Modelo híbrido para el diagnóstico de enfermedades cardiovasculares basado en inteligencia artificial

A hybrid model for the diagnosis of cardiovascular diseases based on artificial intelligence

Guillermo Roberto Salarte Martínez
Ingeniero de Sistemas, magister en Investigación de Operativa y Estadística, estudiante de Doctorado en Informática de la Universidad Pontificia de Salamanca. Docente investigador de la Universidad Tecnológica de Pereira. Pereira, Colombia. Contacto: *roberto@utp.edu.co*

Yanci Viviana Castro Bermúdez
Ingeniera de Sistemas y Computación. Investigadora de la Universidad Tecnológica de Pereira. Pereira, Colombia. Contacto: *yanci_ca@hotmail.com*

Fecha de recepción: 22 de julio de 2011 **Fecha de aceptación: 14 de febrero de 2012**

Clasificación del artículo: Investigación

Palabras clave: databases, diagnosis, disease, Bayesian networks.

Key words: bases de datos, diagnóstico, enfermedades, redes Bayesianas.

RESUMEN

La investigación presentada en el siguiente artículo está orientada hacia el área de la bioinformática, en el campo de minería de datos utilizando la técnica de redes bayesianas y arboles decisión; además de evaluar la utilidad de la metodología bayesiana en la predicción y el diagnóstico médico de enfermedades complejas (cardiovasculares), redes bayesianas se utilizan como representación grafica del conocimiento previo y métodos de razonamiento en los modelos probabilísticos y en la clasificación de los datos, pues desde la base datos todavía existen problemas; de esta manera, la estructura obtenida puede presentar un grado de complejidad innecesario que dificulta la representación e interpretación del conocimiento así como también la eficiencia del proceso de inferencia.

ABSTRACT

The present work pertains in the field of bioinformatics, particularly in the field data mining using Bayesian networks and decision trees. The study also assesses the usefulness of a Bayesian methodology when making medical predictions and diagnosis of non-trivial diseases (cardiovascular). Bayesian networks are used as graphic representations of previous knowledge, and also reasoning methods are applied to probabilistic models. When classifying the data from the database, problems still arise, thus the structure obtained might exhibit an unnecessary complexity degree which makes it difficult to represent and interpret knowledge as well as reducing efficiency in the inference process.

* * *

1. INTRODUCCIÓN

El aumento exponencial de información ha permitido que las organizaciones o empresas diseñen sus propios sistemas de información y estas sirvan de apoyo a la toma de decisiones, así entonces, su propósito es generar la información disponible y necesaria a los ejecutivos de alto nivel, además de brindar un acceso rápido y efectivo a la información critica del negocio, pero en la actualidad la demanda de la información va mas allá de una simple consulta de datos o de reportes consolidados. Como respuestas a dichas inquietudes, se han creado técnicas de almacenamiento y análisis de la información, logrando involucrar diversas áreas de conocimiento como la estadística, inteligencia artificial, computación grafica, [1] bases de datos y el procesamiento masivo, todas estas inquietudes han servido como fundamentos para nuevas técnicas de análisis como la *minería de datos*, que es el proceso de extraer información de los grandes volúmenes de datos, revelando conocimiento innovador a las organizaciones, permitiendo conocer de forma más detallada el comportamiento de variables.

El método de clasificación de las redes bayesianas se ve afectado, ya que a cada nodo de la estructura (grafo) le corresponde una variable que compone el dominio de aplicación aún cuando dicha variable no trascienda de manera directa sobre la tarea de clasificación, de esta manera, la estructura obtenida puede presentar un grado de complejidad innecesario que dificulta la representación e interpretación del conocimiento así como la eficiencia del proceso de inferencia. Por otro lado, las capacidades predictivas de las redes bayesianas están orientadas a pronosticar el valor de cualquiera de las variables pertenecientes al dominio de aplicación en lugar de intentar maximizar el poder clasificatorio. Para solucionar el problema de clasificación de la red bayesiana se utilizarán las ventajas que tienen los arboles de decisión en la clasificación de los datos y su representación grafica; lo que se pretende es hacer un método que combine las ventajas de las técnicas de inducción de los árboles de decisión (TDIDT – IDE.3) con las de las redes bayesianas. Este modelo de aprendizaje estará formado dos por etapas:

- La primera etapa consiste en la preselección de nodos y construcción de la red, es decir que, a partir de los datos que se encuentran en una base de datos, esta se encarga de la selección y clasificación de un subconjunto de nodos para mejorar la capacidad predictiva de la red.
- La segunda etapa consiste en la construcción de la red bayesiana [2] a partir del subconjunto de variables seleccionadas en la etapa previa, aplicando el método de aprendizaje de redes bayesianas (algoritmo de probabilidades). De esta manera se tiene:

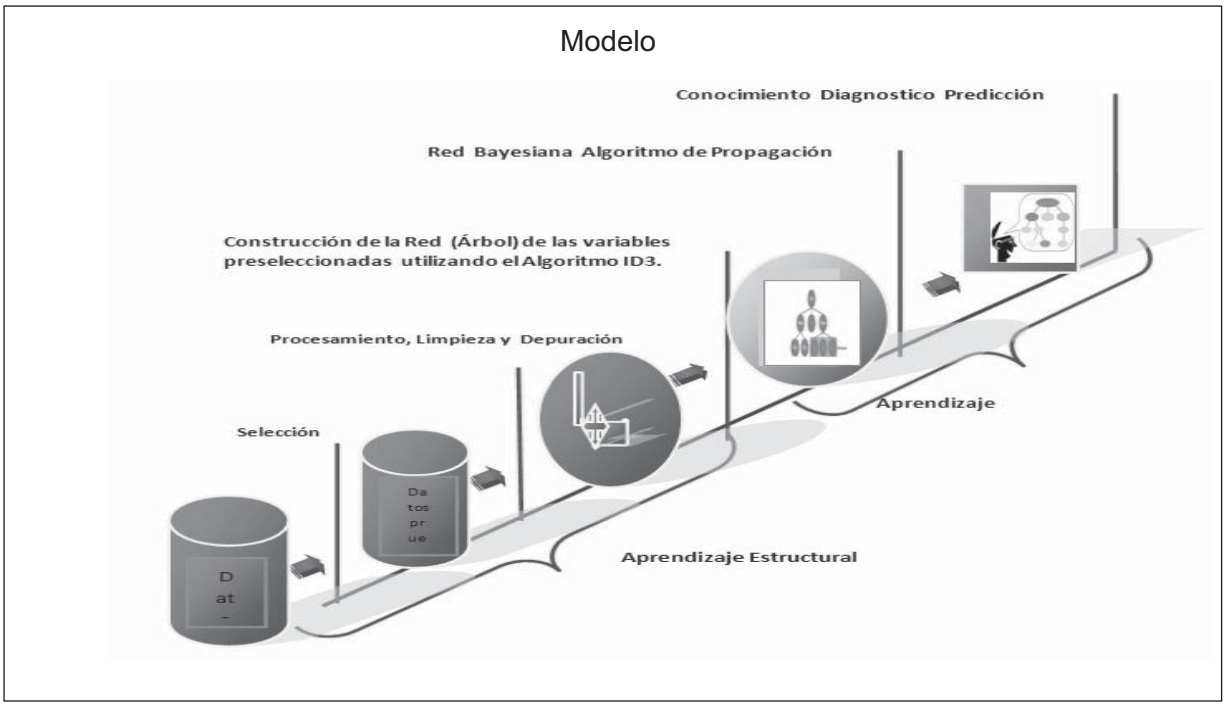

Figura 1. Modelo hibrido de inteligencia artificial.
Fuente: elaboración propia

2. REDES BAYESIANAS

Una red bayesiana es una representación gráfica (grafos dirigidos acíclicos) que contienen información probabilística para realizar un razonamiento probabilístico. Esta red está compuesta por nodos y arcos, donde los nodos representan variables aleatorias que pueden ser continuas o discretas, los arcos interpretan influencias causales, el que un nodo sea padre de otro implica que es causa directa del mismo.

a) Conceptos claves de redes bayesianas

- *Probabilidad a priori:* es la probabilidad de ausencia de evidencia que tiene una variable.
- *Probabilidad a posteriori:* es la probabilidad de una variable condicionada a la existencia de una evidencia determinada.

