

Apéndice D

Redes neuronales y teoría de los conjuntos difusos

Las redes neuronales artificiales y la lógica difusa o borrosa son dos de los avances científicos recientes más importantes del conocimiento. Mediante estas herramientas se ha intentado simular dos de las características más importantes con que cuenta el cerebro humano: la capacidad de aprendizaje y el poder procesar información incompleta o que no es precisa. Estas técnicas se han podido utilizar para la solución de problemas tanto científicos como de la vida diaria. Con una buena combinación de estas dos técnicas es posible un desarrollo tecnológico más amplio en diversos campos, desde la medicina hasta el mejoramiento de electrodomésticos.

D.1. Redes neuronales artificiales

Mediante esta técnica se intenta imitar el proceso de aprendizaje del cerebro humano. El cerebro está formado por miles de millones de neuronas conectadas entre sí. Utiliza información que es percibida, transmitida hasta las neuronas, y allí procesada por ellas para dar una respuesta a cada uno de los diferentes estímulos. Cada neurona tiene tres partes: un cuerpo celular, una estructura de entrada (*Dendrita*) y una de salida (*Axón*). La mayoría de las terminales de los axones se conectan con las dendritas de otras neuronas (*Sinápsis*). El comportamiento de una neurona es el siguiente: recibe una señal de entrada con una fuerza determinada, dependiendo de ellas la neurona emite una señal de respuesta, las *sinápsis* pueden variar en fuerza, algunas pueden dar una señal débil y otras una fuerte. A una neurona pueden llegar miles de señales de entrada, cada una con una fuerza o peso diferente. Matemáticamente el comportamiento de la neurona puede representarse por una lista de sus señales de entrada que son multiplicadas por sus pesos respectivos y posteriormente sumados, el resultado es llamado nivel de activación de la neurona del cual depende la señal de salida que es enviada a cada una de las neuronas a las que está conectada a ella.

Una red neuronal artificial (RNA) es un sistema compuesto de muchos elementos procesadores simples conectados en paralelo, cuya función es determinada por la estructura de la red, la fuerza en las conexiones y el procesamiento realizado por los elementos en los nodos (Jang 1997). Las RNA, igual que las personas, aprenden de la experiencia.

Uno de los primeros modelos de redes neuronales fue el *Perceptrón* diseñado por Rosenbalatt, el cual contaba con tres tipos de neuronas: sensoriales, asociativas y de respuesta. Las sensoriales tomaban entradas de fuera de la red, las unidades de respuesta llevaban señales fuera de la red al mundo externo y las asociativas eran exclusivamente internas. Estos tipos ahora son llamados unidades de entrada, de salida y ocultas. Rosenbalatt desarrolló métodos para que la red alterara los niveles sinápticos de forma que la red aprendiera a reconocer niveles de entrada.

Después de los años 80 la inteligencia artificial se hizo más popular y aumento su campo de aplicación. Se han desarrollado algoritmos para modelar procesos mentales de alto nivel como la asociación de conceptos, deducción, inducción y razonamiento.

En 1986 Rumelhart y McClelland demostraron que algunos problemas imposibles para los Perceptrons simples pueden ser resueltos por redes multi-nivel con funciones de activación no lineales, utilizando procesos simples de entrenamiento (algoritmos *Back-propagation*).

D.1.1. Características generales

Pesos: Las redes neuronales pueden tener factores de peso adaptable o fijo. Las que tienen pesos adaptables emplean leyes de aprendizaje para ajustar el valor de la fuerza de interconexión con otras neuronas. Si se utilizan pesos fijos, su tarea debe estar previamente definida. Los pesos son determinados a partir de una descripción completa del problema a tratar. Los pesos adaptables son muy importantes, ya que no se conoce de antemano su valor correcto.

Aprendizaje: Se utilizan dos tipos de aprendizaje: supervisado y no supervisado. En el primero se le proporciona a la red tanto la salida como la entrada correcta, y la red ajusta sus pesos para disminuir el error en la salida que ella calcula. Este tipo es utilizado principalmente en el conocimiento de patrones. En el aprendizaje no supervisado a la red se le proporcionan únicamente los estímulos, y la red ajusta los pesos de sus interconexiones utilizando solamente los estímulos y la salida calculada por la red. La forma de aprendizaje que se utiliza depende del tipo de problema que se intenta resolver.

Fases de operación: Se presentan dos fases en la operación de una red neuronal artificial: Entrenamiento y recuperación de lo aprendido. En la primera fase se le proporcionan estímulos de entrada y salida (según el caso), para que la red ajuste sus pesos de interconexión y minimice el error en la salida que calcula. En la segunda fase la red solamente calcula la respectiva salida. El aprendizaje de la red disminuye a medida que aumenta el número de veces que se usa.

No son algorítmicas: Las redes neuronales artificiales no se programan sólo para seguir una secuencia de instrucciones, sino que ellas mismas pueden generar sus propias reglas para aprender de su propia experiencia.

Necesitan un patrón: Las redes neuronales no son capaces de reconocer nada que no tenga algún tipo de patrón. Son muy buenas resolviendo problemas de asociación, evaluación y reconocimiento de patrones.

D.1.2. Tipos de redes

Dependiendo de la estructura de la red neuronal existen varias tipologías, algunas de estas son: el *Perceptron*, *Backpropagation*, *Hopfield*, y *Kohonen*. A continuación se presenta una descripción resumida de cada una de estas tipologías:

D.1.2.1. Perceptrón

En 1943 se desarrolló el primer modelo de neurona artificial por McCulloch y Pitts. El *perceptrón* intenta modelar el comportamiento de la neurona biológica, derivado del modelo hecho por McCulloch y Pitts (Looney 1997). Es la base de la arquitectura de las redes neuronales artificiales. El *perceptrón* trabaja con funciones de activación. En la figura D.1 se observa el modelo de un *perceptrón* para reconocimiento de patrones. El cuerpo de la neurona es representado como un sumador lineal de estímulos z_j , seguida por una función no lineal $f(z_j)$ que es la función de activación, esta utiliza la suma de estímulos para determinar la salida de la neurona. En el *perceptrón* cada entrada es multiplicada por un factor de peso W correspondiente, los resultados son sumados y luego evaluados contra el valor de umbral θ , si el valor es mayor al máximo el *perceptrón* se activa.

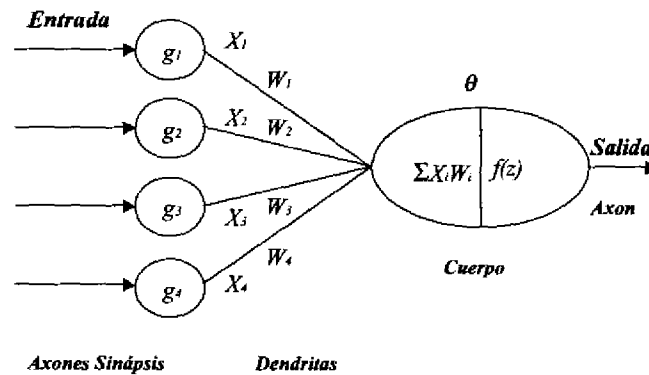


Figura B.1 Modelo de un *Perceptrón*

La primera capa actúa como un grupo de sensores que detectan una señal específica con la importancia correspondiente. La segunda capa toma las salidas de los detectores y clasifica el patrón de entrada dado. El aprendizaje empieza ajustando la fuerza en las conexiones (pesos W_i) y el valor de umbral θ . La función g_i es fija y está predeterminada.

