

Sistema inteligente para el tratamiento de alarmas en anestesiología*

Drozdowicz, Bartolomé**; Hadad, Alejandro***; Evin, Diego**; Salvatelli, Adrian**; Kohan, Diana**

Resumen

Se describe el desarrollo de un Sistema de Información basado en Multiagentes, cuyo objetivo es analizar las señales de monitoreo de procesos anestesiológicos, contextualizarlas con los datos del paciente y de la cirugía, para luego generar las alarmas correspondientes solamente para aquellas situaciones que lo justifiquen. De esta forma se trata de minimizar el número de falsas alarmas y aumentar la calidad de la información disponible para los anestesiólogos. La arquitectura del sistema está conformada por un grupo de Agentes Inteligentes: Diagnosticador basado en Redes Neuronales, Guardián que utiliza algoritmos del Control Estadístico de Procesos Multivariados, Validador que contiene modelado con Redes Bayesianas y el Generador de Alarmas. Para su diseño se utilizó la metodología Gaia, con los modelos de Roles, Actividades y Agentes. Se trabajó sobre casos de pacientes con Hemorragia Aguda de 100, 300, 500, 700 y 900 ml/min, obteniéndose resultados satisfactorios.

Palabras clave: bioingeniería, alarmas en anestesiología, sistema multiagente, redes neuronales, redes bayesianas

*) Este artículo presenta los resultados del PID UNER N° 6066, Facultad de Ingeniería, Universidad Nacional de Entre Ríos; recibido en septiembre 2007; admitido en diciembre 2007.

**) Facultad de Ingeniería, Universidad Nacional de Entre Ríos, Oro Verde, Argentina. bdrozdo@ceride.gov.ar

***) Instituto de Desarrollo y Diseño -INGAR, Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Santa Fe, Argentina, y Facultad de Ingeniería, Universidad Nacional de Entre Ríos, Oro Verde, Argentina.

Intelligent System for the Alarms Treatment in Anesthesiology*

Drozdowicz, Bartolomé**; Hadad, Alejandro***; Evin, Diego**; Salvatelli, Adrian**; Kohan, Diana**

Abstract

The development of an Information System is described. Its aim is to analyze the monitorized signals during the anesthesiological processes, considering the patient and surgery data in order to minimize the false alarms and improve the quality of the information used by the anesthesiologists. The System Architecture is based on Intelligent Agents: Diagnosticator composed by Neural Networks models, Guardian which uses algorithms from the Multivariate Statistic Processes Control, Validator which contains Bayesian Networks models and the Alarm Generator Agent. The Gaia methodology was used for the System design, considering the Roles, Activities and Agents Models. As reference cases, patients with different level of Acute Hemorrhage (100, 300, 500, 700 and 900 ml/min) were considered.

Keywords: Bioengineering, Alarms in Anesthesiology, Multiagent System, Neural Networks, Bayesian Networks

*) This paper presents the results of the research project UNER # 6066, Faculty of Engineering, Universidad Nacional de Entre Ríos; submitted in september 2007; admitted en December 2007.

**) Facultad de Ingeniería, Universidad Nacional de Entre Ríos, Oro Verde, Argentina. bdrozdo@ceride.gov.ar

***) Instituto de Desarrollo y Diseño -INGAR, Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Santa Fe, Argentina, y Facultad de Ingeniería, Universidad Nacional de Entre Ríos, Oro Verde, Argentina.

I. Introducción

En virtud de las permanentes investigaciones, la calidad de los procesos en el ámbito médico ha tenido constantes avances en pos de una mejor y más confiable atención de los pacientes. Específicamente en anestesiología, estas mejoras involucran tres aspectos: conocimiento más profundo respecto de la fisiología humana, surgimiento de nuevas drogas y equipos de anestesia, y finalmente, la aparición de nuevos dispositivos de monitoreo, que permiten una mejor evaluación sobre el estado del paciente. Como resultado de estos avances, se ha aumentado la confiabilidad del proceso anestesiológico.

Sin embargo, existen otros aspectos que pueden incrementar estas mejoras. Estudios de incidentes durante los procesos anestesiológicos, muestran que un porcentaje de los mismos involucran factores humanos (**Block et al.**, 1996), lo que indica que es posible mejorar la confiabilidad del proceso, disminuyendo la probabilidad de incidencia de los mismos.

En sistemas complejos como lo son los procesos anestesiológicos, uno de estos factores humanos se relaciona con situaciones donde, por sobrecarga de información o exigente requerimiento en el tiempo de respuesta, resulta difícil para el profesional evaluar el estado de los mismos, especialmente si existe cierto grado de incertidumbre en sus comportamientos. En consecuencia, la probabilidad de ocurrencia del error humano aumenta cuando los ámbitos donde los profesionales deben desempeñar sus funciones son complejos, tanto por la variabilidad de las situaciones como por la criticidad de las decisiones. Por otro lado, está demostrado que el mayor porcentaje de errores humanos no se debe a problemas intrínsecos de las personas sino a falencias del contexto donde la misma debe desempeñarse, siendo la baja calidad de la información sobre el proceso uno de estos aspectos. Una forma de incrementar la confiabilidad humana es el empleo de Sistemas Informáticos, que brinden soporte a las tareas del anestesiólogo.

Teniendo en cuenta esta problemática, se fundamentó el objetivo del proyecto *Sistema de Información para el Tratamiento Inteligente de Alarmas en Anestesiología*, desarrollado por el Grupo de Inteligencia Artificial de la Facultad de Ingeniería de la Universidad Nacional de Entre Ríos. El

propósito de este artículo es la descripción del desarrollo de ese Sistema de Información.

Algunos estudios indican que generalmente del 40 a 50% de todas las alarmas son falsas (**Block et al.**, 1996; **Wiklund et al.**, 1994; **Cook et al.**, 1996; **Koski et al.**, 1992). Este es un claro ejemplo de que la información, si no es bien presentada, muchas veces no agrega beneficios sino, por el contrario, trae complicaciones. En un informe de la US Food and Drug Administration (FDA) (**Graham et al.**, 1998) se hace referencia a la importancia del problema de alarmas durante el monitoreo. Las alarmas sin sentido, para las situaciones específicas en consideración, son indeseables y sólo llevan a la desconexión de las mismas, como modo normal de operación.

En el diseño de equipamiento para anestesia existe una tendencia hacia la integración de dispositivos y centralización de controles y presentación de la información. La integración de alarmas, sin embargo, no es sencilla debido a la falta de estandarización, falta de consenso en relación a su función, etc.

La implementación de alarmas puede ser mejorada empleando técnicas de Inteligencia Artificial (**Orr et al.**, 1994; **Rau et al.**, 1995; **Oshita et al.**, 1994; **DeGraaf et al.**, 1997; **Mylrea et al.**, 1993; **Schecke et al.**, 1994; **Wolf et al.**, 1996), diseño ergonómico y desarrollo de nuevas metodologías para la medición de variables fisiológicas. Este trabajo considera la primera de estas alternativas, para lo cual en una primera etapa fue necesario establecer cuáles son los datos necesarios, de dónde se obtienen y en qué forma deben estar expresados, para poder brindar información concisa y precisa al Sistema Inteligente.

Para esto se desarrolló una interfaz (hardware / software) capaz de adquirir los datos provenientes del exterior (paciente, anesthesiólogo, etc.) y expresarlo en un formato acorde a lo requerido por el Sistema Inteligente. A partir de los datos del paciente bajo anestesia y de aquellos específicos del estado particular en el que se encuentra, se generan advertencias y alarmas para cada estado crítico, informando al anesthesiólogo las causas más probables de ese estado y en algunos casos las posibles acciones recomendadas.

Además, se pueden diferenciar dos aspectos respecto de la

problemática a considerar en el ámbito anestesiológico: por un lado las alarmas relacionadas con fallas del equipamiento utilizado en los procesos anestesiológicos, y por el otro las alarmas relacionadas con las desviaciones de los parámetros vitales de los pacientes que, bajo muy variadas circunstancias, son sometidos a procesos anestesiológicos específicos.

Por todo lo anteriormente indicado resulta importante el diseño de un Sistema de Información que permita al profesional interpretar mejor cada una de las situaciones que se van desarrollando a lo largo de un proceso anestesiológico. De tal forma se podrán mejorar las decisiones y consecuentemente disminuir la ocurrencia de errores humanos. Para ello resulta muy importante contar con una arquitectura modular, flexible y adaptable a las diferentes situaciones, que además sea capaz de seguir la evolución del proceso y almacenar los datos y eventos del mismo, para el posterior análisis de los diversos casos procesados.

En consecuencia, los principales resultados obtenidos se relacionan con el desarrollo, formalización e implementación de la arquitectura del Sistema, el cual quedó conformado como un Sistema Multiagente, compuesto por Agentes específicos para desarrollar las diferentes funciones requeridas para el procesamiento de la información, relacionado con los procesos anestesiológicos.

I.1. Descripción del ámbito de aplicación

La complejidad de las cirugías actuales es posible, en parte, gracias al desarrollo realizado en el área de anestesiología. Sin embargo, la monitorización de pacientes críticos plantea un problema de elevada complejidad y alto estrés al profesional que la esta llevando adelante.

En los últimos años, la Bioingeniería ha desarrollado un gran número de dispositivos capaces de medir diferentes parámetros vitales de los pacientes. Estas mediciones mejoran la seguridad de los mismos y brindan información al médico acerca de sus diferentes parámetros fisiológicos. Dado que las técnicas clásicas de monitorización consisten en la adquisición masiva de datos y presentación directa de toda la información, pueden producir una sobrecarga sobre el anestesiólogo.