2.1 El aprendizaje en las redes Bayesianas

Se fundamenta en definir redes probabilísticas a partir de datos almacenados en bases de datos para la toma de decisiones, en lugar de utilizar la ayuda de un experto, por lo tanto, este tipo de almacenamiento ofrece la posibilidad de definir las estructuras graficas de la red a partir de los datos observados y permite definir las asociaciones entre los nodos o variables, teniendo en cuenta la información de cada nodo; según Pearl [3] existen dos fases de aprendizaje, a estas dos fases se les puede denominar respectivamente: aprendizaje estructural y aprendizaje paramétrico.

2.2 Aprendizaje estructural

Consiste en obtener la estructura de la red bayesiana [4] a partir de la base de datos, o sea, las

relaciones de dependencia e independencia entre las variables involucradas. Las técnicas de aprendizaje estructural dependen del tipo de organización de la red (árboles, poliárboles o redes multiconectada). Se trabaja con los arboles decisión.

2.2.1 Árboles de clasificación

Los árboles de clasificación, también conocidos como árboles de decisión Quinlan [7] y [8], son una técnica muy sencilla de aplicar y se puede utilizar para diferentes áreas como: reconocimiento de caracteres, señales, sensores sistemas expertos, diagnostico medico, juegos, predicción meteorológica y control de calidad.

2.2.2 Propiedades de los árboles de decisión

Una de las propiedades de esta técnica es que permite una organización eficiente de un conjunto de datos, debido a que los arboles son construidos a partir de la evaluación del primer nodo (raíz) y de acuerdo a su evaluación, o valor tomado, se va descendiendo en las ramas hasta llegar al final del camino (hojas del árbol), donde las hojas representan clases y el nodo raíz representa todos los patrones de entrenamiento que se han de dividir en clases. Los sistemas que implementan arboles de decisión tales como ID3, son muy utilizados en lo que se refiere a la extracción de reglas de dominio. Este método ID3, se construyen a partir del método de Hunt [5]. La heurística de Hunt, consiste escoger la característica más discriminante del conjunto X, para luego realizar divisiones recursivas del conjunto X en varios subconjuntos disyuntos de acuerdo a un atributo seleccionado. Para decidir qué atributo es el más apropiado a usar en cada nodo del árbol, se utiliza una propiedad estadísticas llamada ganancia de información, que mide cómo clasificar ese atributo a los ejemplos, es decir, elije el nodo del árbol que tenga mayor ganancia de información y luego expande sus ramas y sigue igual, pero considerando la nueva partición formada por el subconjunto de ejemplos que tienen ese valor para el atributo elegido.

3. METODOLOGÍA

Si se tiene un grupo de datos S donde contenga valores positivos o negativos sobre un concepto dicotómico en el estudio, para calcular la entropía de S relativa, su clasificación booleana se debe definir ($P = 1 - P_p$)

$P_{p.}$ es la probabildad de que las respuestas sean positivas según el conjunto S.

P_n; es la probabilidad de que las respuestas sean negativas según el conjunto S.

Si X son los datos de ejemplo de administrar tratamiento a los pacientes (ver tabla 1), se puede

Tabla 1. Administración de tratamiento

Paciente	Presión arterial	Gota	Hipotiroidismo	Administrar tratamiento
1	Alta	Si	No	No
2	Alta	Si	Si	No
3	Normal	Si	No	Si
4	Baja	Si	No	Si
5	Baja	No	No	Si
6	Baja	No	Si	No
7	Normal	No	Si	Si
8	Alta	Si	No	No
9	Alta	No	No	Si
10	Baja	No	No	Si
11	Alta	No	Si	Si
12	Normal	Si	Si	Si
13	Normal	No	No	Si
14	Baja	Si	Si	No

Fuente: elaboración propia

observar que de los 14 resultados, 9 tienen resultados positivos y 5 tienen resultados negativos, entonces, si se obtiene la probabilidad de cada resultado tenemos:

$$P_P = \frac{9}{14} = 0,6428 \quad y\ que \quad P_n = \frac{5}{14} = 0,3571$$

La entropía de X se define con base a las probabilidades anteriores, ecuación (1).

$$\mathrm{H}(\mathrm{S}) = -P_P log_2 P_P - P_n log_2 P_n \qquad (1)$$

Según la ecuación (1), la entropía del conjunto de 14 datos respecto a la variable de administrar fármacos se calcularía de la siguiente manera:

$$H(S) = \sum - P_i \log_2 (P_i) = -\frac{9}{14} = 0,6428$$

$$\log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.94028595$$

Si la entropía toma un valor de cero es cuando todos los miembros pertenecen a una misma clase ya sea negativa o positiva debido que $log\,(1)_2 = 0$.

Por tal motivo, la entropía se encuentra siempre un intervalo de cero y uno para los demás casos, alcanzando a un máximo cuando esta proporción es de 0,5 es decir existe una máxima aleatoriedad.

3.1 Concepto de ganancia de información

Como se dijo anteriormente, la entropía es una medida de desorden e impureza en un conjunto de datos de entrenamiento, pero se puede utilizar una medida eficiente para la clasificación de los datos, para ello se utiliza una medida llamada ganancia de información [6], esta medida reduce la entropía obtenida al realizar la división de los datos de entrenamiento.

$$\mathrm{H}(\mathrm{A},\mathrm{S}) \equiv \sum_{V \in Valores(A)} \frac{|S_v|}{|S|} = \mathrm{H}(\mathrm{S_v})$$

Donde S es un grupo de muestras clasificadas en C clases, A son los atributos y S_v es un subconjunto de S Valores (A) es una lista de posibles valores de A, la fórmula de ganancia de información se define como: $G(S,A) = H(S) - H(S,A)$

En la anterior expresión, el primer término corresponde a la entropía de la original de S, el segundo término corresponde al valor esperado de la entropía después de que S sea particionado de acuerdo al atributo A. Como se puede observar, el segundo término de la fórmula de ganancia no es más que la sumatoria de entropías de cada subconjunto S_v, ponderado por la fracción $\frac{|S_v|}{|S|}$

3.1.1 Tabla de contingencia

Para facilitar los cálculos se usan tablas de contingencia, esta técnica es muy útil para realizar cálculos probabilístico y cuando sus datos son categóricos. La tabla de contingencia que se obtiene, según los datos de la tabla 1, para la decisión de administrar medicamento, desde el punto de vista de presión arterial, seria: *Tabla de contingencia presión arterial,* tabla 2.

Tabla 2. Respuestas de presión arterial

PA	Alta	Normal	Baja	Total
Si	2	4	3	9
No	3	0	2	5
Total	5	4	5	14

Fuente: elaboración propia

Cálculo de entropía y ganancia de información con el atributo:

Presión Arterial

Se calcula la entropía de H(S)

Luego se procede a calcular la entropía de cada uno de los valores de A, es decir, la entropía de

presión arterial (alta, media y baja), utilizando la tabla de contingencia de la misma.