El entrenamiento de un *perceptrón* se hace utilizando una función de aprendizaje. Si la respuesta dada por el *perceptrón* es incorrecta, se modifican todos los pesos de las conexiones de acuerdo a la función D.1.

$$\Delta w_i = \eta t_i x_i \tag{D.1}$$

donde η es la tasa de aprendizaje, t_i en la salida deseada, y x es la entrada del *perceptrón*.

La función anterior puede ser utilizada también para la actualización del valor de umbral θ . El valor de η puede ser constante a lo largo del entrenamiento, o puede variar en forma proporcional al error. Un η proporcional al error lleva a una convergencia rápida, pero causa un aprendizaje variable.

El *perceptrón* es capaz tan sólo de resolver funciones definidas por un hiperplano (objeto de dimensión N-1 contenida en un espacio de dimensión N), que corte un

espacio de dimensión N . Es decir sólo puede resolver una función, si todos los posibles resultados del problema pueden separarse de forma que no se combinen entre sí.

Existen varios tipos de *perceptrones* dependiendo del número de capas que tenga. Un *perceptrón* simple o de dos capas, corresponde a una capa de entrada con neuronas lineales y una de salida con función de activación tipo escalón. Este sólo puede establecer dos regiones separadas por una frontera lineal en el espacio de patrones de entrada.

Un *perceptrón* de tres capas puede formar cualquier región convexa en este espacio. Cada elemento se comporta como un *perceptrón* simple. Un *perceptrón* de cuatro capas puede generar regiones de decisión arbitrariamente complejas. Puede resolver una gran variedad de problemas cuyas entradas sean analógicas; la salida sea digital y sea linealmente separable. El rango de tareas que el *perceptrón* puede realizar es más amplio que simples decisiones y reconocimiento de patrones. El *perceptrón* multicapa puede ser usado para la predicción de una serie de datos en el tiempo, como cambios financieros.

D.1.2.2. Backpropagation

Cuando se combinan varios *perceptrones* en una capa y los estímulos de entrada después son sumados, se tiene ya una red neuronal.

La falta de métodos de entrenamiento apropiados para los *perceptrones* multicapa (MLP) hizo que declinara el interés en las redes neuronales en los años 60 y 70. Esto no cambió hasta la reformulación del método de entrenamiento para MLP *backpropagation* a mediados de los años 80 por Rumelhart (1986).

En esta red, se interconectan varias unidades de procesamiento en capas, las neuronas de cada capa se conectan entre sí. Cada neurona de una capa proporciona una entrada a cada una de las neuronas de la siguiente capa. En la figura D.2 se observa la arquitectura de este tipo de red.

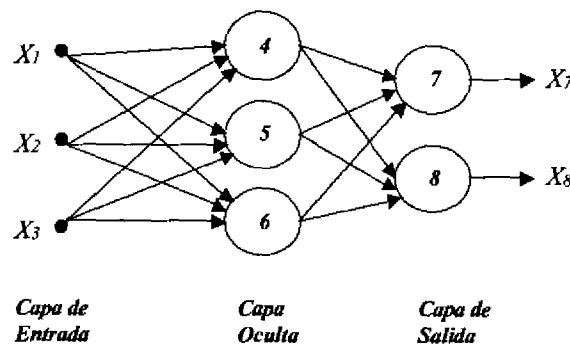


Figura B.2: Modelo de una red tipo *Backpropagation*

El término *backpropagation* se refiere al método para calcular el gradiente de error en una red *feed-forward* (de aprendizaje supervisado), que es una aplicación de la regla de la cadena de cálculo elemental (Jang 1997). Básicamente el entrenamiento de este tipo de red consiste en lo siguiente:

- a) Pasada hacia delante (*forward pass*): las salidas son calculadas y el error en las unidades de salida es calculado.
- b) Pasada hacia atrás (*backward pass*): el error de las salidas es utilizado para alterar los pesos de las unidades de salida. Luego el error en las neuronas de las capas ocultas es calculado mediante propagación hacia atrás del error en las unidades de salida), y los pesos en las capas ocultas son alterados usando esos valores.

D.1.2.3. Modelo Hopfield

Son redes de adaptación probabilística y recurrente. Funcionalmente entrarían en la categoría de las memorias autoasociativas, es decir que aprenden a reconstruir los patrones de entrada que memorizan durante el entrenamiento. Son monocapa con interconexión total y en la que el valor de salida de cada unidad es binario (solo puede tomar los valores 0 o 1) y siguen una regla de aprendizaje no supervisado.

Están formadas por N neuronas interconectadas que actualizan sus valores de activación en forma independiente, todas son a la vez de entrada y salida. Es buena para resolver problemas de optimización. Tiene siempre una restricción por simetría en cada conexión ($W_{jk}=W_{kj}$). Cuando una neurona mantiene su valor de activación se dice que es estable. Se llega a un estado estable cuando todas las neuronas son estables (Looney 1997).

Ante la presentación de un estímulo nuevo se obtiene una configuración inicial más o menos parecida a la de alguno de los estímulos almacenados, el sistema hace iteraciones hasta quedar en una configuración estable. En la figura D.3 se observa la arquitectura del modelo de *Hopfield*.

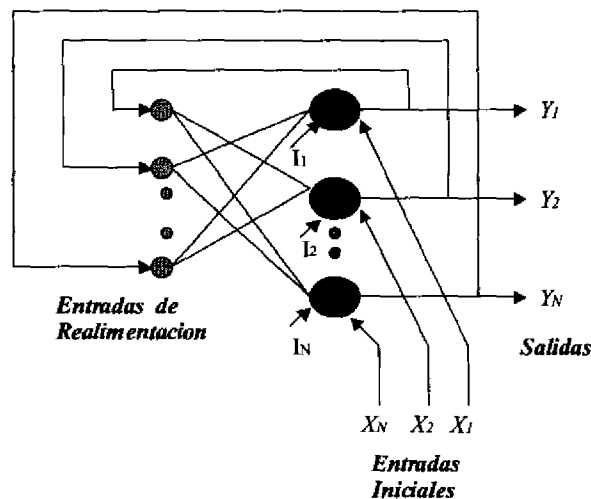


Figura D.3: Modelo de Hopfield

Una característica importante de este modelo es que como se muestra en la figura D.3 una neurona nunca se realimenta a sí misma.

D.1.2.4. Modelo Kohonen

Esta red está basada en la capacidad que tiene el cerebro de formar mapas característicos de la información recibida del exterior. El objetivo de *Kohonen* era demostrar que un estímulo externo, suponiendo una estructura propia y una descripción funcional del comportamiento de la red, era suficiente para forzar la formación de mapas.

La red neuronal *Kohonen* contiene solamente una capa de neuronas y una capa de entrada, que se ramifica para todos los nodos. Pertenece a la categoría de redes competitivas o mapas de auto organización, es decir tiene un aprendizaje no supervisado. Tiene también funciones de activación lineales y flujo de información unidireccional (red en cascada). La red cuenta con N neuronas de entrada y M de salida, cada una de las neuronas de entrada está conectada a todas las de salida.

Las unidades de entrada reciben datos continuos normalizados. La red clasifica los patrones de entrada en grupos de características similares, de tal manera que cada grupo activa siempre las mismas salidas.