Los profesionales médicos deben ser notificados de los cambios

sustanciales en el estado del paciente y fundamentalmente en aquellos casos donde exista una alta probabilidad de riesgo para la vida del mismo. Sin embargo el monitoreo de pacientes produce, en muchas ocasiones, una gran cantidad de alarmas innecesarias para la situación específica del paciente bajo atención, o alarmas falsas provocadas por eventos ajenos al estado real del mismo, que terminan siendo la principal fuente de molestia y confusión para el personal encargado de su cuidado.

Básicamente, las alarmas de los sistemas de monitoreo de pacientes se configuran para producir un alerta en caso que la variable a la cual están atendiendo se encuentre fuera de cierto rango. Sin embargo, estas alarmas no son capaces de considerar la intervariabilidad de los diferentes parámetros, ya que no tienen en cuenta las características particulares del paciente ni las acciones que se puedan estar realizando sobre él. En otras palabras, estos sistemas no incluyen inteligencia, aspecto éste de mucha importancia ya que, en pacientes críticos, ciertos parámetros necesitan ser evaluados considerando las características particulares de los mismos, esto es, edad, sexo, patologías anteriores, etc. Además, los monitores no están en condiciones de interpretar los valores de las variables, contextualizándolas con los casos específicos en tratamiento, por ejemplo, en ciertas ocasiones, una frecuencia cardíaca de 125 latidos por minuto podría no ser considerada taquicardia, cuando generalmente lo es, o una presión sanguínea sistólica de 120 mmHg podría ser considerada elevada, siendo normal para la mayoría de los casos.

En consecuencia, el ámbito general de la anestesiología constituye un sistema complejo con múltiples condicionantes y una muy amplia variabilidad de las situaciones, por lo cual conforman un contexto apropiado para la incorporación de Sistemas Informáticos Inteligentes, que permitan aumentar la calidad de la información en la cual se deben basar los profesionales para tomar sus decisiones.

1.1.1. Ámbito Anestesiológico

El Proceso Anestesiológico

Si bien no existen procedimientos fijos que los anestesiólogos deban seguir paso a paso para cada situación que se les presente, es posible caracterizar un conjunto de similitudes entre los diferentes casos

anestésicos y, en base a ellos, definir un procedimiento general, entendiendo que estos procedimientos pueden tener variantes de acuerdo a los diferentes contextos de aplicación. En la intervención del anestesiólogo dentro de una cirugía se pueden distinguir las siguientes fases: preparación prequirúrgica, inducción, mantenimiento, despertar de la anestesia y/o control posquirúrgico (**Graham et al.**, 1998).

Preparación Prequirúrgica: En esta etapa, el anestesiólogo analiza las características del paciente, realiza un plan anestésico y completa un chequeo del funcionamiento del equipamiento y disponibilidad de los insumos necesarios.

Inducción: El objetivo de esta etapa es llevar al paciente, tan rápido como sea necesario, a un estado de anestesia adecuado para que la cirugía pueda dar comienzo. Involucra además la conexión del paciente a los equipos de monitoreo y/o soporte. Durante esta etapa, el estado fisiológico del paciente se modifica dramáticamente en breves lapsos de tiempo. Además esos cambios se reflejan en distintas señales fisiológicas y con modalidades diferentes.

Mantenimiento: Una vez que el paciente ha sido anestesiado, la tarea del anestesiólogo consiste en mantener su estado, realizando un balance entre sus diferentes subsistemas y modificando el estado del paciente de acuerdo a las demandas del cirujano (nivel de relajación, presión arterial, temperatura corporal, etc.). Estas tareas se logran a través de constantes suministros de drogas, ajustes en los parámetros ventilatorios y del vaporizador, etc. En esta etapa, el anestesiólogo dedica la mayor parte del tiempo a monitorear el estado del paciente y la cirugía. El progreso quirúrgico tiene particular importancia para el anestesiólogo, debido a que le permite planificar las dosis anestésicas, de acuerdo al tiempo que resta para su finalización

Despertar de la anestesia y/o control posquirúrgico: Hacia el final de la cirugía, el paciente emerge de la anestesia profunda, regresando al estado de conciencia y despertar. En esta fase, al igual que durante la inducción, el estado fisiológico del paciente se modifica de manera considerable en breves lapsos y el anestesiólogo debe establecer parámetros de referencia para interpretar los signos clínicos y la información de los monitores. Generalmente al finalizar el proceso, el

anestesiólogo suele registrar las principales observaciones respecto a la cirugía y aquellos eventos que afectaron al paciente.

1.1.2.- Caso de Referencia: Hemorragias Agudas

Los ámbitos profesionales con los que se interactuó fueron el Departamento de Anestesiología del Hospital "San Martín" de Paraná y la Asociación de Anestesiología de Buenos Aires (AABA).

Tomando en consideración la opinión de anestesiólogos de la AABA y con el total acuerdo del grupo de anestesiólogos del Hospital "San Martín", se consideró como caso de referencia específico, para el desarrollo del Sistema Inteligente, a situaciones de pacientes con Hemorragias Agudas (HA), por la criticidad de las consecuencias y la complejidad y cantidad de diferentes tipos de HA. Si bien se trabajó con datos asociados a estas situaciones específicas, los desarrollos del Sistema Inteligente son totalmente generales, pues se consideraron los aspectos relevantes de la problemática en general, relacionada con el diagnóstico, validación y supervisión de situaciones críticas, como así también porque el conjunto de variables medidas que determinan el estado del paciente es común para todas las situaciones que pueden generarse durante los procesos quirúrgicos.

Los pacientes con Hemorragias Agudas plantean múltiples problemas a todo el equipo médico; a menudo necesitan métodos quirúrgicos urgentes para el diagnóstico y el tratamiento, y en ellos interviene directamente el anestesiólogo. Incluso el más experto puede afrontar situaciones muy difíciles, en estas circunstancias extremas y ocasionales.

Esto muestra la complejidad del caso analizado, más aún si se tiene en cuenta la fisiología de manifestación del mismo: Dependiendo de la velocidad de pérdida sanguínea, variará la presión tanto sistólica como diastólica, para poder mantener la oxigenación de los tejidos, o visto de otra manera, disminuirá el anhídrido carbónico acumulado en el fluido. Por otra parte, a fin de facilitar el intercambio gaseoso en los pulmones se aumenta la frecuencia cardíaca, lo que hace variar el gasto cardíaco, entrando el paciente en deterioro continuo de su estado. Desde el punto de vista anestesiológico, el fluido sanguíneo es el medio de transporte de las drogas o gases anestésicos, lo cual deja en claro la importancia

de este evento.

Se debe discriminar entre dos grupos de pacientes con HA. En primer lugar, aquéllos que ingresan a los servicios de urgencia y a los que se diagnostica una posible HA y luego se interviene quirúrgicamente, con lo cual el anestesiólogo actúa con conocimiento de causa, aunque con no menos inconvenientes. En segundo lugar, aquellos pacientes que, durante su intervención quirúrgica, por alguna circunstancia, sufren una HA. Éste fue el grupo de pacientes considerado para el Sistema desarrollado en este proyecto, pues resulta de mayor importancia por sus características y porque es donde se debe evaluar con mayor detalle las condiciones de alarma.

II. Metodologías

II.1. Sistema Multiagente

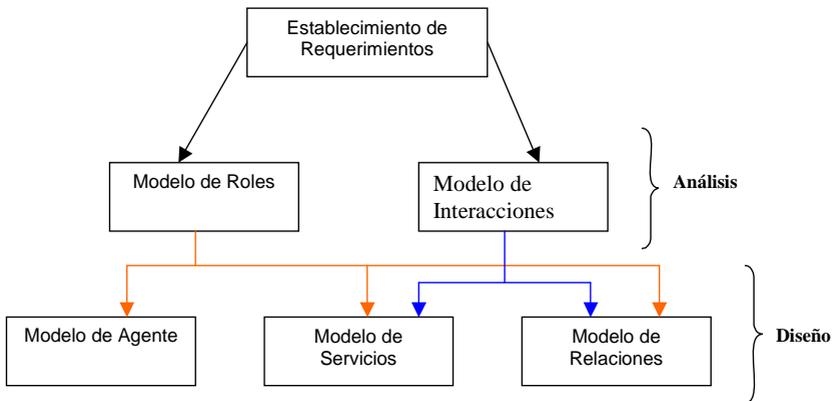
Teniendo en cuenta la necesidad de contar con una estructura general del Sistema de Información Inteligente (SII), modular, flexible y adaptativa, que permita considerar las diferentes situaciones que se generan en un ámbito complejo, como lo son los procesos anestesiológicos, y después de un análisis de las arquitecturas más apropiadas para este tipo de desarrollo, como son las estructuras de Blackboard y Sistema Multi-agente, se consideró a esta última arquitectura como la más apropiada. Esta arquitectura permite una implementación mucho más distribuida del SII, aspecto que puede resultar importante cuando se tengan que considerar las restricciones y requerimientos que impone el Dominio de Aplicación.