Presión arterial alta

$$H(S_{PA=alta}) = \sum_{i=1}^{c} - P_i \log_2(P_i) =$$

$$-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.94028595$$

Presión arterial media

$$H(S_{Pn=normal}) = \sum_{i=1}^{c} - P_i \log_2(P_i) =$$

$$-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0$$

Presión arterial baja

$$H(S_{Pb=baja}) = \sum_{i=1}^{c} - P_i \log_2(P_i) =$$

$$-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.94028595$$

Realizamos el cálculo de ganancia con el atributo.

$$G(S_{,pa}) = 0.94028 =$$

$$-\frac{5}{14}0.94028\ \frac{4}{14}0 - \frac{5}{14}0.9428 = 0.6779$$

De la misma manera, se realizan las tablas de contingencia para cada uno de los atributos de la tabla 1, igualmente se deben realizar los cálculos de entropía y de ganancia de información. Los resultados de ganancia de información de los atributos son los mostrados en la tabla 3:

Tabla 3. Ganancia de Información

Atributos	Ganancia de Información
Presión Arterial	G (S, PA)= 0.6779
Gota	G (S, GO) =0.511
Hipotiroidismo	G (S, HI) =0.0.48

Fuente: elaboración propia

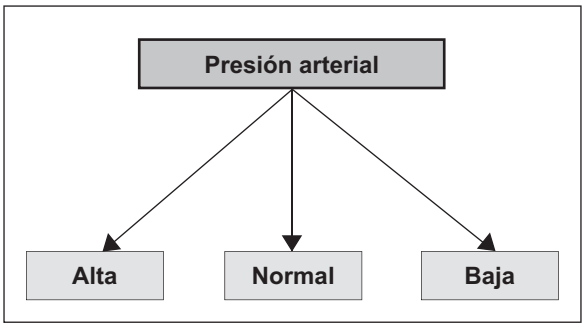

Figura 2. Selección del mejor atributo **"**Presión arterial"

Fuente: elaboración propia

3.1.2 Ensamblaje del árbol

Como se puede observar, el atributo que se debe seleccionar como nodo raíz es la presión arterial ya que, de acuerdo con la medida de ganancia de información, es el más adecuado para ser nodo inicial (raíz) creando así tres ramas.

A continuación se debe aplicar la misma técnica en cada uno de los nuevos nodos creados, pero en cada nodo creado sólo se usa un subconjunto de los datos como se observa en las siguientes tablas:

Tabla 4. Entrenamiento presión arterial alta

Paciente	Presión arterial	Urea en sangre	Gota	Hipotiroidismo	Administrar tratamiento
1	Alta	Alta	Si	No	No
2	Alta	Alta	Si	Si	No
8	Alta	Normal	Si	No	No
9	Alta	Baja	No	No	Si
11	Alta	Normal	No	Si	Si

Fuente: elaboración propia

Datos de entrenamiento presión arterial normal

Tabla 5. Entrenamiento presión arterial normal

Paciente	Presión arterial	Urea en sangre	Gota	Hipotiroi-dismo	Administrar tratamiento
3	Normal	Alta	Si	No	Si
7	Normal	Baja	No	Si	Si
12	Normal	Nor-mal	Si	Si	Si
13	Normal	Alta	No	No	Si

Fuente: elaboración propia

Datos de entrenamiento presión arterial baja

Tabla 6. Entrenamiento presión arterial baja

Paciente	Presión arterial	Urea en sangre	Gota	Hipotiroidismo	Administrar tratamiento
4	Baja	Normal	Si	No	Si
5	Baja	Baja	No	No	Si
6	Baja	Baja	No	Si	No
10	Baja	Normal	No	No	Si
14	Baja	Normal	Si	Si	No

Fuente: elaboración propia

Si se observa el comportamiento de los datos en la tablas, se puede deducir que el atributo presión arterial = normal, la recursión tiende a terminar, ya que en el atributo *decisión administrar fármacos* la mayoría de las variables son positivas, por lo tanto este atributo queda apuntado a una hoja llamada normal con un valor de *sí,* ejemplo "normal = si", sin embargo, las otras dos ramas restantes quedarán en evaluación recursiva, dividiendo el espacio de búsqueda y reduciendo el número de datos de entrenamiento. Igualmente, se realiza el mismo procedimiento recursivo con los demás atributos hasta formar el árbol.

3.2 Aprendizaje paramétrico

Dada una estructura y las bases de datos, obtiene las probabilidades a priori y condicionales requeridas. Uno de los principales trabajos en el campo del aprendizaje de redes bayesianas es el de Herkovits y Copper [9].

El aprendizaje paramétrico se fundamenta en descubrir los parámetros asociados a una estructura dada de una red bayesiana ó árbol de decisión, de acuerdo con los parámetros, lo que se pretende es encontrar las probabilidades a priori de los nodos raíz y las probabilidades condicionales de las demás variables debido a sus padres, si se conocen todas las variables es relativamente sencillo obtener las probabilidades requeridas, ya que las probabilidades previas corresponden a las marginales de los nodos raíz y las condicionales se obtienen del agrupamiento de cada nodo con su(s) padre(s). Para que se actualicen las probabilidades con cada caso observado, en el caso de un árbol las fórmulas de probabilidades previas son:

$$P\left(A_i\right)=\frac{\left(a_i+1\right)}{\left(s+1\right)};\quad i=k$$

$$P\left(A_i\right)=\frac{\left(a_i\right)}{\left(s+1\right)}\quad i\neq k$$

Las probabilidades condicionales

$$P\left(B_j\left|A_i\right.\right)=\left(b_j+1\right)\Big/\left(a_i+1\right)\quad i=k\quad j=1$$

$$P\left(B_j\left|A_i\right.\right)=\left(b_j\right)\Big/\left(a_i+1\right)\quad i=k\quad j\neq 1$$

$$P\left(B_j\left|A_i\right.\right)=\left(b_j\right)\Big/\left(a_i\right)\quad i\neq k$$

Donde:

- s corresponde al número de casos totales,
- i, j los índices de las variables, y
- k,l los índices de las variables observadas

3.3 Inferencia Bayesiana

Es un método de inferencia estadística que permite ingresar nuevas pruebas u observaciones para calcular las probabilidades que alcanzarán el resto de variables, por lo tanto, este método lo que hace es calcular las probabilidades a posteriori $P(XY|=y_i)$ de un conjunto de variables X después de obtener un conjunto de observaciones $Y = y_i$ donde Y es la lista de variables observadas e y_i es la lista correspondiente a los valores observados, también se utiliza para actualizar su probabilidad basada en un cálculo previo.

- Predicción

Si se tienen datos de un suceso anterior y este suceso se está representando en la red como un nodo padre, la red puede conjeturar cuáles serán sus efectos, para lograr esto se debe conservar esta hipótesis en el nodo correspondiente y difundir esta información hacia el resto de los nodos.

- Interpretación de datos

Otra manera de predicción es a través de las mismas relaciones que se muestran en la red [8], donde, conociendo las consecuencias, se puede saber cuáles son sus posibles causas. El conocimiento es el mismo que en el caso anterior: "*si a entonces b*" pero ahora el hecho conocido es "*b*" y el hecho desconocido es "*a*". Las probabilidades a posteriori $P(X|Y=y_i)$ se pueden conseguir desde la probabilidad marginal $P(X|Y)$, que a su vez puede obtenerse de la probabilidad conjunta $P(x_1, x_2,....,x_i)$ sumando los valores para todas las variables que no pertenezcan al conjunto $X \cup Y$.