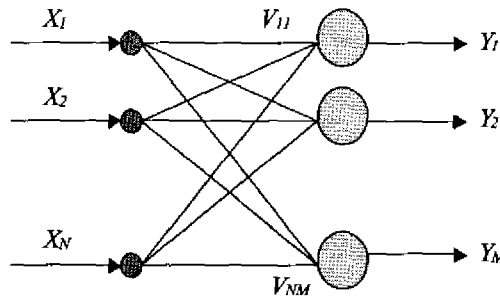


Figura D.4: Modelo de red neuronal *Kohonen*

Puede decirse que entre las neuronas de la capa de salida existen conexiones laterales de inhibición implícitas aunque no estén conectadas. La influencia que una neurona tiene sobre las demás es función de la distancia entre ellas, se hace muy pequeña cuando están muy alejadas.

Cada una de las M neuronas tiene un vector de pesos $V^{(m)}$ de dimensión N , que es la misma dimensión del vector de entradas $X^{(q)}$. Al principio los vectores de pesos son inicializados aleatoriamente. Se calculan las distancias cuadradas entre $X^{(q)}$ y cada $V^{(m)}$, $m=1, \dots, M$ así:

$$D_{qm} = D_{qm}(X^{(q)}, V^{(m)}) = \sum_{n=1, N} (x_n^{(q)} - v_n^{(m)})^2 \quad [D.2]$$

La distancia mínima D_{qm^*} determina la neurona m^* que es ganadora sobre las demás neuronas. Desde este punto es posible tomar diferentes estrategias, una de ellas es actualizar el vector de peso paramétrico de la neurona ganadora utilizando:

$$V^{(m^*)} = V^{(m^*)} + \eta(X^{(q)} - V^{(m^*)}) \quad [D.3]$$

donde η es la tasa de aprendizaje, y las demás neuronas mantienen los valores anteriores. Otra estrategia que puede ser utilizada es actualizar positivamente (recompensar) todas las neuronas que son cercanas a la neurona ganadora, y actualizar negativamente (castigar) las neuronas que están a mayor distancia (Looney1997).

Esta red una vez entrenada puede ser utilizada para clasificar patrones de entrada similares en el espacio N-dimensional.

D.1.3. Aplicaciones

Las redes neuronales pueden ser de diferentes modelos de aprendizaje y tipologías dependiendo del uso que se espere darle a cada una. Pueden ser aplicadas a diferentes campos tales como: Control de eficiencia de máquinas, reconocimiento de firmas, reconocimiento de blancos mediante sonares, predicciones en el tiempo, decisiones, análisis de inversiones, monitoreo, mercadotecnia, planeación estratégica, diagnóstico, diseño, optimización y en aplicaciones (recientemente) de Ingeniería Civil como la valoración de efectos sísmicos, amplificación de onda, solución de problemas de gerencia de construcción, control activo estructural y diagnósticos de daño (Paolucci 2000). Una aplicación reciente ha sido en el pronóstico de caudales diarios y en el cálculo del nivel de agua en ciertos sitios en la época de crecida del río Iguazú. En el sector industrial las redes neuronales son ampliamente utilizadas empleando modelos de ingeniería que incluyen conocimiento científico y tecnológico en la industria cervecera, química, aérea, de alimentos, del acero, el vidrio, el cemento y las telecomunicaciones.

D.2. Teoría de los conjuntos difusos

La lógica de conjuntos difusos o borrosos, como su nombre lo indica, trabaja con conjuntos que no tienen límites perfectamente definidos, es decir, la transición entre la pertenencia y no-pertenencia de una variable a un conjunto es gradual. Se caracteriza por las funciones de pertenencia, que dan flexibilidad a la modelación utilizando expresiones lingüísticas, tales como mucho, poco, leve, severo, escaso, suficiente, caliente, frío, joven, viejo, etc. Surgió de la necesidad de solucionar problemas complejos con información imprecisa, para los cuales la matemática y lógica tradicionales no son suficientes. La lógica difusa es un lenguaje que permite trasladar sentencias sofisticadas del lenguaje natural a un formalismo matemático.

La lógica difusa fue desarrollada a partir de 1960 por Lotfi Zadeh, guiado por el principio de que las matemáticas pueden ser usadas para encadenar el lenguaje con la inteligencia humana. Algunos conceptos pueden ser mejor definidos con palabras, los conjuntos difusos ayudan a construir mejores modelos de la realidad.

D.2.1. Características generales de los conjuntos difusos

Conjunto Difuso: Un conjunto difuso expresa el grado de pertenencia al conjunto que tiene cada uno de los elementos. El conjunto difuso A en X puede definirse como el conjunto de los pares ordenados:

$$A = \{(x, \mu_A(x)) | x \in X\} \quad [D.4]$$

donde $\mu_A(x)$ es la función de pertenencia al conjunto difuso.

Función de pertenencia: Esta función da para cada elemento de X un grado de membrecía al conjunto A . El valor de esta función está en el intervalo entre 0 y 1, siendo 1 el valor para máxima pertenencia. Si el valor de esta función se restringiera solamente a 0 y 1, se tendría un conjunto clásico, o no-difuso. Esta función no es única. Las funciones utilizadas más frecuentemente son las de tipo trapezoidal, singleton, triangular, S, exponencial, tipo Π (forma de campana). En la figura D.5 se pueden presentar diferentes tipos de funciones de pertenencia.

Apoyo: El apoyo del conjunto difuso A es el conjunto de todos los puntos x para los cuales la función de pertenencia ($\mu_A(x)$) es mayor que cero.

Centro: El centro de un conjunto difuso A es el conjunto de todos los puntos para los cuales la función de pertenencia ($\mu_A(x)$) es igual a 1.

Normalidad: Un conjunto difuso es normal si siempre existe un punto para el cual la función de pertenencia es 1, es decir el *centro* no está vacío.

Puntos de Crossover: Son los puntos del conjunto difuso para los cuales $\mu_A(x)=0.5$

Difuso simple (Singleton): Es el conjunto difuso para el cual el *apoyo* es solamente un punto, en el cual el valor de la función de pertenencia es 1.