Cuando se va a desarrollar un sistema software complejo como suelen ser los sistemas Multi-agentes (SMA), es conveniente utilizar metodologías validadas que estructuren tal desarrollo. Para el análisis y diseño del SMA se consideraron dos metodologías específicas: Gaia y MaSE (*Multi-agent Systems Engineering*). Ambas aplican el concepto de rol para guiar el descubrimiento de funciones del sistema. El objetivo en la fase de análisis es desarrollar una comprensión del sistema y su estructura, sin hacer referencia a detalles de implementación. Mientras que el objetivo de la fase de diseño, es transformar los modelos abstractos derivados durante

el estado de análisis, en modelos de suficientemente bajo nivel de abstracción, para que ellos puedan ser implementados fácilmente. Por las características del dominio en consideración, el SII deberá actuar en un ambiente cerrado y acotado, y tendrá un conjunto limitado de Agentes (no más de diez) cuyas habilidades, servicios y relaciones interagentes no cambiarán en tiempo de ejecución. Tales características coinciden con los principales requerimientos de los dominios exigidos por la metodología Gaia, por lo tanto esta última fue considerada la más apropiada.

La metodología Gaia ve al sistema como una sociedad u organización, con los elementos de tal sociedad definido por roles (**Wooldridge et al.**, 2000). Tiene el objetivo de permitir al analista ir sistemáticamente, desde el establecimiento de los requerimientos, hasta llegar a un diseño que sea suficientemente detallado, tal que pueda implementarse directamente. Esta metodología tiene una fase de Análisis y otra de Diseño (**Fig. 1**), que pueden ser pensadas como un proceso de desarrollo de modelos cada vez más detallados, del sistema a construir.

Figura 1: Trabajo: Sistema Inteligente para el Tratamiento de Alarmas en Anestesiología



II.1.1. Análisis

Tiene como objetivo definir la organización del sistema (sin referenciar ningún detalle de implementación). Los pasos en el proceso de Análisis son: Identificación de los roles del sistema; para cada rol, determinar y documentar los protocolos asociados, con lo cual se construye un modelo de interacciones y, finalmente, con éste se elabora el modelo de roles.

II. 1. 2. Diseño

Tiene por finalidad transformar los modelos de la fase de análisis en un nivel de abstracción suficientemente bajo, tal que puedan utilizarse las técnicas tradicionales de diseño para implementar agentes. Se define cómo cooperan los agentes para cumplir los objetivos del sistema y qué requiere cada agente individual para llevarlo a cabo. Los pasos de la fase de Diseño son: definir un modelo de agentes, desarrollar un modelo de servicios y hacer un modelo de relaciones

II.2. Agentes Inteligentes

Para la implementación de los diversos Agentes, se utilizaron diferentes metodologías específicas para cada objetivo particular. De tal forma, el Agente Guardián está basado en metodologías del Control Estadístico de Procesos Multivariados, el Agente Diagnosticador utiliza Redes Neuronales, el Agente Supervisor está basado en Redes Bayesianas y el Agente Evaluador de Profundidad Anestésica se basa en conceptos de Análisis Biespectral y Entropía.

II.3. Redes neuronales

El modelado del Agente Diagnosticador se basó en Redes Neuronales. Las redes neuronales Perceptrón Multicapa son un modelo matemático compuesto por un gran número de elementos simples adaptativos (nodos), interconectados masivamente en paralelo, los cuales procesan información por medio de su estado dinámico, como respuesta a entradas externas (**Haykin**, 1993). Estas redes son capaces de aprender de la experiencia, generalizando casos anteriores a nuevos casos y abstrayendo características esenciales del dominio de trabajo, a partir de

las entradas representativas del mismo.

Numéricamente las redes presentan un vector de entrada de dimensión N , una matriz de pesos de interconexión entre los nodos, de una o más capas ocultas, que almacenan el conocimiento adquirido por la red, y un vector de salidas. Para implementar una red neuronal, se hace necesario contar con una serie de datos sobre los que se llevará a cabo el análisis, así como un conjunto de salida deseada para cada uno de tales datos. Estos datos se dividen en grupos de patrones: de *entrenamiento* y de *evaluación*, que se utilizarán durante las dos fases homónimas, necesarias para dicha implementación.

La *fase de entrenamiento* consiste en la presentación sucesiva de los datos de entrenamiento y la adaptación del valor de pesos de las interconexiones, con el objetivo de minimizar los errores cuadráticos medios (el error consiste en la diferencia entre la salida real tras la presentación de un patrón específico y la salida deseada para ese patrón).

La *fase de evaluación o verificación* consiste en exponer la red ya entrenada (con los pesos de interconexión fijos) a los datos de evaluación. Esta fase tiene el objetivo de comprobar el comportamiento de la red ante muestras que no se habían presentado previamente durante la fase de entrenamiento. Si este proceso arroja un error de clasificación de valor aceptable, se almacena la matriz de pesos que caracterizará completamente la red neuronal.

Para evaluar las prestaciones del detector, se puede utilizar una medida del número de falsas detecciones realizadas (falsos positivos -FP-) y del número de eventos no detectados o detecciones fallidas (falsos negativos -FN).

II.4. Control estadístico de procesos multivariados

Esta metodología fue utilizada en el Agente Guardián. Las herramientas del control estadístico de procesos multivariados más extensamente desarrolladas en la literatura son el estadístico T^2 de Hottelling, MEWMA (Multivariate Exponentially Weighted Moving Average) y MCUSUM (Multivariate Cumulative SUM). Para estas herramientas se asume que vectores de $p \times 1$ (p : número de variables) X_1, X_2, X_3, \dots , son observados y monitoreados a lo largo del tiempo. Estas

observaciones se consideran independientes y distribuidas normalmente con vector de medias μ_i , $i = 1, 2, \dots, p$. También se asume que es conocida la matriz de covarianza Σ y el proceso bajo control es denotado por el vector de medias μ_0 . En la práctica es necesario estimar la matriz de covarianzas Σ y chequear las asunciones de independencia y de distribución normal de las observaciones multivariantes.

El estadístico T^2 está basado en la observación más reciente, siendo insensible a cambios pequeños o moderados, en el vector de medias del proceso. Los métodos MEWMA y MCUSUM tienen como ventaja que van considerando las observaciones anteriores, por lo que *a priori* permiten detectar cambios pequeños o moderados. Sin embargo, en relación a T^2 , requieren definir parámetros de diseño, que no se obtienen directamente de los datos.

Todos estos métodos asumen que los datos utilizados para la estimación del vector de medias y de la matriz de varianza-covarianza se distribuyen normalmente. Otra asunción es que los vectores X_1, X_2, \dots, X_n son estadísticamente independientes, asunción que también suele ser no realista. En este sentido, el gráfico MEWMA puede tratar mejor el efecto de la no independencia estadística. Teniendo en cuenta todas estas consideraciones para el diseño, se consideraron los métodos T^2 de Hotelling y MEWMA con filtrado de outliers y transformación.

II.5. Redes Bayesianas

Una Red Bayesiana es un grafo acíclico dirigido, que codifica relaciones probabilísticas entre eventos distintivos de interés, de un problema de razonamiento bajo incertidumbre. Una red de este tipo combina la potencia del teorema de Bayes, con la expresividad semántica de los grafos dirigidos. Esta metodología fue utilizada en el Agente Supervisor.

Una Red Bayesiana tiene las siguientes características:

- Un conjunto de nodos que representan las variables del modelo, cada uno de los cuales es una variable aleatoria, discreta o continua, que tiene un conjunto exhaustivo de estados χ_i , mutuamente excluyentes.
- Cada par de nodos se conecta entre sí mediante arcos de flechas.

Un arco representa la dependencia entre dos variables del modelo. Una flecha que vaya del nodo X al nodo Y tienen el significado implícito de que X, al que llamamos nodo padre, ejerce una influencia directa sobre Y, nodo hijo.

- Para cada nodo existe una tabla de probabilidad condicional, que cuantifica el efecto que sobre este tiene cada nodo padre. El grafo no tiene ciclos.

Para construir una red bayesiana, en primer lugar se debe establecer el conjunto de variables que va a describir el dominio; para cada una de dichas variables se construye un nodo en la red, y posteriormente se debe establecer la topología de la red. Esto generalmente se hace con ayuda de expertos del dominio, que indican qué relaciones de dependencia condicional directas son válidas (en la definición de la estructura de la red, queda plasmado el conocimiento experto). Podría considerarse a la topología de la red como una base de conocimientos abstractos, válida en una gran variedad de escenarios diversos, al representar la estructura general de los procesos causales del dominio, y no los detalles de la población de sus individuos. Una vez diseñada la estructura de la red, se construye una tabla de probabilidad condicional para cada nodo de la red. En estas tablas, una fila indica la probabilidad condicional del nodo, dado una posible combinación de valores de los nodos padres. Si un nodo no tiene padres, su tabla de probabilidad condicional tiene una sola fila, con el valor de la probabilidad a priori de la variable representada por tal nodo.

El principal objetivo de una red de inferencias probabilista, como lo es una Red Bayesiana, consiste en calcular la distribución de probabilidades *a posteriori* de un conjunto de variables de consulta, con base en determinadas variables de evidencia.

III. Resultados

El desarrollo del Sistema Mutiagente tuvo los siguientes resultados, como consecuencia de la aplicación de la metodología Gaia.

III.1. Etapa 1: Definir la meta del sistema

La meta del sistema, ya establecida desde el principio, es la siguiente:

"Reducir la posibilidad de daños en pacientes sometidos a procesos anestesiológicos, dando soporte al anestesiólogo para mantener el proceso de anestesia en condiciones normales durante una cirugía".

