

Modelo de Evolución entre Esquemas Conceptuales de Bases de Datos basado en una arquitectura ANFIS

Henry Huarsaya Mamani
hhuarsaya@unsa.edu.pe
EPIS - Escuela Profesional de Ingeniería de Sistemas
Universidad Nacional de San Agustín
Av. Venezuela s/n
Arequipa - Perú

Resumen

El presente trabajo tiene como objetivo desarrollar un modelo de evolución entre esquemas conceptuales en un ámbito de migración e integración de bases de datos. Para que el modelo tenga mayor grado de automatización, confiabilidad y flexibilidad, se basa en técnicas de inteligencia artificial, específicamente en un sistema de inferencia difuso de redes adaptables, que aprovechara y analizara la información en términos de modificaciones, para establecer escalas de correspondencias entre los elementos de los esquemas.

Palabras Clave: *Evolución entre esquemas conceptuales, Migración e integración de bases de datos, Árboles n-arios, Grado de correspondencia, Automatización, ANFIS.*

1. Introducción.

Los sistemas basados en software evolucionan continuamente desde el momento de su concepción y en su ciclo de vida son actualizados permanentemente hasta que están obsoletos y deben ser sustituidos [7], a la vez que la necesidad de integrar y migrar sistemas de información cobra mayor importancia a medida que Internet, los avances computacionales y las comunicaciones lo hacen posible, incrementando significativamente su desempeño.

La necesidad de evolución y adaptación a los nuevos requerimientos tecnológicos, empujan a los sistemas a una nueva situación a la que no es posible llegar a través del mantenimiento clásico [6]. Independientemente del concepto que se tenga sobre la evolución del software como los descritos brevemente en [6, 7, 8], lo que sí parece claro es que la evolución que tenga el sistema durante su funcionamiento, no se trata de un proceso perfectamente definido a priori y del que el modelador tenga conocimiento exacto, de forma que pudiera prevenir todas las eventualidades en la fase de modelado.

En la migración de bases de datos (*BD*) la evolución de los esquemas conceptuales (*EC*) tienen un rol de especial interés e importancia significativa, siendo estos los que especifican principalmente los componentes estáticos de la base de datos, es decir son quienes gobiernan la estructura y las restricciones de los datos estáticos almacenados [1].

En este ámbito es fundamental conocer exactamente el grado de evolución sufrido por el *EC* y la *BD* asociada que va a ser migrada, y además se hace indispensable conocer qué modificaciones han sufrido los elementos de los *EC*, así como por ejemplo qué inserciones y borrados se han producido durante el proceso de evolución,

también es una tarea importante determinar qué tipo de información y como se debe usar esta para identificar el origen y el destino de la migración [3].

Entonces, es de especial relevancia una arquitectura en software que no sólo sea capaz de analizar el grado de correspondencia entre los EC origen y destino, sino que también aproveche la información semántica que estas poseen, y resuelva posibles conflictos e inconsistencias.

En la sección 2, se presenta brevemente el problema, como han tratado algunos autores este problema. En la sección 3 se presentan algunos conceptos básicos que usará el modelo como la estructura de datos, los EC y el ANFIS. En la sección 4 se muestra el modelo de la arquitectura propuesta. En la sección 5 el modelo de funcionamiento de la arquitectura ANFIS propuesta con resultados experimentales, sobre la generación de correspondencias. Finalmente en la sección 6 se presentan las conclusiones de este trabajo.

2. El problema.

El problema para obtener los grados de equivalencias y diferencias entre los EC, es tratado actualmente por varias herramientas comerciales que son capaces de identificar la evolución entre los esquemas sin necesidad de comparar, como es el caso de VDIFF de Rational Rose. Sin embargo, estas herramientas guardan registros de las operaciones realizadas y obligan a los EC ser dependientes entre si, restringiéndoles a que sean de un mismo formato o pertenezcan a la misma aplicación [3, 4]. Pero además, estas herramientas no poseen un claro soporte a la migración de datos, siendo un problema cada vez mas frecuente y lamentablemente desatendido por la ingeniería de software [5].

Por otro lado, en la actualidad gracias a los aportes realizados especialmente con carácter de investigación y desarrollo, ya se dispone de una herramienta semi-automatizada como SOSY Modeler que incluye muchas de las características deseadas, la herramienta abarca este problema en su primera fase, basado en algoritmos de comparación y como estos implican independencia entre los EC [3].

