo de regla restringe el valor de los atributos o propiedades de una entidad más allá de las restricciones básicas que sobre las mismas existen.

*Reglas de derivación:* están integradas por aquellas reglas que permiten derivar cierta información a partir de otra, controlan la obtención de información, realización de cálculos, etc.

*Reglas de flujo:* son las que determinan y limitan cómo fluye la información a través de un sistema, indican qué camino recorre la información y obligan a que se sigan solo aquellos que son permitidos.

Esta taxonomía se vincula estrechamente con la forma en que las reglas son implementadas en el SI, dando la visión de los desarrolladores. Las tres primeras categorías conforman un grupo que está dirigido a la estructura y validación de los datos. Considerando este grupo como una categoría, y unida a los dos restantes se observa gran

---

metodología de la investigación, lenguajes y metodologías de la investigación, aprendizaje automático e inferencia inductiva, entre otros. Revisor de numerosas publicaciones, entre las que se encuentra: Information Systems sobre modelación de procesos de negocio. Es miembro del comité del programa del evento "The International RuleML Symposium 2008", sobre el intercambio y aplicación de reglas. Publica numerosos artículos, entre ellos se destaca el que recibe el reconocimiento como best paper (Goedertier et al., 2007b).



similitud con las clasificaciones propuestas por Kovacic (2001) y Goedertier (2004; 2007a; 2007c), diferenciándose básicamente en que estas dan una visión más cercana al negocio.

Las razones antes expuestas hacen que en esta investigación se utilice como guía la clasificación cercana a la implementación propuesta por Soliveres (1997).

#### **1.1.4. Práctica de desarrollo**

Es de interés para la comunidad de desarrolladores de software, investigadores y usuarios del negocio encontrar fuentes seguras para la modelación de RN. Al ubicar el presente trabajo dentro de un escenario propio para RN, se pueden definir etapas de tránsito además de actores involucrados. Entre las etapas iniciales y más importantes en el ciclo de vida de las RN están: captura, análisis y clasificación. Es necesario que el descubrimiento y la identificación de las reglas se vinculen fuertemente con la captación de los requisitos del SI. La unión entre los lineamientos y los pasos es llamada Práctica de Desarrollo y debe realizarse de manera integrada, realizando la continuidad con la implementación de las reglas identificadas en la arquitectura del SI acorde a lo recomendado según la alineación entre las clasificaciones de RN descritas en el epígrafe anterior.

##### **1.1.4.1. Lineamientos para el descubrimiento**

El ambiente de desarrollo de software propuesto por Bajec y Krisper (2005) como un escenario para la administración de RN, integra actividades de modelación del negocio vinculadas al desarrollo de SI. Se destacan las actividades específicas para administrar las RN a través de su ciclo de vida. Entre ellas se encuentra el descubrimiento, descrito dentro de este escenario como el análisis de la información acerca del negocio para el cual se desarrolla el SI; o sea, para el descubrimiento de las RN debe tomarse información de fuentes que permitan derivar los *negocios oscuros*.

En el presente epígrafe se propone seguir cinco lineamientos, formulados en (Weiden et al., 2002), que conforman la estrategia para el descubrimiento de las RN y facilitan la identificación (Martínez Busto et al., 2009). Estos lineamientos, unidos a los pasos propuestos en el siguiente epígrafe, conforman la práctica de desarrollo para la captación

de las RN propuesta en la presente investigación. Cada lineamiento y su correspondiente fundamentación se detallan seguidamente.

*Lineamiento 1:* Asegurarse de que es posible hablar con la persona indicada.

*Fundamento:* Es difícil extraer algunos tipos de RN cuando, al realizar la entrevista, no se cuenta con la persona adecuada. Esto sucede, especialmente, con las reglas que pertenecen a la categoría de administración, que con frecuencia son poco conocidas o dominadas por el personal de la organización.

Para identificar las reglas que intervienen en la ejecución de tareas es recomendable dialogar con quienes las ejecutan, además de los directivos. Esto se debe a que es posible que existan puntos de vista diferentes sobre las reglas que gobiernan una tarea en particular, que deberán ser analizados.

*Lineamiento 2:* Comenzar obteniendo una vista general sobre el proceso de negocio como un todo e identificar las RN del tipo objetivo y valor.

*Fundamento:* Resulta difícil comenzar la extracción de las reglas cuando no se tiene una visión general de qué pasa en la organización y por qué se hace de esa forma. Especialmente, es mejor disponer del modelo de procesos del negocio cuando se realiza una entrevista para la extracción o reconocimiento de RN, siendo utilizado como un mecanismo estructural. El diagrama de actividades UML es una técnica apropiada para representar el modelo de procesos. Las reglas acerca del objetivo y valor de los procesos pueden ser usadas como punto de partida en la extracción de reglas más específicas sobre los objetivos de tareas.

*Lineamiento 3:* Establecer en etapas tempranas las reglas de estructura de objetos de un dominio.

*Fundamento:* Este tipo de RN abarca la estructura de los objetos del vocabulario de dominio. Es importante que los términos usados para los objetos en el dominio sean identificados previamente, de manera que el riesgo de confusión, en cuanto a la terminología durante la extracción de las reglas, sea el menor posible.

*Lineamiento 4:* Si existen dos vías diferentes de modelar una parte del negocio, escoger la que describa el negocio con términos más claros para la persona entrevistada.

*Fundamento:* Esta situación se presenta cuando se decide si una regla pertenece a la categoría flujo de información/control o pre/post-condición. Ambos tipos o categorías pueden ser usados para modelar dependencias entre tareas.

La mayoría de las personas del negocio prefieren usar las reglas del tipo flujo de información/control. Obviamente, durante una entrevista resulta más fácil hablar acerca de flujos entre tareas que tratar de definir pre y post condiciones de tareas. Luego, si se estima apropiado, las sentencias o reglas obtenidas en lenguaje natural de la entrevista, pueden transformarse al tipo pre/post-condición. Para ello se recomienda optar por la categoría pre/post-condición solo si una regla puede ponerse en términos de una tarea. Por otro lado, mientras que la regla contenga una sentencia referente a más de una tarea se recomienda clasificarla como una regla del tipo flujo de información/control.

*Lineamiento 5:* La categoría *conocimiento de la tarea* puede ser usada para aquellas reglas que especifican cómo una tarea debe ser ejecutada, aunque la información del modelo del negocio no debe enfocarse en su comportamiento interno.

*Fundamento:* Las reglas del tipo *conocimiento de la tarea* deben ser detalladas para ser incluidas en un modelo del negocio, pero es conveniente registrarlas aun cuando no lo sean. Estas reglas actúan como base de información utilizada en algún punto posterior del proyecto, y proporcionan entradas importantes para conformar un SI.

#### **1.1.4.2. Pasos para el análisis**

Los lineamientos propuestos guían el descubrimiento de las RN, pero no constituyen una práctica que establezca la forma en que esto debe hacerse. En este epígrafe se proponen los pasos a seguir como método de trabajo para el análisis de las reglas utilizando la notación UML. A través del análisis de las RN son descompuestos los *negocios oscuros* en sentencias precisas y atómicas. Esto incluye la clasificación de las reglas, basadas en un esquema predefinido y cuyo objetivo es proveer estructuras donde puedan ser fijadas respecto a su naturaleza o el tipo de información que proporcionan. En este epígrafe se propone continuar los pasos para el análisis y clasificación de las RN desde una perspectiva de la empresa, extrayéndolas desde el modelado del negocio. Dicha propuesta se apoya en el trabajo de García Molina (2004), donde además se emplea la taxonomía de clasificación semántica por su vínculo directo y natural al propio

modelado. Los pasos son descritos en (Martínez Busto et al., 2009; Machado Padilla, 2009).

### **Paso 1ro: Obtener los roles**

La definición del conjunto de procesos del negocio es una tarea crucial, debido a que se especifican los límites del proceso de modelado posterior. Teniendo en cuenta que los objetivos estratégicos de la organización son generalmente muy complejos y con un nivel de abstracción elevado, estos serán descompuestos en un conjunto de sub-objetivos más concretos que deberán cumplirse para conseguir dicho objetivo estratégico. Estos sub-objetivos pueden a su vez ser descompuestos en otros de manera que se defina una jerarquía, es suficiente obtener dos o tres niveles de descomposición. Para cada sub-objetivo de segundo nivel se define un proceso de negocio que lo soporte. Se propone representar cada proceso de negocio como un caso de uso del negocio.

Una vez identificados los procesos de negocio se plantea encontrar los agentes involucrados en su desempeño, donde cada uno de ellos juega un determinado papel (o rol) cuando colabora con otros para llevar a cabo cualquiera de las actividades que conforman un caso de uso del negocio. Dentro de este primer paso en la obtención del modelo del negocio se propone obtener el diagrama de casos de uso del negocio para lograr una visión más clara y general de los procesos del negocio de la organización.

### **Paso 2do: Obtener colaboración entre roles**

En el segundo paso se propone, primeramente, describir en detalles cada uno de los casos de uso que han sido identificados en el paso anterior. A continuación se determinan los agentes que juegan un rol en cada caso de uso del negocio, para finalmente brindar una especificación detallada de ellos. Por lo que se propone, en este paso, obtener el diagrama de roles (que expresa la colaboración entre roles desde el punto de vista estructural) para desarrollar un caso de uso determinado del negocio. Este diagrama permite expresar el conocimiento que tienen unos roles sobre otros, además de la multiplicidad entre sus relaciones.

### **Paso 3ro: Obtener comportamiento de colaboración**

Como tercer paso se propone crear escenarios para mostrar el comportamiento de la colaboración entre los roles anteriormente identificados. Para este propósito se

recomienda emplear diagramas de secuencia UML, donde los objetos denotan instancias de roles que intervienen en la interacción para cada caso de uso.

#### **Paso 4to: Obtener interacción entre procesos, flujo de información y actores**

En este paso, para mostrar el flujo de trabajo de determinados objetivos de la organización, se propone utilizar un modelo del negocio representado mediante una vista de proceso basada en el estándar IDEF0. Este diagrama muestra el flujo de trabajo, indicando qué rol realiza cada actividad y cuáles son los datos requeridos y producidos por esta. Además, para ello pueden usarse diagramas de actividad de UML con calles, nombrándolos diagramas de procesos.

En cada proceso se puede establecer una distinción entre el flujo básico o normal de la interacción (ejemplo: realización de exámenes de laboratorio) y los posibles flujos alternativos (ejemplo: análisis por sexo). Para mejorar la legibilidad es conveniente asociar varios escenarios a un mismo caso de uso del negocio, en vez de mostrar en una sola secuencia todas las posibilidades.

En este diagrama de procesos existe una calle para cada rol participante (caso de uso del negocio), incluyendo las actividades que realiza el mismo, y la información que necesita y produce cada actividad.

#### **Paso 5to: Identificar y clasificar reglas a partir del modelo del negocio**

En este paso se propone identificar las reglas basándose en el modelo del negocio obtenido de acuerdo a los pasos anteriores. Se propone además, clasificar las RN identificadas según el esquema semántico. Cada RN puede ser enmarcada dentro de una de las categorías, algunas se ven identificadas en el diagrama de procesos de diferentes formas, por ejemplo:

- Las RN de tipo flujo de información se ven a través de las líneas discontinuas orientadas en el sentido de la flecha, indicando qué información necesita una tarea o actividad.
- Otro tipo de reglas, como la de flujo de control, indican el orden entre tareas. Se representan en el diagrama por flechas continuas.

- Las reglas de tipo responsabilidad del actor dicen qué actor es responsable de una determinada tarea; pueden ser extraídas del diagrama al identificar las actividades encerradas en la calle perteneciente a dicho actor (o rol).

Esta taxonomía es útil debido a que las RN pueden ser extraídas desde el propio lugar al que pertenecen, o sea, desde la propia lógica del problema donde juegan un rol en la política de la organización. Es en este quinto paso donde se extraen las reglas de forma natural, a partir del propio modelo del negocio y siguiendo el esquema de clasificación semántico, considerado para esta investigación.

Como práctica de desarrollo para lograr rigor y sistematicidad en la captación de las RN desde el *enfoque de RN* se proponen la guía para el trabajo mediante la asunción de los cinco lineamientos propuestos en el epígrafe 1.1.4.1, el uso de los pasos para el análisis de las RN basado en el modelado del negocio y su clasificación según criterios semánticos.

Por otra parte, la idea de lograr mayor independencia en una aplicación o tarea particular debe verse ligada a recurrir siempre al vocabulario del negocio y modelos ontológicos. Es de interés que en la comunidad de RN se reconozcan las reglas como artefactos que, por la naturaleza del problema (o estrategia del negocio), soportan e infieren la estrategia que es aplicada al problema (Gerrits, 2004). Por lo que las RN que resultan ser capturadas en el contexto de una tarea particular serán menos reusables y más específicas.

### **1.1.4.3. Implementación de las reglas en el sistema sobre el MVC**

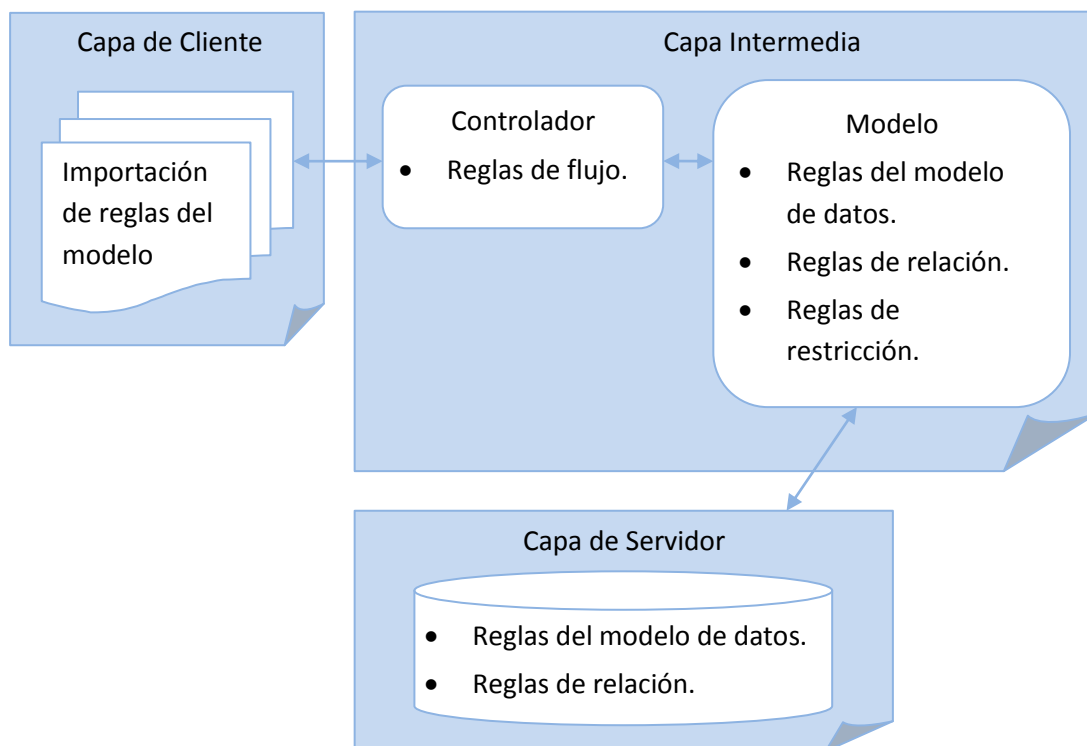
Las RN deben administrarse como un activo independiente de una organización (Ross, 2003b). De estar dispersas y mezcladas con el código de la aplicación se dificulta, entre otras cosas, su localización al momento de modificarlas por algún cambio en el negocio. La motivación para mejorar la ubicación y administración de estas reglas es justamente lograr incrementar el control y el conocimiento en la organización de cómo, por qué, cuándo, dónde y gracias a quién se fortalecen las mismas.

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del MVC corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la

programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí. Esta arquitectura facilita responder a los cambios, permite reutilizar código, simplifica el mantenimiento y hace más fácil la migración a nuevas plataformas.

El patrón MVC básicamente lo que hace es dividir la capa del negocio en dos capas, la capa del *modelo* y la capa del *controlador*; lo que permite resolver el problema de la ubicación de las RN dentro de los SI. Ubicar una determinada RN dentro del MVC puede simplificarse mucho si se tiene en cuenta el tipo de regla tratada.

La Figura 1.1 muestra la ubicación recomendable para las RN según (Mena et al., 2008; Velazco et al., 2010) utilizando la clasificación dada por Soliveres (1997).