III.2. Etapa 2: Definir los procesos y sus actividades

III.2.1. Procesos

Para esta etapa, se propuso, primero, modelar al sistema como un conjunto de procesos definiendo, como ya se mencionó anteriormente, sus entradas, salidas y orden de precedencia de las actividades que los conforman (**Fig. 2**). Aplicando este paso, se determinó que el sistema está conformado por un solo proceso: "Dar soporte al proceso anestesiológico".

Fig. 2. Entradas y salidas de los procesos

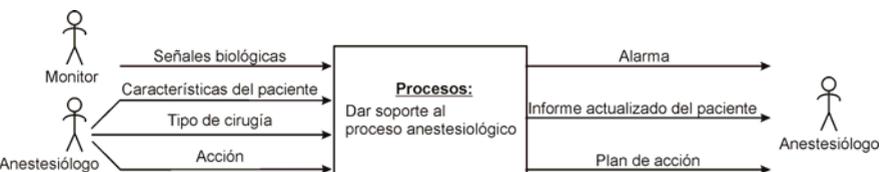


Fig. 2. Entradas y salidas de los procesos.

Las Entradas que recibe son las siguientes:

Señales biológicas: Proviene desde los monitores del quirófano. Son datos muestreados periódicamente de las señales vitales del paciente (frecuencia cardiaca, presión arterial, etc.). Se utilizan para que el sistema deduzca el estado actual del paciente.

Características del paciente: Datos descriptivos del paciente tanto estáticos (edad, sexo, peso) como así también intermitentes (análisis de laboratorio). Son cargados por el anestesiólogo antes de comenzar la cirugía.

Tipo de Cirugía: Indica el tipo de cirugía que se realizará. El anestesiólogo la selecciona a partir de una lista de cirugías conocidas

y previamente cargadas.

Acción: Indica al Sistema la acción llevada a cabo por el anestesiólogo. Esto permite el seguimiento del plan de acción.

El Sistema tiene como Salidas información útil para el anestesiólogo:

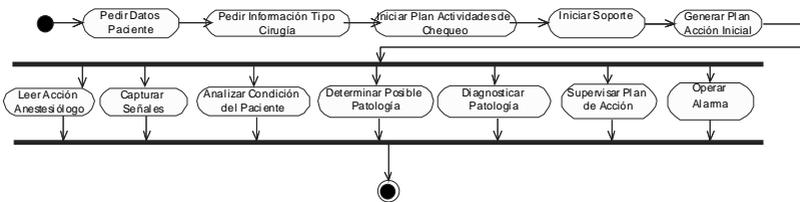
Alarmas: los actuales equipos de monitoreo de pacientes producen frecuentemente alarmas innecesarias para la situación específica del paciente bajo atención o falsas alarmas provocadas por eventos ajenos al estado real del mismo, que terminan siendo la principal fuente de molestia y confusión para el personal encargado de su cuidado. Por lo tanto, el sistema controlará las alarmas, emitiéndolas sólo en los casos necesarios.

Informe actualizado del paciente y plan de acción: el sistema proveerá al anestesiólogo de un informe actualizado del estado del paciente y de un plan de acción sugerido a seguir, tanto en condiciones normales o en condiciones anormales (emergencia).

III.3. Actividades

Teniendo en cuenta las definiciones anteriores, se conformó un diagrama de actividades en UML, que muestra el proceso principal del sistema, a partir del cual se definieron los roles (**Fig. 3**).

Fig. 3. Proceso del sistema



El proceso está dividido en dos fases: en primer lugar la fase de configuración y en segundo lugar, la fase de seguimiento y soporte.

La fase de configuración consiste en las primeras cinco actividades del proceso. En ellas, el anestesiólogo debe configurar el Sistema para comenzar la cirugía. Para tal fin, el Sistema es alimentado con las características propias del paciente y las del tipo de cirugía a realizar.

Luego, se genera un plan de actividades de chequeo, que debe realizar el anestesiólogo para verificar el correcto funcionamiento de todos los equipos. Finalmente, el Sistema presenta un plan de acción inicial a recomendar e inicia las actividades propias de la fase siguiente.

La fase de seguimiento y soporte está conformada, como puede apreciarse en la **Fig. 3**, por un conjunto de actividades que se llevarán a cabo concurrentemente:

Leer Acción del Anestesiólogo: El Sistema es capaz de recibir como entrada una acción del anestesiólogo y luego almacenarla. Esto le permite analizar los posibles efectos de dicha acción, como así también modificar el plan de acción a sugerir.

Capturar señales: Toma periódicamente muestras de las señales vitales provenientes desde los monitores y las guarda para su posterior análisis.

Analizar Condición del Paciente: Lee las señales capturadas y determina si la condición del paciente es normal o anormal. Si es anormal habilita la actividad *Determinar Posible Patología*.

Determinar Posible Patología: Se encarga de leer las señales tomadas del paciente y analizarlas, para determinar si el paciente podría padecer una patología conocida. Si encuentra tal situación, habilita la actividad *Diagnosticar Patología*.

Diagnosticar Patología: Tiene como objetivo analizar las posibles patologías que se encuentran en el paciente y determinar si verdaderamente existe una. Activa la alarma pertinente si fuera necesario.

Supervisar Plan de Acción: A partir del estado del paciente y de las acciones realizadas por el anestesiólogo, define y actualiza el plan de acción a seguir.

Operar Alarma: Activa o desactiva la alarma correspondiente.

Cada una de estas actividades ha sido detallada con el fin de determinar los roles que deberán desempeñar los Agentes del Sistema.

III.4. Modelo de Roles

A partir de las definiciones de las actividades del sistema, los roles que desempeñarán los distintos Agentes del Sistema pueden ser descubiertos. Los roles pueden realizar una o más actividades. No existe

ninguna regla que dicte cómo asignar actividades a los roles. Generalmente, un rol realiza un conjunto de actividades relacionadas entre sí, es decir que, por ejemplo, comparten un mismo recurso. Por otro lado, si la actividad es demasiado compleja, es aconsejable que el rol la realice de manera exclusiva, es decir que el rol tenga esa única actividad.

Ya obtenidos todos los roles del Sistema, se procede a describirlos mediante el Modelo de Roles propuesto por Gaia. Se muestra, a modo de ejemplo, la plantilla del rol DiagnosticadorPatología. (Fig. 4)

Fig. 4. Plantilla del Rol DiagnosticadorPatología

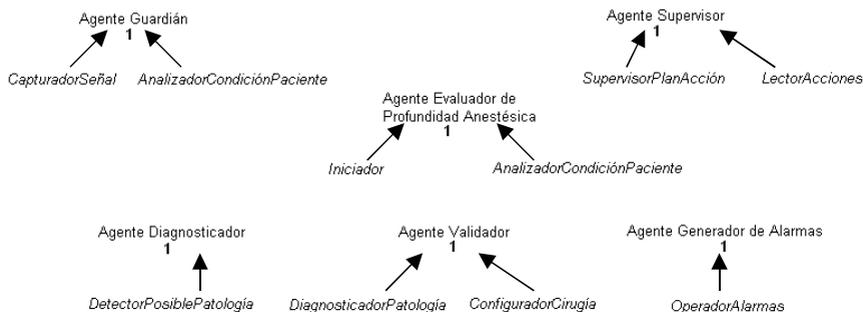
Rol: DiagnosticadorPatología
<p>Descripción:</p> <p>Realiza la actividad. Si determina que el paciente padece una patología real, pone en marcha al rol de OperadorAlarmas. Este rol permanecerá activo hasta que reciba la señal FinProceso</p>
<p>Protocolos y Actividades:</p> <p>EsperarActivaciónOseñalFinProceso, LeerSeñales, LeerPosiblePatología, Leer CaracterísticasPaciente, LeerTipoCirugía, DiagnosticarPatología, GuardarPatología, ActivarOperadorAlarmas, ActivarSupervisorPlanAcción, DesactivarDiagnosticadorPatología.</p>

Este rol realiza exclusivamente una sola actividad, de todo el proceso principal del Sistema. Esta plantilla especifica todas las propiedades del rol *DIAGNOSTICADORPATOLOGÍA*, estableciendo que todo Agente que realice este rol deberá ser capaz de:

- Acceder a las Bases de *PosiblePatología* y *Señales*, a las Bases de *Conocimiento Paciente*, *Cirugía* y *Patologías*.
- Comunicarse con los Agentes que implementen el rol *SupervisorPlanAccion* y *OperadorAlarmas*.

III.5. Modelo de Agentes

Una vez que se tiene el Modelo de Roles completo, es posible realizar el Modelo de Agentes. Un Agente puede realizar uno o más roles. La Fig. 5 muestra los diferentes tipos Agentes del Sistema y los roles que cada uno de ellos desempeñará.

Fig. 5. Modelo de Agentes

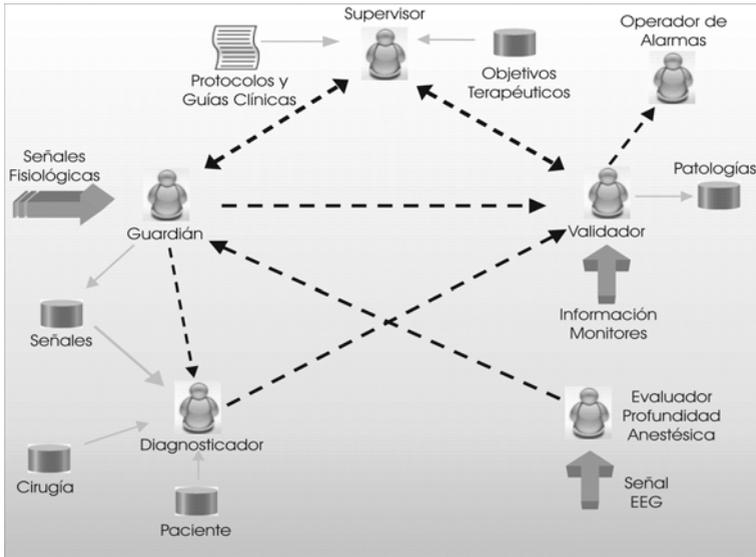
La **Fig. 5** muestra seis árboles distintos, donde las raíces representan los distintos Agentes que conforman el Sistema y sus hojas los roles que realizarán. El número que está debajo de la raíz indica el máximo número de Agentes en el Sistema, que podrán existir en un momento dado.