3.3.1 Algoritmos de propagación en árboles

Cada nodo corresponde a una variable discreta $A=(A_1, A_2,....,A_n)$ con su respectiva matriz de probabilidad condicional $P(B|A) = P(B_j|A_i)$ dada cierta evidencia E (representada por la instanciación de ciertas variables) la probabilidad posterior de cualquier variable B es, por el teorema de Bayes:

$$P(B_i \mid E) = \frac{P(E \mid B_i)\,P(B_i)}{P(E)}$$

Debido a que la tipología de la red es un árbol, el nodo B se divide en dos subárboles, de este modo se puede separar la evidencia en dos grupos:

- E-: Datos en el árbol cuya raíz es B.
- E+: Datos en el resto del árbol.

Por lo tanto se tiene la ecuación (2)**.**

$$P(B_i \mid E) = \frac{P(E^-, E^+ \mid B_i)\,P(B_i)}{P(E)} \qquad (2)$$

Pero teniendo en cuenta las independencias de las dos variables, se aplica nuevamente el teorema de Bayes y se consigue: $P(B_i|E)=aP(B_i|E^+)(E^-|B_i)$ donde a es constante de normalización. De la anterior expresión se puede decir que ésta divide la evidencia para renovar la probabilidad de B, igualmente se observa que no se requiere de la probabilidad a priori, en caso de que el nodo no posea padres se obtiene que:

$$P(A_i \mid E^+) = P(A_i)$$

Para reducir el proceso se definen los siguientes términos que se ven en la ecuación (3).

$$\lambda(B_i) = P(E^- \mid B_i)$$
$$\pi(B_i) = P(B_i \mid E^+)$$

$$\lambda(B_i) = \prod_k P\left(E_k^- \middle| B_i\right) = \prod_k \lambda_k(B_i) \qquad (3)$$

Donde E_k pertenece a la evidencia que se origina del hijo k de B denotado por S_k. Condicionando cada término en la ecuación anterior, respecto de todos los posibles valores de cada nodo hijo, se obtiene que B es condicionalmente independiente de la evidencia bajo cada hijo por lo tanto se usa la siguiente expresión λ en la ecuación (4).

$$\lambda(B_i) = \prod_k \left| \sum_j P\left(E_k^- \middle| B_i, S_J^K\right) P\left(S_J^K \middle| B_i\right) \right. (4)$$

De la misma forma se puede tener una ecuación para π, primero se le condiciona sobre todos los posibles valores del padre, ecuación (5).

$$\pi(B_i) = \sum_j P\left(B_i \mid E^+, A_J\right) P\left(A_J \mid E^+\right) \quad (5)$$

Luego se puede eliminar E+ del primer término, dada la independencia condicional. El segundo término representa la probabilidad posterior de A sin contar la evidencia del subárbol de B, por lo que se puede expresar usando la ecuación para P(BJ | E) y la descomposición de λ en la ecuación (6).

$$\pi(B_i) = \sum_j P(B_i \mid A_J)\left[\alpha\pi(A_j)\prod_k \lambda_k(A_j)\right] \quad (6)$$

Donde k incluye a todos los hijos de A excepto B. Mediante estas ecuaciones se integra un algoritmo de propagación de probabilidades en árboles, donde cada nodo guarda los valores de los vectores π y λ así como las matrices de probabilidad *P,* la propagación se hace por un mecanismo de paso de mensajes en donde cada nodo envía los mensajes correspondientes a su padre e hijos:

Mensaje al padre (nodo B a su padre *A)*

$$\lambda(A_i) = \sum_j P(B_i \mid A_i)\lambda(B_J)$$

Mensaje a los hijos (nodo B a su hijo *k S)*

$$\pi_k(B_i) = \alpha\pi(B_i)\prod_{l\neq k} \lambda_l(B_j)$$

Al instanciarse ciertos nodos, estos envían mensajes a sus padres e hijos y se propagan hasta llegar a la raíz u hojas o hasta encontrar un nodo instanciado, así que la propagación se hace en un solo paso en un tiempo proporcional al diámetro de la red. Esto se puede hacer en forma iterativa instanciando ciertas variables, propagando su efecto y luego instanciando otras variables y propagando la nueva información combinando ambas evidencias.

3.3.2 Algoritmo de propagación de probabilidades

Se han desarrollado varios algoritmos para el cálculo de las probabilidades Neapolitan, Richard [10], de los cuales sólo se explicará en profundidad el algoritmo para redes con forma de árbol. El algoritmo consta de dos etapas:

- *Etapa de inicialización*: en esta etapa se consiguen las probabilidades a priori de todos los nodos de la red, obteniendo un estado inicial de la red que se denotará por S_0.

- *Etapa de actualización*: cuando una variable adquiere un valor nuevo, se actualiza el estado de la red (árbol) obteniendo las probabilidades a posteriori de.

Acorde a las variables del árbol basadas en la evidencia considerada, la red adopta un estado que se denotará por $S_{1.}$

Este proceso se repite cada vez que una variable tome un valor nuevo, obteniendo así los estados sucesivos de la red.

La idea básica del algoritmo de propagación de probabilidades es enviar mensajes a toda la red (árbol), cada vez que existe un cambio en cada uno de los nodos (variables), además se debe modificar sus probabilidades.

3.4 Fórmulas

Para el cálculo de λ y π-mensajes, λ y π-valores y probabilidades P* (esto es la probabilidad a posteriori dada una evidencia observada, (P*(x) = P(x/e)), se tienen las siguientes ecuaciones.

1. Si B es un hijo de A, B tiene k valores posibles y A *m* valores posibles, entonces para j=1,2,...,m, el λ-mensaje de B y A viene dado por la ecuación (7).

$$\lambda_B(a_j) = \sum_{i=1}^{k} P(b_i / a_j) \cdot \lambda(b_i) \qquad (7)$$

2. Si B es hijo de A y A tiene *m* valores posibles, entonces para j=1,2,...,m, el π-mensaje de A y B viene dado por la ecuación (8).

$$\pi_B(a_j) = \begin{cases} \pi(a_j) \prod_{\substack{C \in S(A) \\ C \neq B}} \lambda c \{ (a_j) \; si \; A \; no \; ha \; sido \; instanciada (*) \\ 1 \quad Si \quad A = a_j \\ 0 \quad Si \quad A \neq a_j \end{cases} \qquad (8)$$

Donde *s(A)* denota al conjunto de hijos de *A*. (*) esta fórmula es válida en todos los casos. Hay otra fórmula cuya aplicación resulta a veces más sencilla, pero que sólo es válida cuando todas las probabilidades P*(a_i) son no nulas, que es: P(a_j) / λB(a_j). Esta fórmula da un π-mensaje distinto (pero proporcional al de la otra fórmula) e iguales probabilidades a posteriori.