**Figura 1.1** Ubicación de las reglas de negocio dentro del Modelo Vista Controlador.

## **1.2. Caso de estudio: área de nefrología**

La necesidad de obtener sistemas informáticos que abarquen los procedimientos durante el seguimiento a pacientes de nefrología hace viable la aplicación de SI basados en RN. Los pacientes son tratados en esta área de la medicina, donde la práctica de trasplante ha sido adoptada en los últimos años como primera opción terapéutica. Cada procedimiento a seguir durante el tratamiento es regido por complejos protocolos de trabajo. El paciente puede pasar por diferentes tipos de consultas según sean sus avances y evolución particular. Médicos y paramédicos deben tomar decisiones importantes ante cada nueva situación según los resultados en los exámenes realizados a sus pacientes y de los protocolos que rigen todo el proceso.

Con los avances científicos y los resultados investigativos en este campo de la ciencia, las regulaciones para el seguimiento de los casos en dicha área suelen ser dinámicas, por lo que surge la necesidad de SI capaces de adaptarse, de forma rápida y segura.

La informática médica, en particular, es la aplicación de la informática y las telecomunicaciones a la salud, tiene como objetivo principal prestar servicio a los profesionales de la salud para mejorar la calidad en la atención médica; representa la intersección de las ciencias de la información, ciencias de la computación y la atención de la salud. Esta rama de la informática se ocupa de los recursos, los dispositivos y los métodos necesarios para optimizar la adquisición, almacenamiento, recuperación y utilización de la información en salud y biomedicina. Los instrumentos informáticos de la salud incluyen, no sólo los ordenadores, sino también guías de práctica clínica, terminología médica formal y de sistemas de información y comunicación.

### **1.2.1. Control de los pacientes**

En el AN los pacientes reciben atención especializada insertándose en un sistema de atención secundaria y terciaria personalizada. Se requieren registros permanentes que faciliten gestionar los exámenes y la evolución para cada enfermo. Dicha área incluye diversas especialidades que, en dependencia de cada uno estos controles, tiene sus peculiaridades. La atención al paciente se realiza de acuerdo al seguimiento en tres tipos de consultas:



- *Consulta de progresión o renal crónica avanzada* (externa o evolutiva).
- *Consulta de hemodiálisis.*
- *Consulta de trasplante.*

Cada consulta tiene sus propios requerimientos de información, protocolos de trabajo y restricciones, que deben ser consideradas tanto en las prácticas médicas como para los requisitos y funcionalidades que se esperan de los sistemas computacionales de apoyo. Esto resulta de gran importancia, no solo para el personal médico y paramédico, también para pacientes y familiares relacionados con esta área. Los datos de cada paciente deben estar disponibles para los diferentes especialistas que lo atienden. Además, es deseable brindar información para pacientes y personas allegadas, posibilitando que conozcan -no solo del estado actualizado del enfermo- sino que adquieran cultura en cuanto a los cuidados que se deben tener con pacientes con estas características, donde la atención familiar resulta decisiva en el éxito del tratamiento.

En cada tipo de consulta se registran parámetros diferentes acerca del estado de un paciente, que puede o no haber sido tratado. Esto sugiere manejar al paciente como entidad abstracta, accesible desde cualquier consulta, con la seguridad de disponer de una vista actualizada del mismo. A la vez, debe responder a leyes y regulaciones dependientes del contexto que ofrece cada consulta en particular.

La insuficiencia total o casi total en el funcionamiento del riñón para excretar los desechos, concentrar la orina y regular la electrólisis es considerada como *insuficiencia renal crónica* (IRC). Cuando esta enfermedad está en una etapa terminal, que se considera el desenlace común a múltiples enfermedades que afectan al riñón, se denomina *insuficiencia renal crónica terminal* (IRCT). Esta se presenta cuando los riñones ya no pueden funcionar al nivel necesario para la vida diaria, es decir, que la IRC progresa a un punto tal que la función de los riñones es menor del 10% de su capacidad normal. El AN atiende pacientes en los diferentes *estadios* de esta enfermedad.

Determinar el *estadio* en que se encuentra cada paciente es una tarea fundamental dentro del área y de acuerdo a ello será el tratamiento que reciba.

Los posibles *estadios* de un paciente se encuentran entre los valores del I al V. De encontrarse el paciente en alguno de los dos primeros es reorientado a su área de salud para ser atendido allí en la *consulta de nefrología*. Si se encuentra en uno de los *estadios* III o IV el paciente es atendido en la propia *consulta de progresión*. Cuando el *estadio* del paciente es V se considera que la enfermedad se encuentra en *estado terminal* y el paciente es remitido a uno de los métodos sustitutivos de la función renal: *diálisis peritoneal, hemodiálisis o trasplante renal*.

El recibir terapia con *diálisis peritoneal* puede representar una opción para salvar la vida de aquellos pacientes con IRCT que no pueden, por alguna razón, recibir tratamiento de *trasplante* del órgano; este puede provenir de dos fuentes:

- Un donante familiar vivo: (genéticamente emparentado con el receptor: padres, hermanos o hijos) (Kasiske et al., 2001).
- Un donante muerto o donante con muerte encefálica (persona recientemente fallecida que no ha tenido enfermedad renal crónica) (Ojo et al., 2001).

Al decidir el origen del órgano se establecen dos tipos posibles de donantes para conformar la pareja *receptor-donante potencial*<sup>5</sup>, considerando que es posible tener: uno o varios donantes potenciales y uno o varios posibles receptores. La pareja constituida es sometida a un *protocolo de compatibilidad*.

En caso de tener un resultado favorable, al finalizar el *protocolo de compatibilidad* se envían los pacientes a la *consulta de TR o consulta multidisciplinaria*, o sea al *receptor y posibles donadores o donante potencial* (se pueden proponer uno o varios *donadores candidatos*). Esta consulta se integra de varios especialistas, entre ellos: nefrólogo, cirujano y urólogo, y otros; generalmente *es dirigida por* un nefrólogo (Hernández et al., 2003), y permite evaluar la pareja *receptor-donador potencial*.

El *donante* juega un papel muy importante dentro de este proceso. Identificar el mejor entre los *posibles donadores* es una delicada tarea guiada por un complejo *protocolo*. Cada uno de los exámenes va descartando candidatos para, finalmente, elegir la mejor

---

<sup>5</sup> Se considera "*receptor-donante potencial*" o "*receptor-donador potencial*" a la pareja que se integra por un receptor con un posible donante de riñón.

posibilidad de éxito en la operación. Una vez completado el *protocolo para donante potencial*, si no se rechazan todos, se pasa a seleccionar el donador. En caso exitoso se conforma la pareja *donante-receptor* y se da continuidad al *protocolo de trasplante*.

La *operación de trasplante* asociada a un *donante con muerte encefálica* sigue un procedimiento muy similar a la que es asociada a *donante vivo*, difieren básicamente respecto a la *operación de extracción del órgano del donante* y al tratamiento del *donador*. Los registros respecto al *donante* también difieren, pero en cuanto a otras actividades básicas son exactamente iguales: los protocolos y registros para el órgano donado, la operación de extracción e implante del mismo, y el seguimiento postoperatorio para el receptor.

Con el fin de evitar el rechazo casi todos los *receptores* de trasplante de riñón requieren tratamiento de por vida. El éxito de un trasplante de riñón depende, en parte, del *seguimiento* minucioso y el cumplimiento metódico del régimen de medicamentos. Los pacientes deben ser chequeados periódicamente, controlando un conjunto de parámetros que evidencian posibles padecimientos o complicaciones. El seguimiento al que es sometido el *paciente* posterior a la operación depende de su desarrollo y evolución. Esta consulta es llamada *consulta de progresión, externa o evolutiva*.

Las únicas posibles formas en que un enfermo puede salir del sistema es por la muerte del paciente debido a causas diversas o debido a que cualquier *paciente o posible donante* abandone el tratamiento en cualquier momento de manera voluntaria.

### **1.2.2. Caracterización del caso de estudio**

De la descripción anterior se puede concluir que:

1. Durante la atención a un paciente en esta área se tienen que tomar múltiples decisiones que están muy bien reglamentadas en diferentes protocolos.
2. Las decisiones están fuertemente interrelacionadas, o sea, unas dependen de otras.
3. Cada usuario, médico, paramédico, paciente o familiar tiene requerimientos informativos diferentes. Debe propiciarse el acceso y manejo de la información a cada cual según su rol.

4. Los tratamientos y procedimientos son susceptibles a cambios, e incluso hasta pueden variar de acuerdo a la región o país. Se rigen por normativas o avances científicos a diferentes niveles.
5. Se debe permitir a usuarios avanzados dar mantenimiento a las políticas trazadas en un ambiente flexible, este tipo de usuario puede ser incluso, el propio médico.

Todo lo anterior sugiere que la aplicación del *enfoque de RN* facilita la obtención de soluciones computacionales para este dominio. Debido, fundamentalmente, a la elevada probabilidad de cambios rápidos y constantes que afectan el ambiente de la aplicación, identificándose la necesidad de renovación y adaptación.

El sector de la salud es un área en que los cambios son muy frecuentes dado el vertiginoso desarrollo que tiene la medicina, tanto en nuestro país como a nivel mundial. Estos cambios se originan en la comunidad científica y se canalizan a través del sistema de salud hasta su instrumentación por parte de los especialistas, que incluso pueden aportar, con su experiencia, determinadas variaciones. Lo anterior, refuerza el hecho que deben ser los propios especialistas del negocio, en este caso los médicos, los encargados de crear y dar mantenimiento a sus políticas, expresadas con una terminología que depende en gran medida de cada área o sector. Hay que tener en cuenta las reglamentaciones y normativas que rigen o restringen los diferentes procesos y actividades en un negocio, no solo en la etapa inicial del ciclo de vida de un SI, sino también a lo largo del resto de las diferentes etapas, especialmente en sectores tan sensibles como el de la salud.

Ante el imperativo de obtener un sistema de información con estas características se evidencia la necesidad de:

- Recopilar información en la etapa de análisis de los requerimientos del software que permita crear y mantener los registros relacionados con la actividad y manejo de pacientes del AN y su seguimiento.
- Determinar qué reglamentaciones y normativas permiten la fiscalización de cada etapa en esta área, según los protocolos de trabajo establecidos.

- Determinar el vocabulario médico utilizado por la comunidad profesional ligada a cada etapa o actividad concreta.

### **1.3. Calidad de software**

#### **¿Qué es la calidad del software?**

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

La calidad del software es medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable al nivel de “cero fallas”; un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad; mientras que un producto de software para ser explotado durante un largo período (10 años o más), necesita ser confiable, mantenible y flexible para disminuir los costos de mantenimiento y perfeccionamiento durante el tiempo de explotación.

La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.

#### **Definiciones de calidad del software**

A continuación se refieren algunas definiciones de calidad de software dadas por (Lovelley, 1999).

- “Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”.
- “El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”.

La calidad, al igual que la belleza, está en el ojo de quien lo mira, sin embargo, desde el punto de vista de medición, se debe tener una definición precisa en términos de atributos del software que sean de interés al usuario en general, éstos son atributos externos sin embargo, muchas propuestas miden y analizan atributos internos porque los consideran predictores de aquellos externos, los atributos internos tienen dos ventajas:

1. Están disponibles para medición más temprano.
2. Son más fáciles de medir.

Desde los primeros modelos de calidad en el inicio de la ingeniería de software, se observó que la calidad está dada por un amplio conjunto de características, un modelo de calidad describe entonces estas características y sus relaciones, muchos modelos hacen difusa la distinción entre atributos internos y externos, lo que dificulta la comprensión del concepto de calidad, los modelos que se presentarán a continuación son los que han ganado mayor popularidad en la comunidad, pero no tienen sustento científico, extrayendo los factores comunes a todos ellos es posible derivar modelos propios adaptados a usos específicos

En el desarrollo de software, la calidad de diseño acompaña a la calidad de los requisitos, especificaciones y diseño del sistema. La calidad de concordancia es un aspecto centrado principalmente en la implementación; si la implementación sigue al diseño y el sistema resultante cumple con los objetivos de requisitos y de rendimiento, la calidad de concordancia es alta.

Adicionalmente se puede seguir los siguientes aspectos para evaluar la calidad del software:

- Funcionalidad
- Confiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Portabilidad
- Escalabilidad

Por lo anteriormente analizado se concluye que no se puede medir la calidad del software de forma correcta debido a su naturaleza, la certificación se da a los procesos, la correcta consecución de los mismos garantizaría un buen software. No se puede medir al software como tal, sino los atributos que lo conforman, tales métodos de medida deben ser exactos. El usuario final mide la calidad del software según lo que tenga o no, es en ese sentido de que la calidad del software depende de quien la juzgue. El hecho de que una empresa tenga certificación en calidad de software no garantiza que su software sea de calidad.

- Los requisitos del software son la base de las medidas de calidad.
- La falta de concordancia con los requisitos es una falta de calidad.
- Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se sigue ninguna metodología siempre habrá falta de calidad.
- Existen algunos requisitos implícitos o expectativas que a menudo no se mencionan, o se mencionan de forma incompleta (por ejemplo el deseo de un buen mantenimiento) que también pueden implicar una falta de calidad.
- La calidad de software no se certifica, lo que se certifica son los procedimientos para construir un software de calidad, los procedimientos deben ser correctos y estar en función de la normalización.

#### ***1.4. Conclusiones parciales***

Este capítulo expone la definición, clasificación y propiedades de las RN, haciendo un análisis del uso de estas dentro de los SI. Se plantean las características de dos importantes taxonomías de clasificación: semántica y cercana a la implementación, ya que aunque no son necesarias para lograr el manejo adecuado de las RN facilitan su captación. De las clasificaciones mencionadas, la cercana a la implementación es la usada en la presente tesis ya que facilita al desarrollador determinar la ubicación de las RN en la arquitectura del sistema, debido a que la descripción es cercana al código. Se presenta el caso de uso para el AN y finalmente se concluye abordando el tema de calidad de software.

## CAPÍTULO II. ADAPTABILIDAD DEL SISTEMA DE INFORMACIÓN

Este capítulo se enfoca a la aplicación de los criterios que determinan la ubicación de las RN dentro del SI y su inserción en su capa correspondiente, para lograr la fácil actualización a las nuevas condiciones del negocio, haciendo uso del patrón de diseño MVC. Se presentan las RN implementadas, se argumentan sus ubicaciones y se muestran ejemplos de cómo quedan embebidas en el código del sistema. Se exponen las nuevas características de la versión propuesta centrándose en las herramientas incorporadas, en especial los generadores automáticos de reportes para la extracción y exportación de los datos. Por último se explica el método aplicado en la evaluación del software.

### ***2.1. Implementación de las RN en el SI usando el patrón MVC***

En el sistema se recomienda que la mayor parte de las reglas sean ubicadas en el modelo, lugar ampliamente recomendado en la arquitectura MVC para su localización. Esto se debe a que en este se encuentra la representación específica de la información con la cual el sistema opera. Como excepción están las reglas de flujo que son excelentes candidatas a ser implementadas en el controlador debido a que su complejidad suele ser, por lo general, bastante elevada y su comportamiento se ajusta al objetivo del controlador.

Para una mejor comprensión del tema tratado en esta sección es necesario referirse a un método muy utilizado en la aplicación. Este método es *beforesave ()* declarado en el modelo. Dicho método se ejecuta inmediatamente después de que los datos del modelo han sido satisfactoriamente validados, pero justo antes de que los datos sean guardados. Esta función debe devolver un valor *true* si se desea que continúe la operación de grabado. Este *callback* es especialmente útil para cualquier lógica de tratamiento de datos que necesita ocurrir antes de que los datos sean almacenados, por tal motivo es el lugar recomendado, por el autor de la presente tesis, para insertar funciones de validación dadas por las RN.



### 2.1.1. Reglas implementadas para el caso de estudio

A continuación se refieren RN implementadas en el sistema para el AN. Se ejemplifican agrupándolas según la clasificación dada por Soliveres (1997).

- **Reglas del modelo de datos:**

Con un correcto diseño de la BD quedan automáticamente implementadas muchas de las reglas del sistema en la capa del *servidor*, esto se hace, en primer lugar, escogiendo correctamente el tipo de los campos de cada tabla, pero es muy importante que estas se refuercen en la capa intermedia (modelo) para una mayor independencia del gestor de BD y brindar mayor robustez a la aplicación.

Estas reglas se insertan en el modelo (ubicado en la capa intermedia) en forma de una matriz de condiciones. A continuación se muestran algunos ejemplos de RN, las cuales se implementan en el modelo de historia clínica (*clinical history*) del paciente.