Como puede verse, algunos Agentes realizan más de un rol. Por ejemplo, el *Agente Supervisor* debe realizar el rol *SupervisarPlanAcción*. Este rol está muy relacionado con el de *LectorAcciones*, puesto que el plan de acción varía según las acciones del anestesiólogo. Por lo tanto, se considera apropiado que estos dos roles sean realizados por el mismo Agente.

A su vez, el rol *LectorAcciones* requiere que el Agente que lo implemente esté interrelacionado con todos los Agentes del Sistema, debido a que cuando el anestesiólogo indica la finalización de la cirugía, el Agente debe enviar a todos los demás Agentes la señal *FinProceso*. Aprovechando esta circunstancia, se consideró apropiado que este Agente también realice el rol *Iniciador*, puesto que también requiere de dichas interrelaciones, para indicar el comienzo de la fase de seguimiento y soporte de la cirugía.

La **Fig. 6** muestra los Agentes del Sistema, sus relaciones interagentes y los recursos que utilizan. Esta información se obtiene del Modelo de Roles.

Fig. 6. Arquitectura Multiagentes



El Agente Guardián, como tiene el rol *CapturadorSeñales*, recibe señales desde los monitores y las almacena en la Base Señales. También desempeña el rol de *AnalizadorCondiciónPaciente*, por lo que analiza las señales almacenadas en la Base Señales y guarda el resultado de ese análisis en la Base *Condición del Paciente*. Está relacionado con el Agente Supervisor para darle aviso cuando la condición del paciente cambia. También está relacionado con el Agente Diagnostificador para pedirle que detecte una posible patología, en caso de que el paciente esté en condiciones anormales.

El Agente Diagnostificador, una vez que recibe un pedido de ejecución, analiza la Base de Señales en búsqueda de posibles patologías. En caso de encontrar alguna, la almacena en la Base *Posibles Patologías*. Se relaciona con el Agente Validador para requerirle que implemente una verificación del diagnóstico, en caso de que se haya detectado una posible patología.

El Agente Validador, una vez que recibe un pedido de acción por

parte del Agente *Diagnosticador*, realiza una validación del diagnóstico utilizando información: de la Base *Patologías* (para leer las patologías encontradas en la cirugía), la Base de *Señales*, la Base de *Posibles Patologías*, la Base de *Conocimientos* (BC) del Paciente, la BC de la *Cirugía* y escribe en la Base *Patologías* la patología encontrada (en caso de haberla). Se relaciona con el Agente *Generador de Alarmas* para que active la alarma correspondiente.

Por último, el Agente *Supervisor* define y/o actualiza el plan de acción a sugerir al anestesiólogo, leyendo las Bases: *AccionesUsuario*, *CondiciónPaciente*, *BC Paciente*, *BC Cirugía* y *BC Procedimientos*. Como tiene el rol *Iniciador*, debe conocer a todos los Agentes del Sistema para iniciarlos.

Finalmente una vez confeccionados los diagramas de cada fase de la metodología, se utilizó el software de modelado Enterprise Architect, que es un programa que permite realizar el modelado gráfico de sistemas (en lenguaje UML) y brinda una aproximación hacia la codificación del sistema software.

De su empleo se obtuvo la codificación básica de la arquitectura en lenguaje JAVA, el cual también fue aplicado para la programación de los procesos internos de cada Agente.

III.6. Agente Diagnosticador

III.6.1. Diseño

Tiene por función identificar eventos anormales, a través de los datos provenientes del paciente, mediante un conjunto de Redes Neuronales entrenadas para tal fin. Es habilitado por el Agente *Guardián* y está formado por el siguiente conjunto de objetos que colaboran entre sí:

Temporizador: Teniendo en cuenta que la mayoría de los cambios fisiológicos ocurren en períodos de al menos 5 segundos, tiempo superior al período de muestreo de las señales provenientes de los monitores, se decidió adoptar una frecuencia de activación del Detector de Posible Patología de alrededor de 5 segundos (0,2 Hz), conduciendo esto a un ahorro computacional, aspecto importante en sistemas que trabajan en tiempo real. Cada vez que transcurre un período de tiempo igual al período de análisis, este objeto activa al *Coordinador* con el mensaje

ejecutar `Redes()`, indicándole que es tiempo de una nueva detección sobre los datos del paciente.

Coordinador: envía el mensaje `ejecutarRed()` a toda Red Neuronal de su lista de Redes, cada vez que el *Temporizador* se lo indica. También es capaz de reconfigurar el temporizador alterando el período de análisis.

Preprocesador: cada Red Neuronal específica tiene un preprocesador de datos, que permite obtener los índices derivados de las series de datos temporales. Por otro lado realiza la normalización de datos para llevarlos a un mismo rango de valores.

ConexionDB: cada una de las Redes tiene una conexión a la *Base Señales*, de la cual podrán obtener los datos necesarios para sus nodos de entrada, como así también guardar los resultados.

Redes Neuronales: el Agente posee una lista de Redes Neuronales especializadas. El Sistema, como parte de las actividades de inicialización, carga aquellas Redes que el anesthesiólogo considere adecuadas incluir. Todas ellas heredan las propiedades y los métodos de una *RedNeuronal* abstracta, que establece el único método al que responderán: `ejecutarRed()`. Todos los otros métodos son abstractos y se implementan por cada una de las Redes Neuronales específicas. Como propiedades principales de cada Red, se encuentran el conjunto de matrices y vectores que modelan sus nodos. Cada vez que una Red recibe el mensaje `enviarRed()`, si está activa, realiza las siguientes actividades: Carga las series temporales desde la *Base Señales*; Preprocesa los datos de entrada; Ejecuta la Red Neuronal, asignando los datos de entrada a cada uno de los nodos del vector de entrada y multiplicando por la matriz de pesos de cada Red, obteniendo el vector de salidas correspondiente y finalmente analiza los resultados y los guarda en la *Base Señales*.

Todas las Redes especializadas conforman una biblioteca de Redes Neuronales artificiales, entrenadas cada una para el reconocimiento de casos clínicos particulares. Estas Redes permiten hacer un análisis multivariado no lineal sobre distintos datos provenientes del paciente, y reconocer el tipo de patologías para las cuales fueron entrenadas. Las Redes utilizadas son del tipo *Perceptron Multicapas*.

La topología del conjunto de Redes Neuronales es heterogénea, debido a que cada una es específica para cada patología, buscando la

mejor *performance* en el reconocimiento de patrones. Todas, no obstante requieren como entrada valores numéricos de datos, agrupados en vectores de $1 \times N$, donde N depende de cada problema específico y representa el número de entradas de cada red. Los datos que conforman las entradas de cada Red pueden provenir de dispositivos diferentes. La fuente principal de estos datos la constituyeron los monitores multiparamétricos, que brindan por un lado índices fisiológicos numéricos y por el otro, series temporales que representan señales vitales. El detector implementado utiliza "on-line" ambos tipos de datos.

Para emplear señales dadas por series temporales (ECG, EEG, etc.) como entradas de Redes Neuronales, es conveniente implementar una reducción dimensional como requisito previo. La complejidad de esta tarea depende de las características de cada señal, por lo que se realizó el siguiente análisis: la señal de ECG se puede caracterizar a través de segmentos morfológicos característicos, como intervalo R-R, duración del segmento S-T, altura del complejo QRS, etc., mientras que la señal de EEG no presenta tales características morfológicas y se hace necesario recurrir a técnicas como transformadas de Fourier, análisis de caos, transformada *wavelets*, etc. Específicamente, en el SII se utilizaron los conceptos de Entropía en sus diferentes formas y el Análisis Espectral de la señal de EEG, a fin de inferir la profundidad anestésica.

En la literatura aparecen diferentes tratamientos para la reducción dimensional de señales fisiológicas empleadas por Redes Neuronales; en este caso, para la detección de Hemorragia Aguda, el detector desarrollado a partir de la señal de ECG determina la frecuencia cardíaca y utiliza este índice como una de las entradas.

No sólo es importante considerar una adecuada forma de representación de las series temporales, sino también la calidad del proceso de adquisición de las mismas. Entre los factores que condicionan este proceso se encuentran: tipo de sensores o método de registro empleados para obtener las señales vitales, sistema de adquisición (resolución de los conversores A/D, frecuencia de muestreo, ancho de banda de las señales, etc.), medio donde se realizan las pruebas (que influencia la relación señal ruido de los registros), algoritmo de detección morfológico, etc.