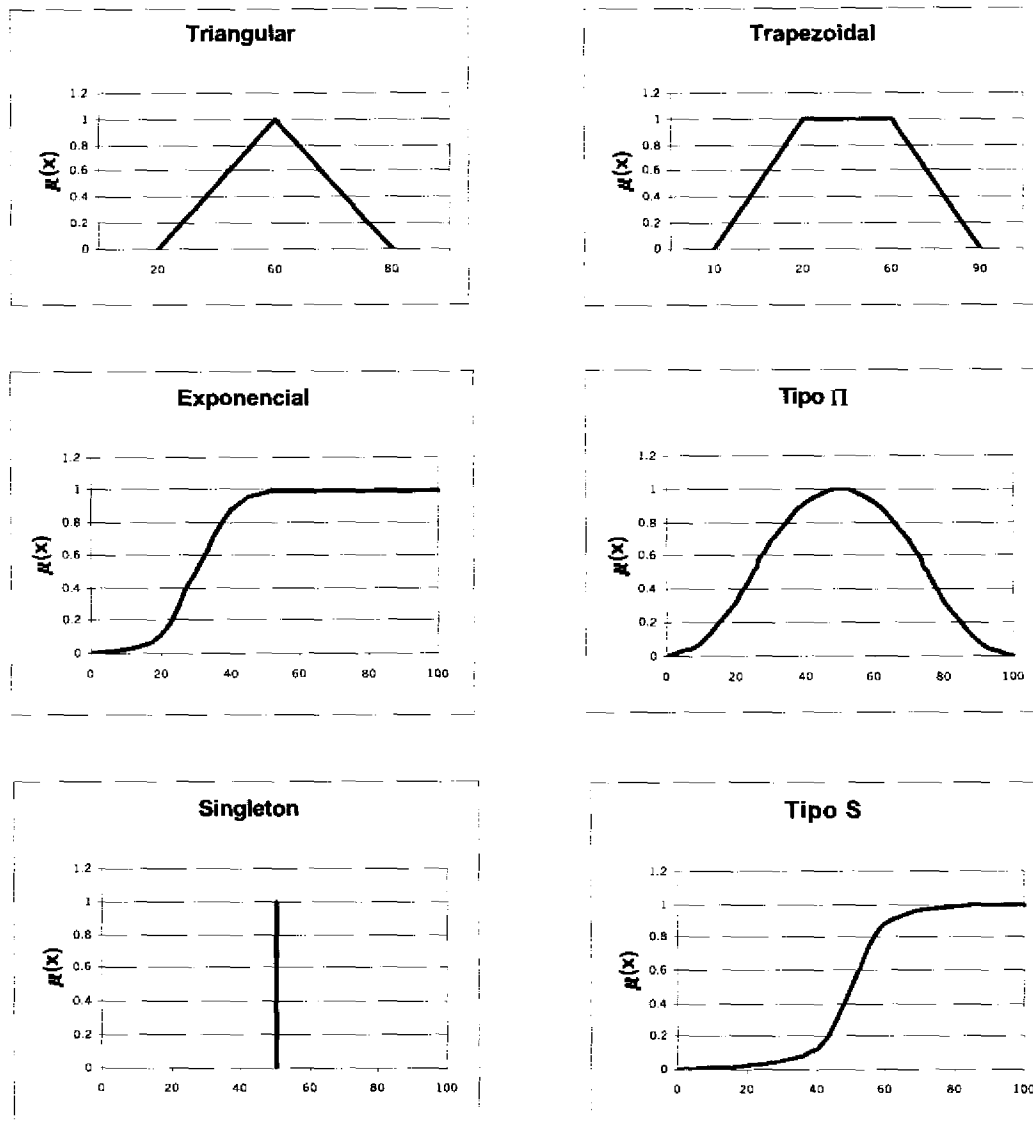


Figura D.5: Tipos de Funciones de Pertencia

En la tabla D.1 se encuentran algunas características matemáticas de los conjuntos difusos.

D.2.2. Operaciones entre conjuntos difusos

En los conjuntos difusos son aplicables las tres operaciones básicas de los conjuntos clásicos (unión, intersección y complemento). En la tabla D.2 se encuentran algunas de las operaciones básicas de los conjuntos difusos.

D.2.3. Desfusificación

Los problemas basados en conjuntos difusos se originan de la utilización de calificaciones lingüísticas. Luego de formar los conjuntos difusos correspondientes a un determinado problema, es necesario obtener una respuesta de estos. En muchos casos es

importante que esta respuesta no sea difusa y se debe pasar de una respuesta difusa a una que no lo es. Para lograr esto se ha desarrollado el concepto de “desfusificación”. Se han propuesto varios métodos para lograr esta transformación; algunos de ellos se explican a continuación. El método de desfusificación debe ser escogido de acuerdo con la aplicación que se desea hacer (Jang 1997).

Tabla D.1: Características matemáticas de los Conjuntos Difusos

Característica Matemática	Definición
α -corte	$A_\alpha = \{x \mu_A(x) \geq \alpha\}$
α -corte marcado	$A'_\alpha = \{x \mu_A(x) > \alpha\}$
Convexidad	Un conjunto es convexo si y solo si para cualesquiera $x_1, x_2 \in X$ y cualquier $\lambda \in [0,1]$ se cumple que: $\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq \min\{\mu_A(x_1), \mu_A(x_2)\}$ Se dice que un conjunto es convexo si todos su conjuntos de nivel α son convexos.
Números Difusos	Es un conjunto difuso en la recta real (R) que satisface las condiciones de normalidad y convexidad.
Ancho de conjuntos difusos normales y convexos	Esta definido como la distancia entre los dos únicos puntos de crossover $\text{Ancho}(A) = x_2 - x_1 $ donde $\mu_A(x_1) = \mu_A(x_2) = 0.5$
Simetría	Un conjunto difuso es simétrico si alrededor de un punto $x=c$ se cumple $\mu_A(c+x) = \mu_A(c-x), \text{ para todo } x \in X$
Conjunto Difuso abierto a la derecha	Un conjunto difuso A es abierto a la derecha si cumple que: $\lim_{x \rightarrow -\infty} \mu_A(x) = 1 \quad \text{y} \quad \lim_{x \rightarrow +\infty} \mu_A(x) = 0$
Conjunto difuso abierto a la izquierda	Un conjunto difuso A es abierto a la izquierda si cumple que $\lim_{x \rightarrow -\infty} \mu_A(x) = 0 \quad \text{y} \quad \lim_{x \rightarrow +\infty} \mu_A(x) = 1$
Conjunto difuso cerrado	Un conjunto difuso A es cerrado si cumple que: $\lim_{x \rightarrow -\infty} \mu_A(x) = \lim_{x \rightarrow +\infty} \mu_A(x) = 0$

Tabla D.2: Operaciones entre conjuntos difusos

Operación	Definición
Contención o Subconjunto	A es subconjunto de B si y solo si $\mu_A(x) \leq \mu_B(x)$, para todo x . $A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x)$
Unión	La unión de los conjuntos difusos A y B es el conjunto difuso C , y se escribe como $C = A \cup B$ ó $C = A \text{ OR } B$, su función de pertenencia está dada por $\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x)$
Intersección	La intersección de los conjuntos difusos A y B es el conjunto difuso C , y se escribe como $C = A \cap B$ ó $C = A \text{ AND } B$, su función de pertenencia está dada por $\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x)$

Tabla D.2: Operaciones entre conjuntos difusos (Continuación)

Operación	Definición
Complemento (negación)	El complemento del conjunto difuso A , denotado por \hat{A} ($\neg A$, $NOT A$), se define como $\mu_{\hat{A}}(x) = 1 - \mu_A(x)$
Producto Cartesiano	Si A y B son conjuntos difusos en X y Y , el producto cartesiano de los conjuntos A y B $A \times B$ en el espacio $X \times Y$ tiene la función de pertenencia $\mu_{A \times B}(x, y) = \min(\mu_A(x), \mu_B(y))$
Co-producto Cartesiano	$A + B$ en el espacio $X \times Y$ tiene la función de pertenencia $\mu_{A+B}(x, y) = \max(\mu_A(x), \mu_B(y))$

D.2.3.1. Centroide de área (COA)

Este método consiste en hallar para cada conjunto difuso su área y la ubicación del centroide. Luego se hace el producto entre ellos y se suman, para posteriormente dividir este resultado por la sumatoria total de las áreas, como se muestra en la ecuación D.5.

$$\text{Valor desfusificado} = \bar{X} = \frac{\sum A_i \bar{x}_i}{\sum A_i} \tag{D.5}$$

$$\text{ó } COA = \frac{\int \mu_A(x) x dx}{\int_x \mu_A(x) dx}$$

D.2.3.2. Procedimiento máximo

Aquí se supone que la función de pertenencia tiene sólo un punto máximo simple, se toma el valor de *desfusificación* en este punto máximo, como se expresa en la ecuación D.6.

$$y_0(B) = \arg \max \{ \mu_B(y) | y \in Y \} \tag{D.6}$$

En caso que la función de pertenencia de la salida tenga varios puntos máximos, se crea un grupo (B_{max}) con estos puntos (soluciones optimas), ecuación B.7.

$$B_{max} = \{ y \in Y | \mu(y) = \max_{z \in Y} \mu_B(z) \} \tag{D.7}$$

De este grupo de máximos se debe obtener un único punto. Esto se puede hacer de varias formas. Un método es escoger aleatoriamente un punto optimo (supone que todas las soluciones son igualmente buenas), pero preferiblemente que sea un punto en la mitad del conjunto de soluciones. La solución puede escogerse también hallando el valor medio del conjunto, si este es un conjunto finito, ecuación D.8.

$$y_0(B) = \frac{1}{N} \sum_{y \in B_{\max}} y \quad [D.8]$$

donde N es el número de elementos en el conjunto.