Pero estas herramientas están enteramente gobernadas por algoritmos discretos y/o clásicos, reduciendo enormemente su flexibilidad y poder de análisis autónomo. Es decir que el proceso tiende a generar posibles inconsistencias al desbordar alguno de sus criterios o métodos de evaluación.

3. Conceptos básicos.

Existen trabajos que han sido desarrollados basados desde el punto de vista de la teoría de grafos, mejor descrito como grafos acíclicos no dirigidos (*árboles*) [11,12], esta técnica puede ser utilizada para representar nuestros EC, pero basándonos en términos mas rigurosos los que mejor se adaptan a nuestra situación son los árboles n-arios [13], a continuación se expondrá brevemente algunos conceptos básicos como los árboles n-arios, esquema conceptuales y la arquitectura ANFIS.

3.1. Árboles n-arios y esquemas conceptuales.

Un árbol n-arios es una estructura recursiva, en el cual cada elemento tiene cualquier número de árboles n-arios asociado [11,12] y un EC esta definido en [15] como, una descripción central única de los distintos contenidos de información que puede coexistir en una BD. En la Fig. 1. se representa mejor estas estructuras. En la figura 1. (b) se hace explicita la raíz, y cada uno de los n subárboles que tiene asociado.

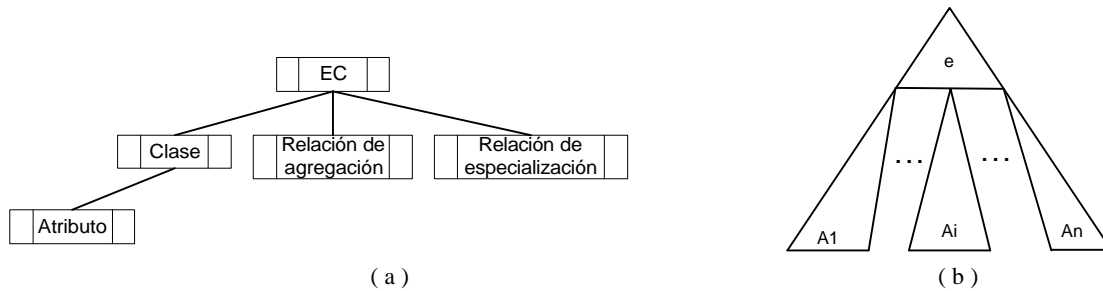


Fig. 1. (a) Representación de los elementos de un EC en un árbol
(b) Formalismo para árboles n-arios

3.2. Las bases de ANFIS¹.

Basada en la lógica difusa y las redes neuro-adaptables estos sistemas producen una sinergia especial que brinda mayores beneficios de desempeño que por separadas. La arquitectura llamada ANFIS, se basa en esas características que se detallan a continuación.

3.2.1. Lógica difusa y los sistemas de Inferencia difusa.

La Lógica Difusa (*Fuzzy Logic*), se basa en la teoría de los conjuntos difusos donde el grado de pertenencia de un elemento a un conjunto va a venir determinado por una función de membresía $\mu(x)$, que puede tomar todos los valores reales comprendidos en un intervalo generalmente entre 0 y 1 [2], como se puede ver en la Fig. 2 y esta definida por la siguiente función:

$$A = \{(x, \mu(x)) \mid x \in X\}$$

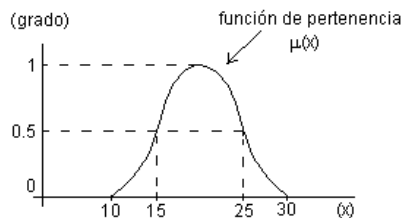


Fig. 2. función de pertenencia a un conjunto difuso

Y los sistemas de inferencia difusa constan básicamente de tres componentes: Una base de reglas, la cual contiene una selección de reglas difusas; una base de datos (*o diccionario*), el cual define las funciones de membresía empleadas con las reglas difusas; y, un mecanismo de razonamiento, que realiza el procedimiento de inferencia sobre las reglas y los hechos proporcionados, para producir una salida [9,10]. La Fig. 3 muestra con más detalles la relación entre todas estas componentes.