**R1:** El número de carné de identidad de un paciente debe cumplir con ciertas condiciones básicas, como que es de tipo entero con una longitud de 11 dígitos.

```
var $validate = array(
    'ci'=> array(
        'isUnique'=> array(
            'rule'=> 'isUnique',
            'required'=> true,
            'message'=> 'Número de carnet repetido'
        ),
        'custom' => array(
            'rule' => array('custom', '/^\d{11}$'/),
            'required' => true,
            'message' => 'Debe introducir un número de 11 dígitos'
        ),
    )
);
```

**R2:** La fecha de ingreso al programa de un paciente debe ser del tipo fecha.

```
var $validate = array(
    'fecha_de_ingreso_al_programa' => array(
        'date' => array(
            'rule' => 'date',
            'message' => 'Ingresa una fecha válida usando el formato AAAA-MM-DD',
        )
    )
);
```

Para evitar envíos erróneos de información y esperas innecesarias por parte del usuario se deben implementar estas validaciones en la capa del cliente. Esto se logra importando del modelo a la vista, las reglas del modelo de datos, haciéndose en la misma gran parte de los controles necesarios para garantizar la validez de la información introducida por el usuario, no es difícil, dado que son chequeos relativamente sencillos, al afectar solo a un campo, y de hecho algunos *plug-in* para el *framework CakePHP* implementan esta funcionalidad.

- **Reglas de relación:**

La validación de estas reglas se puede conseguir en la *capa del servidor* ya que la mayoría de los gestores de BD proporcionan integridad referencial y los mecanismos necesarios para implementar fácilmente estas reglas, lo que proporciona mayor robustez a la BD. Esto se logra utilizando el motor de almacenamiento InnoDB que es una tecnología de almacenamiento de datos de código abierto para MySQL, incluido como formato de tabla estándar en todas las distribuciones de MySQL AB a partir de las versiones 4.0. Su característica principal es que soporta transacciones de tipo ACID y bloqueo de registros e integridad referencial. InnoDB ofrece una fiabilidad y consistencia muy superior a MyISAM, la anterior tecnología de tablas de MySQL. No obstante estas reglas se implementan en los modelos en forma de una matriz de relaciones permitiendo la comprobación de estas en un lenguaje de alto nivel como lo es

PHP, independizando la aplicación del gestor de BD. A continuación se muestran algunos ejemplos:

**R3:** Un posible receptor puede ser asociado al menos a un donante potencial.

Esta regla asegura que un receptor se tenga que asociar a una serie de donantes potenciales (candidatos a donar al realizar el trasplante de donante vivo) antes de la realización del trasplante. Este paso intermedio es importante, pues de la relación de estos se debe insertar, en caso de existir, el parentesco y además el receptor tiene sus datos relacionados como son: Causa de la IRC, Terapia Sustitutiva y Tiempo en Terapia Sustitutiva. Para definir esta regla se incluye la siguiente relación en el modelo donante (*clinical history of donor*):

```
public $belongsTo = array(
    'ClinicalHistoryOfReceptor' => array(
        'className' => 'ClinicalHistoryOfReceptor',
        'foreignKey' => 'clinical_history_of_receptor_id'
    ));
```

En el modelo receptor (*clinical history of receptor*) la relación inversa se representaría como sigue:

```
public $hasMany = array(
    'ClinicalHistoryOfDonor' => array(
        'className' => 'ClinicalHistoryOfDonor',
        'foreignKey' => 'clinical_history_of_receptor_id'
    ));
```

**R4:** Un posible receptor puede ser asociado al menos a un trasplante de donante vivo. Para la implementación de esta regla en el modelo trasplante de donante vivo se inserta la siguiente relación:

```
var $belongsTo = array(
    'ClinicalHistoryOfReceptor' => array(
        'className' => 'ClinicalHistoryOfReceptor',
        'foreignKey' => 'clinical_history_of_receptor_id',
    ));
```

En el modelo receptor (*clinical history of receptor*) la relación inversa se representaría como sigue:

```
var $hasMany = array(
    'TrasplantOfDonorAlife' => array(
        'className' => 'TrasplantOfDonorAlife',
        'foreignKey' => 'clinical_history_of_receptor_id',
    ));
```

**R5:** Todo análisis realizado puede ser asociado a un paciente.

Esta regla establece la relación existente entre el paciente y los análisis realizados a este. Su implementación se realiza en el modelo análisis realizado insertando la siguiente relación:

```
var $belongsTo = array(
    'ClinicalHistory' => array(
        'className' => 'ClinicalHistory',
        'foreignKey' => 'clinical_history_id',
    ));
```

En el modelo historia clínica la relación inversa se representaría como sigue:

```
var $hasMany = array(
    'AnalisyRealize' => array(
        'className' => 'AnalisyRealize',
        'foreignKey' => 'clinical_history_id',
    ));
```

- **Reglas de derivación:**

En casos sencillos pueden implementarse en la *capa de usuario* debido a que no se requiere información adicional y en general toda la información de un registro viaja al mismo tiempo al cliente. En casos complejos es recomendable implementarlas en la *capa servidor*. En la aplicación al tratarse de una fórmula en donde sus valores no precisan de interacción entre varias tablas puesto que es bastante sencilla se implementa

en el modelo, el cual se ubica en la *capa intermedia*, con lo que se logra la centralización en una única capa y facilita su acceso. A continuación se muestra un ejemplo:

**R6:** La hidratación del receptor en sala de trasplante es calculada mediante el peso del receptor en Kg \* (30 cc).

Esta regla se usa para calcular la hidratación del receptor a partir de su peso y se implementa adicionando la siguiente función en el modelo receptor:

```
function rnCalcularHidratacion($pacienteId){
$this->Receptor->id = $pacienteId;
$peso = $this->Receptor->field('peso');
return $peso * 30;
}
```

**R7:** El Índice de Masa Corporal (IMC) de un paciente es calculado mediante el peso en Kg / estatura<sup>2</sup> en metros.

Esta regla se usa para calcular el IMC de un paciente a partir de su peso y estatura. Se implementa adicionando la siguiente función en el modelo historia clínica (clinical\_history):

```
function rnCalcularIMC($pacienteId){
$this->ClinicalHistory->id = $pacienteId;
$peso = $this->ClinicalHistory->field('peso');
$estatura = $this->ClinicalHistory->field('talla');
return $peso / pow($estatura , 2);
}
```

- **Reglas de restricción:**

Estas deben implementarse en el servidor, o en la *capa intermedia*. El lugar ideal podría ser el servidor, pero aquí aparecen las limitaciones del gestor de BD, por lo que ubicar estas reglas en la *capa intermedia* es una buena solución. La ubicación de estas en el sistema se realiza en el modelo, en algunos casos en los que se requiere de acceso a otra tabla se implementa mediante una función de validación y en el caso que involucra la

restricción en valores de un campo en particular se consigue mediante una matriz de condiciones. A continuación se muestran algunos ejemplos:

**R8:** Un paciente solo puede ser receptor una única vez.

**R9:** Un receptor no puede ser donante.

Estas reglas garantizan que al incluir un paciente como receptor este no puede haber sido receptor ni donante previamente. La implementación se realiza en el modelo receptor (*clinical\_history\_of\_receptor*) incluyendo lo siguiente:

```
function beforeSave(){  
    $array_ids = array_merge( $this->getIdReceptors(), $this->getIdDonors());  
    return (!in_array($this->data['ClinicalHistoryOfReceptor']['clinical_history_id'],  
        $array_ids));  
}
```

**R10:** Un paciente solo puede ser donante una única vez.

**R11:** Un donante no puede ser receptor.

Estas reglas garantizan que al incluir un paciente como donante este no puede haber sido donante ni receptor previamente. La implementación se realiza en el modelo donante (*clinical\_history\_of\_donor*) incluyendo lo siguiente:

```
function beforeSave(){  
    $array_ids = array_merge( $this->getIdReceptors(), $this->getIdDonors());  
    return (!in_array($this->data['ClinicalHistoryOfDonor']['clinical_history_id'],  
        $array_ids));  
}
```

**R12:** Un paciente no puede tener dos resultados de un mismo análisis en la misma fecha y hora.

Esta regla es la principal que se impone en los análisis realizados a los pacientes que asegura que no existan colisiones entre los resultados, haciéndolos únicos en una determinada fecha y hora y para un mismo paciente. Una observación importante en esta regla es que se cumple incluso si el análisis pertenece a distintos tipos de estudio.

Esta regla se define en el modelo análisis realizados (*analisy realize*) incluyendo el siguiente código:

```
function beforesave(){
$analisyRealize = $this->AnalisyRealize->find('first', array(
    'conditions' => array(
        'AnalisyRealize.fecha' => $this->data['AnalisyRealize']['fecha'],
        'AnalisyRealize.clinical_history_id' => $this->data['AnalisyRealize']['clinical_history_id'],
        'AnalisyRealize.analisy_id' => $key
    )
));
return empty($analisyRealize);
}
```

**R13:** Un posible receptor de un trasplante renal con donante vivo debe tener una edad entre 15 y 55 años.

Su implementación se realiza en el modelo receptor (*clinical history of receptor*) mediante la aplicación de una regla de validación para la edad, quedando de la siguiente forma:

```
var $validate = array(
    'edad' => array(
        'between' => array(
            'rule' => array('range', 15, 55),
            'message' => 'El receptor debe tener una edad entre 15 y 55 años'
        )
    )
);
```

- **Reglas de flujo:**

Estas reglas, que generalmente suelen ser de elevada complejidad, pueden resultar inmanejables por un gestor de BD por lo que deben ubicarse en la *capa intermedia*. La mejor solución sería implementarlas en el *controlador* que es quien recibe todas las

peticiones determinando qué camino recorre la información y si se necesita de información de otras tareas para poder ejecutarse. A continuación se muestran algunas reglas con su implementación dentro del sistema.

**R14:** Un TR debe realizarse cuando el VIH del paciente fue negativo.

**R15:** Un TR debe realizarse cuando el Hepatitis B del paciente fue negativo.

**R16:** Un TR debe realizarse cuando el Hepatitis C del paciente fue negativo.

**R17:** Un TR debe realizarse cuando el VIH del donante fue negativo.

**R18:** Un TR debe realizarse cuando el Hepatitis B del donante fue negativo.

**R19:** Un TR debe realizarse cuando el Hepatitis C del donante fue negativo.

Para la implementación de estas RN en el sistema ocurre lo siguiente: primeramente en la capa controladora se obtiene el "id" del paciente al que se le quiere realizar el trasplante y se llama a la función "*getLastAnalisis*" del modelo "*AnalisisRealizados*", que devuelve el resultado del análisis para ese paciente. Posteriormente se obtienen los "id" de los donantes potenciales para ese paciente y se pasan los valores de las variables a la vista "*add*" de trasplante, como se indica a continuación:

```
function add($receiver_id) {
    $donante_id = array();
    $vihd = array();
    $hepatitisd = array();
    //Obtengo los datos del receptor
    $receptor = $this->Transplant->Receiver->find('first',
array('conditions' => array('Receiver.id' => $receiver_id)));
    //Cargo el modelo que me haga falta
    $this->loadModel('Patient');
    $patient_id = $receptor['Receiver']['patient_id'];
    $this->loadModel('AnalysisRealize');
    $vih = $this->AnalysisRealize->getLastAnalisis ($patient_id,
'HIV');
```



```

        $hepatitis      =      $this->AnalysisRealize->getLastAnalysis
($patient_id, 'TGP');

        $this->set('vih', $vih);
        $this->set('hepatitis', $hepatitis);
        $this->loadModel('PotentialDonor');

        //Obtengo los datos del donante potencial
        $receptor = $this->PotentialDonor->find('all', array('conditions' =>
array('PotentialDonor.receiver_id' => $receiver_id)));
        foreach($receptor as $donante){
            $donante_id[] = $donante['PotentialDonor']['patient_id'];
        }

        foreach($donante_id as $value_id){
            $vihd[] = $this->AnalysisRealize-
>getLastAnalysis($value_id, 'HIV');
            $hepatitisd[] = $this->AnalysisRealize-
>getLastAnalysis($value_id, 'TGP');
        }

        $c = true;
        $d = true;
        foreach($vihd as $values){
            if($values){
                $c = $values || $c;
            }else{
                $c = false;
            }
        }

        foreach($hepatitisd as $values){
            if($values){
                $d = $values || $d;
            }else{
                $d = false;
            }
        }

```

```

    }
}

// Envío los valores de las variables a la vista
$this->set('c', $c);
$this->set('d', $d);
$this->set('receiver_id', $receiver_id);
.
.
.

```

En la vista “add” de trasplante se toman los valores de las variables que se pasan del controlador de trasplante, y se muestra un mensaje cuando el trasplante no debe ser realizado o el formulario para la inserción de los datos del paciente en TR en caso que los valores sean correctos.

```

<?php
if(($vih || $hepatitis) || ($c || $d)){ ?>
    <div class="contenido">

        <?php if($vih && $hepatitis){ ?>
            <p>A este paciente no se le puede realizar el trasplante porque los siguientes
            exámenes no se han realizado o han dado positivo:
            <?php echo 'HIV, TGP';?></p>
            <?php }elseif($vih){ ?>
                <p>A este paciente no se le puede realizar el trasplante porque el siguiente
                examen no se le ha realizado o ha dado positivo:
                <?php echo 'HIV'; ?></p>
                <?php }elseif($hepatitis){ ?>
                    <p>A este paciente no se le puede realizar el trasplante porque el siguiente
                    examen no se le ha realizado o ha dado positivo:
                    <?php echo 'TGP'; ?></p>
                    <?php } ?>
            <?php if($c && $d){ ?et>

```

<p>A este paciente no se le puede realizar el trasplante porque los siguientes exámenes no se le han realizado a sus donantes o han dado positivo:

```
<?php echo 'HIV, TGP'; ?></p>
```

```
<?php }elseif($c){ ?>
```

<p>A este paciente no se le puede realizar el trasplante porque el siguiente examen no se le ha realizado a sus donantes o ha dado positivo:

```
<?php echo 'HIV'; ?></p>
```

```
<?php }else{ ?>
```

<p>A este paciente no se le puede realizar el trasplante porque el siguiente examen no se le ha realizado a sus donantes o ha dado positivo:

```
<?php echo 'TGP'; ?></p>
```

```
<?php } ?>
```

```
// Mostrar formulario
```

```
<?php else {...}?>
```

Como se pudo apreciar en este epígrafe el sistema implementa una considerable cantidad de reglas, a cada una de las cuales se les aplicaron los criterios que determinan su ubicación dentro de la arquitectura MVC. La Tabla 2.1 muestra una comparativa entre la cantidad de RN implementadas en el sistema actual y en su versión anterior agrupándola según los tipos dados por la clasificación cercana a la implementación.

Tabla 2.1 Cantidad de reglas implementadas.

Tipos de reglas	Versión anterior	Versión actual
Modelo de datos	24	33
Relación	28	32
Restricción	15	24
Derivación	3	5
Flujo	6	8
<b>Total</b>	<b>76</b>	<b>102</b>

## **2.2. Generadores automáticos de reportes**

El sistema brinda herramientas y facilidades para satisfacer las necesidades de los usuarios. Entre ellas se encuentra la implementación de cuatro generadores automáticos de reportes para la extracción de los datos introducidos históricamente. Estos reportadores abarcan las cuatro áreas principales de datos; estas son: historias clínicas, análisis realizados, trasplantes de donantes vivos y trasplantes de donantes cadáver. Estos permiten realizar un completo filtrado por cualquier atributo aplicándole el operador deseado en cada caso. Es posible, además, escoger la columna que se desee visualizar en la salida resultante.

Una vez conformada una búsqueda por parte del usuario es posible guardar su formato (las condiciones usadas) para reutilizarlo. Esto facilita que los reportes más comúnmente utilizados no tengan que rescribirse. En epígrafes posteriores se amplía sobre cómo se implementa esta funcionalidad.

Finalmente, la salida resultante al aplicar el reporte sobre los datos puede ser guardada en diferentes formatos. Entre los formatos permitidos por el generador de reportes se encuentra el PDF (por sus siglas en inglés, *Portable Document Format*) que ofrece un excelente formato para la impresión y no requiere procesos anteriores de ajuste ni de maquetación. El otro formato que ofrece la aplicación para la exportación de los datos es el CSV (por sus siglas en inglés, *Comma-Separated Values*) el cual es un tipo de documento en formato abierto, sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas (o punto y coma, en donde la coma es el separador decimal) y las filas por saltos de línea, este formato es especialmente idóneo para poder, mediante programas más complejos como el Microsoft Office realizar operaciones más específicas con estos datos.

