



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y  
COMPUTACIÓN**

**TESIS DE GRADO**

**“SISTEMA DE CLASIFICACIÓN DE LOS CUESTIONARIOS LLENADOS  
EN EL PROCESO DE ENCUESTA QUE SE REALIZAN PARA LA  
EVALUACIÓN DEL PROFESOR MEDIANTE EL USO DE REDES  
NEURONALES”**

**Previa a la obtención del título de:**

**INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN:  
SISTEMAS DE INFORMACIÓN /  
SISTEMAS TECNOLÓGICOS**

**PRESENTADA POR:**

**DIEGO ALEJANDRO CARRERA GALLEGO  
JOHN GUILLERMO PALOMEQUE AVILA**

**GUAYAQUIL - ECUADOR**

**2007**

## **AGRADECIMIENTO**

*A Dios, nuestros padres,  
y a nuestro director que  
ha sido un gran apoyo  
en cada uno de los  
avances y resultados  
obtenido, siendo bases  
importantes para realizar  
este trabajo.*

## **DEDICATORIA**

*A mis padres que durante muchos años de constante apoyo, consideraciones e incentivos me han ayudado en mi educación universitaria, siendo unos de mis mejores logros tanto personal como profesional.*

*Diego Alejandro Carrera Gallego.*

*A Dios como principal apoyo de mi vida, mis padres como ejemplos de perseverancia, luchadores, a mis hermanos, a los docentes que dejaron semilla durante el paso que tuve en mi trayectoria como estudiante, en fin a cada uno de los que intervinieron en mi formación para llegar a ser un profesional.*

*John Guillermo Palomeque Avila*

**TRIBUNAL DE GRADO**

---

Ing. Holger Cevallos  
SUBDECANO DE LA FIEC  
PRESIDENTE

---

Msc. Fabricio Echeverría B.  
DIRECTOR DE TESIS

---

Ph.D. Katerine Chiliza  
MIEMBRO DEL TRIBUNAL

---

Ph.D. Cristóbal Mera  
MIEMBRO DEL TRIBUNAL

## **DECLARACIÓN EXPRESA**

“La responsabilidad por los hechos, ideas y doctrinas expuestas en esta tesis, nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”

(Reglamento de exámenes y títulos profesionales de la ESPOL)

---

Diego Alejandro Carrera Gallego

---

John Guillermo Palomeque Avila

## **RESUMEN**

El proyecto de tesis tiene como objetivo la automatización de la clasificación de los formularios válidos y no válidos que han sido evaluados por los estudiantes durante el proceso de la evaluación docente realizada en la ESPOL.

La necesidad de la clasificación de estos formularios ha sido detectada durante el transcurso de la implementación de un nuevo sistema de información denominado Censo Académico en Línea - CENACAD, que es utilizado para la evaluación docente en la Escuela Superior Politécnica del Litoral.

En este proceso de evaluación docente, existen estudiantes con características desconocidas cuyos formularios evaluados producen un sesgo en la evaluación docente de los profesores.

En la actualidad estos procesos de clasificación se lo pueden realizar de forma manual, únicamente por un experto en el tema, sin embargo estos procesos conllevan mucha subjetividad ya que depende mucho de la opinión personal del experto, el cual se basa en su experiencia y en pocas ocasiones

por el resultado de uno de los procesos más sencillos que hay para la clasificación de formularios.

Como parte de desarrollo de la solución a nuestro problema, se ha procedido a investigar los diferentes métodos de clasificación existentes en la actualidad, luego del mismo se da inicio al desarrollo de la solución de la tesis usando el método más confiable que determina la teoría, que es utilizando las redes neuronales artificiales (RNAs) como método de clasificación.

Para tener una mejor abstracción de cómo se ha estructurado los contenidos de la tesis se ha dividido la tesis en cinco capítulos.

El primer capítulo introduce al conocimiento de las Redes Neuronales Artificiales (RNA), dando una especificación general de los modelos basados en el tipo de aprendizaje que presentan las mismas y su clasificación, además de los tipos de aprendizaje que se presentan en las Redes Neuronales modernas.

El segundo capítulo presenta el análisis del proceso de aprendizaje para la clasificación de patrones empleando las Redes Neuronales, los diferentes elementos que constituyen la Red, incluye también las especificaciones que

se presentan durante el proceso y la obtención de patrones para el aprendizaje.

El tercer capítulo abarca el proceso de clasificación de formularios, las recomendaciones para obtener un buen clasificador, así como los requerimientos tecnológicos para el funcionamiento del mismo.

El capítulo cuarto presenta el análisis y diseño del sistema, el análisis y diseño del modelo de datos, los diferentes módulos que intervienen, la relación que existe entre módulos y los procesos que se ejecutan dentro de cada módulo.

Por último, en el quinto capítulo se expone los resultados que se obtienen durante el proceso de clasificación, se describe los resultados presentados en el mapa de neuronas, además de comparaciones que se realizaron con otros métodos de clasificación.



## **INDICE GENERAL**

AGRADECIMIENTO .....	ii
DEDICATORIA .....	iii
DECLARACIÓN EXPRESA.....	v
RESUMEN .....	vi
INDICE GENERAL .....	ix
INDICE DE FIGURAS.....	xi
INDICE DE ECUACIONES.....	xiv
INDICE DE TABLAS.....	xv
INTRODUCCIÓN.....	1
CAPÍTULO 1.....	4
1 ESPECIFICACIONES DE LAS REDES NEURONALES.....	4
1.1 Clasificación de las Redes Neuronales Artificiales.....	8
1.1.1 Modelo Supervisado.....	10
1.1.2 Modelo No Supervisado.....	17
1.1.3 Aprendizaje Asociativo.....	21
1.1.4 Redes Competitivas.....	32
CAPÍTULO 2.....	57
2 REDES NEURONALES ARTIFICIALES APLICADO EN LA CLASIFICACION DE FORMULARIOS.....	57
2.1 Definición de la Red Neuronal Artificial y términos a usarse.....	57
2.2 Análisis y elección de la topología de la Red Neuronal Artificial.....	63
2.3 Justificación del tipo de Red Neuronal Artificial.....	67
2.4 Estructura de la Red Neuronal Artificial.....	70
2.5 Funcionamiento de la Red Neuronal Artificial.....	76
2.6 Entrenamiento de la Red Neuronal Artificial.....	78
2.7 Comparación del uso de la Red vs. otras soluciones para clasificación.....	81
CAPÍTULO 3.....	87
3 PROCESO DE CLASIFICACION DE LOS FORMULARIOS.....	87
3.1 Descripción del proceso de clasificación.....	87
3.2 Consideraciones para obtener un buen clasificador.....	91
3.3 Ejemplo de clasificación.....	93
3.4 Modelo(s) Gráfico(s) de la clasificación.....	97
3.5 Componentes que intervienen en la clasificación.....	100
CAPÍTULO 4.....	102
4 DESCRIPCION DEL SISTEMA DE CLASIFICACION DE LOS FORMULARIOS .....	102
4.1 Componentes ó módulos del sistema.....	102

4.2	Análisis y Diseño del Sistema.....	103
4.3	Análisis y Diseño del modelo de datos.....	110
4.4	Módulos externos al Sistema.....	111
4.5	Diagrama explicativo de la relación entre módulos.....	113
4.6	Descripción del funcionamiento de cada módulo.....	113
CAPÍTULO 5.....		116
5	PATRONES Y RESULTADOS.....	116
5.1	Resultados que se obtienen en el sistema.....	116
5.2	Reportes de entrega del sistema.....	123
5.3	Mapa de la red neuronal.....	127
5.4	Descripción del Mapa.....	127
Conclusiones y recomendaciones.....		129
POSIBLES INNOVACIONES.....		131
ANEXOS.....		132
Anexo 1: Diagramas de Interacción de Objetos del Sistema.....		132
Anexo 2: Diccionario de Datos del modelo de la base de datos.....		144
Anexo 3: Instalación del sistema.....		150
Anexo 4: Instalación del sistema en Linux.....		167
Anexo 5: Manual de Usuario.....		173
Anexo 6: Manual del programador.....		187
Anexo 7: Requerimientos.....		190
Anexo 8: Manual operacional.....		191
Anexo 9: Limitaciones en el uso del Sistema.....		202
REFERENCIAS.....		203

## **INDICE DE FIGURAS**

Figura 1. Representación de una Neurona Biológica. (1).....	5
Figura 2. Representación de una Neurona Artificial. ....	6
Figura 3. Representación de redes unidireccionales y realimentadas. ....	9
Figura 4. Entradas y Salidas de una neurona .....	12
Figura 5. Función de transferencia Hardlim. ....	13
Figura 6. Función de transferencia Hardlims .....	14
Figura 7. Función de transferencia lineal .....	15
Figura 8. Función de transferencia sigmoideal .....	16
Figura 9. Representación de imágenes elípticas .....	20
Figura 10. Asociador Lineal. (6) .....	24
Figura 11. Red Instar. (6).....	26
Figura 12. Representación gráfica de la regla de Instar .....	29
Figura 13. Red Outstar. (6).....	30
Figura 14. Conexiones de una red de Kohonen.....	38
Figura 15. Posible evolución de la vecindad en una red de Kohonen. ....	39
Figura 16. on-center/off-surround; para capas biológicas.....	41
Figura 17. Vecindarios. (6) .....	42
Figura 18. Mapa de auto organización (6) .....	43
Figura 19. Red LVQ. (6) .....	43
Figura 20. Comportamiento de las neuronas en una red LVQ .....	45
Figura 21. Red de Hamming .....	49
Figura 22. Instar agrupadas en una capa .....	53
Figura 23. Capa Competitiva.....	54
Figura 24. Representación gráfica de la regla de Kohonen.....	56
Figura 25. Conexiones de una red de Kohonen. (6).....	72
Figura 26. Función gaussiana de distancia de las neuronas. $H(i,k)$ (6).....	73
Figura 27. Mapa de convergencia topológica. ....	76
Figura 28. Distancia Mahalanobis vs Euclidiana.....	85
Figura 29. Ejemplo de clusterización, vectores de entrada. ....	94
Figura 30. Ejemplo de clusterización, vectores de entrada y vector de pesos (6).....	94
Figura 31. Proceso de entrenamiento (6).....	95
Figura 32. Pesos Finales.....	96
Figura 33. Selección de formularios.....	97
Figura 34. Creación de Mapa, en el diagrama 5 x 5 .....	98
Figura 35. Propiedades del Mapa (Patrones de Conglomerado) .....	98
Figura 36. Agrupamiento Total .....	99
Figura 37. Presentación de Formularios Válidos vs. No Válidos .....	99
Figura 38. Componentes del proceso de Clasificación.....	101
Figura 39. Modelo de Base de Datos.....	110
Figura 40. Relación de Módulos del Sistema.....	113
Figura 41. Iteración vs. Error Neuronal (matriz 6x6). ....	117

Figura 42. Iteración vs. Error Neuronal (matriz 8x8).....	118
Figura 43. Iteración vs. Error Neuronal (matriz 12x12).....	119
Figura 44. Presentación de Formularios Válidos vs. No Válidos acorde al rango establecido para la optimización .....	121
Figura 45. Administrador de Discriminadores .....	123
Figura 46. Reporte de Mapa Neuronal de Kohonen .....	124
Figura 47. Pantalla de confirmación para calcular los formularios válidos ..	124
Figura 48. Reporte de Resultados Clasificados .....	125
Figura 49. Muestra el detalle del reporte del discriminador neuronal .....	126
Figura 50. Listado de Formularios Válidos vs. No Válidos.....	126
Figura 51. Mapa de la Red Neuronal. ....	127
Figura 52. Iteración vs. Error (Proyecto R).....	128
Figura 53. Diagrama de secuencia del escenario 1.1 .....	132
Figura 54. Diagrama de secuencia del escenario 1.2.....	133
Figura 55. Diagrama de secuencia del escenario 1.3.....	134
Figura 56. Diagrama de secuencia del escenario 1.4.....	135
Figura 57. Diagrama de secuencia del escenario 2.1 .....	136
Figura 58. Diagrama de secuencia del escenario 2.2.....	137
Figura 59. Diagrama de secuencia del escenario 2.3.....	138
Figura 60. Diagrama de secuencia del escenario 2.4.....	139
Figura 61. Diagrama de secuencia del escenario 3.1 .....	140
Figura 62. Diagrama de secuencia del escenario 3.2.....	141
Figura 63. Diagrama de secuencia del escenario 4.1 .....	142
Figura 64. Diagrama de secuencia del escenario 4.2.....	143
Figura 65. Pantalla de descarga de postgresql.....	151
Figura 66. Pantalla de descompresión de archivos usando winzip .....	152
Figura 67. Asistente de instalación – selecciona idioma.....	152
Figura 68. Asistente de instalación – licencia .....	153
Figura 69. Asistente de instalación – precauciones.....	153
Figura 70. Asistente de instalación – partes a instalar.....	154
Figura 71. Asistente de instalación – usuario.....	154
Figura 72. Asistente de instalación – confirmación clave .....	155
Figura 73. Asistente de instalación – confirmación clave .....	155
Figura 74. Asistente de instalación – elije el lenguaje .....	156
Figura 75. Asistente de instalación – módulos opcionales .....	156
Figura 76. Página oficial de apache .....	157
Figura 77. Icono instalador de apache .....	157
Figura 78. Pantalla de inicio del instalador de apache.....	158
Figura 79. Términos de la licencia de apache.....	158
Figura 80. Pantalla con la información acerca de apache .....	159
Figura 81. Pantalla confirmando Java configurado.....	163
Figura 82. Pantalla con erro, Java no configurado. ....	164
Figura 83. Pantalla para configuración de Variables de entorno.....	165
Figura. 84. Administración de CENACAD .....	173

Figura 85. Asistente para el discriminador de formularios por encuesta ....	174
Figura 86. Pantalla que muestra los diferentes proyectos de encuestas del sistema.....	174
Figura 87. Pantalla que muestra las diferentes encuestas de un proyecto del sistema.....	175
Figura 88. Pantalla que muestra datos completos para proceder a entrenar .....	176
Figura 89. Pantalla de creación de mapa.....	177
Figura 90. Pantalla de creación de mapa.....	177
Figura 91. Creación de Mapa .....	178
Figura 92. Pantalla que ilustra el asistente para el proceso de entrenamiento .....	178
Figura 93. Pantalla que ilustra opción para realizar el proceso de clasificación de formularios.....	180
Figura 94. Pantalla que ilustra el proceso de clasificación de formularios .	181
Figura 95. Pantalla que ilustra el menú de administración del Discriminador de Formularios .....	182
Figura 96. Mapa de Kohonen con formularios asignados a cada nodo.....	183
Figura 97. Pantalla que permite generar el número de formularios.....	183
Figura 98. Pantalla que conocer los patrones o formularios válidos y no válidos .....	184
Figura 99. Pantalla que ilustra el total de formularios de la encuesta. ....	185
Figura 100. Pantalla que ilustra si un formulario cumple con los algoritmos de clasificación.....	186
Figura 101. Procesos Operativos .....	191
Figura 102. Pantalla que ilustra el asistente para el proceso de entrenamiento.....	197
Figura 103. Pantalla que muestra los diferentes proyectos de encuestas del sistema.....	198
Figura 104. Pantalla que muestra las diferentes encuestas de un proyecto del sistema.....	198

## **INDICE DE ECUACIONES**

(Ecuación 1) variación de error para el aprendizaje supervisado.....	10
(Ecuación 2) función de transferencia generalizada .....	12
(Ecuación 3) Función logsig .....	15
(Ecuación 4) Regla de Hebb .....	25
(Ecuación 5) Regla de Hebb con porcentajede olvido. ....	25
(Ecuación 6) Función de Red Instar .....	27
(Ecuación 7) La regla de Hebb con porcentaje de olvido más factor de peso .....	28
(Ecuación 8) Regla Instar. ....	29
(Ecuación 9) Regla Instar Vectorial. ....	29
(Ecuación 10) Función de saturación simétrica “satlins” .....	30
(Ecuación 11) Regla Outstar .....	31
(Ecuación 12) Función de Similitud. ....	37
(Ecuación 13) Función de Entrenamiento de Kohonen. ....	38
(Ecuación 14) Distancia de Mahalanobis .....	83
(Ecuación 15) Semipositividad de Mahalanobis.....	84
(Ecuación 16) Simetricidad.....	84
(Ecuación 17) Desigualdad triangular .....	84

## **INDICE DE TABLAS**

Tabla 1. Funciones de Transferencia (6) .....	17
Tabla 2. Redes Neuronales Monocapa. (4) .....	64
Tabla 3. Redes Neuronales Multicapa más conocidas. (4).....	65
Tabla 4. Comparativos entre modelos.....	82
Tabla 5. Resultados de los errores en las Iteraciones en una matriz, cuyo mapa tiene dimensión de 6 x 6.....	117
Tabla 6. Resultados de los errores en las iteraciones en una matriz, cuyo mapa tiene dimensión de 8 x 8.....	118
Tabla 7. Resultados de los errores en las Iteraciones en una matriz, cuyo mapa tiene dimensión de 12 x 12.....	119
Tabla 8. Tabla Encuesta.....	144
Tabla 9. Tabla Formulario.....	145
Tabla 10. Tabla Neuronal Clúster Kohonen.....	146
Tabla 11. Tabla Neuronal Mapa .....	147
Tabla 12. Tabla Neuronal Neuronas .....	148
Tabla 13. Tabla Neuronal Vector.....	149

## **INTRODUCCIÓN**

En la búsqueda de nuevas soluciones tecnológicas para el proceso de la obtención de información en el área administrativa de una empresa ó corporación, se han presentado nuevas innovaciones, las mismas que son utilizadas o seleccionadas en base al criterio de los directivos y del problema que este encuentre. Cada solución tecnológica puede ser diferente pero buscan un fin común que es brindar de manera entendible respuestas a preguntas frecuentemente realizadas. Por otro lado podemos mencionar que para el manejo de grandes cantidades de información una solución tecnológica muy frecuente es el uso de los modelos de minería de datos, aplicados en procesos que controlan los datos para luego ser transformado en información y/o conocimiento.

El proceso de la evaluación docente que realiza la ESPOL, en cada uno de sus períodos de clases, utilizando el sistema de Información denominado CENSO ACADEMICO en línea (CENACAD), consiste en la realización de una encuesta electrónica a los estudiantes que conforman cada uno de los paralelos que se dictan en un período de clases.

Durante este proceso se produce un fenómeno muy común en estadística que consiste en la aparición de los “errores ajenos al muestreo”, estos errores engloba los diferentes factores que influyen este proceso, siendo el



mismo un estudio de gran magnitud. Sin embargo, para nuestro proyecto de tesis nos enfocaremos en uno de los que se considera de mayor incidencia para este proceso, que consiste en la detección de los formularios de encuesta que están siendo mal evaluados por los estudiantes con características específicas.

Estos formularios de encuesta han producido varias teorías como: la equivalencia de un estudiante que ha reprobado la materia vs la mala evaluación en dicho formulario; otra teoría que existe es que un estudiante que repite la materia también evalúa mal el formulario, entre otros, dando la apertura a un proyecto de tesis para la detección y clasificación de estos formularios válidos y no válidos.

En la actualidad existen procesos de clasificación muy parecidos que se le realiza de manera manual, es decir, que existe un experto en el tema que toma la decisión de clasificar o en muchos casos de discriminar formularios que considere válido o no válido, sin embargo esta decisión en muchos casos es subjetiva, al criterio personal de este experto.

Hace algunos años se consideraba difícil poder realizar tareas de selección, reconocimiento, clasificación en fin poder obtener información de grandes

cantidades de datos, los constantes estudios por mejorar la presentación de información para la toma de decisiones mediante el uso de nuevos y sofisticados sistemas de información han generado la creación de modelos de Minería de datos, siendo uno de ellos las Redes Neuronales Artificiales (RNA).

Las RNAs son las bases para fortalecer el objetivo de nuestro proyecto de tesis que consiste en determinar mediante redes neuronales la identificación y clasificación de los formularios evaluados de manera válida y no válida, durante el proceso de encuesta. Sin embargo el proceso de encuesta trae consigo subjetividad para la determinación de la validez de los formularios, para la cual se ha planteado la comparación con otros métodos que utilizando algoritmos estadísticos. Siendo las RNAs las de mayor efectividad para la clasificación de los formularios.

Esta detección y eliminación de los formularios de encuestas no válidos, producirá una información con una mayor confiabilidad en los permitiendo a los directivos una mejor toma de decisiones para luego poder realizar acciones de importancia sobre la ESPOL y sus profesores.

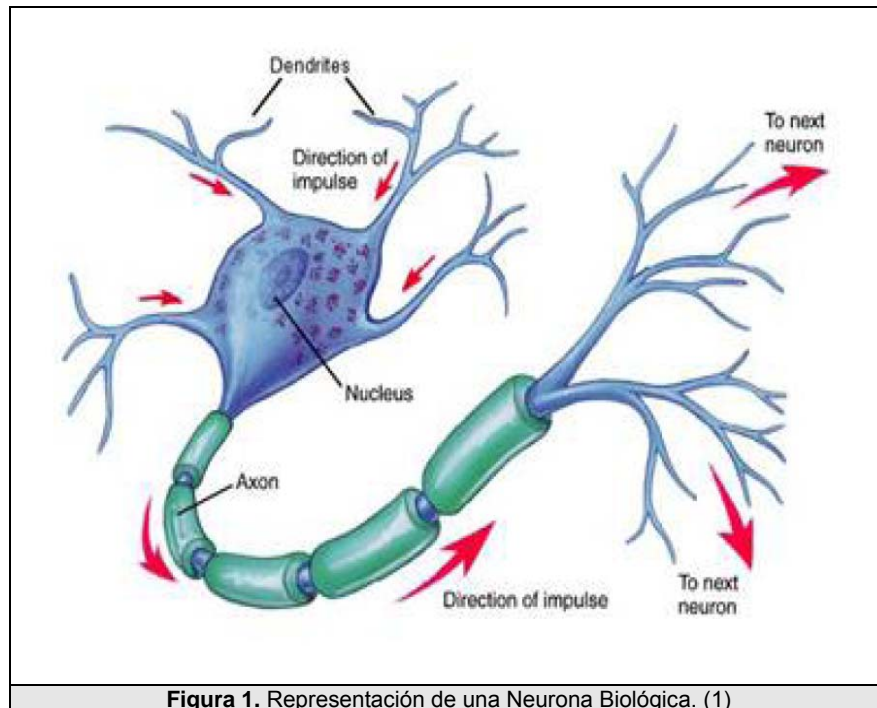
# CAPÍTULO 1

## 1 ESPECIFICACIONES DE LAS REDES NEURONALES

En este primer capítulo se presenta teoría básica para el conocimiento de la Red Neuronal biológica dando a lugar a un mejor entendimiento y analogía sobre la Red Neuronal Artificial (RNA), adicionalmente se presentan diferentes tipos de modelos de Redes Neuronales Artificiales, sus clasificaciones, entre otros. Sin embargo, se profundizará en el modelo de RNA que brinda solución al problema de clasificación que involucra el tema de tesis.