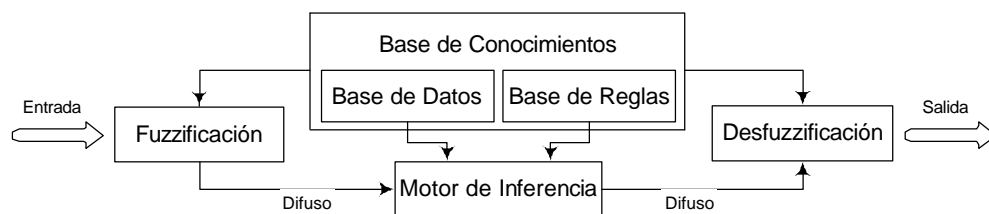


Fig. 3. Sistema de inferencia difusa. [14]

¹ Adaptive-Neuro-based Fuzzy Inference System

3.2.2. Redes Neuro Adaptables.

Basado en las redes neuronales artificiales (ANN), esta red es una estructura cuyo comportamiento global de entrada/salida queda determinado por una colección de parámetros modificables. Específicamente la configuración de una red adaptable se compone de un conjunto de nodos conectados por enlaces direccionados, como se ilustra en la Fig. 4, donde generalmente una función de nodo es una función parametrizada, cambiando dichos parámetros se modifica la función de nodo y en consecuencia el comportamiento global de la red [9,10,14].

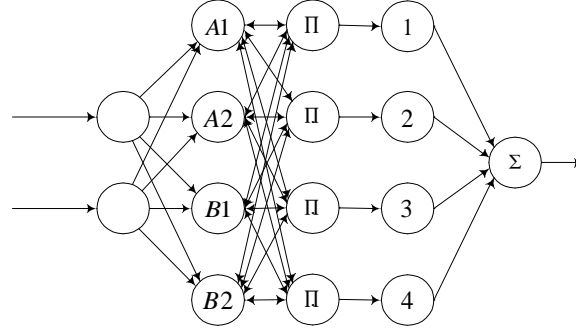


Fig. 4. Arquitectura de una red adaptable.

Donde los nodos en una misma capa realizan funciones similares, a continuación se da una descripción de la función que realiza cada capa en la arquitectura mostrada en donde la salida del i -ésimo nodo en la *capa l* se denota como $O_{l,i}$.

Capa 1. Cada nodo i en esta capa es un nodo adaptable, con una función de nodo definida por una función de membresía:

$$O_{1,i} = m_{Ai}(x), \text{ para } i = 1, \dots, n$$

Capa 2. Los nodos de esta capa son nodos fijos, y se etiquetan con el símbolo Π ; la salida aquí es la operación norma-T de todas las señales que entran a él:

$$O_{2,i} = w_i = T(m_{Ai}(x), m_{Bi}(y)), i = 1, \dots, n.$$

Cada salida de estos nodos representa la intensidad de disparo de una regla.

Capa 3. Esta capa se compone de nodos fijos etiquetados con Π . El i -ésimo nodo calcula la relación de la intensidad de disparo de la i -ésima regla con respecto a la suma de las intensidades de disparo de todas las reglas:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, \dots, n.$$

A cada salida de esta capa se le llama intensidad de disparo normalizada.

Capa 4. Cada nodo i en esta capa es un nodo adaptable con una función definida por:

$$O_{4,i} = \bar{w}_i z_i = \bar{w}_i (p_i + q_i + r_i)$$

Donde \bar{w}_i es una salida de intensidad normalizada de la capa 3, y $\{p_i, q_i, r_i\}$ es el conjunto de parámetros que define a los polinomios z_i en este nodo.

Capa 5. El único nodo presente en esta capa, es un nodo fijo denotado por \sum , el cual calcula la salida global como la suma de todas las señales que entran a él:

$$\text{Salida global} = O_{5,i} = f = \sum_i \bar{w}_i z_i = \frac{\sum_i w_i z_i}{\sum_i w_i}$$

De esta forma, lo que se obtiene es una red adaptable la cual es funcionalmente equivalente a un modelo difuso de tipo Sugeno. La estructura mostrada no es única, ya que se pueden hacer otras.

4. Modelo de la arquitectura propuesta.

El criterio para realizar el análisis será por niveles de forma ascendente, es decir comenzaremos por analizar los atributos y luego las clases. Este orden se debe a que los atributos son los elementos de última instancia que definirán si es una clase correspondiente a otra ó si es que esta sólo fue modificada. Esto ocurrirá aún cuando sus clases hayan sido modificadas. A continuación se puede apreciar el esquema general de la arquitectura.