D.2.3.3. Centro de gravedad

Con este método se tiene en cuenta información relacionada con la función de pertenencia μ_B . Se toma la media de todos los pesos como se indica en la ecuación B.9.

$$y_0(B) = \frac{1}{\sum \mu_B(y)} \sum_{y \in B_{\max}} y \mu_B(y) \quad [D.9]$$

D.2.4 Reglas difusas si-entonces

En una regla difusa se asume como: Si x es A entonces y es B , donde A y B son valores lingüísticos definidos por conjuntos difusos dentro de los universos X y Y . Como en la lógica clásica a menudo “ x es A ” es llamada *premisa* y “ y es B ” es llamada la *conclusión o consecuencia*. Este tipo de reglas puede ser utilizado para modelar y analizar un sistema. Varios métodos para calcular la relación $R = A \rightarrow B$ han sido propuestos. R puede ser visto como un conjunto difuso con una función de pertenencia en dos dimensiones, ecuación D.10.

$$\mu_R(x, y) = f(\mu_A(x), \mu_B(y)) = f(a, b) \quad [D.10]$$

D.2.5. Razonamiento difuso

También llamado razonamiento aproximado es un procedimiento de inferencia que saca conclusiones con reglas *si-entonces* utilizando conjuntos difusos (Jang 1997).

Siendo A , A' , y B conjuntos difusos que pertenecen a X , X , y Y respectivamente. Asume que la implicación difusa $A \rightarrow B$ es expresada como una relación R en $X \times Y$. El conjunto difuso B inducido por “ x es A ” y la regla difusa “si x es A entonces y es B ” están definidos como se indica en la expresión D.11.

$$\begin{aligned} \mu_{B'}(y) &= \max_x \min[\mu_{A'}(x), \mu_R(x, y)] \\ &= \vee_x [\mu_{A'}(x) \wedge \mu_R(x, y)] \end{aligned} \quad [D.11]$$

De acuerdo al problema que se desea resolver se han desarrollado diferentes reglas para el razonamiento difuso, en estas puede variar el número de antecedentes y de consecuencias. Se emplea también diferentes operadores para relacionar los conjuntos difusos como Y , O , *implicación*, *desfusificación*, entre otros.

D.2.6. Aplicaciones

La teoría de conjuntos difusos ha sido amplia mente aplicada en campos como: la Medicina, Economía, Ecología y Biología. Se ha empleado en empresas de producción

de artículos eléctricos y electrónicos como una herramienta de control, se ha utilizado para el desarrollo de procesadores y computadoras.

Los conjuntos difusos son usados para toma de decisiones y estimaciones en Sistemas de Control como son: aire acondicionado, control de automoviles y controladores en sistemas industriales.

El país en el que se ha desarrollado más el uso de la lógica difusa es Japón, donde se encuentra aplicada a múltiple sistemas como son el Metro de la ciudad de Sendai (desde 1987), enfoque automático de cámaras fotográficas y filmadoras digitales, máquinas aspiradoras, máquinas de lavado que evalúan la carga y ajustan por si mismas la temperatura del agua, cantidad de detergente y el ciclo de lavado, televisores SONY que ajustan automáticamente el brillo, el contraste y las tonalidades de color. En tecnología automotriz, en los vehículos Nissan con transmisión automática. Así como también la industria Mitsubishi que la aplica en el control de ascensores. En los Estados Unidos, también, la compañía OTIS también esta desarrollando aplicaciones a esta industria.

En general la lógica difusa es aplicada en cualquier campo donde sea muy difícil o casi imposible crear un modelo, en sistemas controlados por expertos humanos, en sistemas donde se tienen entradas y salidas que son continuas y complejas, en sistemas que utilizan observaciones humanas como entradas o reglas básicas, y en cualquier sistema en el cual se trabaje con conceptos vagos o imprecisos.

En el campo de la Ingeniería Civil, en América Latina, la lógica difusa esta siendo aplicada también en control de cierre de compuertas en presas (Chile), control de tráfico (Puerto Rico), control de secaderos de hoja de tabaco (Cuba), control de balanceo en puentes grúa control de nivel de líquidos en contenedores y se espera que su aplicación se generalizará de manera muy notable en los próximos años.

D.3. Referencias recomendadas

Chapman, O.J.V., Crossland, A.D. (1995): Neural Network in Probabilistic Structural Mechanics, *Probabilistic Structural Engineering Handbook. Theory and Industrial Applications*, Sundararajan, C. (Ed.). Chapman and Hall, New York.

Hadipriono, F.C. (1995): Fuzzy Sets in Probabilistic Structural Mechanics, *Probabilistic Structural Engineering Handbook. Theory and Industrial Applications*, Sundararajan, C. (Ed.). Chapman and Hall, New York.

Hines, J.W. (1997): Fuzzy y Neural Approaches in Engineering, John Wiley & Sons. New York.

Jang, J.-S.R., Sun, C.-T., Mizutani, E. (1999): Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence. Prentice Hall, New York.

Leondes C.T., (Ed.) (1998): *Fuzzy Logic And Expert Systems Applicatios*, Academic Press, San Diego, California.

Looney, C.G. (1997): Pattern Recognition using Neural Networks. Theory and Algorithms for Engineers and Scientists.

- Maldonado R., E, Casas, J.R., Canas, J.A. (1999): Utilización de los Conjuntos Difusos en Modelos de Vulnerabilidad Sísmica. *Monografías de Ingeniería Sísmica*. A. H. Barbat (Ed.). Centro Internacional de Métodos Numéricos en Ingeniería, UPC, Barcelona.
- Peña, C.A. (2000): *Evolutionary Fuzzy Modeling*, Ecole Polytechnique Federale de Lausanne, Logical Systems Laboratory, Universidad de los Andes, Bogotá.
- Paolucci, R., Colli, P., Giacinto, G. (2000): Assessment of Seismic Site Effects in 2-D Alluvial Valleys Using Neural Networks. *Earthquake Spectra*. Earthquake Engineering Research Institute, EERI, Vol.16, No. 3., August.