Los modelos de RNA nacen de la idea de imitar uno de los mejores sistemas de procesamiento, que es el cerebro. El mismo que consta de millones de células denominadas neuronas y que se encuentran conectadas entre sí mediante dendritas. La figura 1 se muestra las principales partes que conforman una neurona con son:

- Dendritas.- Canal de entrada de información.
- Axón.- Canal de salida de información.
- Núcleo.- Órgano de cómputo.

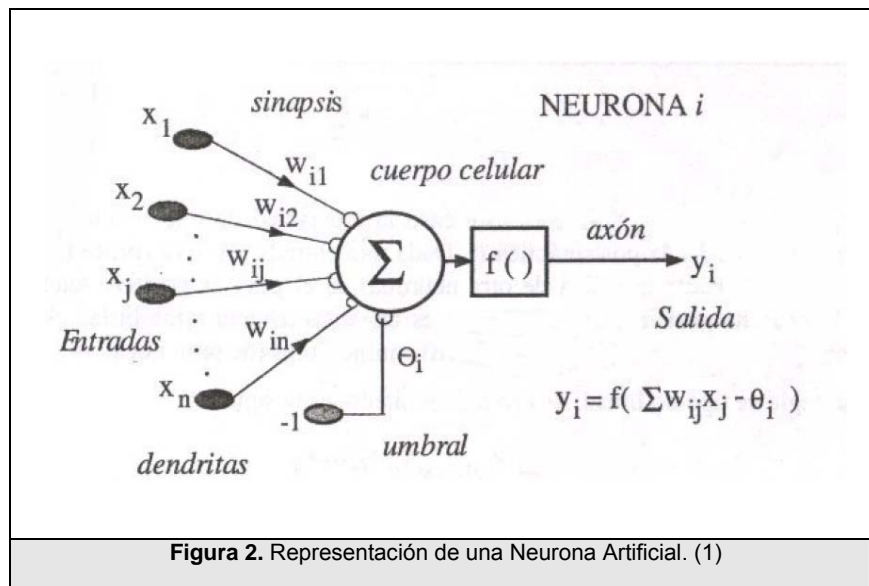


**Figura 1.** Representación de una Neurona Biológica. (1)

Basado en estos antecedentes del humano se construye una nueva forma de computación inspirada en modelos biológicos, denominados RNA, teniendo en cuenta que el procesamiento que desarrolla esta red es mediante operaciones en paralelo y distribuidas; se dice que trabaja en paralelo porque las neuronas procesan información en un mismo instante. Cada neurona realiza procesamientos muy simples y a la vez en conjunto dando lugar a un sistema distribuido. Una ventaja de este sistema es que la información no se encuentra centralizada por lo tanto un daño parcial a las neuronas no afecta en gran magnitud el procesamiento de un problema, es así que se presenta una de las características de mayor importancia para las redes neuronales conocida como “arquitectura de la red” la cual se describirá en la 1.1.

Como parte de las funcionalidades de las RNA podemos mencionar la capacidad de aprendizaje y generalización de la información obtenida por las neuronas.

Se conoce como aprendizaje: “El proceso de adquirir conocimiento, habilidades, actitudes o valores, a través del estudio, la experiencia o la enseñanza” (2) Y generalización: “Hacer común o corriente algo” (3).



En la figura 2 se visualiza los componentes o partes de la RNA en los que se puede destacar:

- Las entradas.- son los datos de otras neuronas y puede estar representada mediante un vector  $x (x_1, x_2, \dots, x_n)$  de entradas.
- Los pesos sinápticos ( $w_{ij}$ ).- establecen el grado de relación entre las diferentes dendritas de entradas y axón de las neuronas vecinas.
- Una regla de propagación ( $f( )$  ).- es un modelo matemático utilizado para el aprendizaje óptimo que obtiene a su vez un valor potencial post-sináptico.

Una estructura de las RNAs desde el punto de vista artificial que entrega una mejor descripción es el siguiente (4) :

- Unidades de procesamiento. Conocida como la neurona artificial.
- Estado de activación de cada neurona. Es un patrón de activación en el conjunto de unidades que lo comprenden a través del tiempo, estos valores pueden ser continuos o discretos, pueden ser limitados o ilimitados.
- Patrón de conectividad entre neuronas. representado por los pesos sinápticos.

- Regla de propagación.
- Función de salida o de transferencia  $f_i(a_i(t))$ . Transforma el estado de activación a una señal de salida. (Ver 1.1.1.2).
- Función o regla de activación  $a_i(t)$ . Regla que combina las entradas con el estado actual de las neuronas para producir un nuevo estado de activación.
- Regla de aprendizaje. Función que controla modificación de los pesos sinápticos.

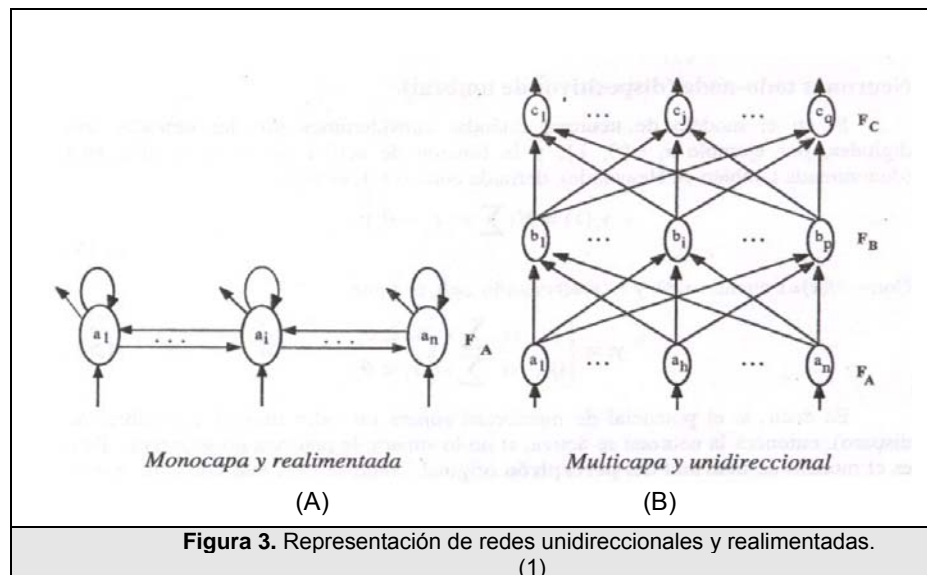
## 1.1 Clasificación de las Redes Neuronales Artificiales

Existen dos criterios de clasificación de mayor relevancia para las RNA, que son:

- Por su arquitectura (topología).

Si los resultados de una neurona no son parte de sus propios valores de entrada se denomina una red unidireccional (feedforward). (Ver figura 3 – B ).

Sin embargo si la neurona tiene como valor de entrada su propio axón esto quiere decir que presenta un ciclo dando a lugar una red denominada recurrente o retroalimentada (recurrent). (Ver figura 3 - A).



- Por su mecanismo de aprendizaje.

Existen dos tipos de RNA acorde a este criterio: supervisadas y no supervisadas.

Una RNA supervisada conoce cuáles son sus entradas y sus salidas correspondientes, es así, que en el momento de entrenamiento de este tipo de redes realiza un ajuste de sus pesos tratando de minimizar el error de salida esperado.

El aprendizaje no supervisado se presenta cuando se proporcionan únicamente los valores de entrada, dando como consecuencia que la red ajuste sus pesos en base a estos y a su propia respuesta.



Ambos tipos de aprendizaje emplean una función de activación o aprendizaje que determina el ajuste de sus pesos utilizando generalmente una función de error.

### 1.1.1 Modelo Supervisado

El modelo realiza el entrenamiento supervisado mediante un ente externo que en muchas ocasiones es representado por un valor de umbral con el cual compara y a su vez determina la respuesta que debería devolver la red a partir de una entrada conocida. Su proceso de entrenamiento consiste en el reajuste de pesos buscando la respuesta óptima o cercana a la salida deseada.

Existen tres tipos de aprendizajes para ser aplicados en los modelos supervisados (4):

- Aprendizaje por corrección de error.- Consiste en ajustar los pesos en función de la diferencia entre los valores deseados y los obtenidos.

$$\Delta w_{ji} = \alpha y_i (d_j - y_j)$$

(Ecuación 1) variación de error para el aprendizaje supervisado.

Donde:

$\Delta w_{ji}$  : Variación en el peso de la conexión entre las neuronas i y j

$$(\Delta w_{ji} = w_{ji}^{\text{actual}} - w_{ji}^{\text{anterior}})$$

$y_i$  : Valor de salida de la neurona i.

$d_j$  : Valor de salida deseado para la neurona j.

$y_j$  : Valor de salida obtenido en la neurona j.

$\alpha$  : Factor de aprendizaje ( $0 < \alpha \leq 1$ ) que regula la velocidad del aprendizaje.

- Aprendizaje por refuerzo.- Consiste en la modificación de pesos en base a la indicación de una señal de refuerzo sin conocer la salida exacta que se desea a partir de una entrada específica (éxito = 1, fracaso = -1), es decir se asemeja más a la supervisión de un crítico (que opina sobre la respuesta) que a la de un maestro (que indica la respuesta concreta que debe generar).
- Aprendizaje estocástico.- Consiste en realizar cambios aleatorios en los pesos de las conexiones y evaluar su efecto a partir de la respuesta deseada y distribuciones de probabilidad.

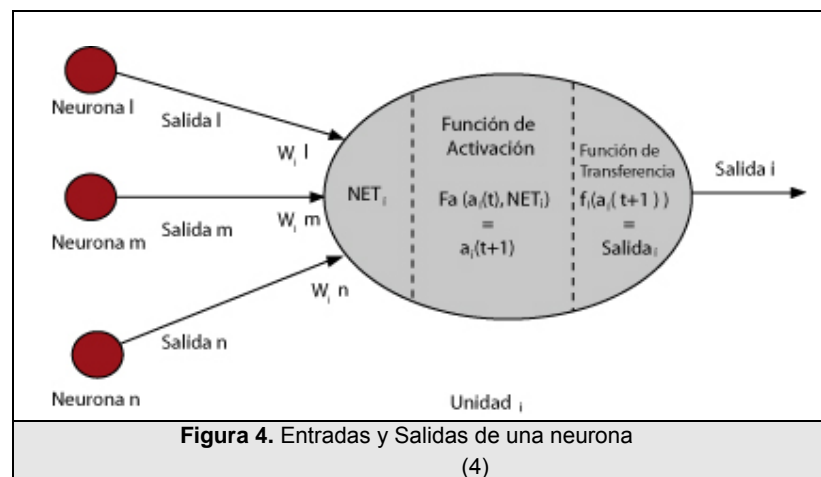
#### 1.1.1.1 Características del Modelo Supervisado

Entre las principales características que este modelo posee se encuentran:

- La necesidad de conocer los valores de salida dada una entrada determinada; es decir, tener pares ordenados  $(x, g(x))$ , en el que  $g(x)$  es una función dependiente del valor  $x$  (5).

- Cuenta con un aprendizaje activo que consiste en dar etiquetas a un grupo de valores o datos, dando a lugar el riesgo a focalizarse en valores no tan importantes entregando una respuesta no esperada.
- Abstracción o tolerancia al ruido: Las redes neuronales artificiales son capaces de extraer o abstraer las características esenciales de las entradas aprendidas, de esta manera pueden procesar correctamente datos incompletos o distorsionados.

### 1.1.1.2 Funciones de Transferencia y Tipos de Redes conocidas.



En la figura 4 podemos encontrar con detalle la función de transferencia cuya funcionalidad es la de transformar el estado actual de activación  $(a_i(t))$  en una señal de salida  $(y_i(t))$ .

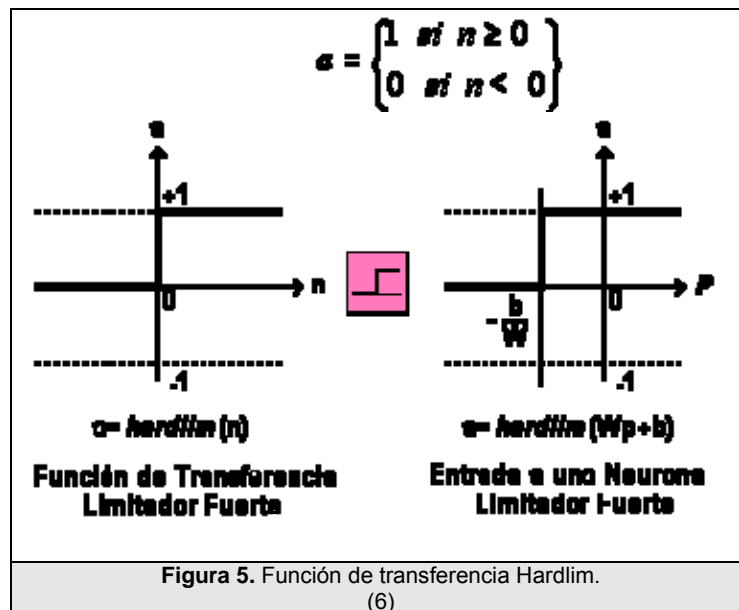
$$y_i(t) = f_i a_i(t)$$

(Ecuación 2) función de transferencia generalizada

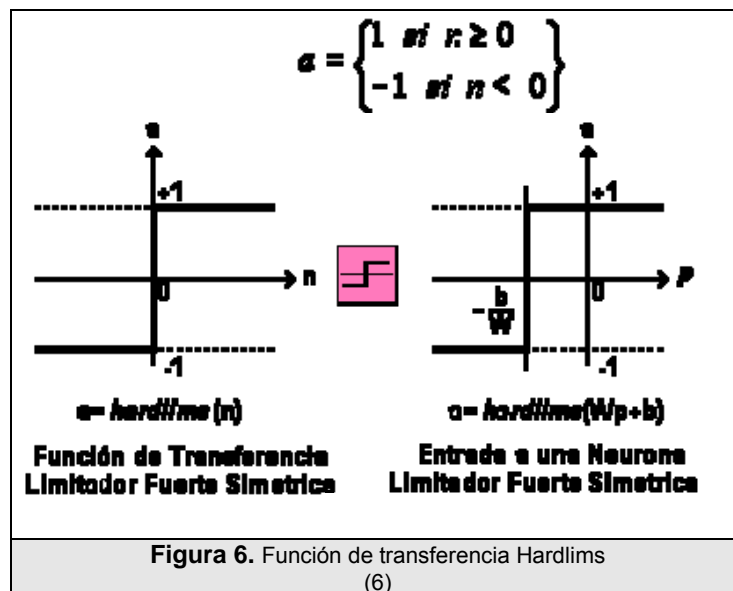
Entre las principales funciones de transferencias podemos mencionar: escalón, lineal, sigmoideal y gaussiana.

Función Escalón - *Limitador fuerte (Hardlim)*: como se puede destacar en la figura 5 esta función de transferencia realiza la activación igual a 1 cuando la suma es mayor o igual que el umbral de la neurona, caso contrario la activación será igual a 0 (4).

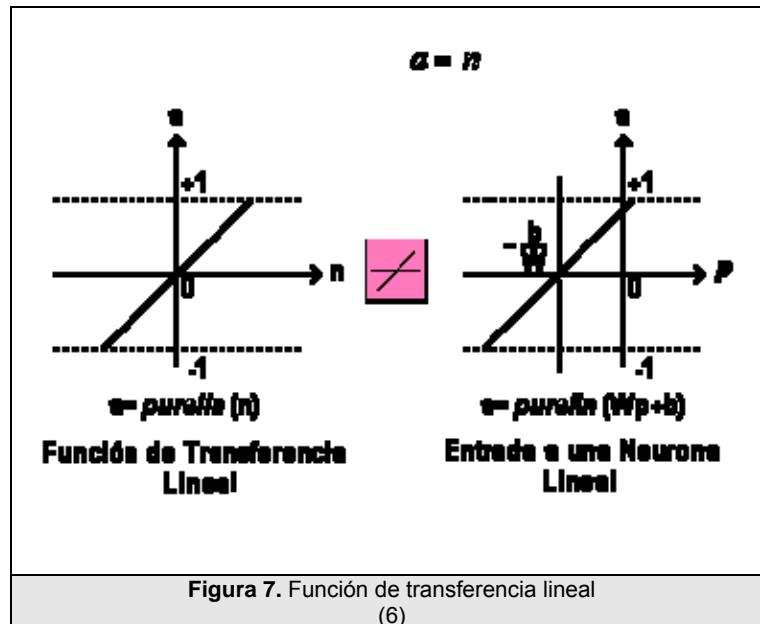
Como se muestra en la figura 5 podemos ver la gráfica de la función de escalón del lado izquierdo, y del lado derecho se muestra la misma función utilizada como función de transferencia cuyos valores de entrada son el peso de la neurona ( $Wp$ ) y un valor de desplazamiento ( $b$ ).



Adicionalmente se encuentra representada en la figura 6 una variación de la función escalón en la cual se amplían los valores de rango entre 1 y -1, cuyos valores se presentan de manera simétrica.



*Función lineal (purelin):* La salida de una función de transferencia lineal es igual a su entrada respondiendo a la expresión  $f(x) = x$ .

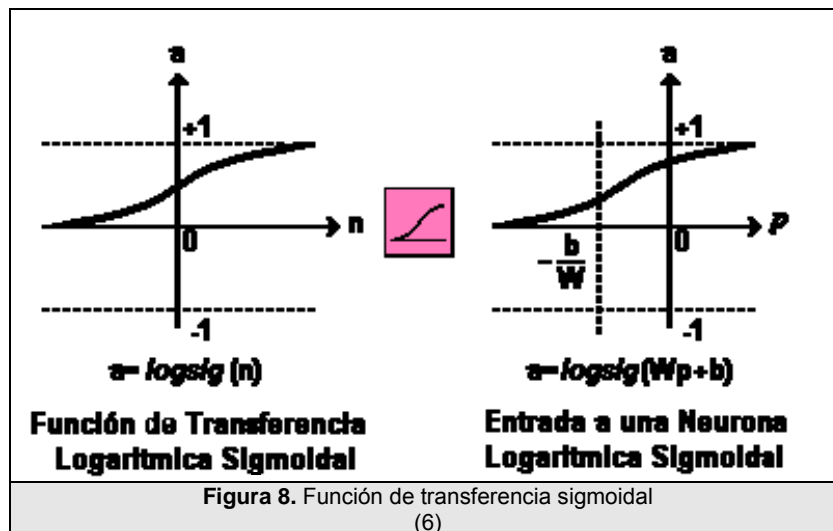


La gráfica derecha de la figura 7 muestra la variante de la función lineal que es aplicada en las redes tipo lineal, por ejemplo redes Adaline (7).

*Función continua sigmoïdal (logsig):* Esta función puede tomar valores de dominio entre más y menos infinito, devolviendo valores de rango entre cero y uno, de acuerdo a la expresión.

$$a = \frac{1}{1 + e^{-n}}$$

(Ecuación 3) Función logsig



La tabla 1 hace una relación de las principales funciones de transferencia empleadas en el entrenamiento de redes neuronales.

Nombre	Relación Entrada /Salida	Descripción Gráfica	Función
Limitador Fuerte (Escalón)	$a = 0 \quad n < 0$ $a = 1 \quad n > 0$		<i>hardlim</i>
Limitador Fuerte (escalón) Simétrico	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		<i>hardlims</i>
Lineal Positiva	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		<i>poslin</i>
Lineal	$a = n$		<i>purelin</i>
Lineal Saturado	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		<i>satlin</i>





Lineal Saturado Simétrico	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = +1 \quad n > 1$		<i>satlins</i>
Sigmoidal Logarítmico	$a = \frac{1}{1 - e^{-n}}$		<i>logsig</i>
Tangente Sigmoidal Hiperbólica	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		<i>tansig</i>
Competitiva	$z = 1 \text{ Neurona con mayor}$ $z = 0 \text{ El resto de neuronas}$		<i>compet</i>

Tabla 1. Funciones de Transferencia (6)

Adicionalmente debemos tomar en cuenta toda red neuronal utilizará una de estas funciones o una variación de las mismas, la cual depende del tipo de función de transferencia que se usa para una red neuronal específica generalmente depende del tipo de red neuronal que se va a utilizar

### 1.1.2 Modelo No Supervisado.

Las redes con aprendizaje no supervisado se basan en los valores de entrada para generar sus propios valores de respuestas. La red no recibe ninguna señal adicional a estos valores y se basa simplemente en determinar si un valor de salida es o no correcto en función al valor de entrada; Por este motivo, se dice que estas redes son capaces de auto organizarse.

Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presenten en



su entrada; puesto que no hay supervisor que indique a la red la respuesta que debe generar ante una entrada concreta (4).

Las neuronas de salida de estas redes presentan respuestas que representan valores de similitud que existe entre los valores de entrada, esto genera un prototipado de la información por medio de la asociación de dichos valores de respuesta; codificando de esta manera los datos en grupos denominados conglomerados.

Algunas redes con aprendizaje no supervisado generan un mapeo de características (featuring mapping), obteniéndose en las neuronas de salida una disposición geométrica que representa un mapa fotográfico de las características de los datos de entrada, de tal forma que si presentan a la red informaciones similares siempre sean afectadas neuronas de salida próximas entre sí, en la misma zona del mapa.

#### **1.1.2.1 Características del Modelo No Supervisado.**

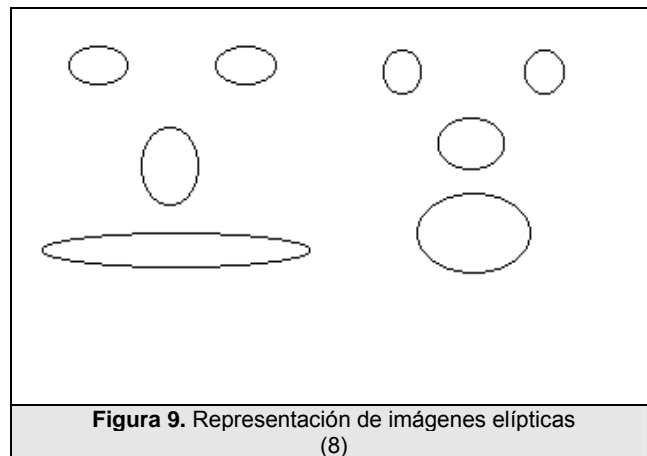
Entre las características de este modelo podemos mencionar:

- La metodología que emplea para aprender, conocido como aprendizaje inductivo que significa que aprende en base a los cambios que encuentra en las variantes del medio, es decir que extrae sus

propias reglas a partir de los ejemplos de aprendizaje, produciendo vectores de salidas consistentes.