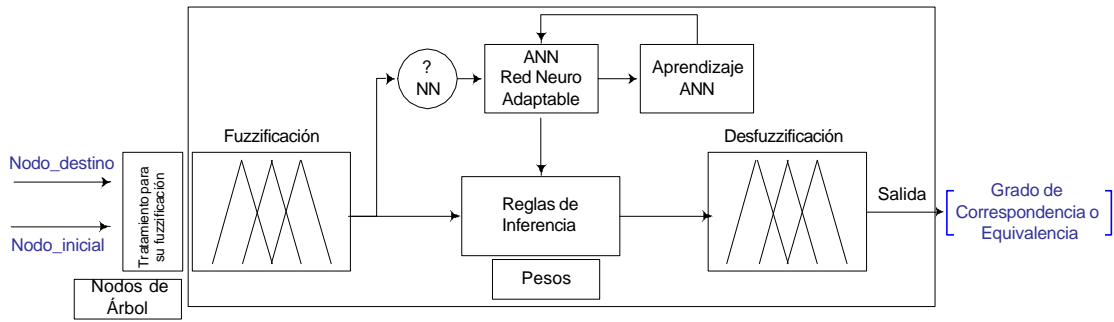


Fig. 5. Esquema general de la arquitectura propuesta.

En el presente trabajo se hicieron uso de estas dos técnicas por poseer características adecuadas, en donde:

Las redes neuro-adaptables, son sistemas que incorporan aspectos de las redes neuronales en el sentido de que son sistemas con la capacidad de aprender y generalizar [13].

La lógica difusa, trabaja con razonamiento lógico basado en reglas de inferencia, incorporan la posibilidad de trabajar con variables lingüísticas, además pueden cambiar el tratamiento binario que hacen las ANN [14].

La información semántica de los atributos ayudará a identificarles adecuadamente, y si es que ocurre una incongruencia, esta será descartada cuando llegue al nivel de clase. Podemos darnos cuenta de que se evaluarán todos los atributos del EC inicial (ECi) con los atributos del EC final (ECf). Esto no sucede así con las clases, ya que el sistema validará la correspondencia sólo entre las clases en la que sus atributos hayan establecido una posible correspondencia, de esta forma se reduce la cantidad de operaciones a realizar, lo suficientes para garantizar e identificar correctamente la evolución de los EC.

El prototipo de la arquitectura que este trabajo propone, ha sido implementado en Visual C++ 6.0 considerando que es el más apropiado para este caso, para poder mostrar algunos resultados que se verán en la siguiente sección.

5. Generación de correspondencias entre los EC's.

La herramienta es capaz de generar las diferentes correspondencias y hacer visible las diferencias entre los esquemas que se están comparando y así lograr ver cual es la evolución sufrida por el esquema conceptual inicial, gracias a la información proporcionada por la arquitectura ANFIS, ya que se dispone de todas las características de los elementos de ambos esquemas. Con esta información es posible mostrar la evolución a un mayor nivel de detalle.

En el siguiente ejemplo se muestra como es que nuestro prototipo funciona, se ha trabajado para este caso experimental solamente con cuatro clases que dos a dos representan al ECi y ECf, donde cada clase consta de un total de cuatro atributos. Se pretende que la herramienta se encargue de la evolución automatizada entre las respectivas, si son ó no correspondientes y en que medida lo son.

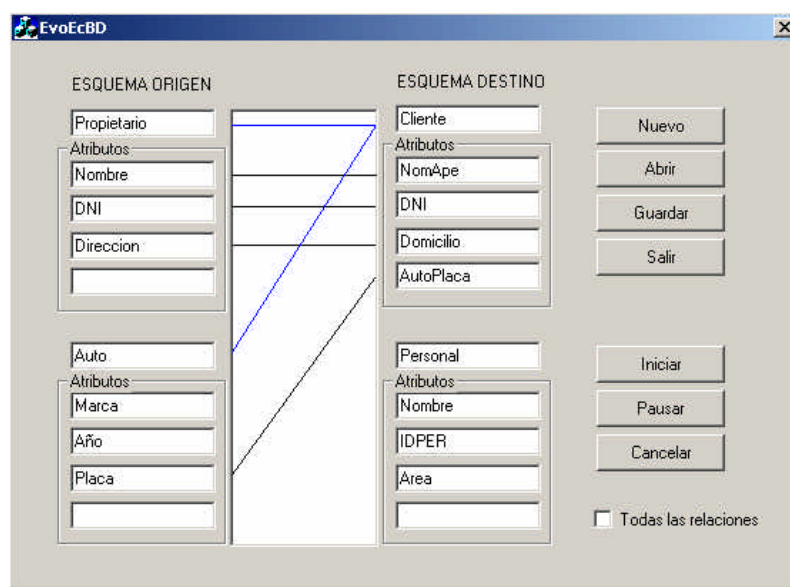


Fig. 6. Evolución de Esquemas Conceptuales.