3. Si *B* tiene *k* valores posibles y *s(B)* es el conjunto de los hijos de *B*, entonces para

i=1,2,...,k, el λ-valor de *B* viene dado por la ecuación (9)

$$\lambda_b(b_i) = \begin{cases} \prod_{C \in S(B)} \lambda c(b_i) \quad si \quad B \; no \; ha \; sido \; instanciada (*) \\ 1 \quad Si \quad B = b_i \\ 0 \quad Si \quad B \neq b_i \end{cases} \qquad (9)$$

4. Si *A* es padre de *B*, *B* tiene *k* valores posibles y *A* tiene *m* valores posibles, entonces para *i=1,2,...,k*, el π-valor de *B* viene dado por la ecuación (10).

$$\pi(b_i) = \sum_{j=1}^{m} P(b_i / a_j) \pi_B(a_j) \qquad (10)$$

Si *B* es una variable con *k* posibles valores, entonces para *i = 1,2,...,k* la probabilidad a posteriori (P*) basada en las variables instanciadas se calcula como en la ecuación (11).

$$P*(b_i) = \alpha \lambda(b_i) \pi(b_j) \qquad (11)$$

Algoritmo de cálculo de probabilidades public doublé [ ][ ] Probabilidad (double b[ ][ ])

```
{  int i,j;
   for (i = 0; i < b.length; i++){
        for ( j = 0; j < b.length; j++) {
           if (i==1) {b[i][j]=b[i][j]/ps;}
           if (i==2) {b[i][j]=b[i][j]/pn;}
        } }        return b;  }
```

Código 1: Calcula Probabilidades

4. RESULTADOS

En esta etapa se tomará como ejemplo la tabla 1 *Administración de tratamiento*, esta tabla está formada por cinco variables médicas, además se tendrá en cuenta que sólo se trabaja con datos categóricos y discretos. Aplicación realizada en Wampserver 2.7 y N Netbeans 6.91.

A continuación se presenta la interface a la que el usuario accede para diagnosticar y realizar el tratamiento médico en pacientes con síntomas de enfermedad cardiovascular, mediante redes bayesianas. Ventana de inicio de la aplicación donde tiene como selección dos opciones: la primera, consulta general, la cual permite visualizar la

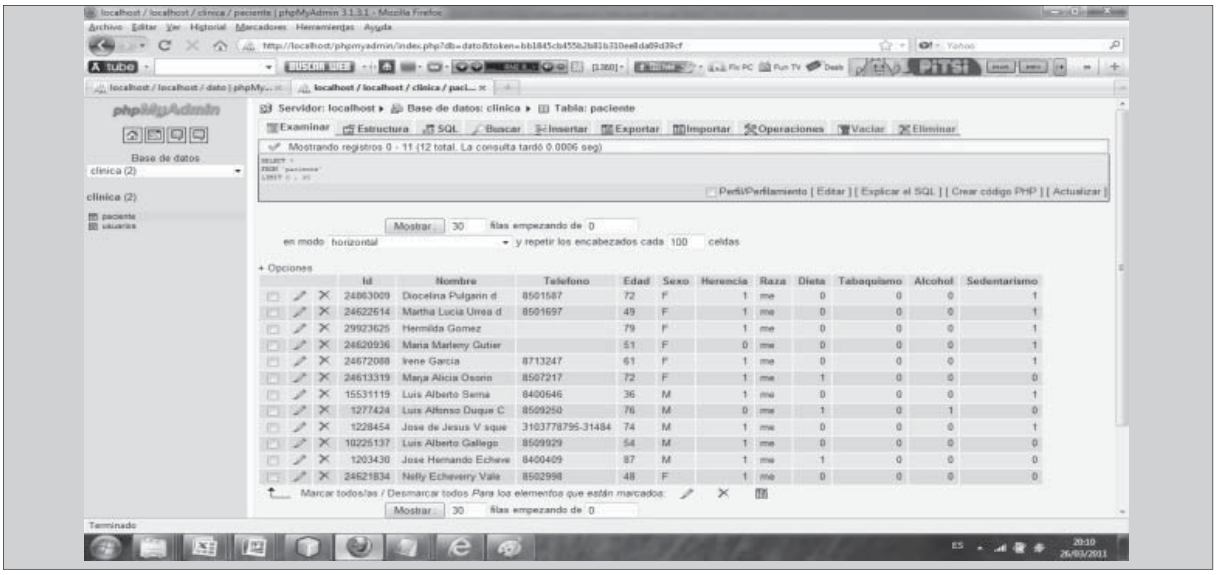

Figura 3. Base de datos en Wampserver [11].
Fuente: elaboración propia

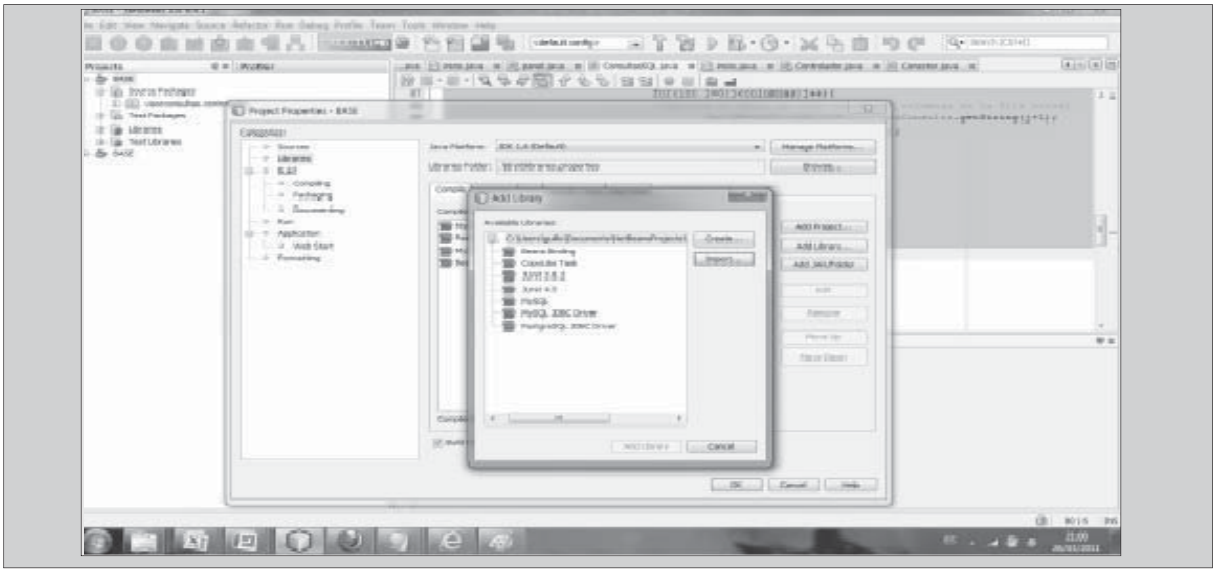

Figura 4. Netbeans 6.91 [11].
Fuente: elaboración propia

base de datos completa; la segunda opción, salir de la base de datos, ver la figura 5.

Ventana de presentación inicial, muestra el esquema de trabajo de este proyecto de investigación. La siguiente es la pantalla de presentación de entrada del aplicativo.

A continuación se muestran algunos de los mensajes que se pueden presentar cuando se trabaja con la aplicación. Ventana de conexión a la base de datos, esta ventana permite conectar una base de datos MySQL desde java.