- Agrupa los comportamientos en conglomerados o clases de patrones.
- Generalización: Esta característica permite a la RNA encontrar respuestas similares a nuevas entradas, que no necesariamente hayan sido ingresadas en el proceso de entrenamiento de la misma.
- Abstracción o tolerancia al ruido: permiten procesar correctamente datos incompletos o distorsionados mediante la abstracción de las características esenciales de las entradas aprendidas.
- Procesamiento paralelo: permite a las neuronas trabajar de manera simultánea dependiendo de su arquitectura y a su vez de la comunicación que exista entre las entradas y salidas de las neuronas.
- Memoria distribuida: el conocimiento acumulado por la red se halla distribuido en numerosas conexiones, esto tiene como consecuencia la tolerancia a fallos: una red neuronal es capaz de seguir funcionando adecuadamente a pesar de sufrir lesiones con destrucción de neuronas o sus conexiones, ya que la información se halla distribuida

por toda la red; Sin embargo, en un programa tradicional un pequeño fallo en cualquier punto puede invalidarlo todo y dar un resultado absurdo o no dar ningún resultado. (8).



Un ejemplo de un modelo no supervisado es la figura 9 donde se muestra varias imágenes formadas por elipses, las cuales podríamos representarlas mediante mapas de bits, pero son posibles otras representaciones más eficientes; una elipse se distingue de otra en cinco atributos: orientación, posición vertical, posición horizontal, longitud y anchura, por consiguiente, la imagen podría describirse usando sólo cinco parámetros por elipse.

#### 1.1.2.2 Tipos de Redes conocidas.

Entre los tipos de Redes para el modelo no supervisado se encuentran dos tipos muy relevantes en base a su aprendizaje:

- Aprendizaje asociativo (Ver Capítulo 1.1.3).

- Aprendizaje competitivo (Ver Capítulo 1.1.4).

Cabe destacar, que en nuestra investigación, el aprendizaje competitivo se encuentra fuertemente representado por el algoritmo de conglomerados de Teuvo Kohonen.

En el aprendizaje competitivo no existe un agente externo que vaya modificando los pesos sinápticos en función de la corrección de la salida de la red; la red no compara su respuesta efectiva con la salida correcta puesto que no especifica cuál ha de ser la salida correcta. En el aprendizaje no supervisado es necesario presentar un mayor número de patrones de entrada y utilizar reglas de aprendizaje distintas a las que nos sirven para el caso anterior; además las arquitecturas de las redes suelen ser distintas, por ejemplo muchas de ellas tienen dos capas; destacando las redes de Kohonen y de Grossberg. (9)

### **1.1.3 Aprendizaje Asociativo.**

Una asociación es cualquier vínculo entre la entrada de un sistema y su salida correspondiente. Cuando dos patrones son vinculados por una asociación, el patrón de entrada es a menudo referido como el estímulo, y la salida es referida como la respuesta.

### **1.1.3.1 Antecedentes.**

A mediados de 1890 se realizaron estudios sobre el tipo de aprendizaje asociativo que junto a las escuelas de Psicología entregaron como resultados uno de los primeros experimentos en este tema, conocido como “ley del reflejo condicionado” (10) o clásico de Petrovich Pavlov.

Otro de los experimentos de gran importancia como aporte a este aprendizaje fue brindado por B.F. Skinner (1975), cuyo experimento desarrollado presenta un entrenamiento que se lo denominó el “Condicionamiento Instrumental” o reflejo condicionado (11).

Basado en estos estudios de comportamiento, Donald Hebb postuló la “regla de Hebb” aplicada en las redes neuronales, dando apertura al aprendizaje no supervisado que es una rama de gran importancia para realizar, clasificación, mapeos, filtrado de información en la reconstrucción de datos, eliminación de ruido, explotación de relaciones entre informaciones similares, facilitación en la búsqueda por contenido en bases de datos y resolución de problemas de optimización.

### **1.1.3.2 Estructura de la Red Neuronal Asociativa.**

Las redes asociativas están conformados por los mismos componentes acorde se describió en las especificaciones de las RNA (Ver Sección 1).

Sin embargo, como parte de este tipo de red asociativa se incorpora a la estructura de la red, los tipos de estímulos se describen a continuación:

- Estímulo condicionado.- El objeto de la asociación, debe ser siempre presentado a la red y ésta debe asociarlo con la salida deseada; al final del proceso de aprendizaje la red debe ser capaz de entregar la respuesta correcta con la presentación de este único estímulo a su entrada, sin importar si el estímulo no condicionado ha sido presentado o no, pues la asociación ya ha sido realizada. “ley del reflejo condicionado de Pavlov”. (10).
- Estimulo No condicionado.- Corresponde a la entrada, que pudiendo ser de carácter escalar o vectorial, refuerza el aprendizaje y ayuda a hacer la asociación con la salida deseada, este estímulo se presenta intermitentemente para simular un real proceso de aprendizaje y memorización de la red; la mayoría de las veces el estímulo no condicionado se convierte en la salida deseada de la red. “Condicionamiento Instrumental”. (11) .

### **1.1.3.3 Regla de Hebb.**

El postulado que se contempla en la regla de Hebb dice:

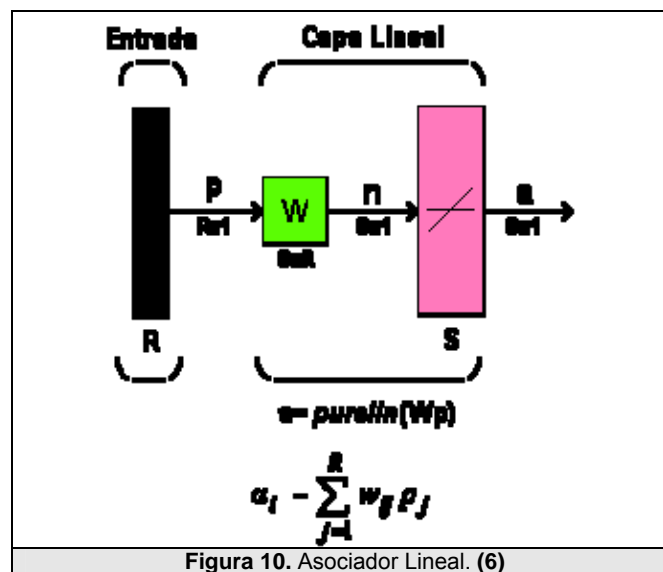
*“Cuando un axón de una celda A está lo suficientemente cerca de otra celda B como para excitarla y repetidamente ocasiona su activación, un cambio*

*metabólico se presenta en una o ambas celdas, tal que la eficiencia de A, como celda excitadora de B, se incrementa".*

Con el término celda, Hebb se refería a un conjunto de neuronas fuertemente conexas a través de una estructura compleja, la eficiencia podría identificarse con la intensidad o magnitud de la conexión, es decir el peso. (9).

Se puede decir entonces que la regla de Hebb presenta un aprendizaje que consiste en el ajuste de los pesos de las conexiones de acuerdo con la correlación de los valores de activación de las neuronas conectadas.

En la figura 10 se muestra un asociador lineal, empleando esta regla.



Donde  $a$ : es la función de activación que utiliza una función lineal determinada por el vector de entrada  $p$ .

De esta forma el postulado de Hebb implica que si un valor positivo “ $p_j$ ” produce un valor positivo “ $a_j$ ” el valor de “ $W_{ij}$ ” debe incrementarse.

$$w_{ij}^{nuevo} = w_{ij}^{anterior} + \alpha(a_i)(p_j)$$

(Ecuación 4) Regla de Hebb

Donde:

“ $p_{jq}$ ” j-ésimo elemento del “q-ésimo” vector de entrada “ $p_q$ ”

“ $a_{iq}$ ” i-ésimo elemento de salida de la red, cuando el q-ésimo vector de entrada es presentado.

$\alpha$ : es el porcentaje de aprendizaje, el cual es un valor positivo constante.

El porcentaje de aprendizaje “ $\alpha$ ” determina cuántas veces un estímulo y su respuesta deben ocurrir simultáneamente, antes de que la asociación sea hecha.

Una forma de mejorar la regla de Hebb, es adicionar un término que controle el crecimiento de la matriz de peso, a esta modificación se le da el nombre de regla de Hebb con porcentaje de olvido.

$$W(q) = W(q-1) + \alpha a(q)p^T(q) - \gamma W(q-1)$$

(Ecuación 5) Regla de Hebb con porcentaje de olvido.

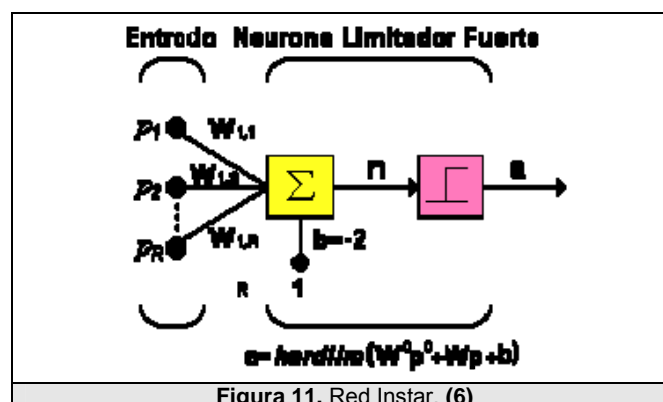
Donde  $\gamma$  es la porcentaje de olvido, la cual es una constante positiva menor que 1; cuando  $\gamma$  se aproxima a cero la ley de aprendizaje se convierte en la



ley de Hebb estándar; cuando  $\gamma$  se aproxima a 1, el porcentaje de aprendizaje olvida rápidamente las entradas anteriores y recuerda solamente los patrones más recientes. El efecto de esta nueva constante, es controlar que el crecimiento de la matriz de pesos no se realice sin límites y así darle un mejor aprovechamiento a la capacidad de memoria de la red.

#### 1.1.3.4 Red Neuronal Instar.

Hasta ahora se han considerado solamente reglas de asociación entre entradas y salidas escalares. Si se examina la red de la figura 11, se nota como esta neurona está enfrentada a un problema de reconocimiento de patrones cuya entrada es de tipo vectorial; esta neurona es el tipo de red más simple capaz de resolver esta clase de problemas y es llamada red Instar. (6)



El desempeño ha sido analizado en diferentes ambientes. Para la Instar no se considerará directamente su característica de decisión, en lugar de ello se analizará la capacidad de la Instar para reconocimiento de patrones a través de asociaciones y aprendizaje no supervisado.

La ecuación para determinar la entrada/salida de la Instar es:

$$a = \text{hardlims}(w^T p + b) \quad (\text{Ecuación 6) Función de Red Instar}$$

La red Instar se activará si el producto punto entre el vector de pesos (fila de la matriz de pesos) y la entrada sea mayor o igual a “-b”

$$w^T p \geq -b$$

Los vectores “w” y “p” son de longitud constante, por lo tanto el mayor producto punto se presentará cuando los dos vectores apunten en la misma dirección; dicho de otra forma cuando el ángulo entre “w” y “p” sea  $\theta = 0$ , esto permite observar que la red instar de la figura 14 se activará cuando “p” y “w” estén muy cercanos, escogiendo un apropiado valor para la ganancia “b” se puede determinar que tan cerca deben estar “p” y “w” para que la instar se active, si se fija

$$b = -|w| \cdot |p|$$

La instar se activará solamente cuando “p” apunte exactamente en la misma dirección de “w”, de esta forma “b” se puede incrementar a valores

ligeramente mayores a  $-|w_i| \cdot |p_i|$ , el mayor valor de “b” se presentará cuando la Instar esté activa. Es importante recalcar que este análisis asume que todos los vectores tienen la misma longitud.

Uno de los inconvenientes de la regla de Hebb con porcentaje de olvido, es que requiere que los estímulos se presenten de forma repetitiva o de lo contrario se perderá la asociación, se desea encontrar una regla alternativa que habilite el término con olvido solo cuando la Instar es activa  $a_i \neq 0$ , de esta forma los valores de los pesos seguirán siendo limitados, pero el porcentaje de olvido será minimizado. Para obtener los beneficios del término de peso con porcentaje de olvido, se adiciona un nuevo término proporcional a  $a_i(q)$ .

$$w_j(q) = w_j(q-1) - \alpha a_i(q) p_j(q) - \gamma a_i(q) w_j^{estable}$$

(Ecuación 7) La regla de Hebb con porcentaje de olvido más factor de peso

El nuevo término de peso se hace proporcional a la salida escalar  $a_i(q)$ , ya que se desea controlar esta salida para que produzca el estímulo no condicionado; si se considera que el porcentaje al cual la red aprende nuevos pesos es igual al porcentaje de olvido  $\alpha = \gamma$ , la ecuación 7 puede simplificarse a:

$$w_j(q) = w_j(q-1) + \alpha a_i(q) (p_j(q) - w_j^{actual})$$

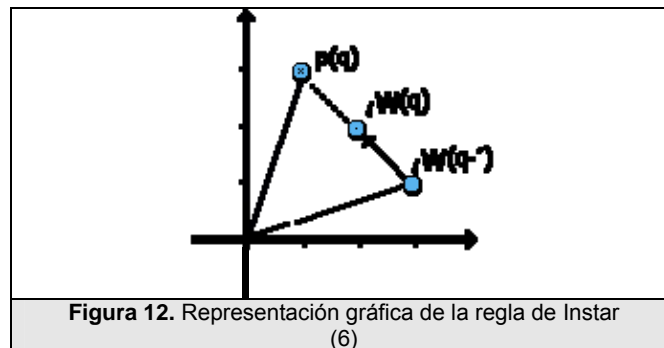
(Ecuación 8) Regla Instar.

Esta ecuación es la llamada regla de Instar, que en forma vectorial teniendo en cuenta el caso en que la instar está activa ( $a_i=1$ ), se convierte en:

$$w(q) = w(q-1) + \alpha (p(q) - w(q-1)) = (1-\alpha) w(q-1) + \alpha p(q)$$

(Ecuación 9) Regla Instar Vectorial.

Esta operación se muestra en la figura 12.



### 1.1.3.5 Red Neuronal Outstar.

La red Outstar, mostrada en la figura 13 tiene una entrada tipo escalar y una salida tipo vectorial y puede recordar patrones por asociación de un estímulo con un vector de respuesta. (6)

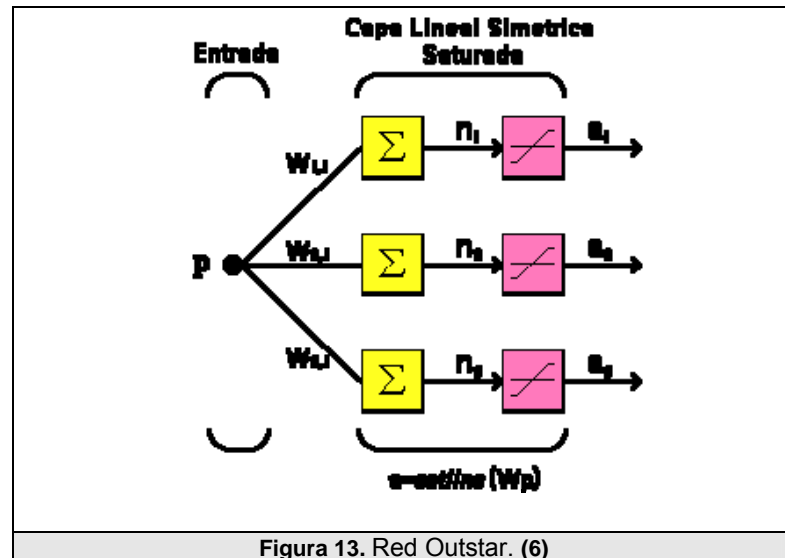


Figura 13. Red Outstar. (6)

La expresión de salida de esta red es:

$$\mathbf{a} = \text{satlins}(\mathbf{Wp})$$

(Ecuación 10) Función de saturación simétrica

“satlins”

Se desea recordar un vector con valores entre  $-1$  y  $1$ , para lo cual se utilizará la función de saturación simétrica “satlins”, aunque pueden usarse otras funciones como por ejemplo “hardlins”.

Para derivar la regla Instar, el problema del olvido presentado por la regla de aprendizaje de Hebb fue limitado por el nuevo término de peso, el cual era proporcional a la salida de la red “ $a_i$ ”. De manera similar, para obtener la regla de aprendizaje Outstar el término con olvido se hará proporcional a la

entrada de la red “ $p_j$ ” ya que la salida de esta red es un vector, con el cual se espera simular el estímulo no condicionado.

$$w_j(q) = w_j(q-1) + \alpha a_i(q) p_j(q) - \lambda p_i(q) w_j(q-1)$$

(Ecuación 11) Regla Outstar

Si se hace el porcentaje de olvido  $\lambda$  igual al porcentaje de aprendizaje  $\alpha$  se obtiene

$$w_{ij}(q) = w_{ij}(q-1) + \alpha (a_i(q) - w_{ij}(q-1)) p_j(q)$$

La regla Outstar tiene propiedades complementarias a la regla Instar; el aprendizaje ocurre cuando una entrada “ $p_j$ ” tiene un valor diferente a cero (en lugar de  $a_i$ ). Cuando el aprendizaje ocurre, la columna “ $w_j$ ”, se acerca al vector de salida.

Las redes Instar y Outstar son empleadas conjuntamente en la red ART (Adaptive Resonance Theory), (12) y cada una en forma independiente son utilizadas en gran cantidad de aplicaciones debido a su fácil implementación y al funcionamiento casi intuitivo de su regla de aprendizaje.

#### **1.1.4 Redes Competitivas.**

En las redes con aprendizaje competitivo (y cooperativo), suele decirse que las neuronas compiten (y cooperan) unas con otras con el fin de llevar a cabo una tarea dada (4).

Las neuronas compiten para activarse quedando finalmente una, o una por grupo, denominada neurona vencedora (winner take all) y el resto quedan anuladas, siendo forzadas a valores de respuesta mínimos.

La competición entre neuronas se realiza en todas las capas de la red, existiendo en estas neuronas conexiones de autoexcitación (signo positivo) y conexiones de inhibición (signo negativo) por parte de neuronas vecinas.

El objetivo de este aprendizaje es agrupar en conglomerados los datos que se introducen en la red, donde los conglomerados o categorías deben ser creados por sí mismo a través de las correlaciones entre los datos de entrada.

El aprendizaje afecta a las neuronas ganadoras de manera directa y en menor grado a las neuronas vecinas más cercanas en un porcentaje determinado por una función de vecindad.

#### **1.1.4.1 Antecedentes.**

El desarrollo de estas redes empieza a principios de 1959 cuando Frank Rosenblatt (13) creó una red de aprendizaje no supervisado basado en el Perceptrón, el cual aprendía a clasificar vectores de entrada en dos clases con igual número de términos.

En 1968 Stephen Grossberg mediante el uso del aprendizaje hebbiano crea un modelo denominado Additive Grossberg, y en 1973 introduce la red shunting Grossberg. Redes utilizadas para la supresión del ruido, aumento del contraste y normalización de vectores.

En el mismo año Christoph Von Der Malsburg introduce la regla del mapa de organización propia, que permitía a la red clasificar entradas en las cuales las neuronas que estuviesen en un vecindario cercano a la neurona ganadora, respondieran a entradas similares (6). Su regla de aprendizaje generó gran interés, pero ésta utilizaba un cálculo no local para garantizar que los pesos fueran normalizados, este hecho hacía este modelo biológicamente poco posible.

Basado en la regla de Von Der Malsburg, Grossberg redescubrió la regla Instar, realizando la corrección en la normalización de los pesos.

En la misma década Teuvo Kohonen desarrolló una versión simplificada de la regla Instar, inspirada en la red de Von Der Malsburg y Grossberg.



En 1985 Rumelhart y Zisper utilizaban redes multicapa dividiendo cada capa en grupos de neuronas.

Fukushima empleó esta idea en 1975 para una red multicapa llamada Cognitron, y en 1980 el mismo Fukushima desarrollo una versión mejorada a la que llamó Necognitron.

En 1982 Teuvo Kohonen crea el mapa de características (Self Organization Map - SOM), aplicado en redes con una disposición bidimensional de las neuronas de salida, que permiten obtener mapas topológicos o topográficos en los que de algún modo estarían representadas las características principales de las informaciones presentadas a la red.

#### **1.1.4.2 Red Neuronal de Kohonen.**

Existen evidencias que demuestran que en el cerebro hay neuronas que se organizan en muchas zonas, de forma que las informaciones captadas del entorno a través de los órganos sensoriales se representan internamente en forma de mapas bidimensionales. Por ejemplo, en el sistema visual se han detectado mapas del espacio visual en zonas del córtex (capa externa del cerebro), también en el sistema auditivo se detecta una organización según la frecuencia a la que cada neurona alcanza mayor repuesta (organización tonotópica).

Aunque en gran medida esta organización neuronal está predeterminada genéticamente, es probable que parte de ella se origine mediante el aprendizaje, esto sugiere que el cerebro podría poseer la capacidad inherente de formar mapas topológicos de las informaciones recibidas del exterior. De hecho, esta teoría podría explicar su poder de operar con elementos semánticos: algunas áreas del cerebro simplemente podrían crear y ordenar neuronas especializadas o grupos con características de alto nivel y sus combinaciones; En definitiva, se construirían mapas especiales para atributos y características.

A partir de estas ideas T. Kohonen presentó en 1982 un sistema con un comportamiento semejante, se trataba de un modelo de red neuronal con capacidad para formar mapas de características de manera similar a como ocurre en el cerebro. El objetivo de Kohonen era demostrar que un estímulo externo (información de entrada) por sí solo, suponiendo una estructura propia y una descripción funcional del comportamiento de la red, era suficiente para forzar la formación de los mapas.

Este modelo tiene dos variantes denominadas LVQ (Learning Vector Quantization) y TPM (Topology Preserving Map) o SOM (Self Organizing Map), ambas se basan en el principio de formación de mapas topológicos para establecer características comunes entre las informaciones (vectores) de entrada a la red, aunque difieren en las dimensiones de éstos, siendo de

una sola dimensión en el caso de LVQ y bidimensional o tridimensional en la red SOM (Ver Sección 1.1.4.3).

El aprendizaje en el modelo de Kohonen es de tipo fuera de línea (off-line) (Ver Sección 2.2), y se distinguen dos etapas que son:

- Etapa de aprendizaje.- en esta se fijan los valores de las conexiones (feedforward) entre la capa de entrada y la de salida, en donde las neuronas compiten por activarse ante un nuevo vector de entrada y solo una de ellas permanece activa, los pesos de las conexiones se ajustan en función de la neurona que haya resultado vencedora y sus neuronas vecinas serán afectadas en base a una función de vecindad.
- Etapa de funcionamiento.- Para efectos de clasificación se procede a ingresar todas las entradas de manera secuencial con la finalidad de agrupar a estos nuevos vectores de entradas acorde a sus características y basados en el mapa previamente entrenado.