Apéndice E

Código fuente del sistema RSU

Se presenta a continuación el programa de ordenador Estimación Holística del Riesgo Sísmico en Centros Urbanos, RSU, que utiliza el modelo basado en una red neuronal y operaciones con conjuntos difusos. Esta aplicación se desarrolló mediante la técnica de *software* basado en objetos, en lenguaje Visual Basic 5.0

FORM 1

```
Private Sub btnAceleracion_Click()  
variable = 18  
ContadorVariable = ContadorVariable + 1  
contador2 = ContadorVariable  
comprobacion (variable)  
btnAceleracion.FontBold = False  
If ncal = 1 Then  
Form6.lblTipo.Caption = "ACELERACIÓN ESPECTRAL EN  
PERIODOS CORTOS"  
Form6.Option1(0).Caption = "Muy Baja"  
Form6.Option1(1).Caption = "Baja"  
Form6.Option1(2).Caption = "Moderada"  
Form6.Option1(3).Caption = "Alta"  
Form6.Option1(4).Caption = "Muy Alta"  
Form6.Show 1  
End If  
End Sub
```

```
Private Sub btnAcueducto_Click()  
variable = 23  
ContadorVariable = ContadorVariable + 1  
contador2 = ContadorVariable  
comprobacion (variable)  
btnAcueducto.FontBold = False  
If ncal = 1 Then  
Form6.lblTipo.Caption = "ROTURA RED DE ACUEDUCTO"  
Form6.Option1(0).Caption = "Muy Pocas"  
Form6.Option1(1).Caption = "Pocas"  
Form6.Option1(2).Caption = "Numerosas"  
Form6.Option1(3).Caption = "Muchas"  
Form6.Option1(4).Caption = "Excesivas"  
Form6.Show 1  
End If  
End Sub
```

```
Private Sub btnAreaDestruida_Click()  
variable = 26  
ContadorVariable = ContadorVariable + 1  
contador2 = ContadorVariable  
comprobacion (variable)  
btnAreaDestruida.FontBold = False  
If ncal = 1 Then  
Form6.lblTipo.Caption = "AREA DESTRUIDA"  
Form6.Option1(0).Caption = "Muy Poca"  
Form6.Option1(1).Caption = "Poca"  
Form6.Option1(2).Caption = "Considerable"  
Form6.Option1(3).Caption = "Muchas"
```

```
Form6.Option1(4).Caption = "Excesiva"  
Form6.Show 1  
End If  
End Sub
```

```
Private Sub btnBlandos_Click()  
variable = 17  
ContadorVariable = ContadorVariable + 1  
contador2 = ContadorVariable  
comprobacion (variable)  
btnBlandos.FontBold = False  
If ncal = 1 Then  
Form6.lblTipo.Caption = "AREA DE SUELOS BLANDOS"  
Form6.Option1(0).Caption = "Mínima"  
Form6.Option1(1).Caption = "Escasa"  
Form6.Option1(2).Caption = "Considerable"  
Form6.Option1(3).Caption = "Extensa"  
Form6.Option1(4).Caption = "Total"  
Form6.Show 1  
End If  
End Sub
```

```
Private Sub btnCamas_Click()  
variable = 5  
ContadorVariable = ContadorVariable + 1  
contador2 = ContadorVariable  
comprobacion (variable)  
btnCamas.FontBold = False  
If ncal = 1 Then  
Form6.lblTipo.Caption = "CAMAS HOSPITALARIAS"  
Form6.Option1(0).Caption = "Excesivas"  
Form6.Option1(1).Caption = "Muchas"  
Form6.Option1(2).Caption = "Considerables"  
Form6.Option1(3).Caption = "Pocas"  
Form6.Option1(4).Caption = "Insignificantes"  
Form6.Show 1  
End If  
End Sub
```

```
Private Sub btnConstruida_Click()  
variable = 12  
ContadorVariable = ContadorVariable + 1  
contador2 = ContadorVariable  
comprobacion (variable)  
btnConstruida.FontBold = False  
If ncal = 1 Then  
Form6.lblTipo.Caption = "AREA CONSTRUIDA"  
Form6.Option1(0).Caption = "Mínima"  
Form6.Option1(1).Caption = "Escasa"
```

```

Form6.Option1(2).Caption = "Considerable"
Form6.Option1(3).Caption = "Extensa"
Form6.Option1(4).Caption = "Total"
Form6.Show 1
End If
End Sub

```

```

Private Sub btnDelincuencia_Click()
variable = 7
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnDelincuencia.FontBold = False
If ncali = 1 Then
Form6.lblTipo.Caption = "TASA DE DELINCUENCIA"
Form6.Option1(0).Caption = "Mínima"
Form6.Option1(1).Caption = "Pequeña"
Form6.Option1(2).Caption = "Significativa"
Form6.Option1(3).Caption = "Alta"
Form6.Option1(4).Caption = "Muy Alta"
Form6.Show 1
End If
End Sub

```

```

Private Sub btnDensidad_Click()
variable = 13
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnDensidad.FontBold = False
If ncali = 1 Then
Form6.lblTipo.Caption = "DENSIDAD POBLACIONAL"
Form6.Option1(0).Caption = "Mínima"
Form6.Option1(1).Caption = "Pequeña"
Form6.Option1(2).Caption = "Significativa"
Form6.Option1(3).Caption = "Alta"
Form6.Option1(4).Caption = "Muy Alta"
Form6.Show 1
End If
End Sub

```

```

Private Sub btnDesarrollo_Click()
variable = 1
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnDesarrollo.FontBold = False
If ncali = 1 Then
Form6.lblTipo.Caption = "DESARROLLO Y ORGANIZACION
DE LA COMUNIDAD"
Form6.Option1(0).Caption = "Muy Alto"
Form6.Option1(1).Caption = "Alto"
Form6.Option1(2).Caption = "Significativo"
Form6.Option1(3).Caption = "Escaso"
Form6.Option1(4).Caption = "Mínimo"
Form6.Show 1
End If
End Sub

```

```

Private Sub btnDeslizamiento_Click()
variable = 15
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnDeslizamiento.FontBold = False
If ncali = 1 Then
Form6.lblTipo.Caption = "AREA CON SUSCEPTIBILIDAD A
OTROS EFECTOS"
Form6.Option1(0).Caption = "Mínima"
Form6.Option1(1).Caption = "Poca"
Form6.Option1(2).Caption = "Importante"
Form6.Option1(3).Caption = "Amplia"
Form6.Option1(4).Caption = "Extensa"
Form6.Show 1
End If
End Sub

```

```

Private Sub btnDisparidad_Click()
variable = 6
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnDisparidad.FontBold = False
If ncali = 1 Then
Form6.lblTipo.Caption = "INDICE DE DISPARIDAD SOCIAL"
Form6.Option1(0).Caption = "Mínimo"

```

```

Form6.Option1(1).Caption = "Pequeño"
Form6.Option1(2).Caption = "Significativo"
Form6.Option1(3).Caption = "Alto"
Form6.Option1(4).Caption = "Muy Alto"
Form6.Show 1
End If
End Sub

```

```

Private Sub btnElectricas_Click()
variable = 19
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnElectricas.FontBold = False
If ncali = 1 Then
Form6.lblTipo.Caption = "SUBESTACIONES ELECTRICAS
AFECTADAS"
Form6.Option1(0).Caption = "Muy Pocas"
Form6.Option1(1).Caption = "Pocas"
Form6.Option1(2).Caption = "Numerosas"
Form6.Option1(3).Caption = "Muchas"
Form6.Option1(4).Caption = "Excesivas"
Form6.Show 1
End If
End Sub

```

```

Private Sub btnEspacioPublico_Click()
variable = 3
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnEspacioPublico.FontBold = False
If ncali = 1 Then
Form6.lblTipo.Caption = "ESPACIO PUBLICO DISPONIBLE"
Form6.Option1(0).Caption = "Extenso"
Form6.Option1(1).Caption = "Amplio"
Form6.Option1(2).Caption = "Importante"
Form6.Option1(3).Caption = "Poco"
Form6.Option1(4).Caption = "Mínimo"
Form6.Show 1
End If
End Sub