Sin dudas esta herramienta constituye un poderoso recurso con que cuenta el especialista, para realizar análisis de los datos almacenados en el sistema, que pueden abarcar años de información acumulada, la cual puede servir para ayudarlo a tomar decisiones.

### **2.2.1. Guardado de la búsqueda mediante la serialización**

Como se mencionaba anteriormente, los generadores de reportes entre las facilidades que brindan, ofrecen la posibilidad al usuario de guardar el formato de búsqueda, dándole un nombre identificativo para la utilización de este más adelante y así evitar pérdida de tiempo en tareas repetitivas. Esta tarea dado el dinamismo en el filtrado que ofrece la herramienta suele ser muy engorrosa, lo que nos lleva al uso de la *serialización*.

La serialización consiste en un proceso de codificación de un Objeto, en un medio de almacenamiento (como puede ser un archivo, o un buffer de memoria) con el fin de transmitirlo a través de una conexión en red, como una serie de bytes o en un formato humanamente más legible como XML o JSON, entre otros. La serie de bytes o el formato, pueden ser usados para crear un nuevo objeto que es idéntico en todo al original, incluido su estado interno. La serialización es un mecanismo ampliamente usado para transportar objetos a través de una red, para hacer persistente un objeto en un archivo o BD, para distribuir objetos idénticos a varias aplicaciones o localizaciones. Esta técnica tiene como ventaja la persistencia completa del objeto que es más conveniente que escribir sus propiedades en formato de texto.

En caso que el usuario mande a guardar el formato de búsqueda que utilizó en el filtrado, el sistema construye una matriz con las condiciones entradas y la serializa, guardando en la BD una copia idéntica de los datos que utilizó el usuario en el filtro. Luego cuando el usuario decide utilizar el formato guardado, el sistema utiliza estos datos serializados para la construcción de una nueva matriz de condiciones, que es un clon de la utilizada en un principio. Es fácil darse cuenta las ventajas y comodidades que ofrece esta técnica a la hora de guardar datos que tienen un formato variable, los cuales de otra forma serían muy difíciles de almacenar y reutilizar de forma eficiente.

### **2.3. Comparación del sistema actual con versiones anteriores**

En el actual SI se realizan diversas modificaciones respecto a la versión anterior. Estos cambios se reflejaron en todas las capas de la aplicación, re-implementándose las principales funcionalidades aprovechando las nuevas tecnologías web, además de haber

identificado e introducido nuevas RN y modificado otras, ya sea en la realización de su función o moviéndolas a otra capa de la aplicación. Las principales modificaciones son descritas en este epígrafe.

### **2.3.1. Modificaciones en el sistema de información**

1. Se cambia la interfaz visual para hacerla más amigable al usuario y lograr mayor compatibilidad con los navegadores más utilizados por los clientes Internet Explorer y Mozilla Firefox.
2. Se utilizaron las nuevas ventajas de la llegada del CSS3 y el HTML5 en la creación de los formularios para reducir su validación mediante *JavaScript*, el cual sobrecarga el código de la página y ralentiza la aplicación. En este caso se usaron los nuevos tipos de campos que traía la nueva versión del HTML5 tales como: *date*, *datetime*, *email*, *number*, *range*, etc. Además de utilizar las nuevas propiedades tales como “*placeholder*” para textos alternativos, propiedades de validación como “*required*” lo cual hace que el campo sea obligatorio y “*pattern*” que permite aplicarle una expresión regular a la validación del campo.
3. Se perfeccionaron las acciones realizadas en los controladores tales como adicionar, editar y listar, esta última fue re-implementada completamente mejorando facilidades tales como: cantidad de filas a mostrar, paginación, cuadro de búsqueda y ordenar columna; todo esto usando tecnología Ajax lo cual reduce considerablemente el tiempo de respuesta de la aplicación y le proporciona al cliente una interacción más amena.
4. Se permite la configuración del sitio con opciones tales como:
  - Nombre del sitio.
  - Correo electrónico.
  - Metadatos para los buscadores web.
  - Mensajes de presentación y pie de página.
  - Sacar el sitio de línea en caso de querer realizar mantenimiento.

5. Se añade la posibilidad de guardar la actividad del sitio, lo cual hace que el sistema cree un registro con todas las acciones realizadas por los usuarios, su IP y el cliente usado en la navegación.
6. Se permite compresión GZIP de las página, para en caso de las redes lentas el servidor las comprima al usuario y mejore el tiempo de respuesta.
7. Se usa cache en la presentación de información poco variable (Ej. ayuda, mensajes, menús, etc.) mejorando la velocidad de carga de las páginas.
8. Se mejora la administración de usuarios debido a que es una parte sensible de la aplicación, que necesita un estricto control y un correcto funcionamiento. En este sentido se creó un rol "Superadministrador" el cual sería un usuario con conocimientos de administración de sitios que tiene acceso ilimitado a todas las partes de este, que incluye: configuración global, registro de actividades del sistema y la información PHP y MySQL del servidor. También se permite desactivar usuarios temporalmente en caso de querer que este no tenga acceso en un momento determinado al sitio y notificar por correo al usuario de la creación de su nueva cuenta, proporcionándole por esta vía un nombre de usuario y contraseña con el que podrá acceder.
9. Se crea un asistente de mensajería interna y privada el cual permite a los usuarios enviarse mensajes facilitando la comunicación entre estos, además de permitir contactar al administrador en caso de tener sugerencias e interrogantes.
10. Se incluyeron mejoras en el ámbito de los nomencladores. Ejemplo de ello es para el caso de "Análisis" que conforman los "Tipos de estudios". Anteriormente el análisis solamente llevaba un nombre y el campo se presentaba como un simple campo de texto. Ahora es posible configurar esto de una forma más completa, siendo posible especificar qué tipo de campo se quiere usar, valor por defecto, tamaño del campo, texto antes del campo, texto después del campo y texto de ayuda. Cabe señalar, además, que al campo fecha se le incluye la hora para lograr mayor exactitud.
11. Se modificaron los formularios facilitando la entrada de los datos al usuario, esto se evidenció en los formularios de receptores, donantes y trasplantes, donde se sustituyeron las simples listas de selección, por ventanas alternativas más complejas

que muestran una tabla con los pacientes posibles a adicionar y sus datos relevantes para facilitar la selección de los mismos. En el caso del campo "Actividad laboral" de la historia clínica del paciente en donde la mayoría de las veces es una actividad laboral ya insertada anteriormente se implementó un campo *autocomplete* que se nos adelanta a lo que se está escribiendo, ofreciendo sugerencias basadas en experiencias anteriores. Esto representa una mejora al considerar la experiencia del usuario en la inserción de los datos.

12. Se añade una ayuda en la cual el usuario puede apoyarse en caso de que presente alguna duda al enfrentarse con el manejo de la aplicación, lo que representa un importante apoyo para usuarios inexpertos.
13. Se incluyeron generadores automáticos de reportes, los cuales fueron tratados en el epígrafe 2.2.
14. Se separan en el sistema, a petición de los usuarios, las consultas de los donantes vivos y los donantes cadáver con sus respectivos trasplantes. Estos deben tomarse como entes diferentes pues desde el punto de vista médico contienen diferencias significativas. Esto provoca modificaciones en el diseño de la BD y por su importancia es abordado en el capítulo III.

### **2.3.2. Modificaciones respecto a las reglas de negocio**

El sistema de nefrología implementado para la presente tesis no es un sistema común como cualquier aplicación de BD en donde las RN quedan embebidas en el código fuente. Este prioriza con la utilización del enfoque RN la fácil actualización a nuevas condiciones del negocio, al estar las reglas bien definidas, documentadas y ubicadas siguiendo los criterios dados por (Mena, 2011), los cuales determinan la ubicación de las RN dentro de un SI usando el patrón de diseño MVC. Por lo anteriormente descrito, podemos señalar la importancia de la correcta ubicación de las reglas en la arquitectura MVC para su mantenimiento, siendo esto un punto fundamental tenido en cuenta en la actual versión del sistema.

A continuación mostramos un resumen de reglas añadidas y modificadas:



Reglas añadidas

**R20:** Un posible donante vivo de un trasplante renal debe tener una edad entre 18 y 55 años.

Esta regla pertenece a la clasificación de *reglas del modelo de datos*. Su implementación se realiza en el modelo del donante (*clinical history of donor*) mediante la aplicación de una regla de validación para la edad, quedando de la siguiente forma:

```
var $validate = array(
    'edad' => array(
        'between' => array(
            'rule' => array('range', 18, 55),
            'message' => 'El donante debe tener una edad entre 18 y 55 años'
        )
    )
);
```

**R21:** Un método depurador puede ser asociado a la operación del receptor.

Esta regla pertenece a la clasificación de *reglas de relación*, para definirla se incluye la siguiente relación en el modelo operación del receptor:

```
var $belongsTo = array(
    'MethodDebug' => array(
        'className' => 'MethodDebug',
        'foreignKey' => 'method_debug_id',
    )
);
```

En el modelo método depurador la relación inversa se representaría como sigue:

```
var $hasMany = array(
    'OperationOfReceptor' => array(
        'className' => 'OperationOfReceptor',
        'foreignKey' => 'method_debug_id',
    )
);
```

Es significativo mencionar que el sistema sufrió un cambio en las condiciones del negocio, tema que se ampliará en el capítulo III, lo cual causó modificaciones en el

diseño de la BD. Esto se reflejó principalmente en las *reglas de relaciones*, algunas de las cuales que tuvieron que ser creadas para adaptar el sistema a las nuevas condiciones. La siguiente regla se muestra como ejemplo:

**R22:** Un posible donante puede ser asociado al menos a un trasplante de donante vivo.

En el modelo trasplante de donante vivo se inserta la siguiente relación:

```
var $belongsToMany = array(
    'ClinicalHistoryOfDonor' => array(
        'className' => 'ClinicalHistoryOfDonor',
        'foreignKey' => 'clinical_history_of_donor_id'
    ));
```

En el modelo donante (*clinical history of donor*) la relación inversa se representaría como sigue:

```
var $hasMany = array(
    'TransplantOfDonorAlife' => array(
        'className' => 'TransplantOfDonorAlife',
        'foreignKey' => 'clinical_history_of_donor_id',
        'dependent' => false,
    ));
```

### Reglas modificadas

Las siguientes reglas fueron modificadas, ya sea en su función o reubicándolas, siempre teniendo en cuenta los criterios que determinan la ubicación de las RN dentro del SI.

**R23:** La fecha del análisis realizado a un paciente debe considerar tanto la fecha como la hora.

Este cambio en la regla se debe a que el análisis del paciente no debía limitarse solo a la fecha en que este fue realizado, también debía incluir la hora para una mayor precisión, pues en algunos casos la exactitud es requerida.

### **VERSIÓN ANTERIOR**

```
var $validate = array(
    'fecha' => array(
```

```

        'date' => array(
        'rule' => 'date',
        'message' => 'Ingrese una fecha válida'
        )
    ));

```

### VERSIÓN ACTUAL

```

var $validate = array(
    'date' => array(
        'rule' => 'datetime',
        'message' => 'Ingrese una fecha y hora válida.'
    )
);

```

La siguiente regla anteriormente descrita en el epígrafe 2.1 pertenece a la clasificación de reglas de restricción, su implementación se realizaba en el controlador “*análisis realizados*”. Aplicando los criterios que determinan la ubicación de las RN dentro del SI se determinó que este tipo de regla se ubica en el modelo, quedando corregido en la actual versión.

**R24:** Un paciente no puede tener dos resultados de un mismo análisis en la misma fecha y hora.

### VERSIÓN ANTERIOR

La comprobación de esta regla se realizaba en el controlador de análisis realizados (*analysy realizes*) a continuación se muestra una fracción de código de su implementación.

```

foreach($results as $key => $value){
    //si este analisis tiene resultados
    if ($value != ""){
        /*Se busca un analisis_realizado con la fecha que se pasa
        que sea de este analisis y que pertenezca a este paciente*/
        $analysyRealize = $this->AnalisyRealize->find('first', array(
            'conditions' => array(

```

```

        'AnalisyRealize.fecha'=> $this->data['AnalisyRealize']['fecha'],
        'AnalisyRealize.clinical_history_id'=>$this-
>data['AnalisyRealize']['clinical_history_id'],
        'AnalisyRealize.analisy_id' => $key
    )
));
//Si existe un analisis_realizado con esta fecha
if($AnalisyRealize){
    $flag = false;
    $this->AnalisyRealize->invalidate('fecha', 'Ya existe un análisis:
'. $AnalisyRealize['key'].' en esta fecha. ');
    break;
}

```

### VERSIÓN ACTUAL

El siguiente código muestra cómo queda implementada la regla una vez ubicada en el modelo mediante una función de validación.

```

function beforesave(){
    $AnalisyRealize = $this->AnalisyRealize->find('first', array(
        'conditions' => array(
            'AnalisyRealize.fecha'=>$this->data['AnalisyRealize']['fecha'],
            'AnalisyRealize.clinical_history_id'=>$this-
>data['AnalisyRealize']['clinical_history_id'],
            'AnalisyRealize.analisy_id' => $key
        )
    ));
    //Si existe un analisis_realizado con esta fecha
    return empty($AnalisyRealize);
}

```

La reubicación de esta regla al modelo facilita en gran medida su actualización en futuras versiones. Esto se debe a que se encuentra aislada en un método y los cambios que se le realicen a esta no alteran el flujo de acciones que se realizan en el controlador. En la Tabla 2.2 se presenta la cantidad de reglas añadidas y modificadas que presenta la nueva versión del sistema.

Tabla 2.2 Reglas modificadas y añadidas.

Tipos de reglas	Reglas Modificadas	Reglas Añadidas
Modelo de datos	13	9
Relación	9	4
Restricción	6	9
Derivación	1	2
Flujo	3	2
<b>Total</b>	<b>32</b>	<b>26</b>

## 2.4. Conclusiones parciales

Se aplicaron a todas las reglas los criterios que determinan la ubicación de las RN dentro de los SI para lograr la fácil actualización a las nuevas condiciones del negocio, utilizando el patrón MVC. De esta forma se soluciona el problema arquitectónico que representa el manejo de las RN dentro de los SI, al ubicar las RN en su capa correspondiente de acuerdo a su clasificación.

Se expusieron algunas de las RN implementadas en el sistema organizándolas por la clasificación cercana a la implementación dada por Soliveres (1997), complementando esto con ejemplos de códigos usados en su implementación. Además, se mostraron diferencias del sistema actual con su versión anterior evidenciando su adaptabilidad mediante la presentación de ejemplos de reglas añadidas y modificadas, finalizando con una tabla de resultados.

Se presentaron las nuevas características y funcionalidades de la versión propuesta, ampliando con más detalles los generadores automáticos de reportes para la extracción y exportación de los datos.

## **CAPÍTULO III. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE NEFROLOGÍA**

En este capítulo se presentan aspectos relacionados con el diseño y la implementación del sistema que recibe el nombre de SINEX (acrónimo de Sistema de Nefrología). Este sistema por sus características muy peculiares se divide en dos partes; un área de publicación de contenidos la cual se referencia con el nombre de *Front-end* y la segunda área dedicada al control de los datos de los pacientes y los trasplantes la cual recibe el nombre de *Back-end*.

Se realiza un estudio a la BD existente proponiendo un cambio en su estructura debido a nuevas condiciones en el negocio, se modelan los casos de uso del sistema y se recrea la utilización de la interfaz gráfica, obteniéndose finalmente un manual de usuario para la nueva versión de la aplicación.