El algoritmo de aprendizaje utilizado para establecer los valores de los pesos de las conexiones entre las “N” neuronas de entrada y las “M” de salida es el siguiente:

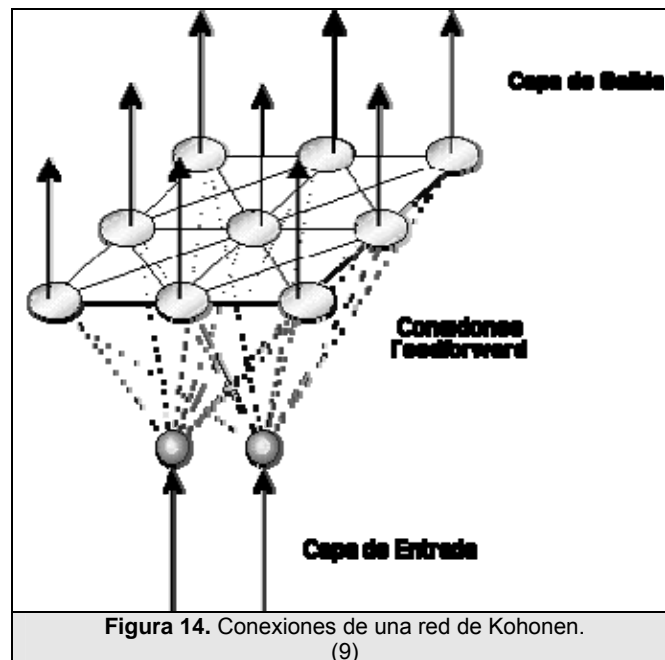
1. En primer lugar se inicializan los pesos ( $w_{ij}$ ) con valores aleatorios pequeños (entre 0 y 1) de las “M” neuronas de salida y se fija la zona inicial de vecindad que depende del coeficiente de aprendizaje.
2. A continuación se presenta a la red los vectores de entrada en forma de  $p = (p_1, p_2, \dots, p_n)$ , cuyas componentes  $p_i$  serán valores continuos utilizados para el entrenamiento.
3. Se determina la neurona vencedora entre el vector ingresado “ $p_1$ ” y las neuronas del mapa previamente inicializado, siendo la neurona ganadora la que obtenga la menor distancia euclidiana (o también mahalanobis) que utiliza la siguiente expresión que es similar a aquella, pero eliminando la raíz cuadrada:

$$d_i = \sum_{j=1}^N (p_j - w_{ij})^2 \quad 1 \leq i \leq M$$

(Ecuación 12) Función de Similitud.

“ $p_j$ ”: Componente “j-ésimo” del vector de entrada

” $w_{ij}$ ”: Peso de la conexión entre la neurona j de la capa de entrada y la neurona i de la capa de salida.



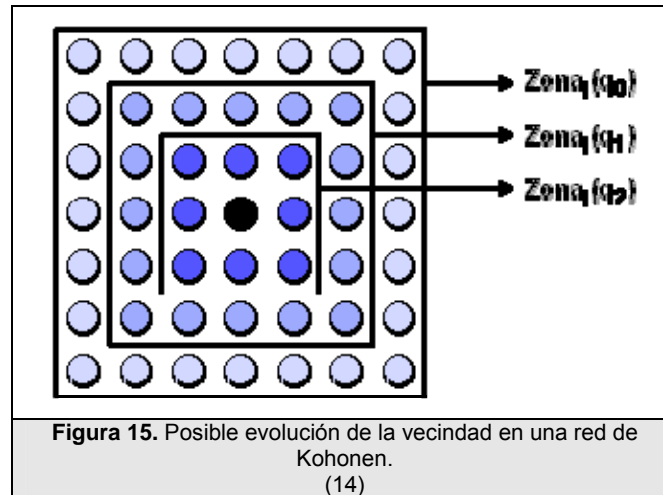
4. Una vez localizada la neurona vencedora ( $i^*$ ), se actualizan los pesos de las conexiones entre las neuronas de entrada y dicha neurona, así como los de las conexiones entre las de entrada y las neuronas vecinas de la vencedora, con esto se logra asociar la información de entrada con una cierta zona de la capa de salida. Esto se realiza mediante la siguiente ecuación.

$$w(q) = w(q-1) + \alpha(q)(p(q) - w(q-1)) \text{ para } i \in X(q)$$

(Ecuación 13) Función de Entrenamiento de Kohonen.

El tamaño de “ $X(q)$ ” se puede reducir en cada iteración del proceso de ajuste de los pesos, con lo que el conjunto de neuronas que pueden considerarse vecinas cada vez es menor como se observa en la figura

15, sin embargo en la práctica es habitual considerar una zona fija en todo el proceso de entrenamiento de la red.



El término  $\alpha(q)$  es el coeficiente de aprendizaje o parámetro de ganancia, con un valor entre 0 y 1 el cual decrece con el número de iteraciones ( $q$ ) del proceso de entrenamiento, de tal forma que cuando se ha presentado un gran número de veces todo el juego de patrones de aprendizaje su valor es prácticamente nulo, con lo que la modificación de los pesos es insignificante.

Para hallar  $\alpha$  suele utilizarse una de las siguientes expresiones:

$$\alpha(q) = \frac{1}{q} \quad \alpha(q) = \alpha_1 \left( 1 - \frac{1}{\alpha_2} \right)$$

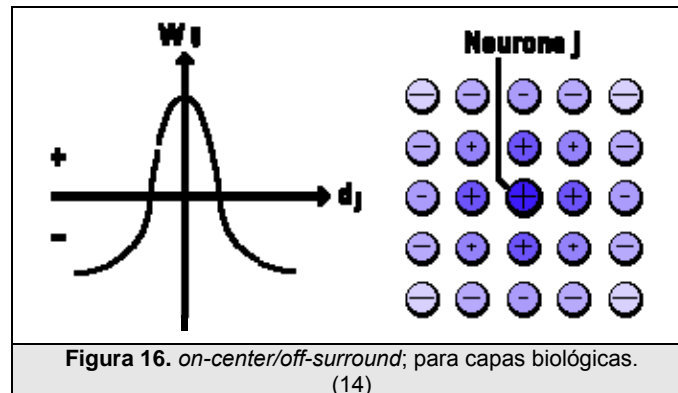
Siendo  $\alpha_1$  un valor de 0.1 ó 0.2 y  $\alpha_2$  un valor próximo al número total de iteraciones del aprendizaje, que por lo general se toma como 10000 para esta red.

5. El proceso debe repetirse, volviendo a presentar todo el juego de patrones de aprendizaje  $p_1, p_2, \dots, p_n$  hasta obtener la salida deseada.

En definitiva lo que hace una red de Kohonen es realizar una tarea de clasificación, puesto que la neurona de salida activada ante una entrada representa la clase a la que pertenece dicha información de entrada, además ante otra entrada parecida se activa la misma neurona de salida, u otra cercana a la anterior debido a la semejanza entre las clases, así se garantiza que las neuronas topológicamente próximas sean sensibles a entradas físicamente similares; por esta causa la red es especialmente útil para establecer relaciones desconocidas previamente entre conjuntos de datos.

#### **1.1.4.3 Mapa de Auto Organización (SOM).**

En las redes competitivas biológicas, una neurona se refuerza a ella misma (center), y refuerza o inhibe las neuronas que están a su alrededor (surround); esta transición entre reforzar las neuronas "vecinas" o "inhibirlas", se realiza suavemente a medida que la distancia entre las neuronas aumenta. Lo cual se ilustra en la figura 16. (14)



Según se describe en la sección 1.1.4.2 sobre redes competitivas, Los mapas de auto-organización (Self Organization Map - SOM) heredan características como determinar la neurona ganadora y encontrar las neuronas en una región cercana (vecindario).

Estas neuronas tanto las ganadoras como las vecinas serán actualizadas mediante la regla de Kohonen en la fase de entrenamiento: (Ver algoritmo en la sección 1.1.4.2)

$${}_i w(q) = {}_i w(q-1) + \alpha (p(q) - {}_i w(q-1)) \text{ para } i \in N_{i^*}(d)$$

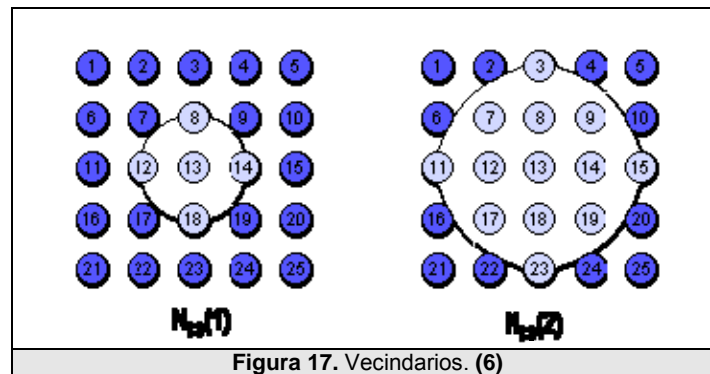
Donde el vecindario  $N_{i^*}$  contiene el índice para todas las neuronas que se encuentren a un radio "d" de la neurona ganadora  $i^*$

$$N_{i^*}(d) = \{j, d_j \leq d\}$$

Cuando un vector "p" es presentado, los pesos de la neurona ganadora y de sus vecinas tenderán hacia "p", el resultado es que después de muchas presentaciones las neuronas vecinas habrán aprendido vectores similares que cada una de las otras.



El concepto de vecindario es ilustrado en la figura 17; para la primera figura se ha tomado un vecindario de radio  $d = 1$  alrededor de la neurona 13; para la segunda figura se ha tomado un vecindario de radio  $d = 2$ .



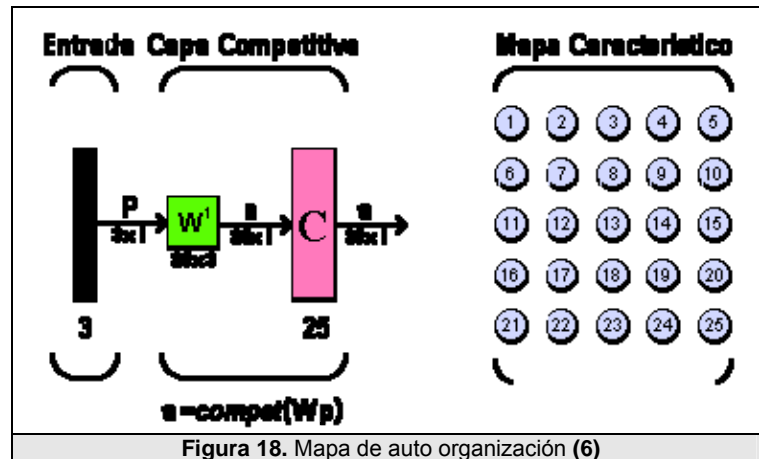
Estos vecindarios pueden definirse como sigue:

$$N_{13}(1) = \{8, 12, 13, 14, 18\}$$

$$N_{13}(2) = \{3, 7, 8, 9, 11, 12, 13, 14, 15, 17, 18, 19, 23\}$$

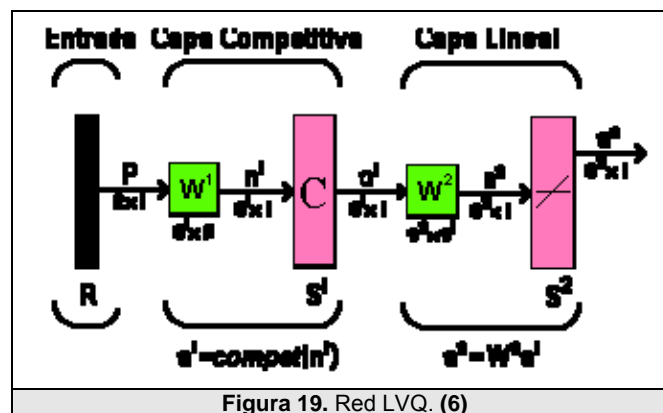
El vecindario puede determinarse en diferentes formas; Kohonen, por ejemplo ha sugerido vecindarios rectangulares o hexagonales para lograr alta eficiencia; es importante destacar que el rendimiento de la red no es realmente sensitivo a la forma exacta del vecindario.

La figura 18. Ilustra un mapa de auto organización de dos dimensiones



#### 1.1.4.4 Learning Vector Quantization (LVQ).

Esta red es un híbrido que emplea tanto aprendizaje no supervisado, como aprendizaje supervisado para clasificación de patrones.



En la red LVQ, cada neurona de la primera capa es asignada a una clase, después cada clase es asignada a una neurona en la segunda capa. El

número de neuronas en la primera capa, S1 debe ser mayor o al menos igual que el número de neuronas en la segunda capa, S2.

Al igual que con redes competitivas, cada neurona en la primera capa de la red LVQ aprende un vector prototipo, el cual permite a la neurona clasificar una región del espacio de entrada, sin embargo en lugar de calcular la distancia entre la entrada y el vector de pesos por medio del producto punto, la red LVQ calcula la distancia directamente. Una ventaja de hacer el cálculo de la distancia directamente, es que los vectores no necesitan ser normalizados, cuando los vectores son normalizados la respuesta de la red será la misma sin importar la técnica que se utilice.

La entrada neta a la primera capa de la red LVQ es entonces,

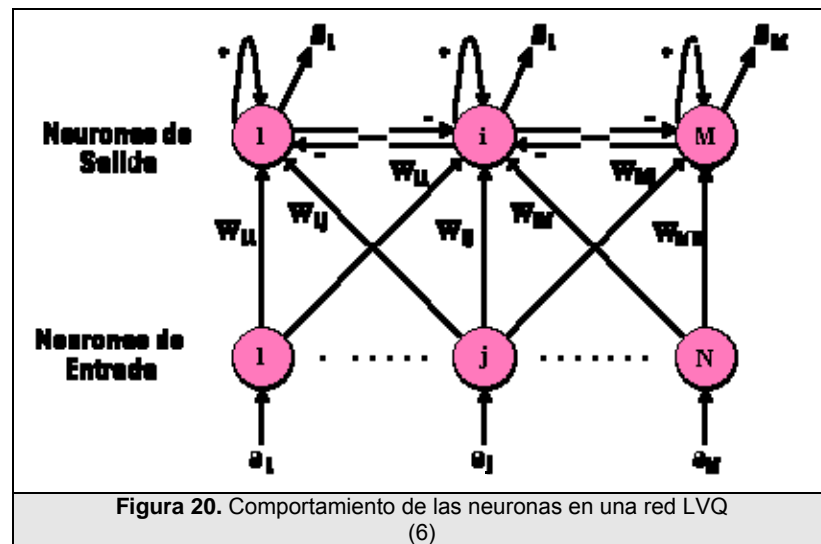
$$R_i^j = \left[ \begin{array}{c} |w_1^j - p| \\ |w_2^j - p| \\ \vdots \\ |w_n^j - p| \end{array} \right]$$

La salida de la primera capa de la red LVQ es,

$$a^1 = \text{compet}(n^1)$$

Así, la neurona cuyo vector de pesos este cercano al vector de entrada tendrá salida 1 y las otras neuronas, tendrán salida 0; en este aspecto la red LVQ se comporta igual a las redes competitivas, la única diferencia consiste

en la interpretación, mientras que en las redes competitivas la salida no cero representa una clase del vector de entrada, para el algoritmo LVQ, indica más bien una sub-clase, y de esta forma muchas neuronas (subclases), conforman una clase.



La segunda capa de la red LVQ es usada para combinar subclases dentro de una sola clase, esto es realizado por la matriz de pesos  $W^2$ . Las columnas de  $W^2$  representan las subclases y las filas representan las clases,  $W^2$  tiene un solo 1 en cada columna, todos los demás elementos son cero, la fila en la cual se presenta el 1 indica cual es la clase a la que la subclase pertenece.

$$W^2_{ki} = 1 \Rightarrow \text{la subclase } i \text{ pertenece a la clase } k$$

Una propiedad importante de esta red, es que el proceso de combinar subclases para formar clases, permite a la red LVQ crear clases más complejas. Una capa competitiva estándar tiene la limitación de que puede

crear solo regiones de decisión convexas; la red LVQ soluciona esta limitación.

La red LVQ combina aprendizaje competitivo con aprendizaje supervisado, razón por lo cual necesita un set de entrenamiento que describa el comportamiento propio de la red

$$p = \{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\}$$

Para ilustrar el desempeño de la red LVQ, se considerará la clasificación de un vector particular de tres elementos dentro de otro de cuatro clases, de esta forma:

$$\left\{ p_1 = \begin{bmatrix} \sqrt{0.74} \\ 0 \\ \sqrt{0.74} \end{bmatrix}, t_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \right\}$$

Antes de que suceda el aprendizaje, cada neurona en la segunda capa es asignada a una neurona de salida, así se genera la matriz  $W^2$ ; por lo general, igual número de neuronas ocultas son conectadas a cada neurona de salida, para que cada clase pueda ser conformada por el mismo número de regiones convexas. Todos los elementos de  $W^2$  son cero excepto los que cumplan la siguiente condición:

$$\text{Si la neurona } i \text{ es asignada a la clase } k \Rightarrow w_{2ki}=1$$

Una vez  $W^2$  ha sido definida, nunca será alterada. Los pesos ocultos  $W^1$  son actualizados por medio de la regla de Kohonen.

La regla de aprendizaje del algoritmo LVQ, trabaja de la siguiente manera:

1. En cada iteración, un vector de entrada  $p$  es presentado a la red y se calcula la distancia a cada vector prototipo.
2. Las neuronas ocultas compiten, la neurona  $i^*$  gana la competición y el  $i^*$ -ésimo elemento de  $a^1$  se fija en 1.
3.  $a^1$  es multiplicada por  $W^2$  para obtener la salida final  $a^2$ , la cual tiene solamente un elemento no cero,  $k^*$ , indicando que el patrón  $p$  está siendo asignado a la clase  $k^*$

La regla de Kohonen es empleada para mejorar la capa oculta de la red LVQ, en dos formas:

Primero, si  $p$  es clasificado correctamente los pesos de la neurona ganadora



se hacen tender hacia  $p$ .

$$i^*w(q) = i^*w(q-1) - a(q) (p(q) - i^*w(q-1)) \text{ si } a_{k^*}^2 = t_{k^*} = 1$$

Segundo, si  $p$  es clasificado incorrectamente una neurona equivocada ganó la competición y por lo tanto sus pesos  $i^*w^1$  se alejan de  $p$ .

$$i^*w(q) = i^*w(q-1) - a(q) (p(q) - i^*w(q-1)) \text{ si } a_{k^*}^2 = 1 \neq t_{k^*} = 0$$

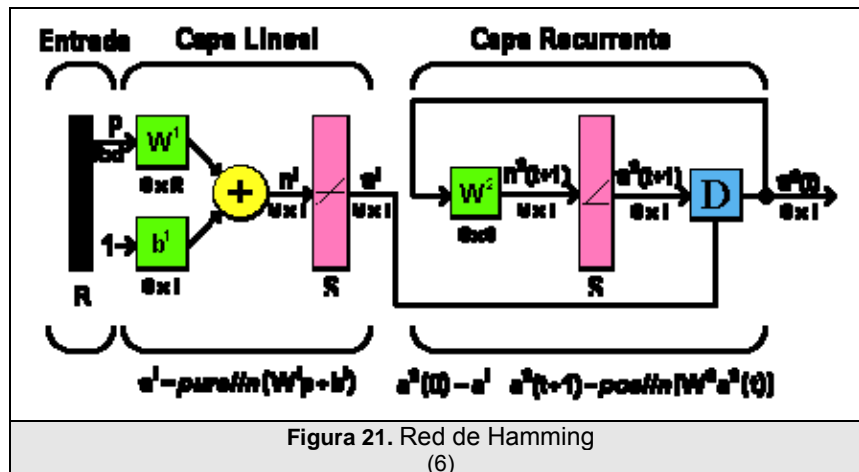
El resultado será que cada neurona se moverá hacia los vectores que cayeron dentro de la clase, para la cual ellos forman una subclase y lejos de los vectores que cayeron en otras clases.

#### **1.1.4.5 Red Neuronal de Hamming.**

La Red de Hamming ilustrada en la figura 21 es uno de los ejemplo más simples de aprendizaje competitivo, a pesar de ello su estructura es un poco compleja ya que emplea el concepto de capas recurrentes en su segunda capa y aunque hoy en día en redes de aprendizaje competitivo se ha simplificado este concepto con el uso de funciones de activación más sencillas, la red de Hamming representa uno de los primeros avances en este tipo de aprendizaje, convirtiéndola en un modelo obligado de referencia dentro de las redes de aprendizaje competitivo. Las neuronas en la capa de salida de esta red compiten unas con otras para determinar la ganadora, la cual indica el patrón prototipo más representativo en la entrada de la red, la competición es implementada por inhibición lateral (un conjunto de conexiones negativas entre las neuronas en la capa de salida).

Esta red consiste en dos capas; la primera capa, la cual es una red Instar, realiza la correlación entre el vector de entrada y los vectores prototipo, la

segunda capa realiza la competición para determinar cuál de los vectores prototipo está más cercano al vector de entrada.



Capa 1:

La red Instar es capaz de clasificar solo un patrón; para que múltiples patrones sean reconocidos se necesitan múltiples Instar y es precisamente de esa forma como está compuesta la primera capa de la red de Hamming. Para una mejor comprensión de su funcionamiento se partirá de unos vectores prototipo que la red debe clasificar

$$\{p_1, p_2, \dots, p_Q\}$$



La matriz de pesos " $W^1$ " y el vector de ganancias " $b^1$ " para la capa uno serán:

$$W^1 = \begin{bmatrix} W^T \\ W^T \\ \vdots \\ W^T \end{bmatrix} = \begin{bmatrix} P_1^T \\ P_2^T \\ \vdots \\ P_Q^T \end{bmatrix}, b^1 = \begin{bmatrix} R \\ R \\ R \\ R \end{bmatrix}$$

Donde cada fila de " $W^1$ " representa un vector prototipo, el cual deseamos reconocer y cada elemento " $b^1$ " es igual al número de elementos en cada vector de entrada ( $R$ ) (el número de neuronas  $S$  es igual al número de vectores prototipo  $Q$ ). Así la salida de la primera capa será:

$$a^1 = W^1 p + b^1 = \begin{bmatrix} P_1^T p + R \\ P_2^T p + R \\ \vdots \\ P_Q^T p + R \end{bmatrix}$$

La salida de la capa 1 es igual al producto punto de los vectores prototipo con la entrada más el vector  $R$ ; este producto indica cuan cercano está cada vector de entrada a los patrones prototipo.

Capa 2:

La red Instar emplea una función de transferencia "poslin" para decidir si el vector de entrada estaba lo suficientemente cerca al vector prototipo. En la capa 2 de la red de Hamming se utilizan múltiples Instar, así se determinara

por medio de una capa competitiva el patrón prototipo más cercano. Las neuronas en esta capa son inicializadas con la salida de la capa en realimentación, la cual indica la correlación entre los patrones prototipo y el vector de entrada. Las neuronas compiten unas con otras para determinar una ganadora; después de la competición solo una neurona tendrá una salida no cero. La neurona ganadora indica cual categoría de entrada fue presentada a la red (cada vector prototipo representa una categoría).