Estas características influyen en los resultados de las Redes Neuronales debido al proceso de aprendizaje que éstas poseen. La red *aprende* o asimila las características del patrón a detectar, a través de la presentación sucesiva de datos de entrada. Así, por ejemplo, si durante el entrenamiento de la Red todas las muestras de una subclase particular están contaminadas por niveles significativos de ruido de características similares, la red aprenderá ciegamente tales características. Al emplear más tarde la red, si el patrón aprendido por error no aparece junto al patrón real, la red puede fallar y no detectarlo. Esto hace indispensable que las características de los datos empleados durante la fase de entrenamiento sean similares a las que mostrarán los datos presentados durante la utilización de la red.

III.6.2. Aplicación al Caso de Referencia

No existe una única variable que evidencie la presencia de una Hemorragia Aguda; estas condiciones del paciente generan la desviación anormal de varios parámetros de manera conjunta. Los monitores convencionales no están preparados para una detección directa de la Hemorragia Aguda y generalmente producen alarmas simultáneas, alertando acerca de las diferentes situaciones de manera independiente. Esto, si bien determina el alerta del profesional médico, no le informa de la causa real del problema, y le puede producir confusión. Además, dado que la Hemorragia Aguda es un caso crítico y ocasional, los profesionales médicos pueden afrontar problemas muy difíciles frente a este cuadro.

A continuación se describen los resultados obtenidos con Redes Neuronales, sobre el caso de referencia Hemorragias Agudas.

Con los datos obtenidos para diferentes casos de HA, 100ml/min, 300 ml/min, 500 ml/min, 700ml/min y 900ml/min, se conformaron los patrones para el entrenamiento y testeo de Redes Neuronales del tipo Perceptrón Multicapa.

Los datos que conformaron los patrones, corresponden a las siguientes variables medidas: Frecuencia Cardíaca, Presión Sistólica, Presión Diastólica, Volumen Eyectado, Saturación de Oxígeno, Saturación de Dióxido Carbónico y Temperatura.

En un primer caso, los parámetros corresponden a un paciente en

condiciones normales y que aproximadamente a los 18 minutos de registro, comienza con una hemorragia de 100 ml/min.

En un segundo caso; se analizó una situación similar, pero ahora el paciente luego de tres minutos de registro, comienza con una Hemorragia de 700 ml/min.

Se destaca que para el primer caso (hemorragia de 100 ml/min), la simulación es de más de treinta minutos, en tanto que la segunda (hemorragia de 700 ml/min) dura poco más de seis minutos.

Para entrenar las Redes se utilizaron los siguientes datos: Paciente sin Hemorragia (229 datos), Paciente con Hemorragia de 100 ml/min (106 datos), de 300 ml/min (35 datos), de 500 ml/min (25 datos), de 700 ml/min (15 datos) y de 900 ml/min (19 datos) y sus derivadas. De tal forma es posible tener en cuenta las tendencias de cada variable y permitir anticipar las situaciones críticas, aspecto este de suma importancia, teniendo en cuenta que se trata de situaciones donde el profesional dispone de un tiempo muy reducido para tomar las decisiones.

Como consecuencia de este entrenamiento, la capa de entrada de la Red posee 16 neuronas. La capa intermedia quedó conformada con 12 neuronas con función de activación logística y la capa de salida con una neurona con función logística. La red se entrenó mediante el método de Backpropagation y se utilizó un valor 0.5 tanto para el parámetro de Aprendizaje, como el de Momento.

Por último, en lugar de colocar la derivada entre los patrones, se trabajó en forma más cualitativa, incluyendo solamente su signo, un uno si la derivada era mayor o igual que cero o un cero si la derivada era menor que cero.

Los mejores resultados se obtuvieron para la segunda configuración y aparecen en la **Tabla I**.

Tabla I

	Porcentaje de Aciertos -Prueba-
Hemorragia Sí / No	98.57%
Clasificación 97.83%	
No hay Hemorragia	95.12%
Hemorragia de 100 ml/min	97.78%
Hemorragia de 300 ml/min	100.00%
Hemorragia de 500 ml/min	96.88%
Hemorragia de 700 ml/min	100.00%
Hemorragia de 900 ml/min	97.22%

En una nueva etapa y para evaluar el comportamiento de la Red frente a situaciones que pudiesen confundirse con Hemorragia, se simularon situaciones de pacientes con cuadros de Hipotensión: Hipotensión leve (HL), Hipotensión media (HM); e Hipotensión severa (HS). La cantidad de datos obtenida para cada caso son las siguientes: HL: 270 datos, HM: 360 datos; HS: 210 datos.

Se formó entonces un nuevo conjunto de entrenamiento que poseía datos de pacientes no anestesiados, sin Hemorragia, con Hemorragias de 100, 300, 500, 700, 900 ml/min (los mismos utilizados en los entrenamientos anteriores) y datos de pacientes con Hipotensión leve, media y severa. Se utilizaron 180 patrones de cada grupo.

El 80% de estos datos se utilizaron para el entrenamiento de la red, en tanto que el 20% restante se utilizó con fines de prueba. Esta Red quedó conformada por 8 neuronas en la capa de entrada con función de activación lineal, 12 neuronas en la capa intermedia con función de activación logística, y una neurona en la capa de salida, también con función de activación logística. Nuevamente la red se entrenó mediante el método de Backpropagation y tanto el parámetro de aprendizaje, como el término de momento utilizados fueron de 0.5. Los resultados obtenidos para este caso se presentan en la **Tabla II**. Se puede observar que el porcentaje de efectividad de la red sigue siendo muy alto.

Tabla II

	Porcentaje de Aciertos -Prueba-
Hemorragia Si / No	98.74%
Clasificación	96.23%
No hay Hemorragia	92.31%
Hemorragia de 100 ml/min	100.00%
Hemorragia de 300 ml/min	92.11%
Hemorragia de 500 ml/min	98.00%
Hemorragia de 700 ml/min	97.00%
Hemorragia de 900 ml/min	97.00%
Hipotensión (Moderada,Media,Severa)	96.00%

Ya demostrada la capacidad de la red para reconocer situaciones de Hemorragia en pacientes no anestesiados, finalmente se evaluó su capacidad para reconocer situaciones de Hemorragia en pacientes anestesiados con fentanilo 100 mg, pancuronio 1 mg, tiopental sodico 280 mg, y suxinilcolina 100 mg.

La simulación se realizó para pacientes anestesiados sin Hemorragia y para pacientes anestesiados con Hemorragias de 100 ml, 500 ml y 900 ml. La cantidad de datos para este caso se distribuyen de la siguiente forma: Pacientes sin Hemorragia (330 datos), Pacientes con Hemorragia de 100 ml/min (660 datos), de 500 ml/min (210 datos), de 900 ml/min (150 datos).

Se formó para este caso un conjunto de entrenamiento que poseía 150 patrones de cada grupo.

El 80% de estos datos se utilizaron para el entrenamiento de la red, en tanto que el 20% restante se utilizó con fines de prueba. La Red quedó conformada por 8 neuronas en la capa de entrada, con función de activación lineal, 12 neuronas en la capa intermedia con función de activación logística, y una neurona en la capa de salida, también con función de activación logística.

Por último se agregó al conjunto de entrenamiento, datos de personas que sufren un cuadro de Hipotensión, por tener sintomatología

similar, el 80% de estos datos se utilizaron para el entrenamiento de la red, en tanto que el 20% restante se utilizó con fines de prueba. Los resultados se presentan en la Tabla III.

Tabla III

	Porcentaje de Aciertos -Prueba-
Hemorragia Si / No	99.00%
Clasificación	98.00%
No hay Hemorragia	97.00%
Hemorragia de 100 ml/min	100.00%
Hemorragia de 500 ml/min	99.00%
Hemorragia de 900 ml/min	98.00%
Hipotensión (Moderada, Media, Severa)	98.00%

Si bien la *performance* de las redes entrenadas con datos de pacientes bajo anestesia pueden parecer sorprendentes, ya que se registra un muy bajo error en la clasificación, este elevado porcentaje de aciertos de las Redes Neuronales Perceptrón Multicapa en pacientes anestesiados se debe a la disminución de la actividad de los diferentes mecanismos fisiológicos reguladores de las variables monitoreadas. Esto puede interpretarse como una disminución del ruido en la señal monitoreada, lo que facilita la tarea de clasificación de estados del paciente por parte de las Redes Neuronales. En consecuencia, la fase de mantenimiento resulta un contexto adecuado para el uso de Redes Neuronales, en la detección de situaciones complejas durante una cirugía.

Los resultados obtenidos en esta primer etapa son buenos, pues las Redes Neuronales entrenadas no solo fueron capaces de diagnosticar una HA, sino también de clasificarlas de acuerdo a la cantidad de flujo de sangre en pérdida, con un alto porcentaje de aciertos.

III.7. Agente Guardián

La función del Agente *Guardián* (AG) se definió como: "poner en funcionamiento los Agentes que tienen relación con funciones de

diagnóstico dentro del Sistema, sólo cuando existan indicios de un evento no esperado o una tendencia que pueda alterar la fase de mantenimiento, respecto del estado de referencia elegido por el anesestesiólogo". La idea motivadora para la inclusión de un Agente de este tipo es que el sistema opere de forma eficiente, realizando las funciones de diagnóstico sólo ante la existencia de indicios de que el proceso tienda a irse fuera de control durante la etapa de mantenimiento y como una forma de reducir las falsas alarmas.

Los Agentes relacionados con la tarea de diagnóstico son el Agente *Diagnosticador*, el Agente *Validador* y el Agente *Supervisor*.