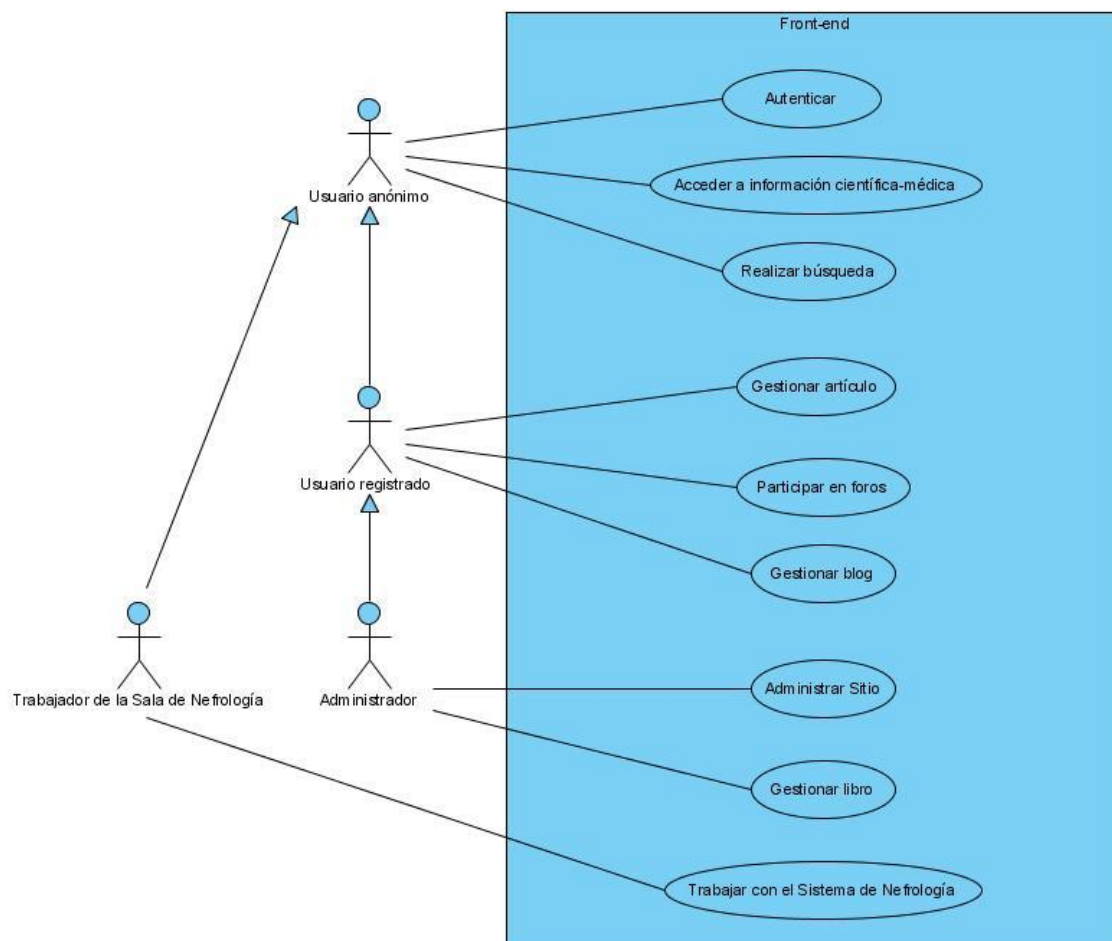
### **3.1. Diseño del sistema**

Los principales resultados del análisis y diseño del sistema, fueron recogidos en diagramas UML (LUM o UML, por sus siglas en inglés, *Unified Modeling Language*). Este es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Su uso facilita la comprensión del software que se está desarrollando y la comunicación entre el desarrollador y el cliente. Para construir los diagramas UML se utilizó el *Visual Paradigm for UML Enterprise Edition 6.0*.

#### **3.1.1. Modelo de casos de uso del sistema**

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso, representando gráficamente uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para lograr un objetivo específico. Esto resulta una técnica muy efectiva a la hora de obtener los requerimientos de un nuevo sistema o la actualización de un software. Un modelo de casos de uso muestra, por tanto, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relacionan con su entorno (usuarios u otras aplicaciones).

Cuando se accede al sitio primeramente se ingresa al *Front-end*, el cual se presenta en forma de portal dinámico para la interacción entre los visitantes y los contenidos publicados por el personal del AN, tales como: artículos, imágenes, foros, blog, libros, etc. Como se muestra en el caso de uso de la Figura 3.1, a esta parte de la aplicación se puede acceder como usuario anónimo, registrado o administrador y en base a ello el sistema determina a que actividades tiene acceso.



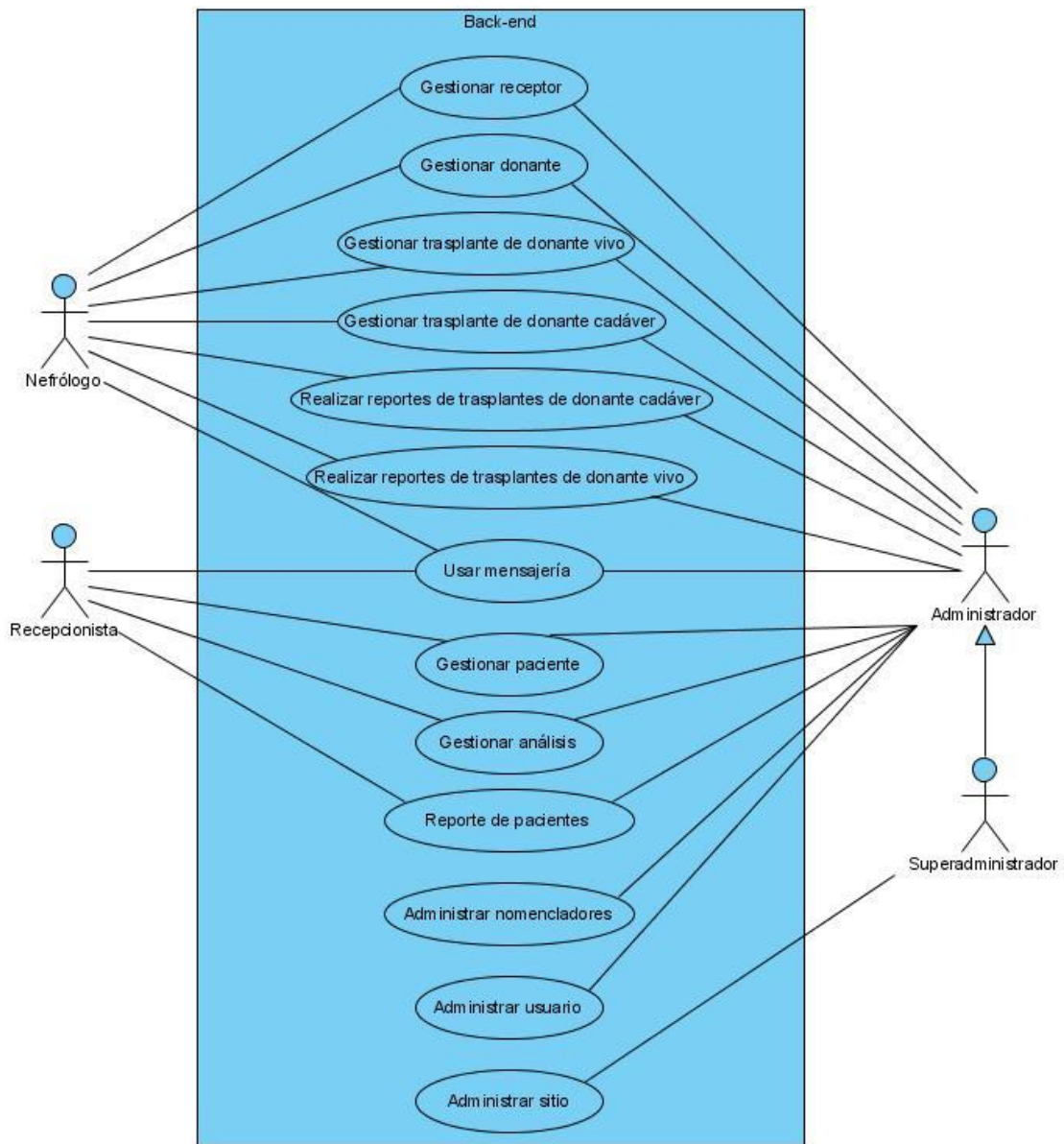
**Figura 3.1** Diagramas de casos de uso del *Front-end*.



Tabla 3.1 Casos de uso del *Front-end*.

Caso de Uso	Descripción
<b>Autenticar</b>	El usuario necesita acceder al sistema e inserta su usuario y contraseña.
<b>Acceder a información científica-médica</b>	El usuario puede ver publicaciones realizadas tales como: artículos, foros, blogs, libros, etc.
<b>Realizar búsqueda</b>	Insertando un término en el cuadro de búsqueda el sistema muestra los artículos en los que este aparece.
<b>Gestionar artículo</b>	Añadir, modificar o eliminar artículos del sistema.
<b>Participar en foros</b>	El usuario puede participar en los foros existentes.
<b>Gestionar blog</b>	Añadir, modificar o eliminar información del blog propio.
<b>Administrar sitio</b>	Modificar opciones de configuración del sitio, gestión de usuarios, etc.
<b>Gestionar libro</b>	Añadir, modificar o eliminar libros del sistema.
<b>Trabajar con el sistema de nefrología</b>	Acceder a la parte de la aplicación dedicada al control de los datos del AN.

Una vez en el *Front-end* si el usuario pertenece al grupo de trabajo de la sala de nefrología y cuenta con la debida autorización puede acceder al *Back-end*. Esta parte del sistema necesita de la autenticación del usuario y en base a su rol se determina que acciones puede o no realizar sobre los datos almacenados en la aplicación. En la Figura 3.2 se muestra el diagrama de casos de uso para el *Back-end* definiendo los tipos de usuarios y las acciones a las que estos tienen acceso.



**Figura 3.2** Diagrama de casos de uso del *Back-end*.

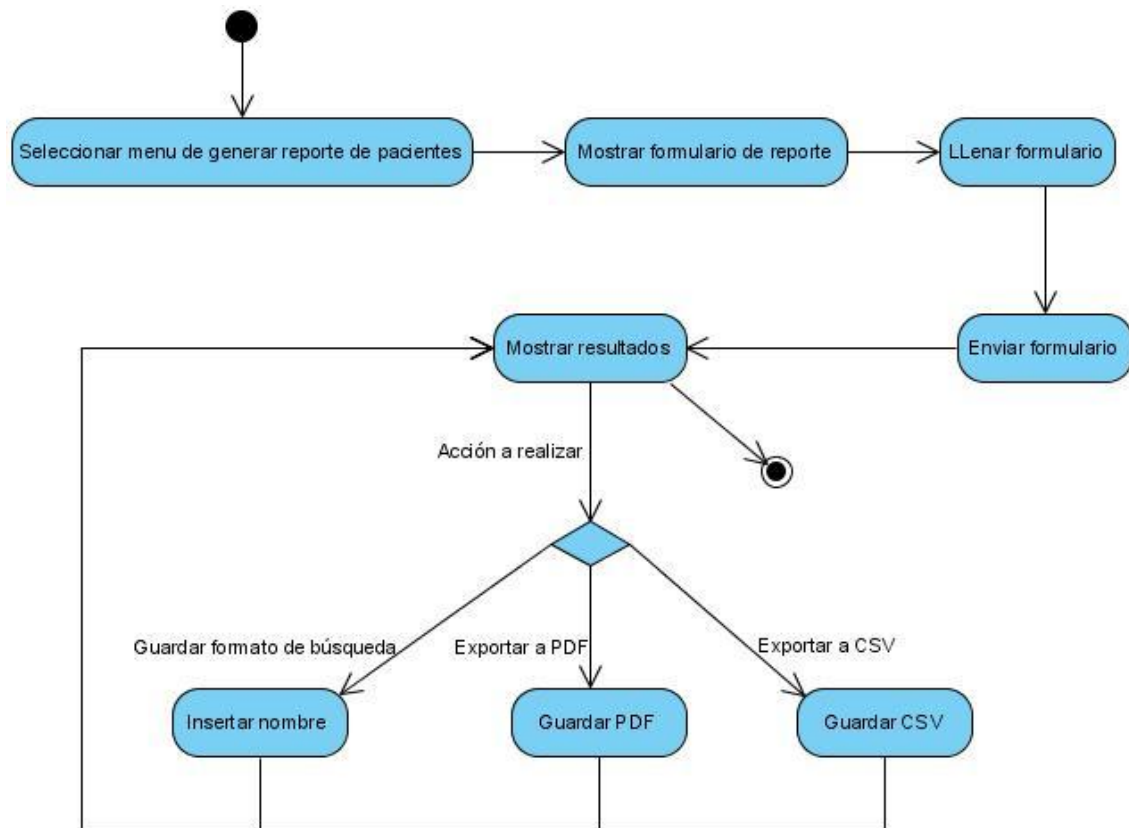
Tabla 3.2 Caso de uso del *Back-end*.

Caso de Uso	Descripción
<b>Gestionar receptor</b>	Añadir, modificar o eliminar receptores en el sistema.
<b>Gestionar donante</b>	Añadir, modificar o eliminar donantes en el sistema.
<b>Gestionar trasplante de donante vivo</b>	Añadir, modificar o eliminar trasplantes de donantes vivos.
<b>Gestionar trasplante de donante cadáver</b>	Añadir, modificar o eliminar trasplantes de donantes cadáver.
<b>Realizar reporte de trasplantes de donante vivo</b>	Generar reportes de los datos de los trasplantes de donantes vivos.
<b>Realizar reporte de trasplantes de donante cadáver</b>	Generar reportes de los datos de los trasplantes de donantes cadáver.
<b>Gestionar paciente</b>	Añadir, modificar o eliminar pacientes en el sistema.
<b>Gestionar análisis</b>	Añadir, modificar o eliminar los datos de los análisis realizados a los pacientes.
<b>Reporte de pacientes</b>	Generar reportes de los datos generales y los análisis realizados a los pacientes.
<b>Usar mensajería</b>	Enviar mensajes a otros usuarios del sistema; leer y borrar mensajes recibidos.
<b>Administrar nomencladores</b>	Incluye añadir, modificar y eliminar términos de los nomencladores del sistema los cuales suman 17 en total.
<b>Administrar usuario</b>	Gestiona los diferentes usuarios que pertenecen al sistema; puede insertar alguno nuevo y editar o eliminar otro usuario con un rol menor que el propio.
<b>Administrar sitio</b>	Cambiar configuración global del sistema.

### 3.1.2. Diagrama de actividad

Los diagramas de actividad se utilizan para modelar los aspectos dinámicos de un sistema. Estos muestran el flujo de control general y son usados para modelar una secuencia de acciones y condiciones tomadas dentro de un proceso.

En la Figura 3.3 se muestra el diagrama de actividad para el caso de uso "Reporte de pacientes" del *Back-end* el cual modela los pasos en la creación de un reporte con los datos generales de los pacientes.



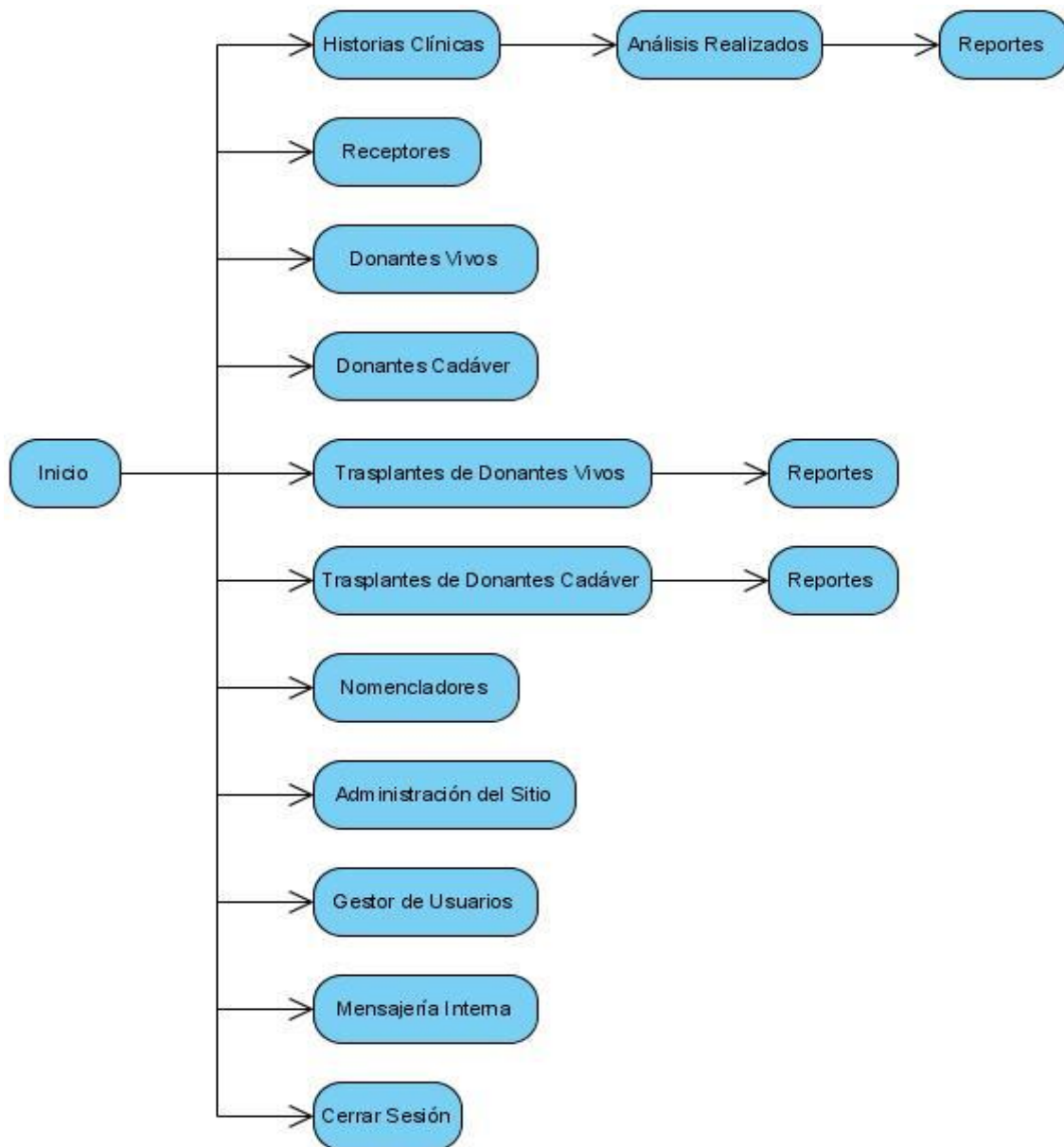
**Figura 3.3** Diagrama de actividad: generar reporte de pacientes.