La salida de la primera capa  $a^1$  es usada para inicializar la segunda capa:

$$a^2(0) = a^1$$

La salida de la segunda capa está determinada de acuerdo a la siguiente relación recurrente:

$$a^2(t+1) = \text{poslin}(W^2 a^2(t))$$

Los pesos de la segunda capa  $W^2$  son fijados de tal forma que los elementos de la diagonal sean 1, y los elementos por fuera de la diagonal tengan pequeños valores negativos.

$$w_{ij}^2 = \left\{ \begin{array}{l} 1, \text{ si } i = j \\ -\epsilon \text{ de otra forma} \end{array} \right\} \quad \text{Donde } 0 < \epsilon < \frac{1}{n-1}$$

Esta matriz produce un efecto inhibitorio, en el cual la salida de cada neurona tiene un efecto inhibitorio sobre todas las otras neuronas. Para ilustrar este efecto, sustituimos los valores de pesos de 1 y  $-\epsilon$  por los apropiados

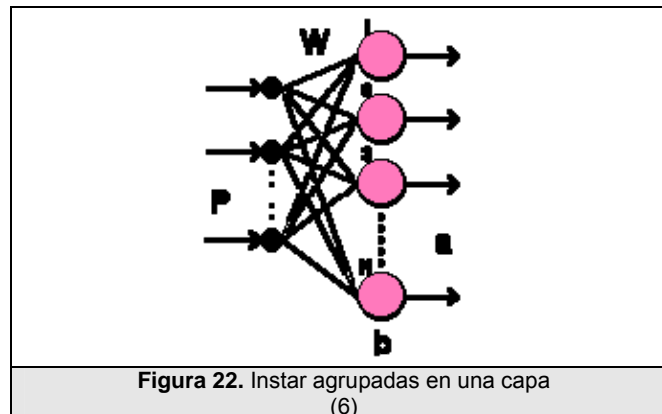
elementos de  $W^2$ . Reescribiendo la ecuación de salida de la red para una sola neurona se tiene:

$$a_i^2(t+1) = \text{poslin} \left( a_i^2(t) - \mu \sum_{j \neq i} a_j^2(t) \right)$$

En cada iteración, cada salida de la neurona se decrementará en proporción a la suma de las salidas de las otras neuronas. La salida de la neurona con la condición inicial más grande se decrementará más despacio que las salidas de otras neuronas; eventualmente cada neurona tendrá una salida positiva y en ese punto la red habrá alcanzado el estado estable.

#### 1.1.4.6 Estructura general de una Red Competitiva.

En las redes asociativas, se vio como la regla instar puede aprender a responder a un cierto grupo de vectores de entrada que están concentrados en una región del espacio. Se asume que se tienen varias instar agrupadas en una capa, tal como se muestra en la figura 22 cada una de las cuales responde en forma máxima a un cierto grupo de vectores de entrada de una región distinta del espacio.



Se puede decir que esta capa de Instars clasifica cualquier vector de entrada, porque la Instar con la mayor respuesta para alguna entrada dada es la que identifica a la región del espacio en la cual yace el vector de entrada. En lugar de examinar la respuesta de cada instar para determinar cuál es la mayor, la labor de clasificación sería más fácil si la Instar de mayor respuesta fuera la única unidad que tuviese una salida no nula; esto se puede conseguir si las instar compiten unas con otras por el privilegio de la activación, este es el principio de las redes competitivas.

Las neuronas de la segunda capa de la red de Hamming, están en competición porque cada neurona se excita a sí misma e inhibe a todas las otras neuronas, para simplificar la discusión se definirá una nueva función de transferencia que hace el trabajo de una capa recurrente competitiva

$$a = \text{compet}(n)$$

Donde  $a$  es la salida total de la red y  $n$  es la entrada neta a la función de transferencia,  $\text{compet}$  es una función de transferencia que encuentra el

índice  $i^*$  de la neurona con la entrada neta más grande y fija su salida en uno, todas las otras neuronas tienen salida 0.

$$a_i = \begin{cases} 1, & i = i^* \\ 0, & i \neq i^* \end{cases} \quad \text{Donde } n_p \geq n_i, \forall i, \quad i^* \leq i, \forall n_i = n_p$$

Reemplazando la capa recurrente de la red de Hamming, con una función de transferencia competitiva, la presentación de una capa competitiva se simplifica de la siguiente manera.

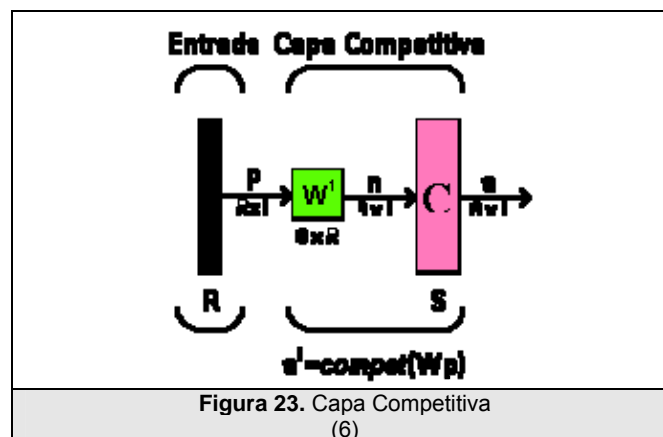


Figura 23. Capa Competitiva  
(6)

Como con la red de Hamming, los vectores prototipo son almacenados en las filas de  $W$ . La entrada neta  $n$  calcula la distancia entre el vector de entrada  $p$  y cada prototipo  $w_i$  (asumiendo que los vectores tienen longitudes normalizadas  $L$ ). La entrada neta " $n_i$ " de cada neurona es proporcional al ángulo  $\theta_i$  entre  $p$  y el vector prototipo  $w_i$ :

$$n = Wp = \begin{bmatrix} 1w^T \\ 2w^T \\ Bw^T \end{bmatrix} p = \begin{bmatrix} 1w^T p \\ 2w^T p \\ Bw^T p \end{bmatrix} = \begin{bmatrix} L^1_{out} & 1 \\ L^2_{out} & 2 \\ L^B_{out} & B \end{bmatrix}$$

La función de transferencia competitiva asigna una salida de 1 a la neurona cuyo vector de pesos apunte en la dirección más cercana al vector de entrada.

$$a = \text{compet}(n)$$

#### 1.1.4.7 Regla de Aprendizaje.

En este punto es posible diseñar una red competitiva que realice clasificaciones correctas fijando el valor de las filas de  $W$  en los valores del vector prototipo esperado, sin embargo es deseable tener una regla de aprendizaje que pueda entrenar los pesos en una red competitiva sin conocer los vectores prototipo, una de estas reglas es la Instar estudiada en el numeral 1.1.3.4

$${}_1w(q) = {}_1w(q-1) + a(q)(p(q) - {}_1w(q-1))$$

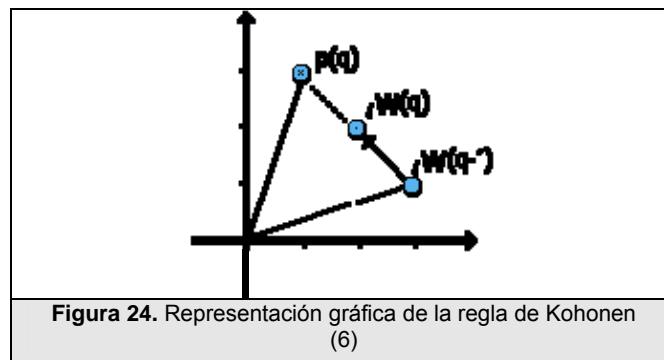
Para redes competitivas, "a" tiene un valor diferente de cero solamente para la neurona ganadora ( $i=i^*$ ), de esta forma los mismos resultados serán obtenidos utilizando la regla de Kohonen

$$w_i(q) = w_i(q-1) + \alpha (p(q) - w_i(q-1)) = (1 - \alpha) w_i(q-1) + \alpha p(q)$$

y

$$w_i(q) = w_i(q-1) \quad i \neq i^*$$

Así, la fila de la matriz de pesos que esté más cerca al vector de entrada (o tenga el producto punto más grande con el vector de entrada) se moverá hacia el vector de entrada. Este se mueve a lo largo de la línea entre la fila anterior del vector de pesos y el vector de entrada, como puede verse en la figura 24.



# CAPÍTULO 2

## 2 REDES NEURONALES ARTIFICIALES APLICADO EN LA CLASIFICACION DE FORMULARIOS

En este capítulo se presenta las diferentes clasificaciones de redes neuronales, según su aplicación, topología, aprendizaje; adicionalmente se detallará la justificación del uso de la red tipo “Kohonen” utilizado en el desarrollo de la tesis y la terminología utilizada.

### 2.1 Definición de la Red Neuronal Artificial y términos a usarse.

Se define como redes neuronales artificiales (RNA o en inglés como “ANN”) a los modelos matemáticos que simulan un comportamiento similar al cerebro humano, estos modelos se basan en fundamentos teóricos sobre el funcionamiento de las neuronas en acciones y reacciones que son provocadas por agentes externos, además de respuestas que se generan durante ese proceso, con la finalidad de recopilar información.

Estas redes neuronales son parte de lo que se conoce como “Minería de datos”, que consiste en un conjunto de técnicas para la inducción de conocimiento útil a partir de masas existentes de datos, teniendo gran



importancia en otras disciplinas como la estadística tradicional, el reconocimiento de patrones, la inteligencia artificial, etc.

Tomando en consideración que las redes neuronales artificiales surgen a partir de estudios realizados en el campo de la psicología, podemos abstraer términos de vital importancia para su definición en el campo de la inteligencia artificial, entre ellos podemos mencionar: comportamiento, aprendizaje, entrenamiento y conocimiento.

Se dice que una vez escogido el modelo de red neuronal a utilizar con sus respectivas características (capas y estructura), esta red debe tener una técnica de aprendizaje sobre los datos a entrenar para su posterior procesamiento y poder obtener el conocimiento de manera correcta.

Entre las técnicas de aprendizaje podemos mencionar el aprendizaje duro y aprendizaje blando, las cuales se construyen a partir de la combinación los siguientes factores:

- Una función de adyacencia
- Función de vecindad,
- Un factor de aprendizaje,
- Capas de una red,
- Tamaño de la red,
- Número de patrones de ingreso

Entre las diferentes funciones de adyacencia podemos mencionar:

- *La función de adyacencia Gaussiana.*- Es el grado de medida comparativa en entre una neurona con sus vecinas, presenta un aprendizaje en forma de sombrero mexicano como se lo conoce a este estilo, y obteniendo una onda descendente desde la neurona ganadora hacia sus vecinas, por ende se observa así un aprendizaje blando. (función aplicada en nuestro caso para el desarrollo de nuestra tesis).
- *La función de adyacencia escalón.*- Es el grado de medida comparativa en el que a diferencia del estilo anterior este opta por solo re-entrenar a la neurona vencedora provocando así un aprendizaje más duro, restringe a que las demás aprendan.

*El factor de aprendizaje.*- Es un coeficiente que permite realizar el aprendizaje de manera más acelerada o más lenta, pero se debe cuidar mantener un aprendizaje óptimo, este valor debe estar en un rango de 0 a 1. Para nuestro tema usaremos un valor de 0.2 y cuya recomendación generalizada es que sea mayor a 0.1. (4)

Capa: se lo define como un contenedor o caja negra en el cual se realiza un proceso específico con los datos, generalmente se dice que la capa de entrada es para el ingreso de datos y la capa de salida es para la emisión de los resultados de la neurona. En algunas Redes existen capas ocultas que modifican los datos para poder obtener resultados interpretables. Una capa puede contener 1 o más neuronas.

Una neurona debe recibir las entradas de las neuronas vecinas y calcular un valor de salida, el cual es enviado a todas las neuronas restantes. Toda la información que la red recibe, almacena, transforma y envía, se expresa en términos matemáticos, en la forma de cantidades y de fórmulas aplicadas a cantidades.

El tamaño de la red: comprende el número de neuronas y sus diferentes capas que serán utilizados en la red neuronal que se utilizará para resolver el problema que se plantee.

Para nuestro caso específico, el tamaño de la red será el equivalente a mencionar el número de neuronas que se utilizarán como centroides en el mapa de Kohonen de dimensiones  $N \times N$ , (el valor de  $N$  se define en el capítulo 5). Es de importancia considerar que no se recomienda un valor de  $N$  demasiado pequeño debido a la excesiva aglomeración de patrones en

pocas neuronas, ni tampoco se recomienda un valor de N demasiado extensos porque podría existir demasiadas neuronas vacías.

El término que se menciona de manera común es el “mapa”, este es un hiperplano, cuya característica principal es la de reducir problemas de dimensiones mayores y llevarlas a dos dimensiones manteniendo sus propiedades.

*El número de patrones de ingreso.*- El valor de patrones ingreso depende del tamaño de la población, también se lo denomina tamaño de la muestra. El rango de obtención de muestra está entre (500, 10000) para poblaciones mayor a 500, y en un 75% de la misma para tamaño de muestra menor a 500.

Entre otros términos utilizados podemos mencionar:

Formulario de encuesta: será equivalente al “cuestionario evaluado” por el usuario (estudiante, profesor) que para la RNA se considerará como “patrón de ingreso”.

Patrones neuronales: son los vectores de pesos asociados a cada neurona entrenada, nos indican el patrón de los diferentes comportamientos de los

formularios de encuesta evaluados durante el proceso de evaluación docente.

Para entender el término de conglomerado se indica brevemente un concepto más genérico. Las llamadas técnicas de agrupaciones o conglomerados, intentan agrupar objetos similares mediante algoritmos matemáticos. Estas técnicas aplicadas al ámbito de la recuperación de información consiguen crear de forma automática la clasificación de patrones considerando ciertas similitudes en sus pesos.

Una vez conocido los diferentes términos a utilizar durante el transcurso de esta tesis, es de importancia destacar que la representación interna de la información se genera en las sinapsis o conexión de las entradas con las neuronas del mapa y en la actividad de las neuronas con sus vecinas. Dado que las redes modifican sus estados a través del tiempo como consecuencia de las distintas entradas de información, es imprescindible también introducir en la descripción de los estados de activación la referencia al tiempo, esto se analiza más adelante en la elección del tipo de Red seleccionada. (Ver Sección 1.1.2.2)

## **2.2 Análisis y elección de la topología de la Red Neuronal Artificial.**

Como parte del análisis realizado sobre las diferentes topologías de las redes neuronales podemos encontrar que para seleccionar una red neuronal acorde a nuestro requerimiento (sobre clasificación de formularios), se deberá tomar en cuenta los tipos de RNAs según su aplicación, según el tipo de conexión, según el número de capas, según el tipo de aprendizaje, cada una de estas redes neuronales tienen un modelo matemático diferentes, pero en ciertos casos son ligeramente diferentes.

Una vez elegido la red neuronal ideal para el problema planteado, se deberá proceder a definir los parámetros de aprendizaje (como son el factor de aprendizaje, # patrones para entrenamiento), los cuales siempre serán mediante prueba y error.

Para una mejor explicación de los tipos de redes artificiales existentes y analizando su comportamiento de acuerdo a su categoría, hacemos referencia las siguientes tablas:

La tabla 2 presenta las redes neuronales según sus tipos de conexiones en redes monocapa, en la cual se clasifican principalmente por su

comportamiento de Auto recurrente, (Se define como recurrente que la salida de una neurona sea entrada de la misma neurona para la siguiente iteración).

<b>Tipos de Conexiones</b>		<b>Modelo de red</b>
Conexiones Laterales Explícitas	Conexiones Autorrecurrentes	Brain-State-In-a-Box
		Additive Grossberg (AG)
		Shunting Grossberg (SG)
		Optimal Linear Associative Memory
	No autorrecurrentes	Hopfield
		Boltzmann Machine
Cauchy Machine		
CROSSBAR		Learning Matrix (LM)

**Tabla 2.** Redes Neuronales Monocapa. (4)

En los modelos monocapa se debe obtener previamente patrones con una salida conocida para el entrenamiento. Es utilizado para regeneración información de entrada que se presenta en la red, es decir información incompleta o distorsionada.

La tabla 3 detalla los diferentes tipos de redes para capas mayor a 2, y cuyos tipos de conexiones pueden variar entre las conexiones hacia adelante o hacia atrás, conexiones recurrentes o no recurrentes, con conexiones laterales o no.

En la misma tabla se presenta el grupo al que pertenece el modelo SOM (Self Organization Map - Kohonen), una red de dos capas con conexiones laterales que se basa en la competitividad para ganar y determinar el vecindario que puede aprender, las conexiones laterales se generan en base a la función de adyacencia elegida para el proceso.

La tabla 4-1 presenta modelo de Redes según el tipo de aprendizaje supervisado, sub-clasificados en tipo off-line y on-line, que indica el momento que se ejecuta el proceso de Red (tanto del entrenamiento como de la clasificación), es decir que sea como proceso batch o durante la ejecución del mismo.

<b>Nº de capas</b>	<b>Tipo de Conexiones</b>		<b>Modelo de Red</b>
<b>2 C A P A S</b>	Conexiones Hacia adelante  (FEEDFORWARD)		Adaline/Madaline
			Perceptron
			Linear/Assoc. Rewar Penalty
			Linear Associative Memory
			Optimal Linear Assoc. Mem.
			Drive-Reinforcement (DR)
		Conexiones Laterales implícitas y Autorrec.	Learning Vector Quantizer
			Topology Preserving Map (TPM o SOM)
	Conex. Adelante/ Atras  (FEEDFORWARD/ FEEDBACK)	Sin Conexiones Laterales.	Bidirectional Assoc. Mem (BAM)
			Adaptive BAM
Temporal Assoc. Memory (TAM)			
Fuzzy Associative Memory (FAM)			
Competitive Adaptive BAM			
	Con Conexiones Laterales y Autorrec.	Adaptive Resonance Theory (ART)	
<b>3</b>	Conexiones hacia adelante  (FEEDFORWARD)	Sin Conexiones Laterales	Adaptive Heuristic Critic (AHC)
			Boltzmann/Cauchy Machine
		Con Conexiones Laterales	Counter Propagation
			Boltzmann/Cauchy Machine
		Conexiones Adelante/Atrás y Laterales	Boltzmann/Cauchy Machine
<b>N</b>	Conexiones hacia adelante		Back-Propagation (BPN)
	FEEDFORWARD-FEEDBACK (Jerarquía de niveles de capas bidimensionales)		Cognitron/Neocognitron

**Tabla 3.** Redes Neuronales Multicapa más conocidas. **(4)**



Tipo de Aprendizaje Supervisado		Modelo de Red
Aprendizaje por corrección de error	OFF LINE	Perceptron
		Adaline/Madaline
		Backpropagation
		Bain-State-In-A-Box
		Counterpropagation
Aprendizaje por refuerzo	ON LINE	Linear Reward Penalty
		Associative Rew. Penalty
		Adaptive Heuristic Critic
Aprendizaje Estocástico	OFF LINE	Boltzmann Machine
		Cauchy Machine

**Tabla 4-1.** Tipos de redes con aprendizaje supervisado más conocidos. (4)

Tipo de Aprendizaje NO Supervisado		Modelo de Red
Aprendizaje Hebbiano	O F F  L I N E	Hopfield
		Learning Matrix
		Temporal Assoc. Memory
		Linear Associative Memory (LAM)
		Optimal LAM
		Drive-Reinforcement
		Fuzzy Associative Memory
	O N  L I N E	Additive Grossberg
		Shunting Grossberg
		Bidirectional Associative Memory (BAM)
Aprendizaje Competitivo /Cooperativo	O F F	Learning Vector Quantizer
		Cognitron / Neocognitron
	L I N E  O N	Topology Preserving Map
		Adaptive Resonance Theory

**Tabla 4-2.** Redes con aprendizaje no supervisado más conocidas. (4)

La tabla 4-2 detalla las redes neuronales del tipo de aprendizaje no supervisado, con su respectiva sub-clasificación de online y offline.

Tomando como referencia estas cuatro tablas mencionadas, le debemos adicionar la recomendación del uso ideal de cada modelo de red neuronal, ya que no existe una regla o teorema específico sobre el tipo de red se debe aplicar para un tema a desarrollar.

En cuyos datos de entrenamiento no es necesario conocer su salida, ya que éste se auto clasifica determinando los patrones de comportamientos de los patrones ingresados.

### **2.3 Justificación del tipo de Red Neuronal Artificial.**

Entre los requerimientos básicos que se necesitan para la utilización de esta tesis para la discriminación de formularios se presentan:

- La automatización de la clasificación de formularios.
- El proceso de clasificación debe realizarse sin ningún ente humano.
- Determinar los formularios necesarios para el entrenamiento.
- Realizar el proceso al finalizar el levantamiento de los formularios.
- Clasificar un cuestionario de N preguntas.

Existiendo otros factores como:

- La falta de conocimiento de los formularios válidos o no válidos por un humano.
- La subjetividad de validación de los formularios
- Las diferentes teorías que existen sobre los formularios válidos en la evaluación docente.

Tomando en consideración los requerimientos y los factores encontrados, junto con las 4 tablas presentadas en el capítulo 2.2, sobre las posibles soluciones de red neuronal, podemos definir las características de la red que utilizaremos, las cuales son:

- Tipo offline: ya que este proceso se deberá realizar una vez al finalizar una evaluación docente que puede durar entre 1 semana a 3 meses de intervalo para este proceso de evaluación docente.
- Aprendizaje no supervisado: debido a que no existe ente humano que pueda ser capaz de definir los formularios válidos y no válidos, por lo tanto una red de tipo no supervisado, podría detectar estos patrones de manera automática.

Una manera efectiva para detectar estos patrones (formularios válidos y no válidos) es mediante la creación de conglomerados o grupos de formularios,

que se agruparían según su comportamiento, el cual es representado por el modelo denominado Kohonen. Este tipo de red se basa en la competencia, es decir, no existen dos valores de respuesta para un patrón ingresado, simplemente existe uno que es el conglomerado que más se asocia al patrón de ingreso, en otras palabras cada formulario de encuesta evaluado pertenecerá a un conglomerado, y cada conglomerado se diferencia de otra por el patrón aprendido dentro de él, que técnicamente sería los pesos que se han readecuados en el proceso de entrenamiento.

Cabe destacar que previamente se realizaron pruebas con una red supervisada, pero se obtuvo un inconveniente en el desarrollo de la aplicación, ya que el modelo supervisado obliga a tener un valor de supervisión (humana), con el que se determina si un patrón de ingreso o formulario de encuesta evaluado pertenece o no a un conglomerado; es decir, además de no conocer cómo obtener los conglomerados se debía indicar a cuál de ellos pertenecería un patrón de ingreso sin conocerlos aún, Es evidente el problema que se presentó; usando una red supervisada en lugar de dar una solución generaba otra interrogante, este método de red supervisada necesitaba la fase inicial del modelo de Red de Kohonen, para la obtención de categorías o conglomerados.