```

```

Private Sub btnFallecidos_Click()
variable = 25
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnFallecidos.FontBold = False
If ncali = 1 Then
Form6.lblTipo.Caption = "FALLECIDOS"
Form6.Option1(0).Caption = "Muy Pocas"
Form6.Option1(1).Caption = "Pocas"
Form6.Option1(2).Caption = "Numerosas"
Form6.Option1(3).Caption = "Muchos"
Form6.Option1(4).Caption = "Excesivos"
Form6.Show 1
End If
End Sub

```

```

Private Sub btnGas_Click()
variable = 22
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnGas.FontBold = False
If ncali = 1 Then
Form6.lblTipo.Caption = "ROTURA RED DE GAS"
Form6.Option1(0).Caption = "Muy Pocas"
Form6.Option1(1).Caption = "Pocas"
Form6.Option1(2).Caption = "Numerosas"
Form6.Option1(3).Caption = "Muchas"
Form6.Option1(4).Caption = "Excesivas"
Form6.Show 1
End If
End Sub

```

```

Private Sub btnHeridos_Click()
variable = 24
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnHeridos.FontBold = False
If ncali = 1 Then
Form6.lblTipo.Caption = "HERIDOS"
Form6.Option1(0).Caption = "Muy Pocas"

```



```

Form6.Option1(1).Caption = "Pocos"
Form6.Option1(2).Caption = "Numerosos"
Form6.Option1(3).Caption = "Muchos"
Form6.Option1(4).Caption = "Excesivos"
Form6.Show 1
End If
End Sub

Private Sub btnIndustrial_Click()
variable = 11
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnIndustrial.FontBold = False
If ncal1 = 1 Then
Form6.lblTipo.Caption = "AREA INDUSTRIAL"
Form6.Option1(0).Caption = "Mínima"
Form6.Option1(1).Caption = "Poca"
Form6.Option1(2).Caption = "Importante"
Form6.Option1(3).Caption = "Amplia"
Form6.Option1(4).Caption = "Extensa"
Form6.Show 1
End If
End Sub

Private Sub btnInstitucional_Click()
variable = 10
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnInstitucional.FontBold = False
If ncal1 = 1 Then
Form6.lblTipo.Caption = "AREA INSTITUCIONAL"
Form6.Option1(0).Caption = "Mínima"
Form6.Option1(1).Caption = "Poca"
Form6.Option1(2).Caption = "Importante"
Form6.Option1(3).Caption = "Amplia"
Form6.Option1(4).Caption = "Extensa"
Form6.Show 1
End If
End Sub

Private Sub btnLicuacion_Click()
variable = 16
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnLicuacion.FontBold = False
If ncal1 = 1 Then
Form6.lblTipo.Caption = "AREA CON POTENCIAL DE
LICUACION"
Form6.Option1(0).Caption = "Mínima"
Form6.Option1(1).Caption = "Escasa"
Form6.Option1(2).Caption = "Considerable"
Form6.Option1(3).Caption = "Extensa"
Form6.Option1(4).Caption = "Total"
Form6.Show 1
End If
End Sub

Private Sub btnMarginales_Click()
variable = 9
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
btnMarginales.FontBold = False
If ncal1 = 1 Then
Form6.lblTipo.Caption = "AREA DE BARRIOS MARGINALES"
Form6.Option1(0).Caption = "Mínima"
Form6.Option1(1).Caption = "Escasa"
Form6.Option1(2).Caption = "Considerable"
Form6.Option1(3).Caption = "Extensa"
Form6.Option1(4).Caption = "Total"
Form6.Show 1
End If
End Sub

Private Sub btnMenu_Click(Index As Integer)
'Declaracion de Variables temporales
Dim ap As Integer
Dim OPC As Integer
Dim acceso As String
Dim ubicacion As String
Dim vacmas As String
Dim i As Integer
n = 0
Loc = 0
If Index = 0 Then
'Boton Guardar
definidos 'ver si todos los pesos han sido ingresados
If n = 1 Then
MsgBox "Ingrese todos los pesos"
Else
'Desca guardar o corregir los datos
OPC = MsgBox("Todos los datos ingresados son correctos?",
vbYesNo, "Datos Ingresados")
If OPC = vbYes Then
'Determinacion del número del último registro
Form2.dtaResultados.Recordset.MoveLast
ultimoreg = Form2.dtaResultados.Recordset.Fields(0).Value
ultimoreg = ultimoreg + 1
'Determinacion del número del último registro de pesos
Form2.DtaPesos.Recordset.MoveLast
ultimopesos = Form2.DtaPesos.Recordset.Fields(0).Value
ultimopesos = ultimopesos + 1
' llamado de funcion guardar
ap = guardarpesos(ultimopesos, vecentrada)
' llamado a la funcion de evaluacion total
a = guardacalRiesgo(nmp)
b = guardacalAmenaza()
c = guardacalExposicion()
d = guardacalFragilidad()
e = guardacalResistencia()
evaluatotal (nump)
'cargar formulario de resultados
Form7.Show 1
'Descargar formulario de Nueva Evaluacion
Unload Form1
End If
End If
Else
'Boton Cancelar
OPC = MsgBox("Está usted seguro que desea Salir ? Perdiera la
informacion suministrada hasta ahora ", vbYesNo, "Datos
Ingresados")
If OPC = vbYes Then
Unload Form1
End If
End If
End Sub

Private Sub btnMortalidad_Click()
variable = 8
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
'Guardar numero de la variable
'vecentrada(contador2) variable = variable
btnMortalidad.FontBold = False
If ncal1 = 1 Then
Form6.lblTipo.Caption = "TASA DE MORTALIDAD"
Form6.Option1(0).Caption = "Mínima"
Form6.Option1(1).Caption = "Pequeña"
Form6.Option1(2).Caption = "Significativa"
Form6.Option1(3).Caption = "Alta"
Form6.Option1(4).Caption = "Muy Alta"
Form6.Show 1
End If
End Sub

Private Sub btnMovilidad_Click()
variable = 28
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable
comprobacion (variable)
'Guardar numero de la variable
btnMovilidad.FontBold = False
If ncal1 = 1 Then
Form6.lblTipo.Caption = "MOVILIDAD POR LA RED VIAL"
Form6.Option1(0).Caption = "Muy Baja"
Form6.Option1(1).Caption = "Baja"
Form6.Option1(2).Caption = "Moderada"
Form6.Option1(3).Caption = "Alta"
Form6.Option1(4).Caption = "Muy Alta"
Form6.Show 1
End If
End Sub

Private Sub btnOperatividad_Click()
variable = 0
ContadorVariable = ContadorVariable + 1
contador2 = ContadorVariable