Una problemática presente en diferentes ámbitos la constituyen los distintos criterios sobre lo que se considera "normal", como para poder formalizarlo e incorporarlo a un sistema como éste. Sin embargo, es usual en el ámbito anestesiológico tener como referencia el comienzo del proceso o el comienzo de la etapa de mantenimiento, una vez que los parámetros sensados del paciente se encuentran estabilizados. Teniendo en cuenta tanto la funcionalidad planteada para este Agente como las consideraciones antes mencionadas, es que se propone construir el modelo para el AG de forma tal que capture la referencia al comienzo de la etapa de mantenimiento, para cada paciente y proceso en particular. Por lo tanto, la función del AG será indicar, lo más tempranamente posible, algún indicio de que el paciente se aleje de dicha referencia, dejando la función de especificar el motivo de dicho alejamiento a los demás Agentes. En relación a esto último, se considera apropiado utilizar las herramientas disponibles del Control Estadístico de Procesos Multivariados. Estas herramientas permiten construir un modelo estadístico que refleje la interacción de las variables a partir de los datos adquiridos para el paciente y el proceso anestesiológico monitoreado.

Un inconveniente muy frecuentemente señalado en la literatura del tema es la interpretación de señales de fuera de control, ya que en cualquiera de los procedimientos multivariados descriptos es bastante dificultosa. El problema más importante que surge es la identificación de la(s) variable(s) responsable(s) de esa señal. En el caso del AG, este inconveniente no afectó su diseño, ya que la tarea de identificar la/s señal/es responsables está a cargo del Agente *Diagnosticador* y el Agente *Validador*.

El estadístico T^2 está basado en la observación más reciente, siendo insensible a cambios pequeños o moderados en el vector de medias del proceso. El principal problema práctico que surge con estos gráficos es su falta de robustez, ya que son sensibles a los *outliers* multivariados. Éstos son observaciones (X) que tienen grandes valores de T^2 , aunque ninguna componente del vector sea considerada un outlier en un test univariado. Para resolver este problema, una vez adquiridas un número de muestras N ($N > 30$), se procedió a filtrar los *outliers* multivariados utilizando la estadística Chi-cuadrado.

Los métodos MEWMA y MCUSUM tienen como ventaja que van considerando las observaciones anteriores, lo que a priori permite detectar cambios pequeños o moderados. Sin embargo, en relación a T^2 , requieren definir parámetros de diseño, que no se obtienen directamente de los datos. Teniendo en cuenta todas estas consideraciones para el diseño del AG, se consideraron los métodos T^2 de Hotelling y MEWMA con filtrado de outliers y transformación.

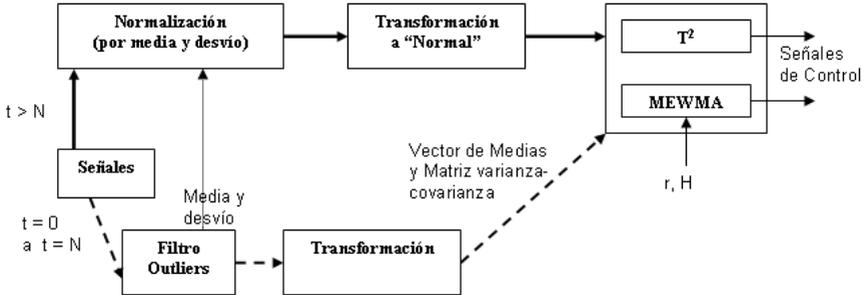
III.7.1. Diagrama de bloques del Agente

En función del análisis del dominio del problema, de la funcionalidad del AG y de las metodologías, se propuso el diagrama de bloques de la **Fig. 8**.

III.7.1.1.- Filtro *Outliers*

Los datos adquiridos durante el comienzo de la etapa de mantenimiento, pueden contener una o más observaciones multivariable inusuales o que no pertenecen al patrón de variabilidad producido por las otras observaciones, a las que se denominan *outliers*.

Fig. 7. Diagrama de Bloques del Agente Guardián



Dentro del AG se deben detectar los *outliers* automáticamente, luego de adquiridas las observaciones. En este sentido, un valor "grande" de la distancia estadística es un indicador de que una observación es un *outlier* multivariable, definiendo distancia estadística "grande" mediante un percentil apropiado de la distribución Chi-Cuadrado, con p grados de libertad. En nuestro caso, p es el número de variables y el percentil apropiado está determinado por el nivel de confianza *alfa* utilizado. En función de lo antes mencionado, el filtro es aplicado iterativamente:

$$d^2 = (x_j - \bar{x})' \bar{S} (x_j - \bar{x}) < \chi_p^2(\alpha) \text{ (FO)}$$

hasta que todas las observaciones a considerar para construir \bar{S} , \bar{x} , cumplen con la ecuación FO.

Los métodos descriptos para establecer si el proceso controlado tiende a irse fuera de control asumen que las observaciones poseen una distribución multivariada aproximadamente normal. En la mayor parte de las aplicaciones son suficientes análisis unidimensionales y/o bidimensionales. Además, en general, conjuntos de datos "patológicos" se distribuyen normalmente en bajas dimensiones, pero no normalmente en altas dimensiones.

Una alternativa para cumplir con esta condición es aplicar una transformación a las observaciones, para que la distribución de las mismas transformadas se aproxime a la distribución normal. Esta transformación

se incluye, dado que *a priori* no se sabe cuál será la distribución de los datos para cada caso. Lo que si se sabe *a priori* es que durante el proceso anestesiológico se intenta mantener los valores de las observaciones alrededor de algún valor y con una variabilidad propia de cada paciente. La transformación utiliza el comportamiento de expansión y/o compresión de funciones matemáticas, para modificar la distribución original de frecuencias, de manera que ésta se aproxime a la distribución normal, reduciendo la tasa de falsas alarmas.

III.7.2. Aplicación sobre el Caso de Referencia

Los casos utilizados para el desarrollo del AG consisten en secuencias temporales, en los cuales los parámetros están estables hasta que ocurren los siguientes eventos:

- Caso 1: Hemorragia Aguda estimada en 500 ml/min.
- Caso 2: Hemorragia Aguda estimada en 500 ml/min. simultáneamente con una Embolia
- Caso 3: Neumotórax y luego de un minuto una Hemorragia Aguda estimada en 500 ml/min.

Las señales adquiridas corresponden a las siguientes variables: Frecuencia cardíaca (HF en lpm.), Saturación parcial de Oxígeno (SPO2%), Dióxido de carbono (CO2%), Frecuencia respiratoria (RR en rpm), Temperatura (T en °C) y Presión invasiva (PA mmHg), de la cual a su vez se extrae las presiones sistólicas y diastólicas (PAS y PAD).

Los datos adquiridos, para las primeras N muestras de señal (se tomo N = 40), en general no presentaban *outliers*, salvo en el Caso 3, en el cual el filtro eliminó 2 muestras multivariantes, lo que no afectó significativamente los valores de media y desvío de las variables. En cuanto a los resultados de la transformación, cabe aclarar que la mayor parte de las variables tienen una distribución aproximadamente normal. Se aplicó el esquema descrito anteriormente, con tres niveles de ARL: 200, 500 y 1000 (con $\delta = 3$), a los tres casos 1, 2 y 3. En la **Tabla IV** se presentan los tiempos de detección para cada uno de los casos, para los diferentes ARL

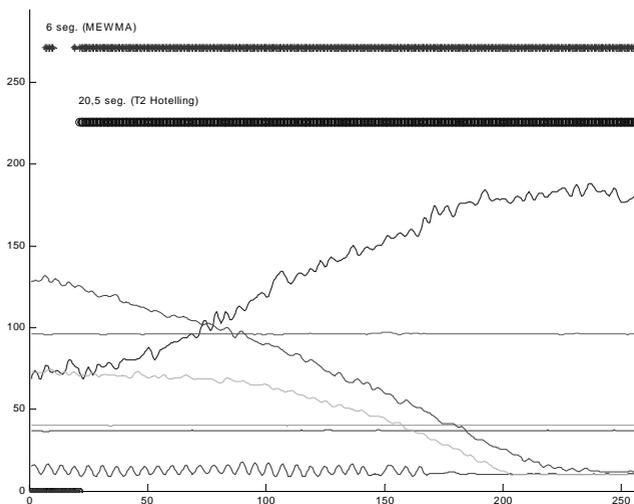
Tabla IV. Tiempos de detección (en segundos)

ARLo	200		500		1000	
	T2	MEWMA - $r = 0.4$	T2	MEWMA - $r = 0.46$	T2	MEWMA - $r = 0.43$
Caso 1	20.5	6	20.5	20.5	20.5	20.5
Caso 2	20	2	20	19.5	20.5	19.5
Caso 3	21.5	2.5	21.5	19.5	21.5	19.5

En la **Fig. 8** se observa la evolución de las señales del Caso 1 y las señales de control para el T² y MEWMA.

En los tres casos se observa que, para la condición de diseño de un nivel mínimo de falsas alarmas de 200 muestras (alfa = 1/200), la técnica MEWMA se anticipa a la detección del T² entre 14 y 18 seg. La detección mediante el estadístico T² ocurre aproximadamente en forma simultánea con el comienzo de signos claros de presencia de un evento crítico, como se observa en la **Fig. 8**, por lo que la detección con MEWMA permitió ganar tiempo para analizar las causas del evento, las cuales segundos más adelante se van a evidenciar más claramente.