Este modelo Kohonen, emplea dos fases completamente definidas, la primera realiza la creación de conglomerados, denominada como fase de aprendizaje o entrenamiento, y la segunda usa los grupos creados por la primera fase para generar la clasificación, esto nos evita el uso de tener un “**supervisor interno**” (humano) en la red y ayuda a que los resultados estén directamente ligados a los patrones de ingreso.

Es así, que se logra obtener una mejor visión sobre los resultados obtenidos en la clasificación de los formularios de encuesta evaluados, aplicando el modelo de Red de Kohonen y determinando las diferentes clases de patrones basados en los cuestionarios que se encuentren en un mismo grupo o población seleccionada de la encuesta.

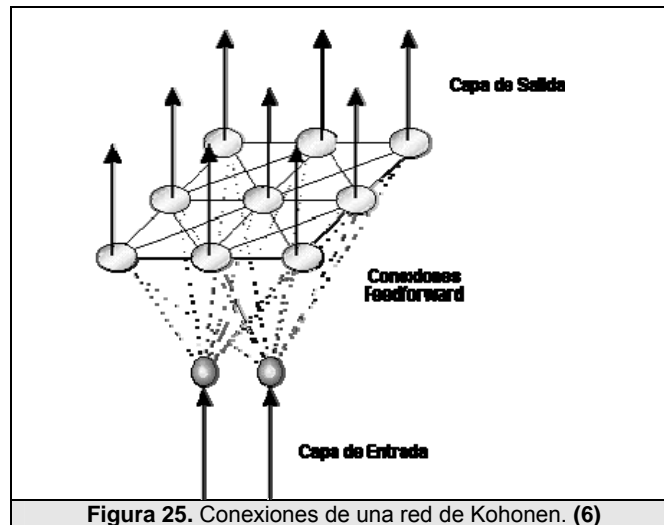
Una vez que se ha justificado su uso procedemos a detallar la estructura de Red que presenta el modelo SOM de Kohonen.

#### **2.4 Estructura de la Red Neuronal Artificial.**

Se conoce que el tipo de Red es un modelo NO SUPERVISADO, por lo tanto dentro de su estructura no constará de valor alguno para determinar la tendencia de uno u otro grupo de valores de respuesta, es decir, lo que en los modelos supervisados se conoce como “umbral” no será necesario debido a que por la característica competitiva del mismo, permite que se

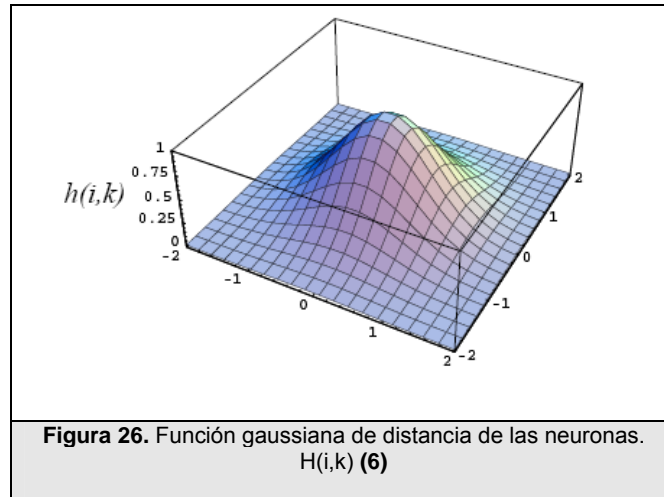
determine la respuesta de la función de activación. El modelo de Kohonen presenta una función de activación característica que determina la neurona ganadora dentro del mapa en base a la relación de semejanza que exista entre un patrón de ingreso y el correspondiente vector de pesos asociado a la neurona evaluada, como se ve la relación de los términos presentados dan a conocer la estructura de la Red Neuronal y determinan su funcionalidad.

Se determina la estructura lógica de la RNA SOM de Kohonen de la siguiente manera: Se utiliza dos capas, *la primera* es la *capa de entrada* que consta de un grupo de neuronas, las mismas que presentan los patrones de aprendizaje (formularios de encuesta evaluados de una encuesta) a la red, y envía una respuesta mediante las conexiones hacia delante a *la segunda capa* conocida como *capa de salida* ó mapa de características de Kohonen (Featuring map), y que mediante el proceso o fase de aprendizaje se modifican hasta alcanzar un estado estable, producido por la ausencia de cambios.



La figura 25 presenta la estructura del modelo de Red de Kohonen, la capa de entrada relacionada a los patrones de ingreso, son estructuras de datos similares a los cuestionarios evaluados, las neuronas del mapa tienen asociados vectores, estos vectores presentan una estructura similar a los patrones de ingreso, pero estos valores a diferencia de los de ingreso son modificables.

Se ha mencionado que el tipo de aprendizaje puede ser blando o duro según sea el tipo de función de adyacencia que se halla empleado para el proceso de entrenamiento.



La figura 26 muestra una función gaussiana la misma que se emplea para el aprendizaje blando, y es la que se usa en el modelo de Red de este estudio.

Como parte de la estructura de la Red se encuentra el algoritmo o el modelo matemático que genera la funcionalidad del mismo, en este caso el algoritmo de aprendizaje se diferencia al algoritmo presentado en la sección 1.1.4.2 por el uso de la función gaussiana. Siendo la siguiente:

1. En primer lugar, se inicializan los pesos ( $W_{ji}$ ) con valores aleatorios en un rango de (0 a 1), valores determinados por el algoritmo como vectores acordes a los cuestionarios evaluados, aunque durante el entrenamiento estos se modifican hasta llegar a adaptarse a valores coherentes, valores que sean semejantes a los patrones de ingreso.



2. A continuación se presenta a la red una información de entrada (patrón de ingreso en la fase de entrenamiento, y patrón a ser clasificado en la fase de aplicación) en forma de vector  $E_k = (e_i^{(k)}, \dots, e_n^{(k)})$ , cuyas componente  $e_i^{(k)}$  serán valores continuos, como se ha indicado respuestas de los cuestionarios evaluados de una encuesta seleccionada.
3. Puesto que se trata de un aprendizaje competitivo, se determina la neurona vencedora de la capa de salida. Esta será aquella  $j$  cuyo vector de pesos  $W_j$  (vector cuyas componentes son los valores de los pesos de las conexiones entre esa neurona y cada una de las neuronas de la capa de entrada) sea el más parecido a la información de entrada  $E_k$  (patrón o vector de entrada).

Para ello, se calculan las distancias o diferencias entre ambos vectores, considerando una por una todas las neuronas de salida. Suele utilizarse la distancia euclídea o la siguiente expresión, que es similar a aquella, pero eliminando la raíz cuadrada:

$$d_j = \sum_{i=1}^N (e_i^{(k)} - W_{ji})^2 \quad i \leq j \leq M$$

Siendo:

$e_i^{(k)}$ : Componente  $i$ -ésimo del vector  $k$ -ésimo de entrada.

$W_{ji}$ : Peso de la conexión entre la neurona  $i$  de la capa de entrada y la neurona  $j$  de la capa de salida.

4. Una vez localizada la neurona vencedora ( $j^*$ ), se actualizan los pesos de las conexiones entre las neuronas de entrada y dicha neurona, así como los de las conexiones entre las de entrada y las neuronas vecinas de la vencedora (este paso y el 5to solo se lo realiza en la fase de entrenamiento).

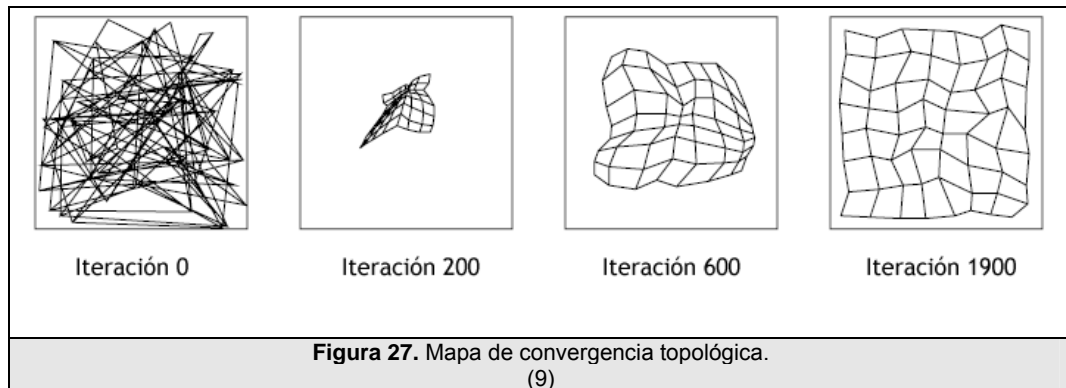
$$w_i(n+1) = w_i(n) + \Lambda_{i,i^*}(n) \eta(n) (x(n) - w_i(n))$$

$$\Lambda_{i,i^*}(n) = e^{\left( \frac{-d_{i,i^*}^2}{2\sigma^2(n)} \right)}$$

Donde  $d$  es la distancia entre las neuronas y  $\sigma$  disminuye con la iteración

El parámetro lambda es la función gaussiana que genera el aprendizaje blando, función que varía en función del número de iteraciones y que es inversamente proporcional a la distancia que exista entre la neurona ganadora y las neuronas vecinas.

5. El proceso se debe repetir, volviendo a presentar todo el juego de patrones de aprendizaje  $E_1, E_2, \dots$  un mínimo de 500 veces, hasta encontrar invariación en los valores de pesos.



La figura 27 presenta el proceso iterativo de manera gráfica, se puede notar como el algoritmo va generando una estructura de malla que está conformado por las neuronas de Red.

## 2.5 Funcionamiento de la Red Neuronal Artificial.

Recordando que la Red Neuronal de Kohonen trabaja en dos instancias, una es la fase de entrenamiento y otra que se conoce como fase de aplicación o funcionamiento. Ambas fases se realizan para mantener el correcto proceso de la Red, y en el orden indicado, primero se realiza el entrenamiento y luego se emplea la red entrenada en la fase de aplicación.

Sin embargo, en muchas ocasiones es necesario la existencia de una tercera fase conocida como “fase de pruebas” (testing) que determina si los valores de los coeficientes como son: el coeficiente de entrenamiento, coeficiente de

la función de activación, son acordes a lo esperado en la salida de la red. Generalmente, estos valores son determinados por el experto en redes neuronales y es mayormente empleado en el entrenamiento de redes supervisadas.

Como se conoce, la obtención de datos que se generan está dada por un módulo de formularios, los mismos que serán tomados como parte de la capa de entrada de la primera fase de la Red Neuronal, y por ende no consta como parte de las fases de este proceso.

Es sencillo determinar cómo es el funcionamiento de una RNA debido a que todas tienen funcionamiento similar; es decir, comienzan por la fase de entrenamiento y concluyen en la fase de aplicar el entrenamiento pero se diferencian en las especificaciones de los algoritmos que intervienen en las fases. Cada modelo cambia su metodología de funcionamiento en las fases que implementa, es por ello entonces que se debe enfocar más en esas especificaciones. Se ha determinado un ítem de este capítulo para el proceso de entrenamiento y un capítulo completo para la etapa de aplicación.

Por lo tanto a continuación se desarrollara el proceso de entrenamiento de la Red Neuronal de Kohonen de manera más detallada.

## **2.6 Entrenamiento de la Red Neuronal Artificial.**

Sin dejar a un lado la etapa de clasificación, cabe mencionar que esta fase del proceso Neuronal de la Red es de mayor importancia en comparación con el proceso de clasificación, esto se debe a que los valores que se emitan en los resultados serán los que determinen una buena o mala clasificación, como todo proceso de aprendizaje, la práctica depende de lo bien que se aprende.

La fase de entrenamiento también conocida como aprendizaje, obtiene los datos del módulo de formularios como parte de la capa de entrada, estos datos son obtenidos mediante la selección de la encuesta a clasificar y especificando el cuestionario asociado a la misma, es necesario que la encuesta haya concluido su proceso de recolección de datos para poder obtener los patrones de ingreso, debido a que el proceso de Kohonen es off-line, y se necesita tener el tamaño aproximado de la población para la selección del tamaño de la muestra.

Adicionalmente a la encuesta seleccionada se obtiene un cuestionario que se aplicó en la misma, que contiene un N número de preguntas las cuales son equivalentes a las N entradas del patrón de ingreso y a la dimensión N de los vectores de pesos de cada neurona que se inicializan para el entrenamiento.

Luego, el usuario selecciona el número de patrones de ingreso (formularios de encuesta evaluados) acorde a las recomendaciones realizadas en la Sección 2.1.

Estos patrones seleccionados de la encuesta son ingresados uno a uno y pasados a ser evaluados (comparados) por medio de las conexiones que existe entre la capa de entrada y capa de salida para obtener un valor de activación que determine si el patrón ingresado es similar o semejante al patrón asociado en las conexiones de ambas capas, en otras palabras una comparación entre el patrón de ingreso y el vector relacionado a la neurona del mapa, este proceso se lo realiza con todas las neuronas.

Una vez realizado este proceso, la comparación entre el patrón de ingreso con todas las neuronas, los valores de comparación son utilizados para determinar cuál es el valor mínimo y mediante la elección escoger a la neurona que obtuvo dicho valor y designarla como “vencedora”, que por ende tiende a ser la que más se asemeja al patrón de ingreso.

Es así que procedemos a iniciar la fase de entrenamiento, que desde el punto de vista neuronal es recalcular los pesos asociados a la neurona vencedora y además los de las neuronas vecinas, según lo permita la función gaussiana. Como se indicó en la sección 1.1.4.2 el modelo matemático es:

$$w_i(n+1) = w_i(n) + \Lambda_{i,i^*}(n) \eta(n) (x(n) - w_i(n))$$

$$\Lambda_{i,i^*}(n) = e^{\left(\frac{-d_{i,i^*}^2}{2\sigma^2(n)}\right)}$$

Donde  $d$  es la distancia entre las neuronas y  $\sigma$  disminuye con la iteración

Donde  $w_i(n+1)$ : es el vector de pesos recalculado que resultan de la evaluación del vector anterior más una función lambda multiplicada por el factor de adyacencia y por la diferencia que existe entre el patrón de ingreso y el vector anterior.

Lambda es la función gaussiana que genera un aprendizaje blando tomando como parámetros la distancia de la neurona vecina y el número de iteraciones que se van suscitando en el proceso, es evidente observar que el valor negativo de la distancia en la función gaussiana provoca que a una mayor distancia menor sea el aprendizaje.

Con la obtención de los nuevos valores el mapa queda listo para recibir un nuevo patrón y realizar nuevamente el proceso anterior, este se repite hasta que todos los patrones hayan sido ingresados obteniéndose una malla con valores de vectores característicos invariantes, si la variación es notoria se debe ingresar nuevamente los valores para obtener un mapa óptimo. Se determina el mapa óptimo mediante el uso de la fórmula de Error Cuadrático el mismo que debe ser menor a 0.01.

Una vez que se han procesado todos los patrones seleccionados, la red queda lista para el proceso de clasificación de formularios de encuesta evaluados.

## **2.7 Comparación del uso de la Red vs. otras soluciones para clasificación.**

Se ha mencionado que existen herramientas similares que realizan el proceso de clasificación y que se basan en procesos estadísticos, entre ellas podemos mencionar el proceso de clasificación mediante la aplicación de la distancia de mahalanobis, y la aplicación de una regresión logística sobre los formularios de encuesta evaluados de un paralelo, además de la existencia de procesos que encuentran grupos de conglomerados con modelos supervisados como se dio a conocer anteriormente, pero conociendo de antemano los conglomerados.

Se puede determinar un grado de comparación entre modelos diferentes de clasificación basado en el tiempo de procesamiento, además de comparar el modo de resolución de los mismos, y finalmente demostrar las ventajas de usar el modelo de Red Neuronal.

La siguiente tabla muestra el tiempo que demanda cada proceso para resolver el mismo número de datos, hasta llegar a generar la solución.



<b>Modelo Discriminador</b>	<b>Tamaño de la Muestra</b>	<b>Tiempo de Resolución</b>	<b>Aplicable a</b>
Modelo discriminante mediante el uso de la distancia de Mahalanobis	500 cuestionarios evaluados	20 min	Cualquier tamaño de muestra
Modelo discriminante mediante el uso de Regresión logística	500 cuestionarios evaluados	20 min.	Aplicable a muestras con tamaño $n \geq 30$ (por el teorema del límite central)
Modelo discriminante mediante el uso de Red Neuronal (Kohonen)	500 cuestionarios evaluados	4 horas.	Aplicable a cualquier tamaño de muestra, generalmente usado para muestras extensas.

**Tabla 4.** Comparativos entre modelos

Sin embargo haciendo referencia a la tabla 5, se podría dar apertura a un incógnita sobre: “Si la solución a nuestro problema de tesis se pudo haber realizado por estos dos métodos diferentes y en menor tiempo”, ya que se aparentemente tendría una solución igual.

Como respuesta a esta incógnita debemos aclarar que existe una gran diferencia en la efectividad de discriminación mediante estos dos métodos, en donde los textos referentes a clasificación son claramente explícitos que los resultados de clasificación mediante redes neuronales son más exactos, ya que su capacidad para delimitar los diferentes conglomerados es más

precisa, en comparación a los otros métodos que utilizan la distancia de mahalanobis o la euclidiana.

A continuación se presenta la definición sobre las distancias aplicadas en el algoritmo para discriminación de formularios y su explicación sobre la forma para delimitar los conglomerados:

**Distancia Mahalanobis.-** Es una medida de distancia introducida por Mahalanobis en 1936. Su utilidad radica en que es una forma de determinar la similitud entre dos variables aleatorias multidimensionales. Se diferencia de la distancia euclídea en que tiene en cuenta la correlación entre las variables aleatorias. (18)

Formalmente, la distancia de Mahalanobis entre dos variables aleatorias con la misma distribución de probabilidad  $\vec{x}$  y  $\vec{y}$  con matriz de covarianza  $\Sigma$  se define como:

$$d_m(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})}.$$

(Ecuación 14) Distancia de Mahalanobis

### **Propiedades:**

La distancia de Mahalanobis cumple las siguientes propiedades, necesarias para ser una distancia:

$$d(a, b) \geq 0 \quad \forall a, b \in X \qquad d(a, b) = 0 \text{ si } a = b$$

(Ecuación 15) Semipositividad de Mahalanobis

Es decir, la distancia entre dos puntos de las mismas coordenadas es cero, y si tienen coordenadas distintas la distancia es positiva, pero nunca negativa.

$$d(a, b) = d(b, a) \quad \forall a, b \in X$$

(Ecuación 16) Simetricidad

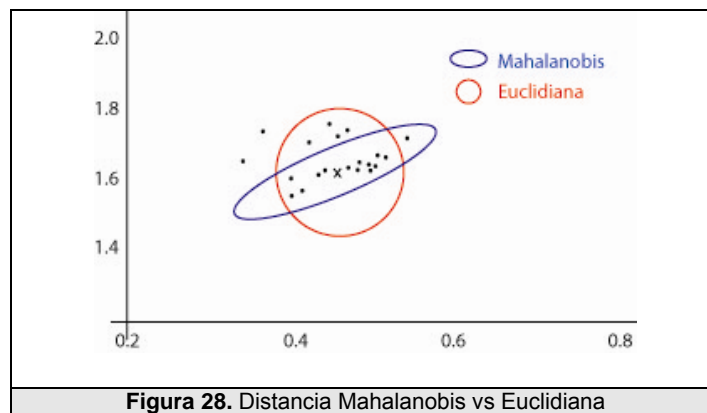
Intuitivamente, la distancia ente a y b es la misma que entre b y a.

$$d(a, b) \leq d(a, c) + d(c, b) \quad \forall a, b, c \in X$$

(Ecuación 17) Desigualdad triangular

Como se observa el modo de resolución es la creación de grupos basados en la medida de una distancia la cual solo es positiva y sin indicar el grado real de diferencia de un formulario de encuesta a otro, es decir pueden existir cuestionarios completamente distantes, pero por los módulos de las distancias se declaran semejantes, en contraste con el modelo de Red Neuronal, este determina de manera real la semejanza de un cuestionario con otro en vista que no se usan normas de alteración de valores y emplea patrones para diferenciarlos.

En la utilización de esta distancia como método de discriminación se representa de manera gráfica con la forma de una elipse (Ver figura 28), en donde los formularios válidos serán todos los formularios que se encuentre dentro del perímetro de la misma y los formularios no válidos se encontrarán fuera del mismo.



**Distancia Euclidiana.-** Es la raíz cuadrada de la suma de las diferencias al cuadrado entre los valores de dos casos para cada variable

$$d(p_1, p_2) = \sqrt{\text{pow}(x_2 - x_1, 2) + \text{pow}(y_2 - y_1, 2)}$$

En la utilización de esta distancia como método de discriminación se representa de manera gráfica con la forma de un círculo (Ver figura 28), en donde los formularios válidos serán todos los formularios que se encuentre

dentro del perímetro de la misma y los formularios no válidos se encontrarán fuera del mismo.

**Regresión logística.-** permite la elaboración de un análisis de regresión para estimar y probar la influencia de una variable sobre otra, cuando la variable dependiente o de respuesta es de tipo dicotómico.

# CAPÍTULO 3

## 3 PROCESO DE CLASIFICACION DE LOS FORMULARIOS

El tercer capítulo trata sobre el proceso de clasificación de formularios o cuestionarios evaluados, las recomendaciones para obtener un buen clasificador, así como los requerimientos tecnológicos para el funcionamiento del mismo.

Este proceso es el que determina qué cuestionarios dentro de una encuesta se clasificarán en válidos o no válidos basados en teorías de distancias, y luego de haber sido considerada una categorización de las mismas.

### 3.1 Descripción del proceso de clasificación.

Para iniciar el proceso de clasificación debemos considerar los siguientes puntos:

- La encuesta debe ser seleccionada para la obtención de los cuestionarios a clasificar, algo similar que el proceso de entrenamiento.

- La clasificación se podrá aplicarse para un formulario con N número de preguntas y M número de alternativas que utilicen escala Likert.
- Para cada encuesta deberá generarse un mapa con sus respectivos pesos iniciales, para luego ser entrenado.
- Un mapa neuronal que corresponde a una encuesta, que ha sido previamente entrenado, y todas sus neuronas mantienen un vector de pesos asociados como patrón.

Una vez verificado estos puntos previos, la fase de clasificación se presenta de manera similar a la de entrenamiento, diferenciándose que en lugar de buscar algún patrón por neurona, éste ya existe y mediante el uso de dicho patrón los cuestionarios se van agrupando y formando los conglomerados.