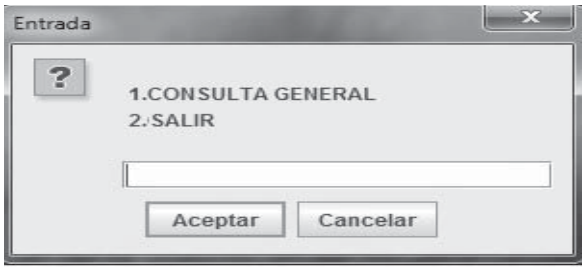

Figura 5. Ventana de Inicio del programa.
Fuente: elaboración propia

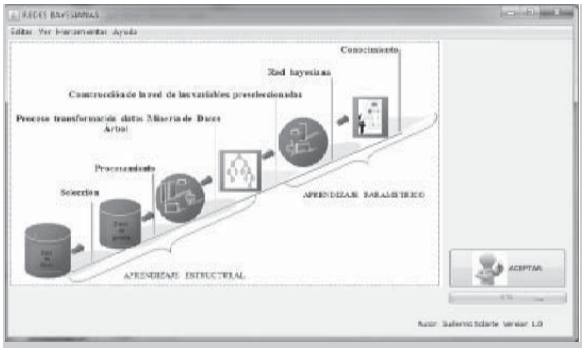

Figura 6. Ventana de presentación.
Fuente: elaboración propia

A continuación se presenta la ventana de trabajo y visualización. Esta ventana está compuesta por

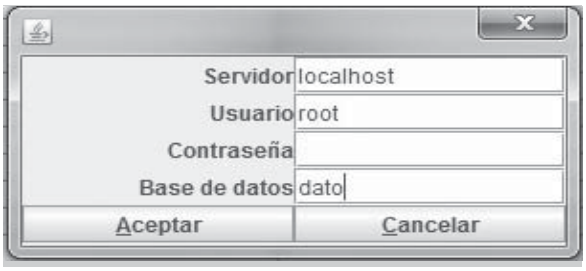

Figura 7. Ventana de Conexión.
Fuente: elaboración propia

Figura 8. Ventana de mensajes 1.
Fuente: elaboración propia

cuatro botones y un área de texto. El primer botón sirve para consultar en la base de datos, el segundo botón muestra el análisis de redes bayesianas, el tercero muestra la estructura de los datos y el último botón es para salir de la aplicación, ver la figura 9.

Figura 9. Ventana de trabajo y presentación de datos.
Fuente: elaboración propia

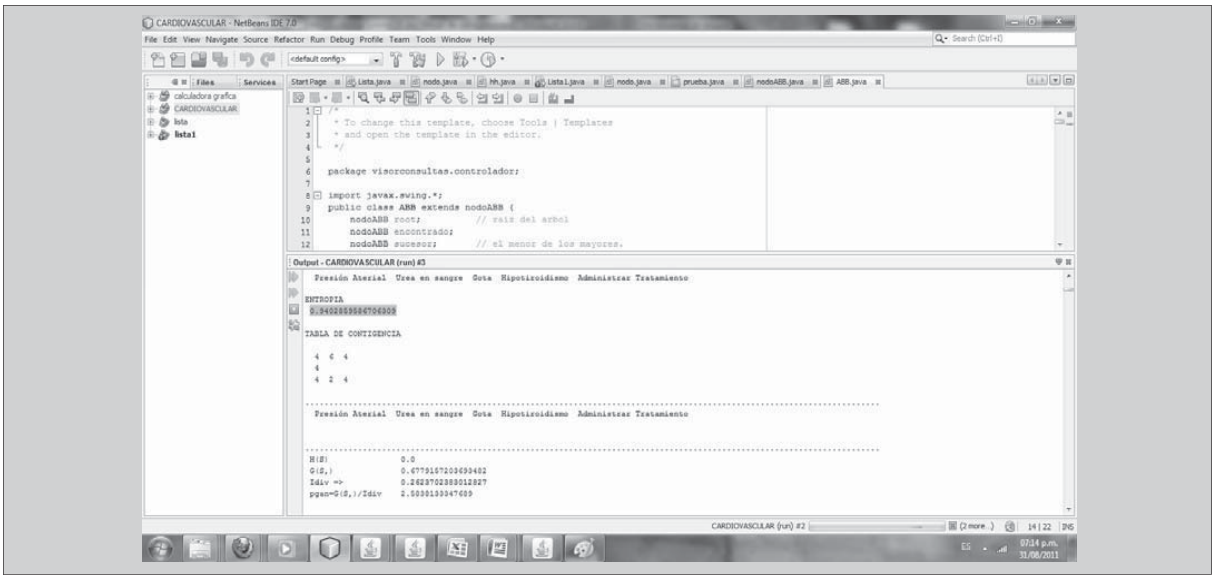

Figura 10. Ventana de trabajo y presentación de datos.
Fuente: elaboración propia

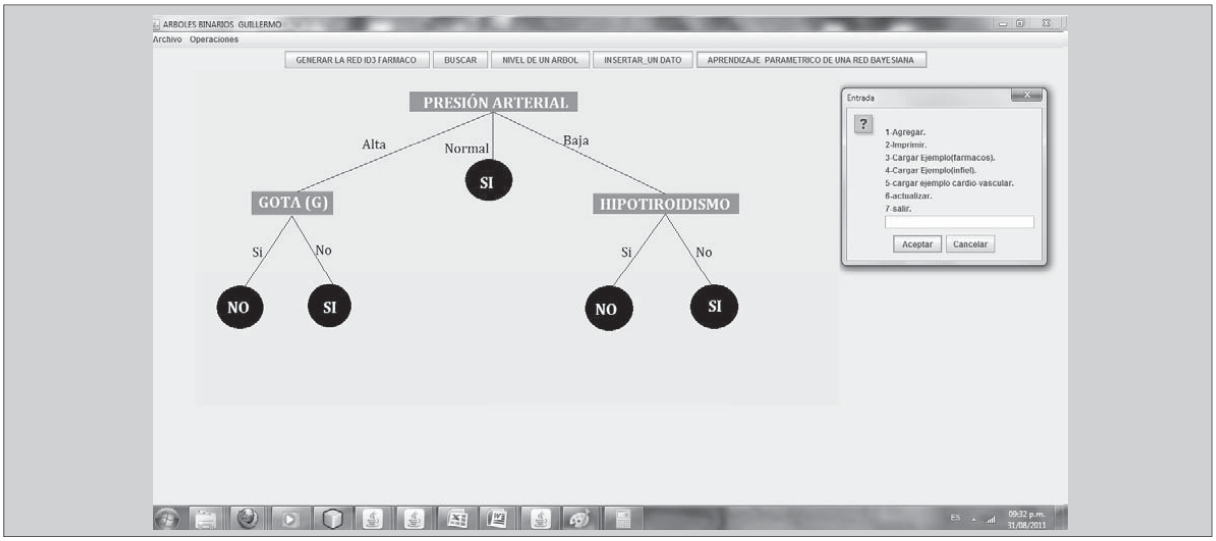

Figura 11. Ventana de trabajo y presentación de datos.

Fuente: elaboración propia

Esta ventana visualiza el árbol generado a partir de los datos que se encuentran en la base de datos, este caso está en Wapmserver Generación del árbol ID3. Ver la figura 11.

Resultados mostrados por aplicación Cálculo de ganancia de información

Tabla 7. Contingencia de presión arterial

PA	Alta	Normal	Baja	Total
Si	2	4	3	9
No	3	0	2	5
Total	5	4	5	14

Fuente: elaboración propia

(S,) 0.6779157203693482

Tabla 8. Contingencia Gota

US	Si	No	Total
Si	3	5	8
No	4	2	6
Total	7	7	14

Fuente: elaboración propia

G(S,) 0.5117145300992023

Tabla 9. Contingencia Hipotiroidismo

HI	Si	No	Total
Si	3	2	5
No	3	6	9
Total	6	8	14

Fuente: elaboración propia

G(S,) 0.0487145300992023

Tabla 10. Resultados de ganancia de información

Atributos	Ganancia de información
Presión Arterial	G (S, PA)= 0.6779
Gota	G (S, GO) =0.511
Hipotiroidismo	G (S, HI) =0.0.48

Fuente: elaboración propia

Se observa que los resultados de la tabla 10 de ganancia de información, el atributo que se debe seleccionar como nodo raíz es la *presión arterial* ya que, de acuerdo con la medida de ganancia de información, es el más adecuado para ser nodo inicial (raíz) creando así tres ramas: Alta, Normal y Baja. A continuación se debe aplicar la misma técnica recursiva en cada uno de los nuevos nodos creados.