Fig. 8. Evolución de las señales del Caso 1 y de las señales de control



III.8. Agente Validador

Para desarrollar el Agente Validador, se trabajó empleando el concepto de análisis de los datos desde diferentes niveles de interpretación. Este enfoque consiste en estudiar los datos de los que se obtiene información, desde diferentes puntos de vista, buscando inconsistencias en alguno de ellos. Se parte considerando los datos de manera aislada y se intenta determinar la validez de los mismos buscando detectar artefactos que hayan podido ser responsables de los hallazgos del Diagnosticador, o eliminando aquéllos que no hayan sido filtrados previamente, para reducir la posibilidad de falsas alarmas. En el nivel de detección de artefactos, se trata de establecer si los valores que se miden son válidos desde una interpretación fisiológica. El primer concepto que se emplea para determinar dicha validez es el de rango fisiológico admisible, que representa los umbrales de los índices vitales aceptados como compatibles con la vida. Así, el rango fisiológico permite inferir que una señal es inválida, si sus valores se encuentran fuera de ese límite.

Además, en este nivel de interpretación se hace un análisis de los datos en sus aspectos temporales, dado que aun cuando se encuentren dentro de los rangos fisiológicos, su dinámica puede no ser factible fisiológicamente. Para ello se consideran características morfológicas de la señal.

Los artefactos (componentes no deseados de la señal medida) pueden ser producto de ruido que interfiere con la señal o estar relacionados con distorsiones introducidas por los dispositivos de medición. En ambos casos se debe establecer si los diagnósticos efectuados no fueron sesgados, producto de estos componentes indeseados.

Para realizar estas tareas de validación, el Agente *Validador* dispone de tres módulos: el módulo de Validación de Lecturas, el módulo de Validación Contextual y el módulo Constructor de Alarmas y Advertencias. El primero se encarga de constatar la validez de los resultados del diagnóstico, analizando la calidad y posibles artefactos en las señales empleadas por el *Diagnosticador*. El segundo módulo determina la viabilidad del diagnóstico hallado, a la luz del contexto en el que se da el proceso anestesiológico (tipo de cirugía, drogas suministradas, estado

actual del paciente, probabilidad de ocurrencia de dicho evento, etc.), o si ello no es posible, una explicación de los mecanismos más probables, que podrían haber llevado al evento alarmable, mediante el modelado con Redes Bayesianas. Finalmente, el módulo Constructor de Alarmas y Advertencias es el encargado de elaborar los mensajes que este agente enviará al Agente *Generador de Alarmas*.

El riesgo de la presencia de ruido en las señales debido a una falla en los sensores, cables, interferencias externas o internas y fallas del equipo de monitoreo, se incrementa a medida que se utilizan más señales simultáneamente. El objetivo de este módulo es servir de filtro que reduzca la probabilidad de falsas alarmas, haciendo un análisis basado en conocimiento de las señales para establecer su validez.

El módulo propuesto para esta tarea consiste en una serie de procesadores de señales fisiológicas en paralelo, integrados a un Sistema Experto Borroso. Los procesadores fisiológicos toman los datos que vienen en forma de señales fisiológicas y las analizan extrayendo índices que caracterizan dichas señales. Estos índices y los datos que originalmente tenían la forma de índices, son representados en forma cualitativa. Entre los procesadores de señales fisiológicas se encuentran el detector de intervalos RR (a partir de la señal de ECG), detectores de presión arterial sistólica, presión arterial diastólica, presión arterial media y frecuencia de pulso (a partir de la señal de presión arterial invasiva) y la frecuencia de pulso (a partir del oxímetro de pulso). Además de estas características se extraen parámetros referentes a la dinámica de las señales y vinculados a sus tendencias.

IV. Conclusiones

Comenzando con los aspectos informáticos del desarrollo del Sistema de Información, se concluye que la elección de una Arquitectura Multi Agente, fue una decisión totalmente apropiada, por la modularidad y adaptabilidad que esta estructura ha permitido.

De tal forma, durante todo el transcurso del desarrollo del Sistema se han visto facilitados los cambios y modificaciones que necesariamente surgen en una implementación con tanta complejidad, como

consecuencia de la variabilidad de las situaciones, que requirieron, en algunos casos, de la verificación de diferentes metodologías.

Estos cambios en las metodologías de modelado de diferentes Agentes, se realizaron sin mayores complicaciones en la arquitectura general del Sistema desarrollado, dada la modularidad de la misma. Se destaca que estas modificaciones realizadas durante el avance del desarrollo, prácticamente tampoco complicaron lo desarrollado en los restantes Agentes, ni a la Lógica de Funcionamiento de los mismos, como así tampoco el proceso de intercomunicación entre Agentes.

Por otro lado, la utilización de la metodología Gaia permitió estructurar el desarrollo del Sistema, aplicando el concepto de roles para guiar el descubrimiento de funciones del Sistema, a través de la descripción de procesos y sus respectivas actividades, para alcanzar la meta. Esta herramienta resultó útil, al permitir obtener buenos resultados mediante la identificación de detalles que hubiese sido muy difícil determinar aplicando un modelo *ad-hoc*. Además, permite desarrollar una comprensión del Sistema y su estructura, sin hacer referencia a detalles de implementación. Por su parte, hizo posible que el desarrollo de cada Agente resuelva una problemática independiente, con aspectos de procesamiento de datos, metodologías y objetivos propios, siendo el Sistema Mutiagente lo que permitió la total integración y complementación de los resultados de cada Agente, que resultó en una consolidación de las conclusiones que muestra el Sistema al usuario.

Finalmente el diseño de las interfases gráficas, basadas en necesidades del usuario, presentan y toman en cuenta la forma habitual de trabajo de los usuarios. La interfaz apropiada consideró una eficiente visualización de las conclusiones, advertencias y alarmas generadas por el Sistema, teniendo en cuenta la variabilidad de las situaciones en desarrollo del Sistema Informático. Para ello se tuvieron en cuenta aspectos relevantes de los procesos anestesiológicos en sus diferentes fases (inducción, mantenimiento y despertar), más que en la resolución de una situación específica. También se analizó la eficiencia de la ejecución del Sistema, desde el punto de vista informático, tratando que la activación de algunos de los Agentes se produzca solamente cuando la situación así lo requiera. De tal forma se mantienen solamente activos,

en forma permanente, aquéllos que realizan un seguimiento del estado general del proceso anestesiológico. Sin embargo se ha tratado de anticipar, dentro de lo posible, las situaciones de emergencia, mediante la identificación de tendencias de desvíos no deseados.

Referencias bibliográficas

- BLOCK, F; SCHAAF C. (1996). Auditory alarms during anesthesia monitoring with an integrated monitoring system, en: **International Journal of Clinical Monitoring and Computing**, Vol 13, Nº 2: 81-84.
- COOK R., WOODS D. (1996). Adapting to new technology in the operating room, en: **Human Factors**, Vol 38, Nº 4:593-613.
- DE GRAAF, P., VAN DEN EIJKEL G., VULLINGS H., DE MOL B. (1997). A decision-driven design of a decision support system in anesthesia, en: **Artificial Intelligence in Medicine** Vol. 11, Nº 2:141-153.
- GRAHAM, A. (1998). US Food and Drug Administration: adverse event reports associated with anesthesia devices, en: **CRNA** Vol. 9 Nº 4:39-142.
- HAYKIN S.. (1999) **Neural Networks. A comprehensive Foundation**. New Jersey: Macmillan College Publ.Co.:50-111.
- KOSKI E., MAKIVIRTA A., SUKUVAARA T., KARI A. (1992). Frequency and reliability of alarms in the monitoring of cardiac postoperative patients, en: **International Journal of Clinical Monitoring and Computing**, Vol 7, Nº 1: 129-133.
- MYLREA K., ORR J., WESTENSKOW D. (1993). Integration of monitoring for intelligent alarms in anesthesia: neural networks, can they help?, en: **Journal of Clinical Monitoring**, Vol 9, Nº 1:31-37.
- ORR J., WESTENSKOW D. (1994). A breathing circuit alarm based on neural networks, en: **Journal of Clinical Monitoring**, Vol 10, Nº 2:101-109.
- OSHITA S., NAKAKIMURA K., SAKABE T. (1994). Hypertension Control During Anesthesia, en: **IEEE Engineering in Medicine and Biology**, Vol. 13, Nº5:667-670.
- RAU G., BECKER, K., KAUFMANN R., ZIMMERMANN H. (1995). Fuzzy logic and control: principal approach and potential applications in medicine, en: **Artificial Organs** Vol. 19, Nº 1:105-112.
- SCHECKE T., LANGEN M., POPP H., RAU G. KASMACHER H., KALFF G. (1994). Knowledge-based decision support for patient monitoring in cardioanesthesia, en: **International Journal of Clinical Monitoring and Computing**, Vol 9, Nº 1:1-11.
- WIKLUND L., HOK B., STAHL K, JORDEBY J. (1994). Postanesthesia monitoring revisited: frequency of true and false alarms from different monitoring devices, en: **Journal Clinical Anesthesia**, Vol. 6, Nº 3: 182-188.
- WOLF M., KEEL M., BUCHER H.; GEERING K., LEHAREINNGER Y., NIEDERER F. (1996). Improved monitoring of preterm infants by Fuzzy Logic, en: **Technol. Health Care**, Vol. 4, Nº 2: 193-201.
- WOOLDRIDGE M., JENNINGS N.R., KINNY D. (2000). The GAIA Methodology for Agent-Oriented Analysis and Design, en: **J. of Autonomous Agents and Multi-Agents Systems**. Vol 3(3): 285-312