De igual manera todos los patrones de ingreso deben ser pasados uno a uno al algoritmo clasificador, el mismo que determina la semejanza que existe entre el patrón ingresado y el patrón asociado a la neurona entrenada, de existir semejanza el patrón queda relacionado a la neurona y el número de patrones se incrementa dentro de la misma, generando así categorías pobladas. De no existir semejanza entre algún patrón de ingreso y neurona del mapa, la categoría queda vacía, y se la define como neurona sin clase, aunque tenga asociada un patrón de entrenamiento. Como se observa la técnica para determinar conglomerados es más sencilla en contraste a la de

entrenamiento debido a que el proceso más fuerte se lo realiza en la primera fase.

Asociado al proceso de creación de conglomerados existen las técnicas de visualización de resultados en el mapa, estas técnicas son empleadas para generar una diferenciación gráfica en la discriminación realizada por el algoritmo.

Mediante técnicas de visualización apropiadas se pueden construir mapas bidimensionales del espacio de entrada donde se ilustran todas las muestras disponibles. Esto facilita, entre otras cosas, la identificación de conglomerados, traslapes entre muestras de diferentes clases, fronteras entre clases y, también el muestreo y partición de los datos en conjuntos de entrenamiento y validación.

En la versión básica los datos se representan en un arreglo bidimensional, ubicándolos en las celdas correspondientes a los prototipos más cercanos. Adicionalmente se pueden utilizar representaciones que permitan distinguir la distancia entre prototipos, las formaciones de grupos, y visualizar la clasificación, tales como las celdas vacías, celdas pobladas.



Añadiendo un factor discriminador que permite determinar la existencia de grupos válidos y no válidos dentro de cada conglomerado nos permite completar el proceso clasificatorio obteniendo de esta manera la solución deseada.

Se determina un valor de clasificación de los formularios, aplicando un radio del 90% (modificable), con la finalidad de obtener los formularios más cercanos a su neurona, con este factor y el uso de las técnicas de visualización nos permite presentar resultados que se pueden interpretar de una manera menos compleja.

Cabe destacar que dentro de la teoría estadística, se considera que un formulario se considera “no válido”, siempre y cuando el ente humano considere que este formulario X, tendrá una dispersión considerable con respecto a los demás formularios. Esto conlleva a una subjetividad la cual está sujeta a la opinión personal del responsable de esta tarea. Y adicionalmente se deberá realizar un máximo del 10% de discriminación dentro de este proceso. Es así que se ha procedido a realizar una agrupación (clusterización) de los datos y luego se realiza la discriminación de los formularios hasta un máximo del 10% dentro de cada conglomerado.

Entre otros puntos considerados en la investigación fue la previa entrevista de personas expertas en el tema, el cual en muchas ocasiones no podían establecer que formularios pueden ser considerados como válidos y no válidos, ya que humanamente es muy difícil su tarea, sin embargo acoplándose a las normas de la estadística, junto con la necesidad de automatización los únicos parámetros de clasificación de los formularios, es la existencia de los cuatro métodos más comunes que son aplicados a la clasificación, como es el caso de:

- Clasificación usando la distancia de Mahalanobis
- Clasificación usando la distancia euclidiana
- Clasificación usando regresión logística
- Clasificación usando redes neuronales.

Siendo este último el más ideal acorde a la teoría estadística.

### **3.2 Consideraciones para obtener un buen clasificador.**

Un buen clasificador usando redes neuronales se lo define como aquel que determina de manera clara y confiable los grupos de cuestionarios de la encuesta, para tales motivos se deben considerar los siguientes puntos:

- La red debe entrenarse con un número mínimo de patrones entre 500 a 10.000, sin embargo si el número de neuronas es menor a 500,

entonces se tomará como base las  $\frac{3}{4}$  partes de la población de formularios de una encuesta.

Recomendación para evitar la falta de conocimiento en la red, y además que al momento de crear grupos no se generen agrupaciones que presenten la generalización hacia unos pocos cuestionarios.

- El número de neuronas dentro del mapa de Kohonen es un factor importante para decidir qué tan bueno es el clasificador, cada neurona del mapa presenta un patrón asociado (vector de pesos).

La dimensión del mapa puede variar en base a las pruebas, debido a que si tenemos demasiadas neuronas podría provocar información demasiado dispersa, existirían categorías que sean bastante similares. Por otro lado, si obtenemos lo contrario, es decir muy pocas neuronas, podremos obtener un grupo demasiado condensado, por lo tanto una sola neurona del mapa sería activada por todos los formularios.

Es por esto que se recomienda realizar pruebas para poder llegar a obtener un número de neuronas adecuado para categorizar. (Ver Capítulo 5 Determinación de Mapa).

- Otro requisito a considerarse para que el clasificador funcione de manera eficaz es la normalización, es decir que al momento de ingresar las respuestas de los formularios deben ser llevados a un rango de 0 a 1 pues de esta manera lo utiliza la red, debido a que la matriz de pesos asociada a las neuronas se aleatoriza con valores en este rango.

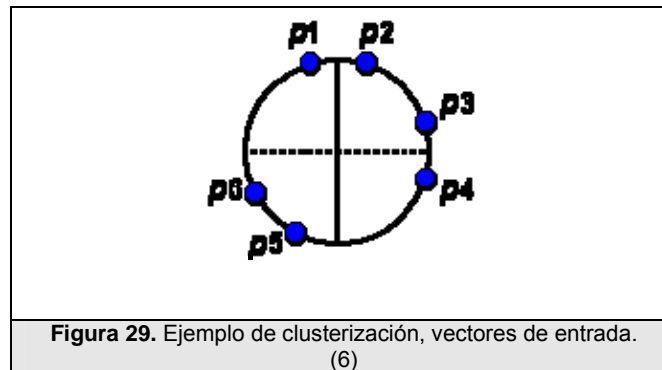
### 3.3 Ejemplo de clasificación.

En esta demostración de clasificación, el proceso se realiza con la evaluación de un factor de comparación, que indica el formulario a que categoría (neurona) pertenece y luego determina que tan distante se encuentra este formulario del patrón asociado a la neurona en la que se ha categorizado. Tomemos un caso en particular.

Para demostrar cómo trabaja una red competitiva, se creará una red que clasifique los siguientes vectores:

$$P_1 = \begin{bmatrix} -0.216 \\ 0.993 \end{bmatrix}, P_2 = \begin{bmatrix} 0.216 \\ 0.993 \end{bmatrix}, P_3 = \begin{bmatrix} -0.993 \\ 0.216 \end{bmatrix}$$

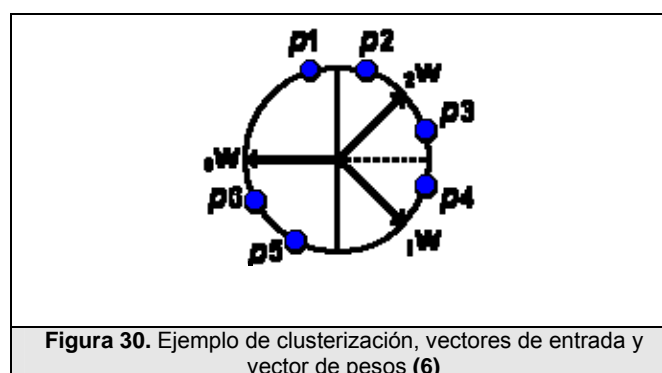
$$P_4 = \begin{bmatrix} 0.993 \\ -0.216 \end{bmatrix}, P_5 = \begin{bmatrix} -0.622 \\ -0.873 \end{bmatrix}, P_6 = \begin{bmatrix} -0.873 \\ -0.622 \end{bmatrix}$$



La red tendrá tres neuronas, por lo tanto los vectores serán clasificados en tres clases o grupos, esta es una de las principales características de las redes competitivas, ellas pueden agrupar los patrones de entrada en clases que no se conocen. Los pesos normalizados escogidos aleatoriamente son:

$${}_1w = \begin{bmatrix} 0.7071 \\ -0.7071 \end{bmatrix}, \quad {}_2w = \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix}, \quad {}_3w = \begin{bmatrix} -1.000 \\ 0.000 \end{bmatrix}, \quad W = \begin{bmatrix} {}_1w^T \\ {}_2w^T \\ {}_3w^T \end{bmatrix}$$

Los vectores de datos y los pesos asignados pueden visualizarse en la figura 30.



Se presenta a la red el vector p2

$$s = \text{compet}(W, p2) = \text{compet} \left( \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \\ -1.000 & 0.000 \end{bmatrix}, \begin{bmatrix} 0.216 \\ 0.993 \end{bmatrix} \right)$$

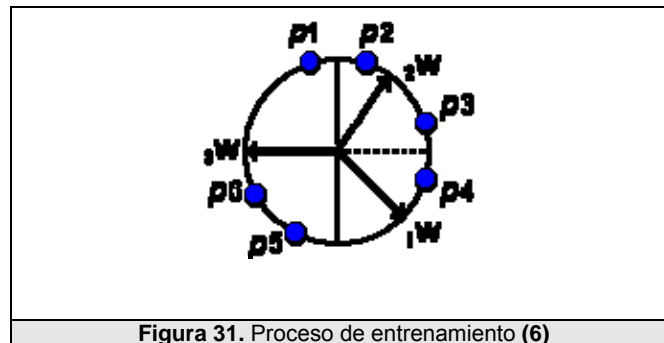
$$s = \text{compet} \left( \begin{bmatrix} -0.5494 \\ 0.8549 \\ -0.2160 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

El vector de peso de la segunda neurona estaba más cercano a p2, por lo tanto ganó la competición ( $i^*=2$ ) y su salida es 1. Ahora se aplicará la regla de Kohonen a la neurona ganadora con una tasa de aprendizaje  $\alpha=0.5$

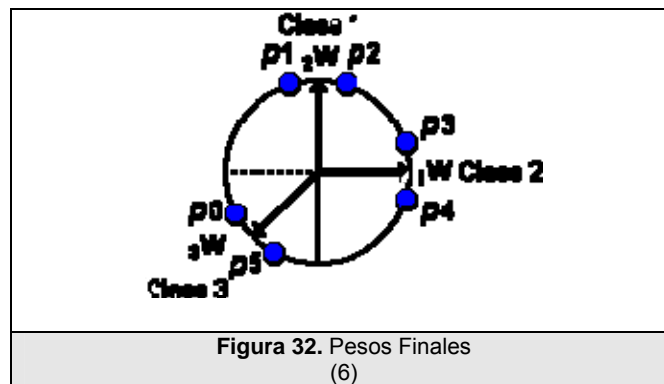
$${}_2W^{\text{nuevo}} = {}_2W^{\text{anterior}} + \alpha (p2 - {}_2W^{\text{anterior}})$$

$${}_2W^{\text{nuevo}} = \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix} + 0.5 \left( \begin{bmatrix} 0.216 \\ 0.993 \end{bmatrix} - \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix} \right) = \begin{bmatrix} 0.9527 \\ 0.5641 \end{bmatrix}$$

La regla de Kohonen hace que  ${}_2W$  tienda hacia p2 como puede verse en la figura 30, si continuamos escogiendo vectores de entrada aleatoriamente y presentándoselos a la red, en cada iteración el vector de pesos se acercará más al vector de entrada.



Cada vector de pesos apuntará hacia una clase diferente del vector de entrada, convirtiéndose en un prototipo para esa clase. Cuando se termine el proceso de entrenamiento, los pesos finales se verán como aparece en la figura 32.



Aquí están preparadas las categorías halladas por el grupo de patrones de entrenamiento, se han organizado las neuronas y asociado a sus respectivos patrones, Ahora para clasificar formularios se realiza la evaluación de los mismos y se observa que se categorizan en una neurona en particular, es decir si el formulario se encuentra a una distancia considerada razonable del patrón asociado a la neurona.

Este es un ejemplo que demuestra la resolución para un problema de dos dimensiones, pero aplicado a la tesis el ingreso de valores o patrones de entrada se encuentra en un rango mucho mayor y además variable ya que

depende del número de preguntas que tenga un cuestionario que fue aplicado en la encuesta.

### 3.4 Modelo(s) Gráfico(s) de la clasificación.

- a. Selección de formularios de encuesta para emplearlos como patrones de entrenamiento de la red.

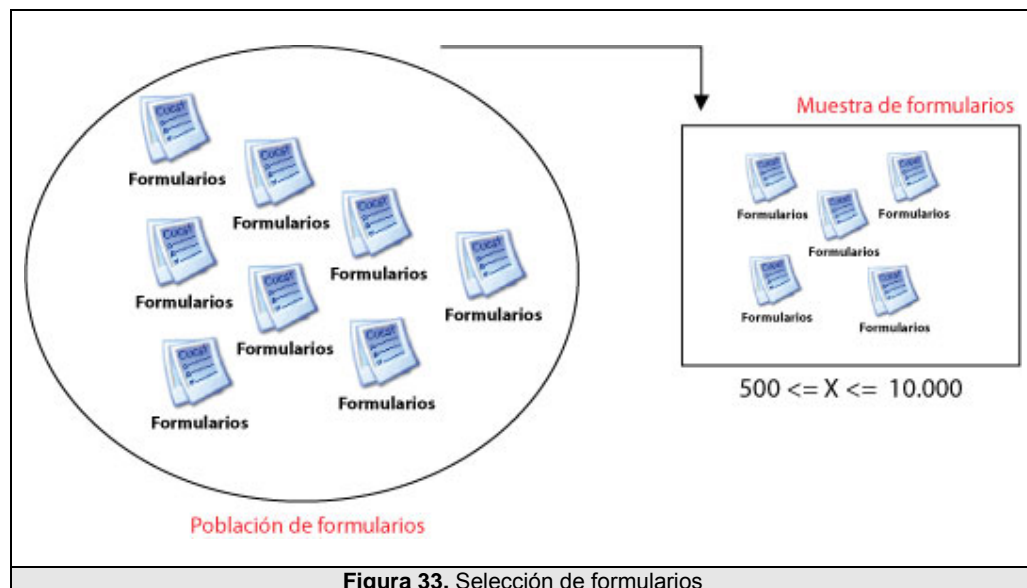
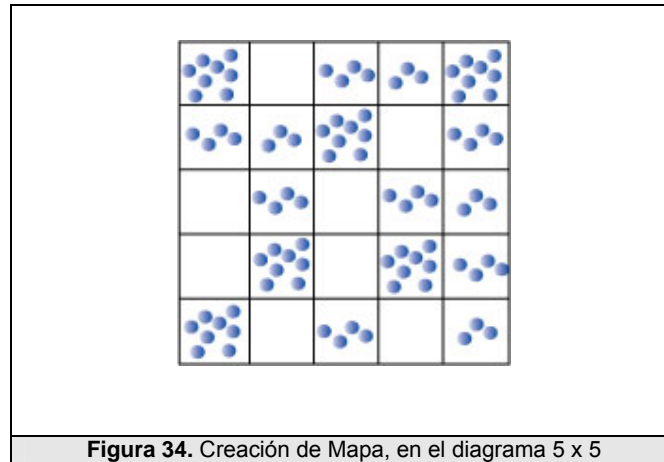


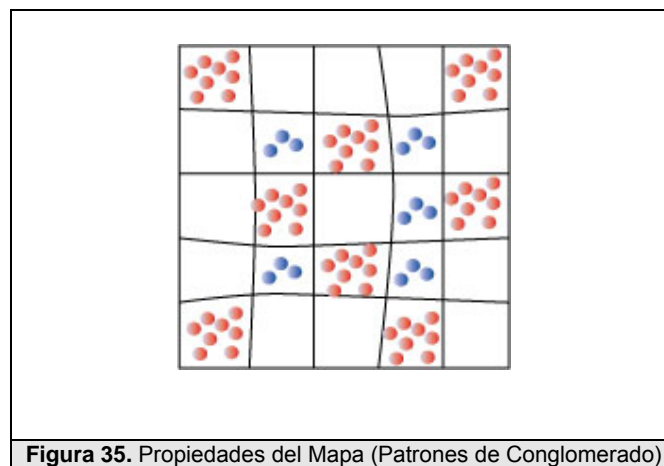
Figura 33. Selección de formularios

- b. Se realiza la creación del mapa para su respectivo entrenamiento.

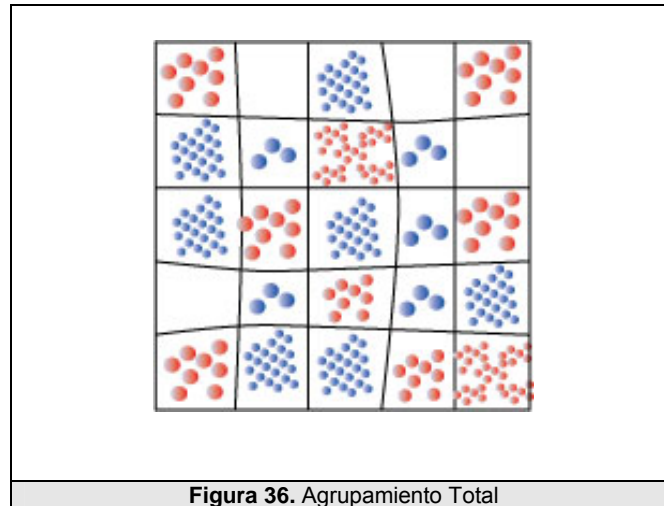




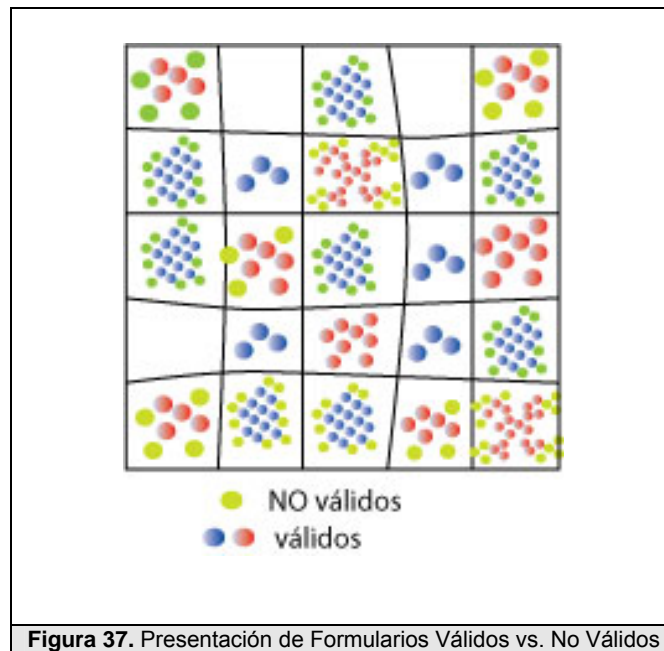
- c. Finalizando el entrenamiento se puede ver el mapa con sus nuevas propiedades.



- d. Encontramos los grupos con todos los formularios de la muestra total.



- e. Clasificamos entre formularios válidos y no válidos, aplicando el 10% de discriminación.



### **3.5 Componentes que intervienen en la clasificación.**

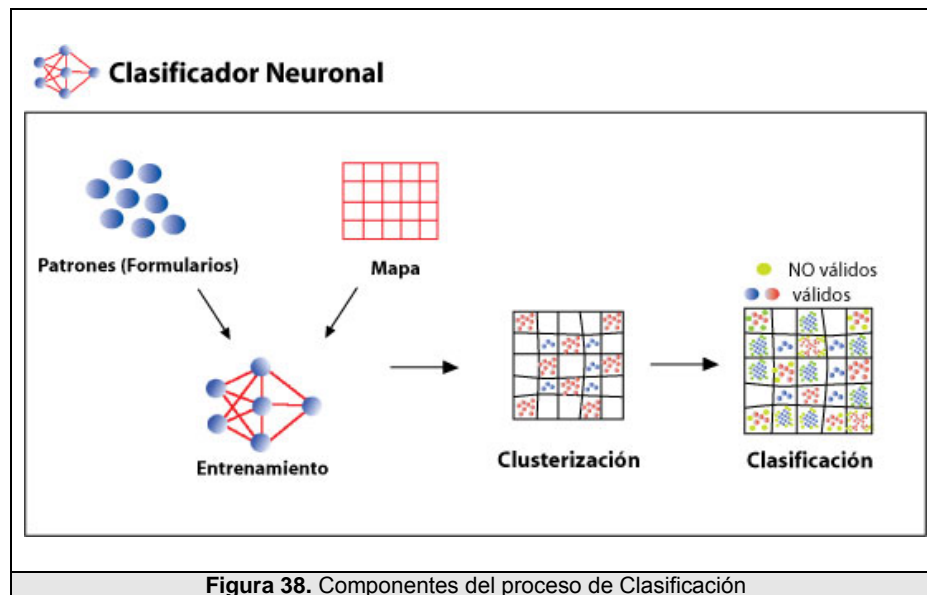
Se ha denominado componente a elementos que brindan funcionalidad a la red. El componente inicial de la clasificación son los patrones de ingreso que son generados por el proyecto de encuesta, este proyecto contiene asociado una o varias encuestas realizadas.

Este es el punto de partida del proceso de Red Neuronal, cabe dar a notar que con la encuesta seleccionada se determinan los patrones de entrenamiento que se utilizarán en la Red.

El grupo de formularios a clasificar determina el propósito de la tesis, por tal motivo tiene una relevancia al momento de catalogarlo como componente.

Otro de los componentes importantes de la clasificación, es el cuestionario asociado a la encuesta seleccionada, que permite determinar el tamaño del vector de pesos, y el uso de identificadores para permitir una relación cuestionario-vector de pesos que estará referenciado por cada neurona, basado en el número de preguntas del cuestionario en cuestión.

Uno de los componentes necesarios dentro de la clasificación es el mapa de neuronas ó mapa de características luego del proceso de entrenamiento, un mapa con patrones claramente definidos permiten realizar una clasificación confiable.



Finalmente hablar de patrones neuronales es nombrar a cada componente dentro de la neurona de un mapa entrenado (para este caso los patrones serán equivalentes a los formularios), también conocidos como vectores de pesos, cumplen una función particular para el proceso de clasificación, debido que se basa en ellos para determinar los buenos o malos formularios.

# CAPÍTULO 4

## 4 DESCRIPCION DEL SISTEMA DE CLASIFICACION DE LOS FORMULARIOS

En este capítulo se presenta el análisis y diseño del sistema, el análisis y diseño del modelo de datos, los diferentes módulos que intervienen, la relación que existe entre módulos y los procesos que se ejecutan dentro de cada módulo.

### 4.1 Componentes ó módulos del sistema.

El sistema emplea los siguientes módulos:

1. Módulo de encuestas.- este módulo nos permite elegir la encuesta a la cual se realizará la discriminación de los formularios aberrantes.
2. Módulo de obtención de formularios.- permite realizar la toma de muestra de datos recomendada para el entrenamiento del mapa de Kohonen.
3. Módulo de entrenamiento.- permite realizar el entrenamiento inicial para obtener los pesos adecuados.