Generación de la reglas

De acuerdo con la figura 12 se deduce que para administrar tratamiento a un paciente se deben tomar en cuenta las siguientes reglas generadas. Se puede administrar tratamiento un paciente

SI:

- Si tiene presión alta y no tiene gota
- Si tiene presión arterial normal
- Si tiene presión baja y no tiene Hipotiroidismo

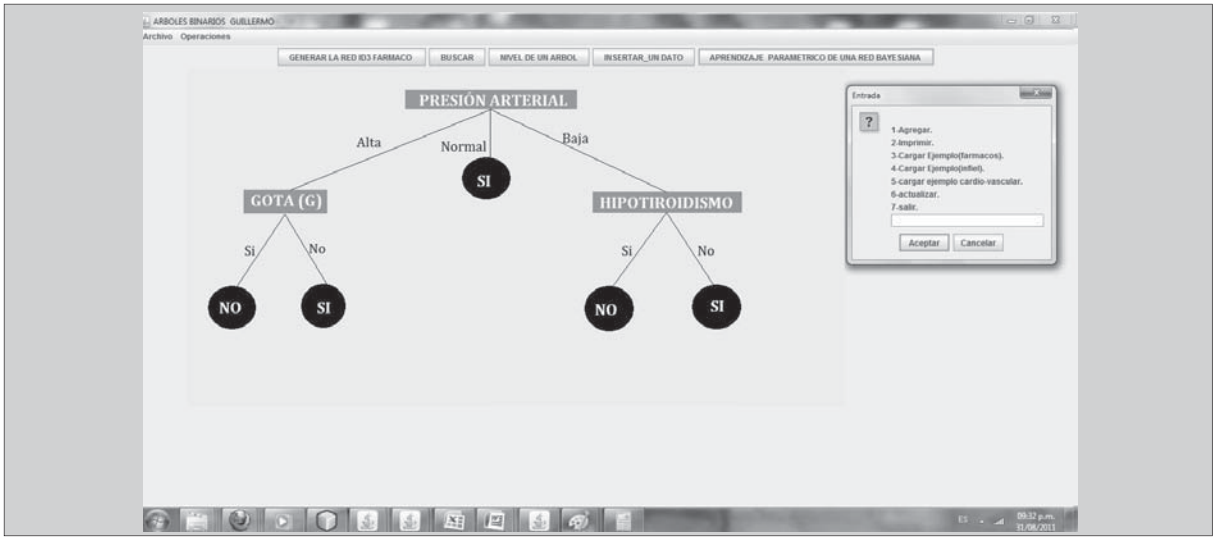

Figura 12. Generación de reglas con el algoritmo IDE3

Fuente: elaboración propia

NO:

- Si tiene presión alta y tiene gota.
- Si tiene presión baja y tiene Hipotiroidismo

A continuación se visualizan las probabilidades iníciales, generadas por la aplicación:

Probabilidad presión arterial

Tabla 11. Probabilidad presión arterial

Probabilidad	Alta	Normal	Baja
SI	0.38	0.31	0.31
NO	0.62	0.69	0.69

Fuente: elaboración propia

Tabla probabilidad de gota

Tabla 12. Probabilidad índice de gota

Probabilidad	SI	NO
SI	0.46	0.54
NO	0.54	0.46

Fuente: elaboración propia

Tabla probabilidad de Hipotiroidismo

Tabla 13. Probabilidad Hipotiroidismo

Probabilidad	SI	NO
SI	0.38	0.62
NO	0.62	0.38

Fuente: elaboración propia

Tabla 14. Probabilidad administrar tratamiento

Probabilidad	
SI	0.69
NO	0,31

Fuente: elaboración propia

A continuación, con el objeto de comparar los resultados obtenidos con el código Java arriba propuesto por el autor de este trabajo, se procesan los mismo datos con otra herramienta libre llamada la herramienta Elvira, esta herramientas tiene un conjunto de software open-source para investigación y desarrollo usando modelos gráficos de probabilidad. Para mayor información consultar en: http://www.ia.uned.es/~elvira/manual/manual.html

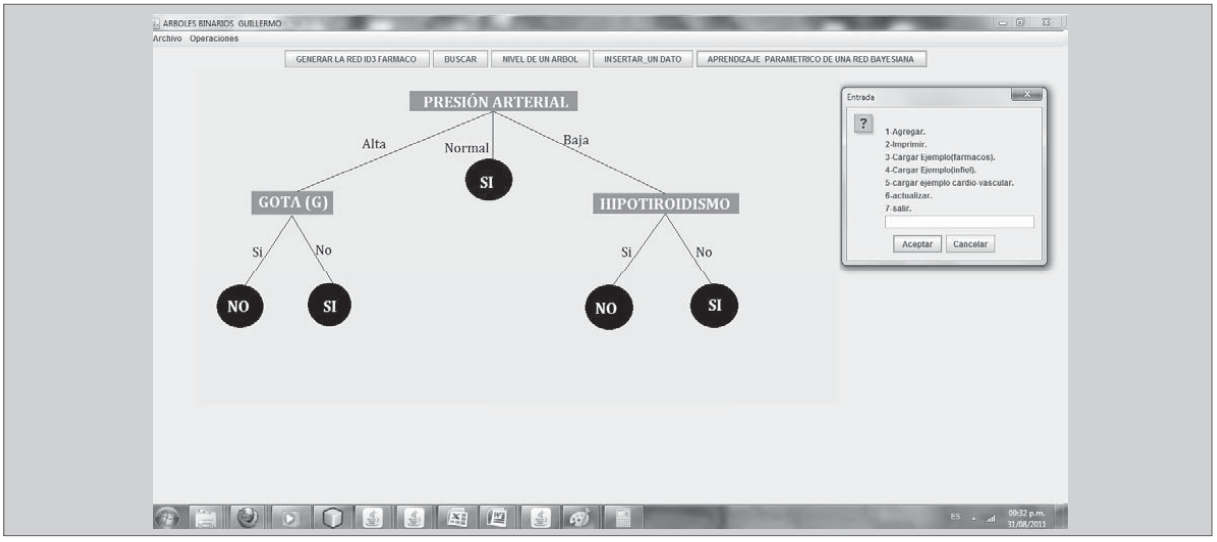

Figura 13. Ventana Elvira
Fuente: Elvira

Utilizando el algoritmo de propagación se calcula el estado inicial del árbol. Como los valores anteriores son los valores iniciales, ahora se van a recalcular los valores haciendo uso del algoritmo de propagación (Algoritmo de propagación hecho por el autor).

```
public void lamdaValor(nodoABB nodo){

if(nodo.instanciada){

nodo.lamda_valor[0] = nodo.probabilidad[0];

nodo.lamda_valor[1] = nodo.probabilidad[1];

}else{CalcLamdaValor(nodo,nodo.lamda_valor,0);}

}

public void CalcLamdaValor(nodoABB nodo,double []lamda_v,int flag){

if(nodo != null){

if(flag==0){lamda_v[0] = 1;          lamda_v[1] = 1; flag++;

                                CalcLamdaValor(nodo.izq,lamda_v,flag);          CalcLamdaValor(nodo.der,lamda_v,flag);

}                                       else{lamda_v[0] *= nodo.lamda_msg[0];    lamda_v[1] *= nodo.lamda_msg[1];

                                CalcLamdaValor(nodo.izq,lamda_v,flag);          CalcLamdaValor(nodo.der,lamda_v,flag);

} } }
```

Código 3: Calcula el lambda valor del nodo si este no ha sido instanciado.

```
public void lamdaMsg(nodoABB nodo){

if(nodo.padre!=null){ nodo.lamda_msg[0]=nodo.matriz[0]*nodo.lamda_valor[0]+nodo.matriz[2]*nodo.lamda_valor[1];

                                nodo.lamda_msg[1]=nodo.matriz[1]*nodo.lamda_valor[0]+nodo.matriz[3]*nodo.lamda_valor[1];

} }
```

Código 4: Calcula el lamda-msg del nodo que le enviemos

```
public void piMsg(nodoABB nodo){

if(!nodo.instanciada){      CalcPiMsg(nodo,nodo.pi_msg,0);

}else{ nodo.pi_msg[0] = nodo.probabilidad[0]; nodo.pi_msg[1] = nodo.probabilidad[1];

          }          }
```

Código 5: Función recursiva que calcula el pi_msg del nodo

Función de control elimina marcas utilizadas en la recursión.

```
public void EliminarMarca(nodoABB p){

if (p!=null){ p.marca = false;  EliminarMarca(p.izq); EliminarMarca(p.der);

  } }
```

Código 6: control elimina marcas utilizadas en la recursión.