Las líneas azules representan a nivel de clases, un nivel de correspondencia alto (ejemplo: 60%-90%) y las líneas negras de forma análoga para los atributos, y curiosamente como ya se esperaba que en realidad la herramienta encuentra mas correspondencias pero estas con un nivel bajo (ejemplo: 20%-50%), pero esta información es fácil de suprimir, dependiendo de dos características importantes los valores del umbral cada nodo de la red y aplicando a nuestro resultado un factor de aceptación para la resultante final, no siempre colocando un valor alto en la umbral se obtiene el mejor resultado, pero si sucede así con el factor de aceptación aquí el problema es a mayor grado se suprime varias correspondencias correctas.

En la siguiente figura se puede apreciar este caso, donde la herramienta muestra todas las correspondencias encontradas bajas y altas.

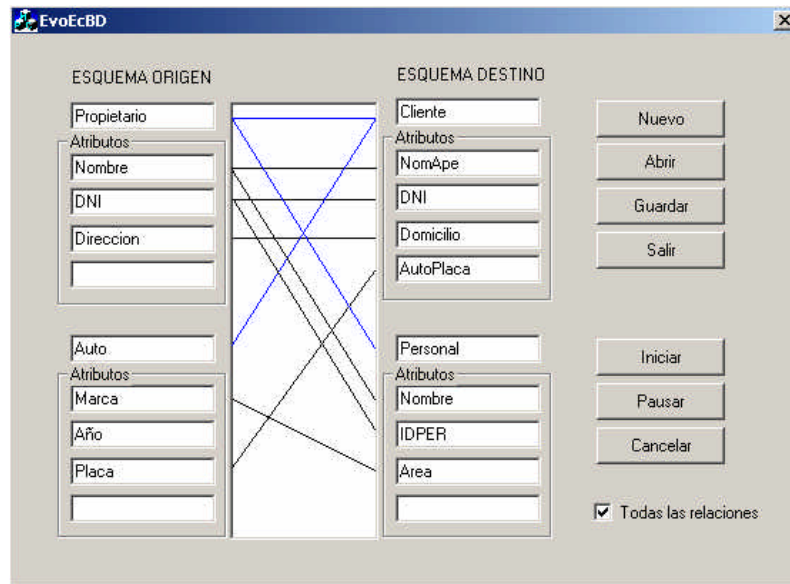
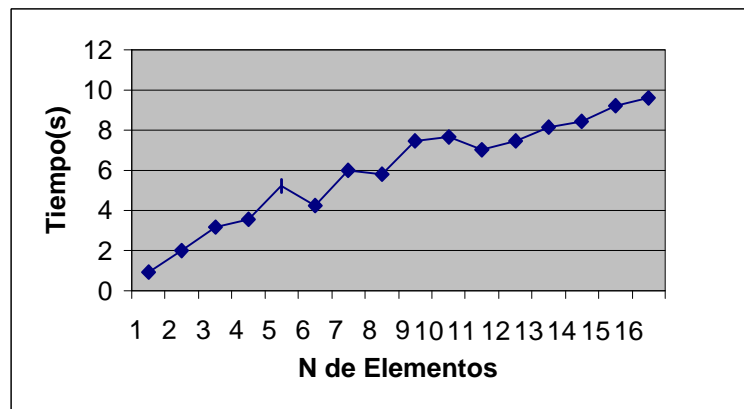


Fig. 7. Evolución de Esquemas Conceptuales con todas las relaciones de correspondencia.

Aquí se aprecia mejor todas las correspondencias, y las que son correspondencia baja son suprimidas por la herramienta según el criterio de correspondencia y los factores antes mencionados.

Un factor crítico aun por resolver es el tiempo de procesamiento dependiendo esta de la cantidad de neuronas, número de elementos a comparar (no necesariamente el total), valor del umbral, arquitectura del software.

Para esta herramienta se trabajo con un total de quince neuronas por esquema, nueve reglas de inferencia y dieciséis elementos en total como máximo que son nuestros datos de entrada. El siguiente cuadro muestra los intervalos de tiempo utilizados por la herramienta en este caso, pero este puede ser variante de acuerdo a los elementos y/o grado de dificultad para ser identificados



6. CONCLUSIONES.

Se ha presentado la solución a nivel experimental, al proceso que corresponde a la identificación y clasificación de los nodos para establecer sus respectivas correspondencias de evolución según al tipo de elemento.