### 3.1.3. Diagrama de navegación

Los mapas de navegación proporcionan una representación esquemática de la estructura del hipertexto, indicando los principales conceptos incluidos en el espacio de la información y las interrelaciones que existen entre ellos. Un mapa es, por ejemplo, una representación completa (o resumida) del sitio web para orientar al usuario durante el recorrido o facilitarle el acceso directo al lugar que le interese. Refleja la estructura de la

web por medio de enlaces a los nodos principales, y estos también pueden desarrollarse para mostrar sub-nodos. Se puede concluir entonces que el diagrama de navegación es el encargado de modelar como se efectúa la navegación por el sitio.

La Figura 3.4 muestra el diagrama de navegación del *Back-end* para el usuario "superadministrador". Este usuario tiene todos los privilegios, por lo que puede acceder a todas las áreas, pudiendo modificar tanto las opciones del sistema como los datos que guarda.

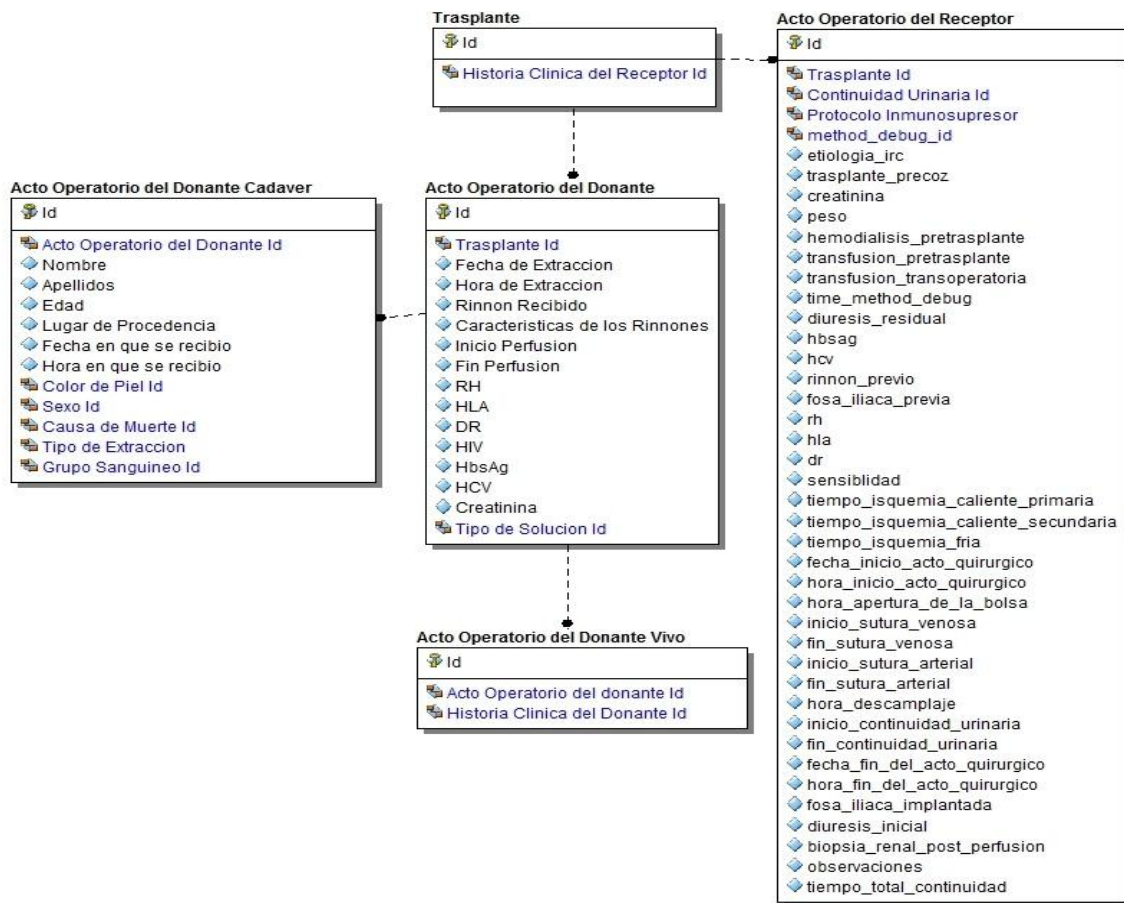


**Figura 3.4** Diagrama de navegación para el *Back-end* del usuario "superadministrador".

### 3.1.4. Cambio en el diseño de la base de datos

Dado que existía con anterioridad una BD implementada para la antigua versión del sistema, este epígrafe se enfoca en los cambios realizados para la adaptación a las nuevas condiciones del negocio.

En la implementación anterior de la BD se consideraba el trasplante como una entidad global y abarcadora. Como se observa en la Figura 3.5 la entidad trasplante se relacionaba con la operación del donante, de la cual heredaban la operación de donante vivo y cadáver.



**Figura 3.5** Diseño anterior del trasplante.

Desde el punto de vista de interfaz de usuario existía una tabla general de trasplantes en la cual estaban mezclados los trasplantes de donantes vivos y cadáver. Una vez puesto el sistema en etapa de prueba los especialistas en nefrología se percataron que esta condición del negocio estaba errónea y causaba confusión, pues en el ámbito de la



medicina los trasplantes de donantes vivos y cadáver guardan diferencias significativas. Por tal motivo se determinó que la diferencia no se encuentra en el tipo de donante sino en el trasplante en sí, lo que implica que el manejo de esta información se debe tomar por separado.

Esta nueva forma de comprender la instancia trasplante fue corregida en la nueva versión del sistema, quedando separado por completo el trasplante del donante vivo del donante cadáver. Esto ofrece como ventajas un manejo más claro de la información por parte del usuario, reduce el nivel de relaciones en las tablas de la BD lo que facilita las consultas y desde el punto de vista técnico se ajusta más a la arquitectura MVC, al dedicarse un modelo y un controlador a cada uno por separado. En la Figura 3.6 se muestra cómo queda implementada la instancia trasplante y las tablas relacionadas. En el Anexo1 se muestra el diagrama Entidad Relación (E/R) para TR.

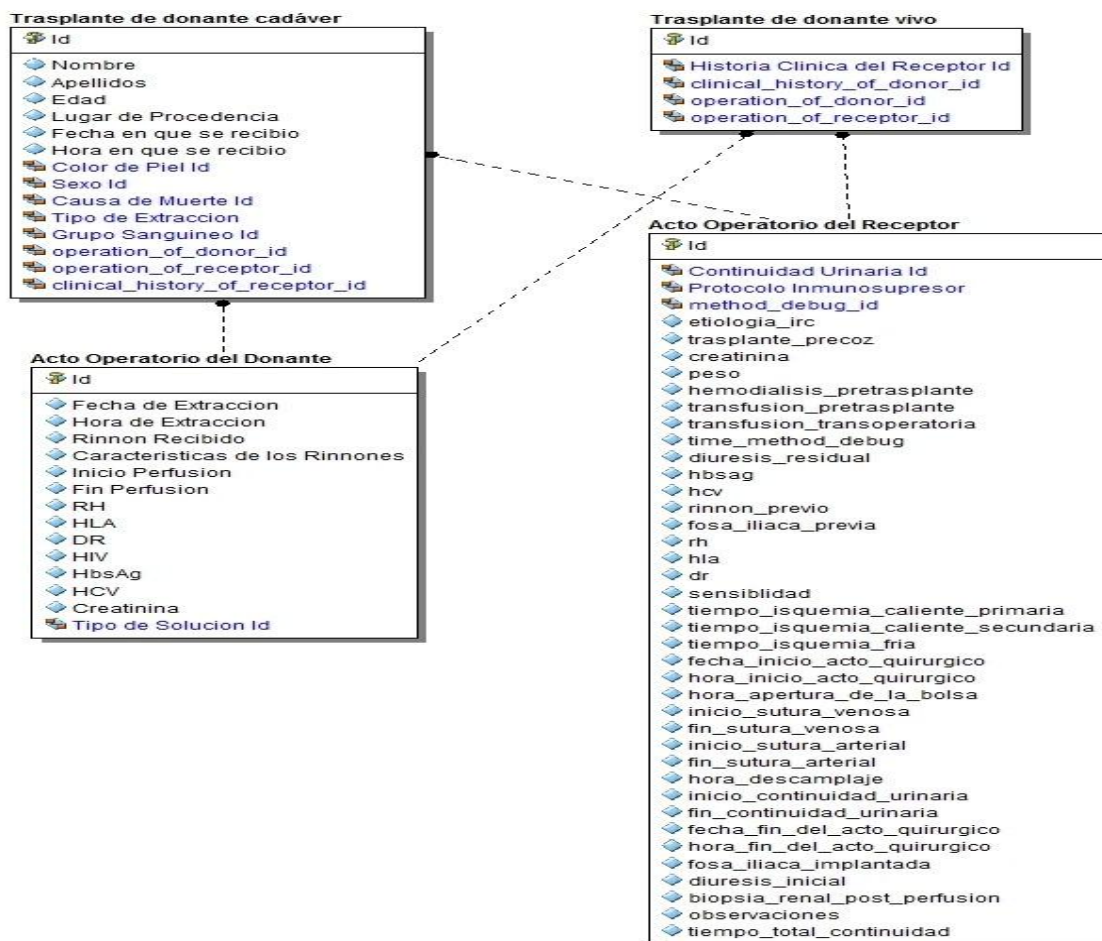


Figura 3.6 Diseño actual.

## **3.2. Estructura de la aplicación**

SINEX está ideado para facilitar el trabajo de los médicos del Hospital Universitario “Arnaldo Milián Castro”, de Villa Clara. Este permite mediante una interfaz web amigable publicar contenidos y servir como portal interactivo a los usuarios. También y como rol principal está el de gestionar toda la información referente al AN que incluye el control de las historias clínicas de todos los pacientes, los análisis realizados a estos, el manejo de la información de los receptores y los donantes y tramitar los trasplantes de donantes vivos y cadáver realizados en el centro. Todo esto con el objetivo de que los especialistas de esta área encuentren un lugar en la red donde comunicarse y gestionar la información que se maneja en el AN. Al sentarse en la máquina el usuario podrá acceder a este sistema mediante la web. En los posteriores epígrafes se hace un resumen visual de la navegación por el sitio.

### **3.2.1. Front-end**

#### **3.2.1.1. Página principal**

Cuando se inicia la aplicación esta se presenta en forma de portal informativo (ver Figura 3.7) por la cual el usuario común puede navegar e informarse de las noticias y contenidos publicados en este. En caso de tener acceso como usuario registrado o administrador se puede publicar contenidos o manejar el sitio según su rol (ver epígrafe 3.1.1 para más detalles sobre las acciones permitidas).





**Figura 3.7** Portal informativo.

Esta página principal del portal informativo muestra las últimas publicaciones realizadas por el equipo de especialistas del AN, permitiendo acceder de forma fácil a la información más actualizada. Además se visualizan los principales menús del sitio desde los cuales se accede a las diferentes partes de la aplicación.

### 3.2.1.2. Tipos de contenidos

Como se aprecia en la Figura 3.8 el sistema en su área informativa permite a los usuarios administradores y registrados, la publicación de diversos tipos de contenidos tales como: artículos, imágenes, foros, blog, libros, encuestas, etc.



**Figura 3.8** Tipos de contenidos.

## 3.2.2. Back-end

### 3.2.2.1. Inicio de sesión

Una vez que se accede al área destinada al control de pacientes se presenta una página de autenticación, donde se debe introducir su nombre de usuario y su contraseña (ver Figura 3.9 **Inicio de sesión**). Esta es una zona muy restringida y exclusiva para unos pocos usuarios, por lo que estos no están relacionados con los del área informativa. Este usuario y contraseña permite que la aplicación establezca un control sobre quién se ha registrado y qué privilegios tiene, o sea, a qué partes de la aplicación tiene acceso. Es importante destacar que cada usuario tiene un rol, el cual define la navegación por la aplicación.

**milian**  
HOSPITAL UNIVERSITARIO  
Arnaldo Milian Castro

Nombre de Usuario\*

Contraseña\*

**Acceso**

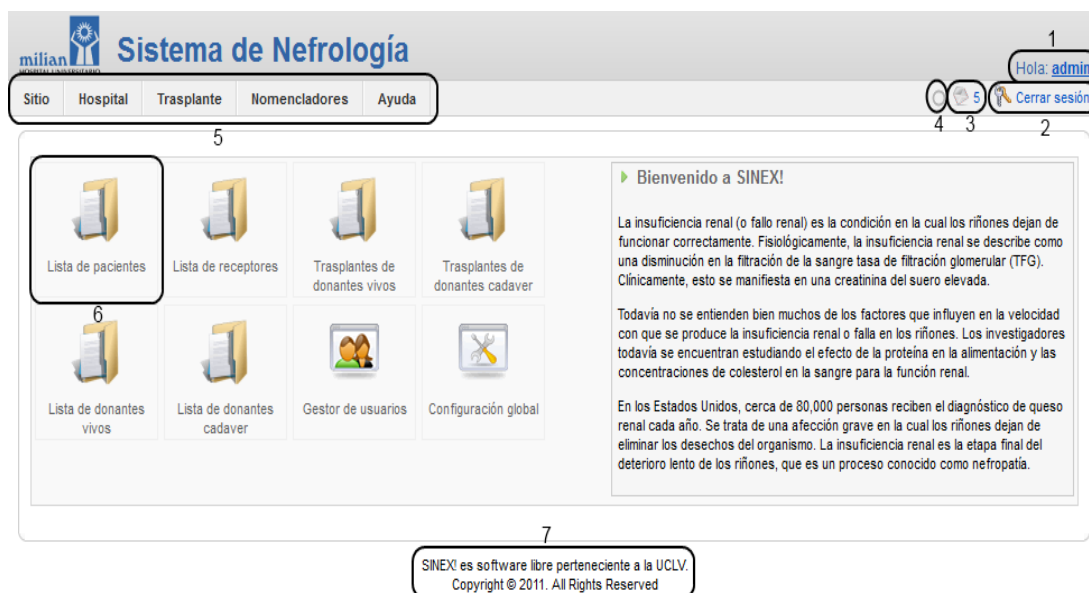
**Aviso:** Está accediendo a una parte privada del sitio, entre solo si cuenta con la autorización.

**Figura 3.9** Inicio de sesión.

Luego de insertar el usuario, su contraseña y pulsar el botón **Acceso** se accede a la aplicación en sí.

### 3.2.2.2. Página principal

Habiendo accedido, aparece la página principal del área de gestión de pacientes (ver Figura 3.10). En esta se muestra en forma de menú y enlaces las principales áreas a la que el usuario se puede dirigir. Para realizar una mejor explicación de la misma a continuación se muestra una leyenda de sus partes.