Resultados del algoritmo de propagación hecho por el autor, como podemos ver los resultados muy similares con los resultados hecho con la herramientas Elvira.

Nodo ("A**dministrar Tratamiento**" id = 8){
Matriz: {0.71 | 0.0 | 0.29 | 0.0 |}

lamda_valor {1.0, 1.0} pi_valor {0.71, 0.29}
lamda_msg {1.0, 1.0} pi_msg {0.69, 0.31}

Probabilidad: **Verdad: 0.69 Falso: 0.31** }

Nodo ("**Presión Arterial**" id = 6){
Matriz: {0.4 | 0.1 | 0.6 | 0.9 | }

lamda_valor {1.0 ,1.0}pi_valor {0.38,0.62}
lamda_msg {1.0 , 1.0}pi_msg {0.38,0.62}

Probabilidad: **Verdad: 0.42 Falso: 0.58}**

Nodo ("**Hipotiroidismo** " id = 4){
Matriz: {0.5 | 0.4 | 0.5 | 0.6 | }

lamda_valor {1.0,1.0}pi_valor {0.44, 0.56}
lamda_msg {1.0, 1.0}pi_msg {0.44 , 0.56}

Probabilidad: **Verdad: 0.38 Falso: 0.62}**

Si se tiene una nueva evidencia de que un paciente tiene presión arterial alta y tiene gota, en este caso, se tomaría un valor de uno (1), las probabilidades se actualizan para contabilizar esta información.

A continuación se muestran los resultados obtenido con la aplicación hecha en java por el autor.

Nodo ("**Administrar tratamientos**" id = 8){
Matriz: {0.69 | 0.0 | 0.31 | 0.0 | }

lamda_valor {0.63, 0.168}
pi_valor {0.71, 0.29

lamda_msg {1.0 , 1.0 }
pi_msg {0.428 , 0.11}

Probabilidad**: Verdad: 0.31 Falso: 0.69** }

Nodo ("**Presión Arterial**" id = 6) {
Matriz: {0.4 | 0.1 | 0.6 | 0.9 | }

lamda_valor {0.6 , 0.4}
pi_valor {1.0 , 0.0}

lamda_msg {0.48 , 0.42000000000000004 }
pi_msg {0.313 , 0.687}

Probabilidad**: Verdad: 1.0 Falso: 0.0**}

Nodo ("**Gota**" id = 5){
Matriz: {0.4 | 0.4 | 0.6 | 0.6 | }

lamda_valor {1.0, 1.0}
pi_valor {0.622 , 0.38}

lamda_msg {1.0 , 1.0 }
pi_msg {0.62 , 0.38}

Probabilidad: **Verdad: 0.62 Falso: 0.38**}

Nodo ("**Hipotiroidismo "** id = 4){
Matriz: {0.5 | 0.4 | 0.5 | 0.6 | }

lamda_valor {1.0 ,1.0}
pi_valor {1.0 , 0.0}

lamda_msg {1.0 , 1.0}
pi_msg {0.24 , 0.36}

Probabilidad:**Verdad: 1.0 Falso: 0.0**}

Ahora, las probabilidades de los nodos cambian de la red. Esto es así porque la evidencia disponible de presión arterial y gota aumentó en 1, por lo tanto la probabilidad de administrar tratamiento disminuye, a la vez cambia las probabilidades de Hipotiroidismo. En este caso, la probabilidad de administrar tratamiento es baja. De igual forma se puede realizar para cada uno de los nodos de red ya que el algoritmo recorre todos los nodos hacia abajo y hacia arriba.

5. CONCLUSIONES

Se demostró que es posible determinar la administración, o no, de un procedimiento clínico a un paciente con síntomas de enfermedad cardio-

vascular, usando las variables tales como: presión arterial, gota, Hipotiroidismo; mediante la utilización del modelo híbrido, utilizando las ventajas de los arboles de decisiones en la clasificación de los datos y las probabilidades en la redes bayesianas.

La propuesta del algoritmo de propagación (inferencia de los datos) con la aplicación que se creó, genera resultados coherente con otras herramientas (Bayesian Netword tool, Elvira)

5.1 Validación e impacto del desarrollo de la propuesta

Esta propuesta aporta una serie de algoritmos que se pueden utilizar y mejorar para posibles aplicaciones en minería de datos.

Este trabajo deja una aplicación que se puede usar como apoyo para el personal médico en la toma de decisiones médicas, pero para datos discretos y variables de decisiones dicotómicas.

REFERENCIAS

[1] M. Campell, Base IV Guía de Autoenseñanza. España: Editorial McGraw Hill Interamericana (4 Mar 2009). 1990. pp110/111,121/122, 16, 169, 179-191/192.

[2] M. A. Torres Rios, "Analysis of Electrical Industrial Systems Using Probabilistic Networks", in G. A. Ramos, Member, IEEE, A. Torres, Senior Member, IEEE and M. A. Rios, Member, IEEE, *Latin America Transactions*, Vol. 8, No. 5, September 2010.

[3] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA, USA: Morgan Kaufmann Publishers, 1998

[4] A.C. Pifer e L. A., "Guedes. Aprendizagem Estrutural de Redes Bayesianas Utilizando Métrica MDL Modificada", in A.C. Pifer e L.A. Guedes, *IEEE Latin America Transactions*, Vol. 5, No. 8, December 2007.

[5] E.B. Hunt, J. Marin, P.J. Stone, *Experiments in Induction*. New York: Academic Press, 1966.

[6] R. Arias Montoya, (octubre de 2010), Detección Temprana De Fallas En La Red De Internet Banda Ancha Aplicando Minería De Datos [Tesis]. Disponible e-mail: reinel.arias@gmail.com

[7] J.R. Quinlan, "Induction of Decision Trees", en *Machine Learning*, capítulo 1, 1990, p. 81-106. Morgan Kaufmann.

[8] J.R. Quinlan, *Generating Production Rules from Decision trees. Proceeding of the Tenth International Joint Conference on Artificial Intelligence,* pp. 304-307. San Mateo: CA, Morgan Kaufmann, 1987.

[9] G.F. Cooper and E. Herskovits, *A Bayesian Method for the Induction of Probabilistic Networks*, Volume 9, 1992, pages 309-47.

[10] C. Carmona Márquez, G. Castillo Jordán y E. Millán Valldeperas, Modelo Bayesiano del Alumno basado en el Estilo de Aprendizaje y las Preferencias, IEEE-*Rita* Vol. 4, Núm. 2, May. 2009, p. 139.

[11] R. E. Neapolitan, (1990), Probabilistic reasoning in expert systems: Theory and Algorithms. Available: http://www.getcited.org/pub/102782375