El presente modelo, hace uso de los elementos del EC_f y el EC_i para una identificación de correspondencia y ver su evolución de forma biyectiva. Permitiendo así, un mayor nivel de flexibilidad e independencia del sistema, a diferencia del análisis discreto.

Una correcta identificación de la evolución real para este análisis propuesto, radica directamente en el tercer nivel del árbol n-ario, es decir en los atributos. En la red adaptable, son las que tendrán mayores pesos que sus semejantes, que serán determinantes para el resultado.

La determinación de la función de membresía es muy importante para lograr una correcta asignación de los grados de pertenencia. La función escogida para este modelo es tipo campana debido a la suavidad que muestra en los puntos cercanos al núcleo.

El valor de la umbral puede ser mejorado por entrenamiento y el rendimiento de la herramienta de igual forma con un mayor número de neuronas y el mejoramiento de los algoritmos para reducir el tiempo de procesamiento para un mayor número de elementos, tengamos en cuenta que la prioridad aquí es la identificación de la evolución real.

7. REFERENCIAS.

- [1] Varas, M. Diseño conceptual de Bases de Datos: Un enfoque basado en la medición de Calidad. presentado en las JCC, 2001. <http://www.inf.udec.cl/~mvaras/papers/2001/mvaras-wisw-cr2.htm>
- [2] Urrutia, A., Rodríguez, C. Comparación del álgebra Relacional en Bases de Datos Relacionales Clásicas y Difusas. presentado en el VIII Encuentro Chileno de Computación en las JCC, 2001.
- [3] Silva, J., Carsí, J., Ramos, I. Comparación Automática de Esquemas Conceptuales Orientados a Objeto. Presentado a las JISBD, 2001. www.inf.udec.cl/revista/edicion7/jsilva.htm
- [4] Abad, S., Carsí, J., Ramos, I. Obtención de un Orden de Migración para los elementos de un esquema conceptual orientado a objetos. Presentado en Taller de Evolución del Software, 2001. <http://aporia.ugr.es/evol/docs/5DEF.pdf>
- [5] Pérez, J., Carsí, J., Ramos, I., Anaya, V., Silva, J. Migración de Datos automática a partir de la información de los esquemas conceptuales. Presentado en Taller de Evolución del Software, 2001. <http://aporia.ugr.es/evol/docs/4DEF.pdf>
- [6] Rodríguez, A. Márquez, A. Toro, M. Gestión de la evolución del software. El eterno problema de los *legacy system*. Presentado en Taller de Evolución del Software, 2001. <http://aporia.ugr.es/evol/docs/1DEF.pdf>
- [7] Torres, C., Jesús J., Parets J. Estadios y Condiciones del Software en Evolución. Presentado en Taller de Evolución del Software, 2001. <http://aporia.ugr.es/evol/docs/2DEF.pdf>
- [8] Gómez, C., López, J. Evolving Temporal Conceptual Schemas: The Partition Case. Presentado en Taller de Evolución del Software, 2001. <http://aporia.ugr.es/evol/docs/6DEF.pdf>
- [9] Arellano, O. Procesamiento y clasificación de señales utilizando Lógica Difusa. Centro de investigación y estudios avanzados del IPN México .Tesis Doctoral, 2000. Págs: 1-33. www.ie.cinvestav.mx/biblioteca/tesis/xd00.2.pdf
- [10] Shing, J., Jang, R. Anfis: Adaptive-Network-Based Fuzzy Inference System .University of California, Berkeley, USA.1993.

- [11] Kolman, B., Busby, R. Estructuras de Matemáticas Discretas para la Computación. Naucalpan, México. Editora Prentice – Hall Hispanoamericana S.A. Primera Edición en Español. 1986.
- [12] Villalobos, J. Diseño y Manejo de Estructuras de Datos en C. Santafé de Bogotá, Colombia. McGraw-Hill 1996.
- [13] Hilera, J., Martínez V. Redes Neuronales Artificiales: fundamentos, modelos y aplicaciones. Wilmington Delaware, USA. Addison-Wesley Iberoamericana, S.A. Edición RA-MA. 1995.
- [14] Rao, V., Rao, H., C++ Neural Networks and Fuzzy Logic. New York, USA Editora Miss:Press, Second Edition, 1995.
- [15] Piatinni, M., De Miguel, A., Marcos, E. Diseño de Bases de Datos Relacionales. México. Alfa y Omega S.A. 2000.