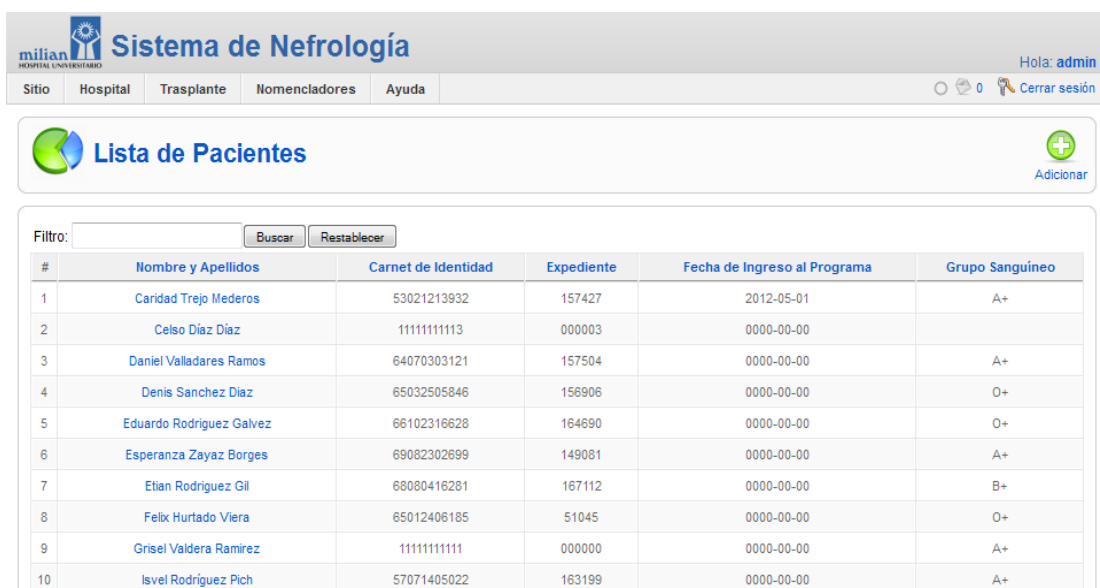


**Figura 3.10** Página principal.

1. Brinda información del usuario registrado.
2. Cierra la sesión activa y sale del sistema.
3. Informa sobre la cantidad de mensajes sin leer que tiene en la Mensajería Privada, si da clic sobre él muestra el buzón de entrada.
4. Icono de actividad del sistema, se activa si la aplicación está ejecutando una acción en segundo plano.
5. Menú desplegable principal.
6. Iconos de acceso directo los cuales proveen un acceso visible y rápido a distintas partes de la aplicación, también se puede utilizar el Menú Principal.
7. Mensaje de pie de página que aparece en la parte inferior de todas las páginas del sitio, se usa para informar sobre derecho de autor.

### **3.2.2.3. Manejo de historias clínicas**

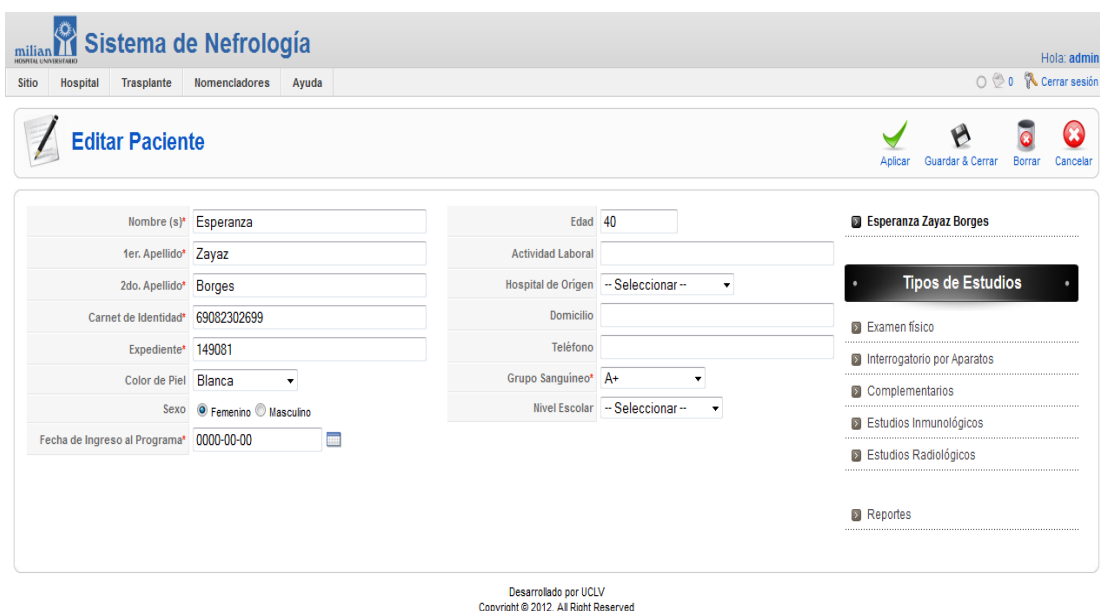
Una parte importante del sistema y la cual es el centro de toda la información es la entidad Historia Clínica, esta cuenta con una serie de datos generales sobre un paciente y que normalmente están en cualquier historia clínica de atención secundaria. Con esta entidad base están relacionadas todas las demás que también contienen información general acerca del paciente; lo cual permite que los sub-modelos se simplifiquen considerablemente, pues tan sólo contendrán la información específica de ellos. Esta parte de la aplicación puede ser accedida por los usuarios administradores y los recepcionistas. En la Figura 3.11 se muestra una tabla de ejemplo con datos reales de pacientes trasplantados en el año 2010.



#	Nombre y Apellidos	Carnet de Identidad	Expediente	Fecha de Ingreso al Programa	Grupo Sanguíneo
1	Caridad Trejo Mederos	53021213932	157427	2012-05-01	A+
2	Celso Díaz Díaz	11111111113	000003	0000-00-00	
3	Daniel Valladares Ramos	64070303121	157504	0000-00-00	A+
4	Denis Sanchez Diaz	65032505846	156906	0000-00-00	O+
5	Eduardo Rodriguez Galvez	66102316628	164690	0000-00-00	O+
6	Esperanza Zayaz Borges	69082302699	149081	0000-00-00	A+
7	Etian Rodriguez Gil	68080416281	167112	0000-00-00	B+
8	Felix Hurtado Viera	65012406185	51045	0000-00-00	O+
9	Grisel Valdera Ramirez	11111111111	000000	0000-00-00	A+
10	Isvel Rodriguez Pich	57071405022	163199	0000-00-00	A+

**Figura 3.11** Lista de pacientes.

En la Figura 3.12 se muestra el formulario de edición de los datos generales de un paciente real, al cual se le aplicó un trasplante renal en el año 2010. Como se puede ver el formulario cuenta con un menú lateral a su derecha el cual muestra los tipos de estudios. Este formulario de edición como la mayoría cuenta con un panel superior derecho con las principales funciones de edición como son: aplicar cambios, guardar y cerrar, borrar, eliminar y cancelar.



**Editar Paciente**

Aplicar Guardar & Cerrar Borrar Cancelar

Nombre (s)*	Esperanza	Edad	40
1er. Apellido*	Zayaz	Actividad Laboral	
2do. Apellido*	Borges	Hospital de Origen	-- Seleccionar --
Carnet de Identidad*	69082302699	Domicilio	
Expediente*	149081	Teléfono	
Color de Piel	Blanca	Grupo Sanguíneo*	A+
Sexo	<input checked="" type="radio"/> Femenino <input type="radio"/> Masculino	Nivel Escolar	-- Seleccionar --
Fecha de Ingreso al Programa*	0000-00-00		

**Esperanza Zayaz Borges**

**Tipos de Estudios**

- ☐ Examen físico
- ☐ Interrogatorio por Aparatos
- ☐ Complementarios
- ☐ Estudios Inmunológicos
- ☐ Estudios Radiológicos
- ☐ Reportes

Desarrollado por UCLV  
Copyright © 2012. All Right Reserved

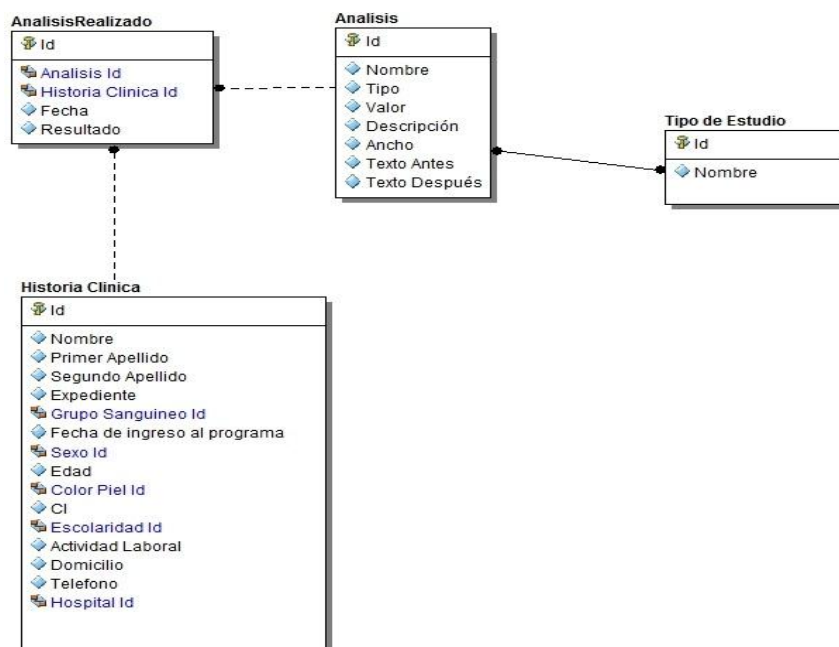
**Figura 3.12** Editar Paciente.

### 3.2.2.4. Tipos de estudios

Cuando un paciente entra en atención secundaria se le realizan una serie de tipos de estudios, que básicamente están compuestos por análisis, pero estos tipos de estudios cambian frecuentemente y mucho más en el AN, por tanto, es necesario que la aplicación sea capaz de adaptarse a estos cambios, sin llevar consigo reprogramación, ni perder los resultados de los análisis a pesar de que se modifiquen los tipos de estudios.

Esto se resuelve mediante el diseño que se observa en la Figura 3.13 el cual soluciona los requisitos planteados anteriormente, ya que en la tabla "AnálisisRealizado" se guardan todos los análisis realizados a un paciente, independientemente del tipo de estudio a que pertenecen, por lo que se pueden modificar los tipos de estudios sin que esto conlleve pérdida de información. De esta manera la BD se encuentra preparada para adaptarse a los cambios que se efectúen en los tipos de estudios que se realizan a los pacientes.

Como se aprecia en la Figura 3.13 la entidad "análisis" incluye campos para su configuración visual lo que permite especificar opciones tales como: qué tipo de campo HTML se desea utilizar para la inserción del valor del análisis, valor por defecto, ancho, texto antes y después.



**Figura 3.13** Submodelo de tipos de estudios.

A continuación se muestra como el usuario puede conformar sus propios tipos de estudios a partir de los análisis que ellos mismos insertan en el sistema. La solución para ello se observa en la Figura 3.14 donde el usuario puede escoger los análisis que conforman tipo de estudio que está editando.

**Figura 3.14** Editar tipo de estudio.

Cuando se definen los tipos de estudio, y se le van a agregar a un paciente el formulario se construye dinámicamente a partir de los análisis que lo conforman y de las configuraciones de estos, esto se puede observar en la Figura 3.15.

**Figura 3.15** Adicionar estudio: examen físico.

En caso de que se inserte algún análisis sin resultado este no será guardado evitando valores nulos.

### 3.2.2.5. Generadores automáticos de reportes

Una de las nuevas funcionalidades más significativas, la cual fue abordada en el epígrafe 2.2, fue la creación de reportes de todos los datos almacenados. Esto se logra implementando un reportador dinámico en cada área, abarcando así todos los datos de la BD. Estos componen un total de 4; uno para los análisis, uno para los datos de los pacientes, uno para los trasplantes de donantes vivos y otro para los trasplantes de donantes cadáver. De esta forma el especialista puede filtrar los datos guardados en el sistema de una forma sencilla.

En la Figura 3.16 se puede observar el formulario de reportes para los análisis realizados a un paciente, este ofrece la ventaja de realizar reportes de un paciente por sus análisis pudiendo especificar un rango de fechas.

The screenshot displays the 'Sistema de Nefrología' interface. At the top, there's a navigation bar with links: 'Sitio', 'Hospital', 'Trasplante', 'Nomencladores', and 'Ayuda'. A user greeting 'Hola: admin' and a 'Cerrar sesión' link are on the right. Below this is a search bar titled 'Buscar análisis de Esperanza Zayaz Borges'. The main form area contains a dropdown menu for 'Análisis' (currently set to '-- Seleccionar --') and a date range selector with 'Desde' and 'Hasta' fields (both showing 'AAAA-MM-DD'). To the right of the form is a sidebar titled 'Tipos de Estudios' with a list of study types: 'Examen Físico', 'Interrogatorio por Aparatos', 'Complementarios', 'Estudios Inmunológicos', 'Estudios Radiológicos', and 'Reportes'. The footer indicates 'Desarrollado por UCLV' and 'Copyright © 2012. All Right Reserved'.

**Figura 3.16** Reporte por análisis.

También existe la posibilidad de crear reportes sobre los datos de las historias clínicas de los pacientes. Como se puede apreciar en el formulario mostrado en la Figura 3.17, este



generador automático de reportes permite realizar búsquedas que pueden llegar a ser bastante complejas de forma muy fácil e intuitiva para el usuario.

**Sistema de Nefrología**

Hola: admin [Cerrar sesión](#)

[Sitio](#) [Hospital](#) [Trasplante](#) [Nomencladores](#) [Ayuda](#)

**Generar Reporte de los Pacientes** [Filtrar](#) [Restablecer](#)

Campo	Operador	Valor	Mostrar
Nombre	Igual que ▼	<input type="text"/>	<input checked="" type="checkbox"/>
Primer Apellido	Igual que ▼	<input type="text"/>	<input checked="" type="checkbox"/>
Segundo Apellido	Igual que ▼	<input type="text"/>	<input checked="" type="checkbox"/>
Carnet de Identidad	Igual que ▼	<input type="text"/>	<input checked="" type="checkbox"/>
Expediente	Igual que ▼	<input type="text"/>	<input checked="" type="checkbox"/>
Color de Pie	Igual que ▼	-- Seleccionar -- ▼	<input type="checkbox"/>
Sexo	Igual que ▼	-- Seleccionar -- ▼	<input type="checkbox"/>
Fecha de Ingreso al Programa	Igual que ▼	<input type="text" value=""/>	<input checked="" type="checkbox"/>
Edad	Igual que ▼	<input type="text"/>	<input type="checkbox"/>
Actividad Labora	Igual que ▼	<input type="text"/>	<input type="checkbox"/>
Hospital de Origen	Igual que ▼	-- Seleccionar -- ▼	<input type="checkbox"/>
Domicilio	Igual que ▼	<input type="text"/>	<input type="checkbox"/>
Teléfono	Igual que ▼	<input type="text"/>	<input type="checkbox"/>
Grupo Sanguíneo	Igual que ▼	-- Seleccionar -- ▼	<input checked="" type="checkbox"/>
Nivel Escolar	Igual que ▼	-- Seleccionar -- ▼	<input type="checkbox"/>

Desarrollado por UCLV  
Copyright © 2012. All Right Reserved

**Figura 3.17** Formulario de reporte de historias clínicas.

Una vez realizada la búsqueda el sistema muestra la salida en forma de tabla como se aprecia en la Figura 3.18, en la cual se muestra un ejemplo de reporte realizando un filtrado por los pacientes que pertenecen al grupo sanguíneo A+. Esta salida puede ser exportada en los formatos permitidos (CSV y PDF) y los parámetros del filtrado pueden ser guardados para una posterior reutilización.