```

```

comprobacion (variable)
'Guardar numero de la variable
btnOperatividad.FontBold = False
If ncali = 1 Then
    Form6.lblTipo.Caption = "CAPACIDAD OPERATIVA EN EMERGENCIA"
    Form6.Option1(0).Caption = "Muy Alta"
    Form6.Option1(1).Caption = "Alta"
    Form6.Option1(2).Caption = "Notable"
    Form6.Option1(3).Caption = "Reducida"
    Form6.Option1(4).Caption = "Minima"
    Form6.Show 1
End If
End Sub

Private Sub btnPesAmenaza_Click()
Dim OPC As Integer
Dim I As Integer
btnPesAmenaza.FontBold = False
If ame = 0 Then
    Form4.hsPesAceleracion.Value = 44
    Form4.hsPesASueBlandos.Value = 24
    Form4.hsPesAPotLicuacion.Value = 16
    Form4.hsPesASuscOtrosEfectos.Value = 16
    Form4.lblPesAceleracion.Caption = Str$(Form4.hsPesAceleracion.Value)
    Form4.lblPesAPotLicuacion.Caption = Str$(Form4.hsPesAPotLicuacion.Value)
    Form4.lblPesASueBlandos.Caption = Str$(Form4.hsPesASueBlandos.Value)
    Form4.lblPesASuscOtrosEfectos.Caption = Str$(Form4.hsPesASuscOtrosEfectos.Value)
    Form4.frmPesRiesgo.Visible = False
    Form4.frmPesAmenaza.Visible = True
    Form4.Show 1
Else
    OPC = MsgBox("Ud. ya ha ingresado los pesos o influencias de las variables de AMENAZA SISMICA, desea corregirlos?", vbYesNo)
    If OPC = vbYes Then
        For I = 0 To 28
            If vcentrada(I).variable = 15 Then
                Form4.hsPesASuscOtrosEfectos.Value = vcentrada(I).peso(3)
            ElseIf vcentrada(I).variable = 16 Then
                Form4.hsPesAPotLicuacion.Value = vcentrada(I).peso(3)
            ElseIf vcentrada(I).variable = 17 Then
                Form4.hsPesASueBlandos.Value = vcentrada(I).peso(3)
            ElseIf vcentrada(I).variable = 18 Then
                Form4.hsPesAceleracion.Value = vcentrada(I).peso(3)
            End If
        Next I
        Form4.lblPesAceleracion.Caption = Str$(Form4.hsPesAceleracion.Value)
        Form4.lblPesAPotLicuacion.Caption = Str$(Form4.hsPesAPotLicuacion.Value)
        Form4.lblPesASueBlandos.Caption = Str$(Form4.hsPesASueBlandos.Value)
        Form4.lblPesASuscOtrosEfectos.Caption = Str$(Form4.hsPesASuscOtrosEfectos.Value)
        Form4.frmPesRiesgo.Visible = False
        Form4.frmPesAmenaza.Visible = True
        Form4.Show 1
    End If
End If
End Sub

Private Sub btnPesExposicion_Click()
Dim OPC As Integer
Dim I As Integer
btnPesExposicion.FontBold = False
If expo = 0 Then
    Form9.hsPesPoblacion.Value = 25
    Form9.hsPesDensPoblacional.Value = 20
    Form9.hsPesAreaIndustrial.Value = 25
    Form9.hsPesAreaConstruida.Value = 15
    Form9.hsPesAreaInstitucional.Value = 15
    Form9.lblPesPoblacion.Caption = Str$(Form9.hsPesPoblacion.Value)
    Form9.lblPesDensPoblacional.Caption = Str$(Form9.hsPesDensPoblacional.Value)
    Form9.lblPesAreaIndustrial.Caption = Str$(Form9.hsPesAreaIndustrial.Value)
    Form9.lblPesAreaConstruida.Caption = Str$(Form9.hsPesAreaConstruida.Value)
    Form9.lblPesAreaInstitucional.Caption = Str$(Form9.hsPesAreaInstitucional.Value)
    Form9.Show 1
Else
    OPC = MsgBox("Ud. ya ha ingresado los pesos o influencias de las variables de EXPOSICIÓN DEL CONTEXTO, desea corregirlos?", vbYesNo)
    If OPC = vbYes Then
        For I = 0 To 28
            If vcentrada(I).variable = 10 Then
                Form9.hsPesAreaInstitucional.Value = vcentrada(I).peso(2)
            ElseIf vcentrada(I).variable = 11 Then
                Form9.hsPesAreaConstruida.Value = vcentrada(I).peso(2)
            ElseIf vcentrada(I).variable = 12 Then
                Form9.hsPesAreaIndustrial.Value = vcentrada(I).peso(2)
            ElseIf vcentrada(I).variable = 13 Then
                Form9.hsPesDensPoblacional.Value = vcentrada(I).peso(2)
            ElseIf vcentrada(I).variable = 14 Then
                Form9.hsPesPoblacion.Value = vcentrada(I).peso(2)
            End If
        Next I
        Form9.lblPesPoblacion.Caption = Str$(Form9.hsPesPoblacion.Value)
        Form9.lblPesDensPoblacional.Caption = Str$(Form9.hsPesDensPoblacional.Value)
        Form9.lblPesAreaIndustrial.Caption = Str$(Form9.hsPesAreaIndustrial.Value)
        Form9.lblPesAreaConstruida.Caption = Str$(Form9.hsPesAreaConstruida.Value)
        Form9.lblPesAreaInstitucional.Caption = Str$(Form9.hsPesAreaInstitucional.Value)
        Form9.Show 1
    End If
End Sub

Private Sub btnPesFragilidad_Click()
Dim OPC As Integer
Dim I As Integer
btnPesFragilidad.FontBold = False
If frag = 0 Then
    Form5.hsPesABarrMargin.Value = 40
    Form5.hsPesTasaMortalidad.Value = 10
    Form5.hsPesTasDelincuencia.Value = 10
    Form5.hsPesIndDisparidad.Value = 40
    Form5.lblPesABarrMargin.Caption = Str$(Form5.hsPesABarrMargin.Value)
    Form5.lblPesTasaMortalidad.Caption = Str$(Form5.hsPesTasaMortalidad.Value)
    Form5.lblPesTasDelincuencia.Caption = Str$(Form5.hsPesTasDelincuencia.Value)
    Form5.lblPesIndDisparidad.Caption = Str$(Form5.hsPesIndDisparidad.Value)
    Form5.frmPesResiliencia.Visible = False
    Form5.frmFragilidad.Visible = True
    Form5.Show 1
Else
    OPC = MsgBox("Ud. ya ha ingresado los pesos o influencias de las variables de FRAGILIDAD SOCIAL, desea corregirlos?", vbYesNo)
    If OPC = vbYes Then
        For I = 0 To 28
            If vcentrada(I).variable = 6 Then
                Form5.hsPesIndDisparidad.Value = vcentrada(I).peso(1)
            ElseIf vcentrada(I).variable = 7 Then
                Form5.hsPesTasDelincuencia.Value = vcentrada(I).peso(1)
            ElseIf vcentrada(I).variable = 8 Then
                Form5.hsPesTasaMortalidad.Value = vcentrada(I).peso(1)
            ElseIf vcentrada(I).variable = 9 Then
                Form5.hsPesABarrMargin.Value = vcentrada(I).peso(1)
            End If
        Next I
        Form5.lblPesABarrMargin.Caption = Str$(Form5.hsPesABarrMargin.Value)
        Form5.lblPesTasaMortalidad.Caption = Str$(Form5.hsPesTasaMortalidad.Value)
        Form5.lblPesTasDelincuencia.Caption = Str$(Form5.hsPesTasDelincuencia.Value)
        Form5.lblPesIndDisparidad.Caption = Str$(Form5.hsPesIndDisparidad.Value)
        Form5.frmPesResiliencia.Visible = False
        Form5.frmFragilidad.Visible = True
        Form5.Show 1
    End If
End Sub

Private Sub btnPesResiliencia_Click()
Dim OPC As Integer
Dim I As Integer

```