Nombre y Apellidos	Carnet de Identidad	Expediente	Sexo	Fecha de Ingreso al Programa	Grupo Sanguíneo
Daniel Valladares Ramos	64070303121	157504	Masculino	0000-00-00	A+
Caridad Trejo Mederos	53021213932	157427	Femenino	0000-00-00	A+
Grisel Valdera Ramirez	11111111111	000000	Femenino	0000-00-00	A+
Maximiliano Campanioni Masdeu	11111111112	000002	Masculino	0000-00-00	A+
Luis Manuel Fernández Rodríguez	11111111114	161397	Masculino	0000-00-00	A+
Esperanza Zayaz Borges	69082302699	149081	Femenino	0000-00-00	A+
Juan Cordova Hurtado	61051315146	162666	Masculino	0000-00-00	A+
Omar Valido Macias	67060521342	162664	Masculino	0000-00-00	A+
Justo Morales Glez	11111111115	163207	Masculino	0000-00-00	A+
Isvel Rodríguez Pich	57071405022	163199	Masculino	0000-00-00	A+
Santiago Chavez Medina	11111111116	000004	Masculino	0000-00-00	A+
Nemesio Hernández Olivera	34070504600	163390	Masculino	0000-00-00	A+
Mirta Peralta Lopez	11111111117	165126	Femenino	0000-00-00	A+

Desarrollado por UCLV  
Copyright © 2012. All Right Reserved

**Figura 3.18** Salida de los datos del reportador.

### 3.2.2.6. Trabajo con trasplantes

Los trasplantes son las entidades que más datos almacenan en el sistema, esto se debe a que durante la intervención quirúrgica en la que intervienen un donante (puede ser vivo o cadáver) y un receptor los especialistas necesitan guardar una gran cantidad de datos relacionados con las operaciones realizadas a estos. Esta gran cantidad de datos, como es de suponer, necesita un formulario extenso pero que en la medida de lo posible se muestre organizado, para facilitar la inserción de los datos por parte del usuario. La Figura 3.19 muestra el formulario de los datos de un trasplante de donante vivo.

**Sistema de Nefrología**

Hola: admin

Sitio Hospital Trasplante Nomencladores Ayuda

**Adicionar Trasplante de Donante Vivo**

Guardar Guardar & Nuevo Cancelar

Receptor\*  Buscar...

Donante\*  Buscar...

**Datos de Donante**

Fecha de Extracción

Riñón Recibido ☐ Izquierdo ☐ Derecho

Hora de Extracción 2 47 am

Características del Riñón

**Quirugia de Banco**

Inicio de la Perfusión 2 47 am

RH ☐ Negativo ☐ Positivo

DR

HbsAg ☐ Negativo ☐ Positivo

Creatinina  umol/l

Fin de la Perfusión 2 47 am

HLA

HIV ☐ Negativo ☐ Positivo

HCV ☐ Negativo ☐ Positivo

Tipo de Solución --Seleccionar--

**Datos del Receptor**

Peso  Kg

Diuresis Residual

Trasplante Precoz ☐ No ☐ S

Transfusión Pre Tx ☐ No ☐ S

HCV ☐ Negativo ☐ Positivo

Método Depurador --Seleccionar--

RH

DR

Etiología IRC

Creatinina

Hemodialis Pretrasplante ☐ No ☐ S

Transfusión Transoperatoria ☐ No ☐ S

HbsAg ☐ Negativo ☐ Positivo

Tiempo en Métodos Depuradores

HLA

Sensibilidad

**Acto Quirúrgico**

Calcular duración del acto quirúrgico

Tiempo de Isquemia Caliente Primaria

Fecha de Inicio del Acto Quirúrgico

Inicio de la Sutura Venosa Pretrasplante 2 47 am

Inicio de la Sutura Arterial 2 47 am

Hora de Apertura de Bolsa 2 47 am

Inicio Continuidad Vía Urinaria 2 47 am

Tiempo Total Continuidad

Fecha de Fin del Acto Quirúrgico

Sitio de Implante Renal: Fosa Ilica

Biopsia Renal post-reperfusión ☐ No ☐ S

Observaciones

Tiempo de Isquemia Caliente Secundaria

Hora de Inicio del Acto Quirúrgico 2 47 am

Fin de la Sutura Venosa 2 47 am

Fin de la Sutura Arterial 2 47 am

Hora de Desclampaje

Fin Continuidad Vía Urinaria 2 47 am

Tipo de Continuidad Vía Urinaria --Seleccionar--

Hora de Fin del Acto Quirúrgico 2 47 am

Diuresis Inicial ☐ No ☐ S

Protocolo Inmunosupresor Inicial --Seleccionar--

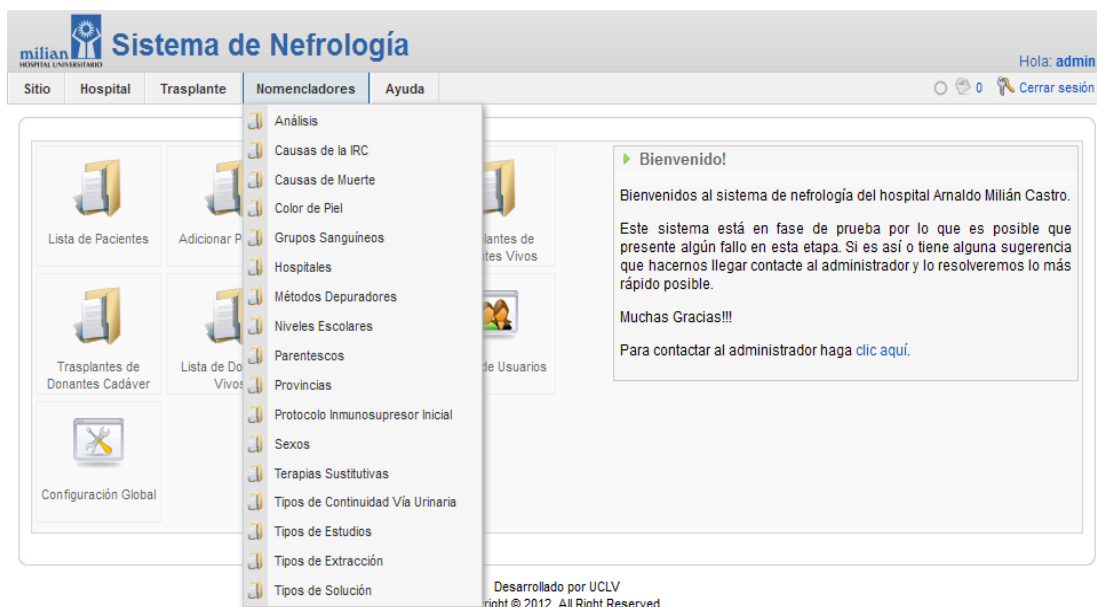
Desarrollado por UCLV  
Copyright © 2012. All Right Reserved

**Figura 3.19** Adicionar trasplante de donante vivo.

### 3.2.2.7. Configuración y nomencladores

Esta sección pretende exponer una concepción que se tiene en todo momento en el sistema y es la idea de convención sobre configuración, por lo que 17 de las 35 tablas con que actualmente cuenta el sistema son nomencladores. Esto posibilita que el sistema sea completamente configurable en cuanto a los términos que utiliza dejando en manos de los especialistas del área (los cuales deben tener rol de administrador) la creación y

modificación de estos. La forma de acceder a los nomencladores es mediante el Menú Principal como se puede apreciar en la Figura 3.20.



**Figura 3.20** Menú nomencladores.

También como se puede apreciar en la Figura 3.21 si el usuario cuenta con el rol “superadministrador”, puede cambiar los parámetros de configuración del sitio mediante las opciones del siguiente formulario.

**Configuración Global**

Nombre del sitio web\* Sistema de Nefrología

E-mail address\* jdbellon@uclv.edu.cu

Mensaje de pie de página  
Desarrollado por UCLV  
Copyright © 2012. All Right Reserved

Sitio Desactivado ☒ NO ☐ SI

Mensaje para el sitio web desactivado  
<br /><br /><br /><p align="center">Info: El sitio está desactivado por tareas de mantenimiento. Por favor, vuelve más tarde. </p>

Guardar Actividad ☒ NO ☐ SI

Compresión GZIP de las páginas ☒ NO ☐ SI

Mensaje de página principal  
<p>Bienvenidos al sistema de nefrología del hospital Arnaldo Milián Castro. </p>  
<p align="justify">Este sistema está en fase de prueba

META-Descripción del sitio  
Sistema de nefrología

META-Palabras clave del sitio  
nefrologia,hospital,Arnaldo,Milián,Castro

Desarrollado por UCLV  
Copyright © 2012. All Right Reserved

**Figura 3.21** Configuración global.

### **3.3. Conclusiones parciales**

Como resultado de este capítulo se puede concluir que:

- Se analizó el diseño de la BD anterior y se propuso modificaciones.
- Mediante el uso del lenguaje UML fue modelado el entorno Web, atendiendo siempre a los requerimientos de los usuarios. Al respecto, se definieron los casos de uso, los diagramas de navegación y actividad.
- Se expone, en forma de manual de usuario, las principales secciones del sitio y operaciones a realizar por el cliente en el manejo del sistema.

## **CONCLUSIONES**

Como resultado de la investigación:

- Se aplicaron los criterios que determinan la ubicación de las RN en la arquitectura del SI y se ubicaron estas en su capa correspondiente permitiendo una fácil actualización a nuevas condiciones de negocio.
- Se analizó el diseño de la BD existente y se propusieron modificaciones.
- Se obtuvo una nueva versión del sistema para el control de pacientes del AN.
- Se logró la puesta en marcha del sistema.
- Se validó el sistema desde el punto de vista de las RN, al lograrse implementar un significativo número de reglas, realizándose un análisis comparativo con sistemas que le anteceden, evidenciando su adaptabilidad.
- Se pobló la BD con datos de pacientes trasplantados en el 2010.

## RECOMENDACIONES

Para dar continuidad al presente trabajo, se recomienda:

- Adicionar otras áreas del hospital al sistema para potencializar su uso.
- Poblar la BD con todos los pacientes registrados en el hospital y los trasplantes históricos realizados en el AN.
- Realizar un módulo que, mediante una interfaz visual, permita al usuario construir sus propias reglas, insertándolas automáticamente en el sistema en su capa correspondiente aplicándole los criterios que determinan la ubicación de las RN.

## REFERENCIAS BIBLIOGRÁFICAS

- Appelton, D. S. (1984) Business Rules - The Missing Link. *Datamation*, 30, Nro 16, 145-150pp.
- Bajec, M. & Krisper, M. (2005) A methodology and tool support for managing business rules in organisations. *Information Systems*, Sep 2005, 30, no. 6, 423-443pp.
- Bajec, M., Krisper, M. & Rupnik, R. (2000): Using Business Rules Technologies To Bridge The Gap Between Business and Business Applications. The IFIP 16th World Computer Congress 2000, Information Technology for Business Management. Peking, China. RECHNU, G. E. 77-85pp
- Bajec, M. & Marjan, K. (2001) Managing business rules in enterprises. *Electrotechnical Review*, Ljubljana, Slovenija Elektrotehniški vestnik 68(4) 236-241pp.
- BRG (2000): Defining Business Rules ~ What Are They Really? Business Rules Group. Seattle. 77pp
- Ceri, S. (1997) Designing Database Applications with Objects and Rules: The IDEA Methodology.
- Date, C. J. (2000): What Not How: The Business Rules Approach to Application Development. pp
- García, J., Ortín, M. J., Moros, B., Nicolás, J. & Toval, A. (2004) De los Procesos del Negocio a los Casos de Uso.
- Gerrits, R. (2004) Business Rules, Can They Be Re-used? *Business Rules Journal*, 5.
- Goedertier, S. (2008): Declarative Techniques for Modeling and Mining Business Processes. p
- Goedertier, S., Haesen, R. & Vanthienen, J. (2007a): EM-BrA2CE v0.1: A Vocabulary and Execution Model for Declarative Business Process Modeling. K.U.Leuven.
- Goedertier, S., Mues, C. & Vanthienen, J. (2007b): Specifying Process-Aware Access Control Rules in SBVR. The International RuleML Symposium on Rule Interchange and Applications (RuleML2007). 39-52pp
- Goedertier, S. & Vanthienen, J. (2007c): A Vocabulary and Execution Model for Declarative Service Orchestration. 2nd Workshop on Advances in Semantics for Web services (semantics4ws'07) en Business Process Management Workshops. 496-501pp
- Hay, D. (1997) GUIDE Business Rules Project, Fial Report - revision 1.2. GUIDE International Corporation.
- Hernández, P. E., López, J., Cruz, R. E., Pérez, R. A., López, J., Batista, R. & Hernández, O. (2003) Protocolo de Trasplante Renal Donador Vivo Relacionado: Hospital Universitario "Arnaldo Milián Castro"
- Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B. & Dean, M. (2004) *SWRL: A Semantic Web Rule Language Combining OWL and RuleML* W3C Member Submission.
- Kasiske, B., Cangro, C., Harihanan, S., Hricik, D., Kerman, R., Roth, D., Rush, D., Vazquez, M. & Weir, M. (2001) The evaluation of renal transplant candidates: Clinical practice guidelines. *J Am Transplant* 2(suppl 1), 5-95pp.
- Kovacic, A., Groznik, A. & Krisper, M. (2001) Business renovatios: from business process modelling to infomation system modelling. *International Journal of Simulation*, 2.
- Layzell, P. J. & Loucopoulos, P. (1988): A rule-based approach to the construction and evolution of business information systems. Fourth IEEE International Conference on Software Maintenance. Phoenix, Arizona, EEUU. 258-264pp
- Lovelle, J. M. C. (1999) *Calidad del Software*. 12pp.
- Machado Padilla, C. A. (2009): Modelación de Reglas de Negocio. Departamento de Computación Universidad Central de Las Villas. Santa Clara, Villa Clara. 66p
- Martínez Busto, M. E., Moreno Montes de Oca, I., Machado Padilla, C. A. & González González, L. (2009): De los procesos de negocio a la modelación de reglas de negocio para un



- caso de estudio. Memorias del IV Encuentro de la Red Iberoamericana de evaluación y Decisión Multicriterio. Zapopan, Jalisco, México. 9-15 de Noviembre.
- Mena, J. A. G. (2011) Inserción e implementación de las reglas de negocio dentro de la arquitectura de los sistemas de información. 75pp.
- Mena, J. A. G. & Vellón, M. D. (2008): Modelación de reglas de negocio como apoyo para sistemas de información en el área de nefrología. Departamento de Programación. Universidad Central de las Villas. Santa Clara, Villa Clara. 94p
- Morgan, T. (2002): Business Rules and Information Systems: Aligning IT with Business Goals. Addison Wesley. pp
- Ojo, A., Hanson, J., HU., M.-K., Okechukwu, C., Wolfe, R., Liechtman, A., Agodoa, L., Kaplan, B. & Port, F. (2001) Survival in recipients of marginal cadaveric donor kidneys, compared to other recipients and wait-listed transplant patients. J Am Soc Nephrol 12, 589-597pp.
- Redín, A., Larrea, B., Osés, B., Linacero, M. C., Berantegui, M. & Labairu, E. (2006) Informatización del protocolo de cuidados del trasplante renal. Clínica universitaria de Navarra. Pamplona.
- Rosca, D., Greenspan, S. J., Wild, J. C., Reubeinstein, H. B., Maly, K. & Feblowitz, M. (1995): Application of a Decision Support Mechanism to the Business Rules Lifecycle. KBSE. 114-121pp
- Ross, R. (1997): The Business Rule Book: Classifying, Defining and Modelling Rules. Ross Method, version 4.0. Texas, Huston. pp
- Ross, R. G. (2003a) The Business Rule Approach. IEEE Computer, 36, 85-87pp.
- Ross, R. G. (2003b): Principles of the Business Rule Approach. Addison-Wesley. pp
- Ross, R. G. (2005): Business Rule Concepts ~ Getting to the Point of Knowledge Business Rule Solutions, LLC. pp
- Ross, R. G. (2010): Business Rule Concepts. Business Rule Solutions, LLC. 158 pp
- Schreiber, G., Akkemans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W. & Wielinga, B. (1999): Knowledge Engineering and Management: The CommonKADS Methodology. The MIT Press. 471 pp
- Soliveres, P. A. (1997): Desarrollo Cliente/Servidor: ubicación de las reglas de negocio. Revista Profesional para Programadores (RPP).  
<http://www.softwarementors.com/arti/csbr/csbr.htm>. Última visita:
- Struck, D. L. (1999): Business Rule Continuous Requirements Environment. PhD Thesis. Colorado Technical University. Colorado Springs, Colorado. p
- Velazco, A. R. & Ulloa, R. J. V. (2010) Sistema de Bases de Datos para el control de pacientes en el área de Nefrología guiado por reglas de negocio. 98pp.
- Weiden, M., Hermans, L., Schreiber, G. & van der Zee, S. (2002): Classification and Representation of Business Rules. Última visita:
- Weiden, M. R. (2000): A Critique of the Business-Rule Approach. Department of Social Science Informatics. University of Amsterdam. Amsterdam. p
- Youdeowei, A. (1997): The B-Rule Methodology: A Business Rule Approach to Information Systems Development. Department of Computation UMIST, . Manchester, United Kingdom. p

# ANEXOS

## Anexo1

Diagrama entidad relación (E/R) para Trasplante Renal.