4. Módulo de clasificación.- permite realizar la determinación de los formularios válidos y no válidos de la encuesta acorde a la optimización realizada por el usuario.
5. Módulo de reportes.- muestra los diferentes resúmenes de los formularios válidos y no válidos usando los tres métodos discriminadores, que son el método usando la distancia de mahalanobis, usando la distancia euclidiana, usando regresión logística y usando el método neuronal.

Cada módulo contiene procesos que se deben ejecutar de manera secuencial.

## **4.2 Análisis y Diseño del Sistema.**

### **Casos de uso**

1. Generar mapa
2. Entrenar red neuronal
3. Procesar datos
4. Clasificar formularios

## Especificación de casos de usos

### Caso de Uso 1. Generar mapa

**Nombre:** Generar mapa

**Descripción:** El operador elije la encuesta y las dimensiones a la cual será generada el mapa para su red neuronal.

**Notas:**

- La encuesta tiene formularios evaluados.

### Caso de Uso 2. Entrenar red neuronal

**Nombre:** Entrenar red

**Descripción:** El realiza el proceso para entrenamiento de la red.

**Notas:**

- No podrá ser entrenado si no tiene un mapa generado

### Caso de Uso 3. Procesar datos en la red

**Nombre:** Procesar datos en la red

**Descripción:** El operador procesará los datos usando la red neuronal.

**Notas:**

- La red deberá estar previamente entrenada

Caso de Uso 4.- Clasificación de formularios

**Nombre:** Clasificar formularios

**Descripción:** El operador determinará que formularios y/o conglomerados serán válidos o no, en la encuesta

**Especificación de escenarios**

**1. Caso de uso: Generar Mapa**

Escenario 1.1: Generar Mapa exitosamente

**Suposiciones:**

- Existen encuestas disponibles
- Existen formularios en las encuestas
- Existen formularios evaluados

**Resultados:**

- Se genera un mapa con los pesos iniciales para la red neuronal.

Escenario 1.2.- Generar Mapa fallidamente por que no hay encuesta

**Suposición:**

- No existe encuesta disponible

**Resultado:**

- No se puede crear mapa, por falta de datos.



Escenario 1.3.- Generar Mapa fallidamente por que no hay formularios

**Suposición:**

- No existe formularios disponibles

**Resultado:**

- No se puede crear mapa, por falta de datos.

Escenario 1.4.- Generar Mapa fallidamente por cancelación del usuario

**Suposición:**

- Usuario cancela la operación

**Resultado:**

- No se puede crear mapa, por falta de datos.

**2. Caso de uso: Entrenar red neuronal**

Escenario 2.1: Entrenar red neuronal exitosamente

**Suposiciones:**

- Existen un mapa creado.
- Existen formularios evaluados, para entrenar.

**Resultados:**

- Se genera una red entrenada, con los pesos adecuados para luego ser aplicado en los datos.

Escenario 2.2.- Entrenar red neuronal fallidamente porque el mapa no está creado

**Suposición:**

- No existe un mapa creado.

**Resultado:**

- No se puede entrenar la red.

Escenario 2.3.- Entrenar red neuronal fallidamente porque cancela el usuario

**Suposición:**

- El usuario cancela la operación.

**Resultado:**

- No se puede entrenar la red.

Escenario 2.4.- Entrenar red neuronal fallidamente porque no existen formularios

**Suposición:**

- No existen formularios existentes para entrenar la red.

**Resultado:**

- No se puede entrenar la red.

### **3. Caso de uso: Procesar datos con la red neuronal**

#### Escenario 3.1: Procesar datos con la red neuronal exitosamente

##### **Suposiciones:**

- Existen un mapa creado.
- Existe una red neuronal entrenada.

##### **Resultados:**

- Entregar datos procesados por la red neuronal, con su respectivo conglomerado.

#### Escenario 3.2.- Procesar datos con la red neuronal fallidamente porque el usuario cancela.

##### **Suposición:**

- El usuario cancela la operación.

##### **Resultado:**

- Datos sin procesar.

### **4. Caso de uso: Clasificación de formularios**

#### Escenario 4.1: Clasificación de los formularios

##### **Suposiciones:**

- Existen formularios
- Existen conglomerados de datos
- Existe usuario que clasifica los formularios y/o conglomerados.

**Resultados:**

- Entrega los formularios clasificados en válidos y no válidos

Escenario 4.2: Clasificación de los formularios fallida por cancelación del usuario

**Suposición:**

- El usuario cancela la operación.

**Resultado:**

- Formularios / conglomerados no están clasificados.

\* Los diagramas de interacción a objetos se encuentran en el Anexo 1.

### 4.3 Análisis y Diseño del modelo de datos.

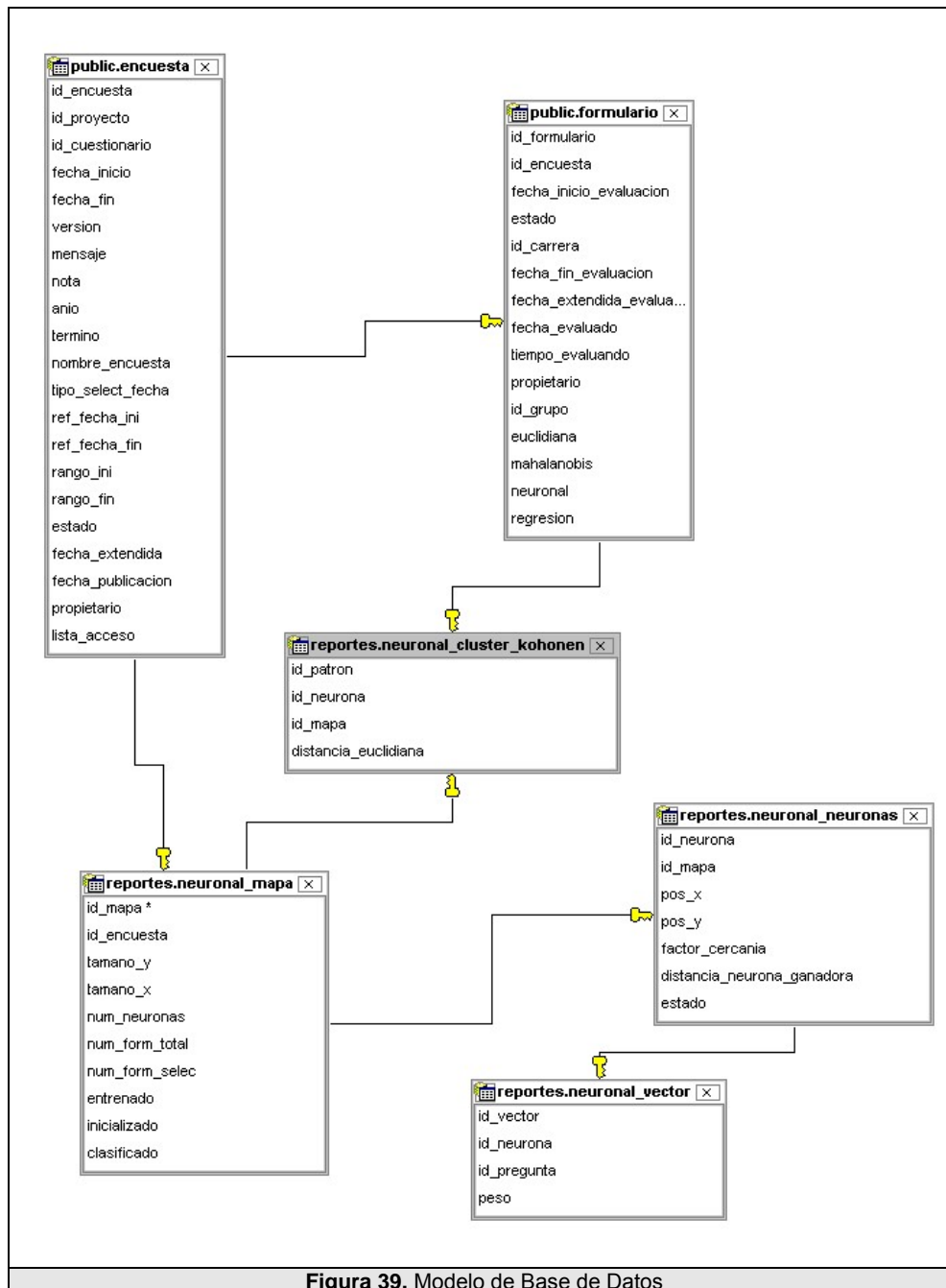


Figura 39. Modelo de Base de Datos

**NOTA:** El diccionario de datos se encuentra dentro del Anexo 2.

#### 4.4 Módulos externos al Sistema.

##### Módulo WEB

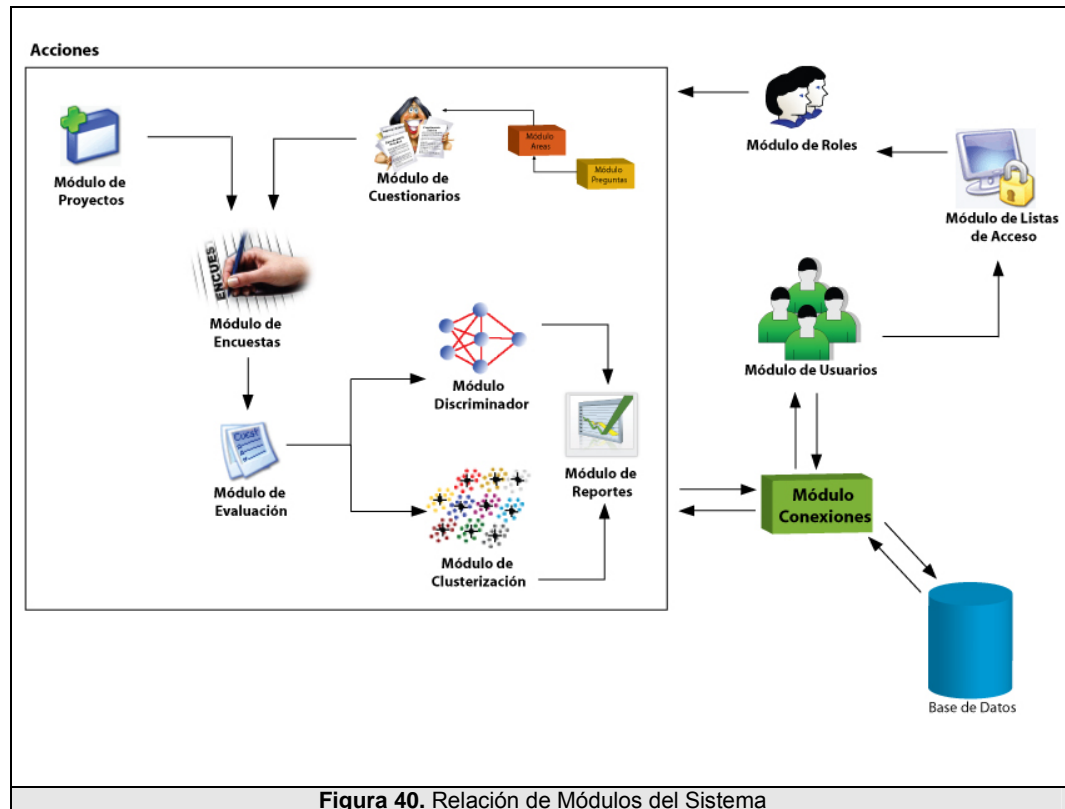
- Módulo de Usuarios.- contiene los usuario del sistema
- Modulo Cuestionario.
  - Sub-Módulo Áreas.- tiene las áreas que conforman el cuestionario
  - Sub-Módulo Preguntas.- tiene las preguntas que conforman un área.
  - Sub-Módulo Alternativas.- tiene las alternativas de una pregunta
- Modulo de Encuestas (Evaluado-Evaluador).- contiene las encuestas con sus diferentes formularios generados, ya sean evaluados o no.
- Módulo categorizador de comentarios.- categorizador manual sobre los comentarios de una encuesta.
- Módulo clusterización de profesores.- agrupa a los profesores en 5 diferentes comportamientos.
- Módulo Roles.- establece los diferentes accesos y permisos que tendría cada rol del sistema.
- Módulos Listas de Acceso.- agrupa un grupo de personas para manejar uno o más roles.
- Módulo discriminadores.- clasifica los formularios válidos y nó validos
  - Sub-Modulo Mahalanobis, utiliza la distancia de mahalanobis

- Sub-Módulo Euclidiana, utiliza la distancia euclidiana
- Sub-Módulo Regresión logística, utiliza regresión logística como discriminador.
- Sub-Modulo Red neuronal, utiliza redes neuronales
  - Submódulo optimización
  - Submódulo clasificación
- Módulo de Reportes.- Entrega los diferentes reportes empresariales sobre los discriminadores, encuestas, proyectos, entre otros.
- Módulo de Estadísticas de visitas, permite verificar el uso y acceso al sistema en los diferentes horarios de los usuarios.
- Log de Actividades.- registra cada actividad realizada en el sistema.
- Conexiones con ADODB.- permite la utilización independiente de otros tipos de base de datos.

### **Módulo Aplicación**

- Discriminador Neuronal
  - Sub-módulo encuestas
  - Sub-módulo entrenamiento
  - Sub-módulo clusterización
- Discriminador Mahalanobis
  - Sub-módulo mahalanobis
  - Sub-módulo euclidiana

#### 4.5 Diagrama explicativo de la relación entre módulos.



#### 4.6 Descripción del funcionamiento de cada módulo.

- Módulo de Usuarios.- contiene los usuarios del sistema, y el permiso de los mismos.
- Modulo Cuestionario.
  - Sub-Módulo Áreas.- tiene las áreas que conforman el cuestionario



- Sub-Módulo Preguntas.- tiene las preguntas que conforman un área.
- Sub-Módulo Alternativas.- tiene las alternativas de una pregunta
- Módulo de Encuestas (Evaluado-Evaluador).- contiene las encuestas con sus diferentes formularios generados, ya sean evaluados o no.
- Módulo categorizador de comentarios.- categorizador manual sobre los comentarios de una encuesta.
- Módulo clusterización de profesores.- agrupa a los profesores en 5 diferentes comportamientos.
- Módulo Roles.- establece los diferentes accesos y permisos que tendría cada rol del sistema.
- Módulos Listas de Acceso.- agrupa un grupo de personas para manejar uno o más roles.
- Módulo discriminadores.- clasifica los formularios válidos y no válidos
  - Sub-Módulo Red neuronal, utiliza redes neuronales
  - Sub-Módulo Mahalanobis, utiliza la distancia de mahalanobis
  - Sub-Módulo Euclidiana, utiliza la distancia euclidiana
  - Sub-Módulo Regresión logística, utiliza regresión logística como discriminador.
- Módulo de Reportes.- Entrega los diferentes reportes empresariales sobre los discriminadores, encuestas, proyectos, entre otros.

- Módulo de Estadísticas de visitas, permite verificar el uso y acceso al sistema en los diferentes horarios de los usuarios.
- Log de Actividades.- registra cada actividad realizada en el sistema
  
- Conexiones con ADODB.- permite la utilización independiente de otros tipos de base de datos.
  
- Discriminador Neuronal
  - Submódulo encuestas.- se elige la encuesta cuyos formularios serán discriminados.
  - Submódulo entrenamiento.- permite el entrenamiento del mapa para su correcta clusterización.
  - Submódulo clusterización.- permite clusterizar todos los formularios.
  
- Discriminador Mahalanobis
  - Sub-módulo mahalanobis.- determina los formularios válidos basándose en la distancia de mahalanobis, y cuyo centro en la media de todas las preguntas de todos los formularios.
  - Sub-módulo euclidiana.- determina los formularios válidos basándose en la distancia de euclidiana, y cuyo centro en la media de todas las preguntas de todos los formularios.

# CAPÍTULO 5

## 5 PATRONES Y RESULTADOS

En este capítulo se expone los resultados que se obtienen durante el proceso de agrupamiento, optimización y clasificación, describiendo los resultados presentados en el mapa de neuronas, además de comparaciones que se realizaron con otros métodos de clasificación.

### 5.1 Resultados que se obtienen en el sistema.

Podemos presentar diferentes reportes de los cuales se puede llegar a una conclusión clara sobre los resultados. Para la cual se tomaron en cuenta diferentes puntos como son:

#### 1. Sobre la determinación de la dimensión del mapa.

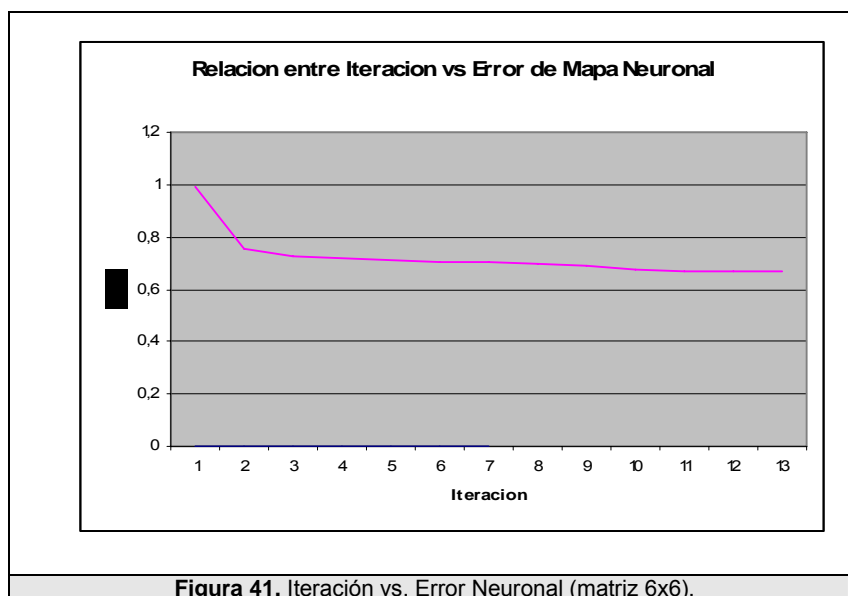
Acorde a la teoría sobre redes neuronales, este factor suele medirse acorde a las pruebas realizadas en las redes neuronales creadas, es así que luego de varias pruebas se ha encontrado lo siguiente:

Tomando como muestra los formularios que se han evaluado en encuesta 2005-1T (Primer Trimestre del 2005) en diferentes dimensiones del mapa de Kohonen, se obtienen las siguientes gráficas:

### Matriz de error del mapa neuronal, dimensión 6x6

Iteración	Error
1	0.008971652060395917
2	0.007325224175022246
3	0.006795419897304975
4	0.006781701904741523
5	0.006706696499765775
6	0.0067603309183211
7	0.006692360673557969

**Tabla 5.** Resultados de los errores en las Iteraciones en una matriz, cuyo mapa tiene dimensión de 6 x 6



\*Tiempo de cálculo: 3 días

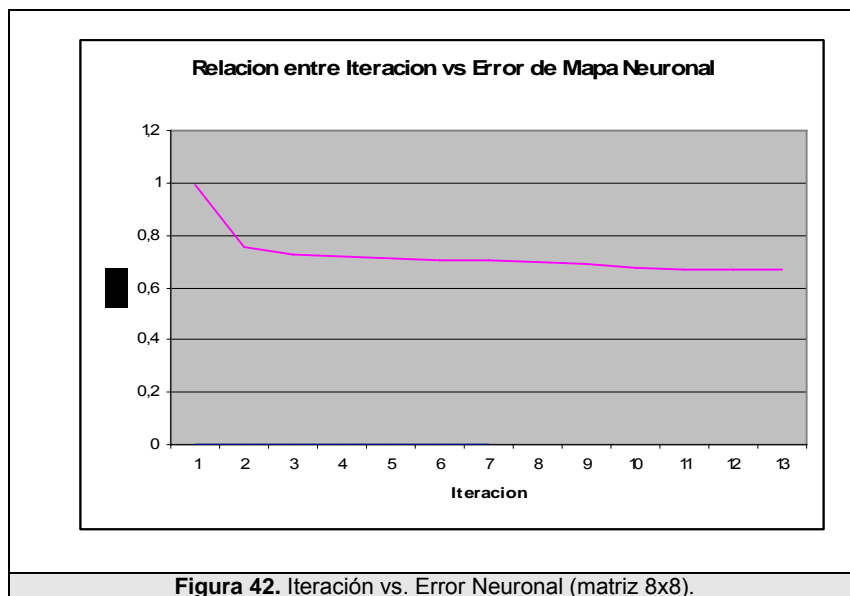
### Matriz de error del mapa neuronal, dimensión 8x8

Iteración	Error
1	0,00991557276
2	0,00751380785
3	0,00725214367
4	0,00715826333
5	0,00710500033

6	0,00707062489
7	0,00704738193
8	0,00697155962
9	0,00688043157
10	0,00676458239
11	0,00670095627
12	0,00667402249
13	0,0066565136

**Tabla 6.** Resultados de los errores en las iteraciones en una matriz, cuyo mapa tiene dimensión de 8 x 8

\*Tiempo de cálculo: 5 días.



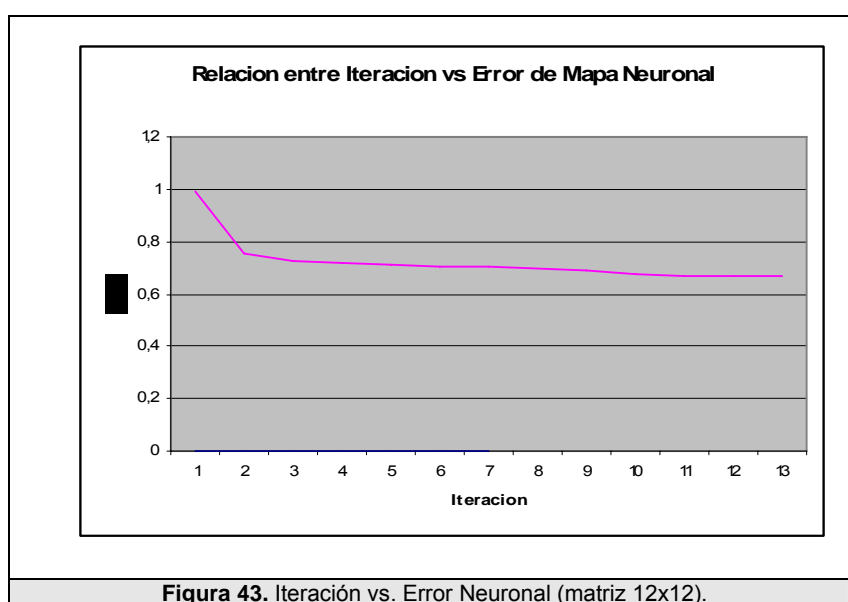
### Matriz de error del mapa neuronal, dimensión 12x12

Iteración	Error
1	0.008811017323342771
2	0.0075938881093618
3	0.007349796781147685
4	0.006938495948514117

5	0.006315413972172933
6	0.005725815198831117
7	0.005684254051967886
8	0.005627839266179439
9	0.005601572194800154

**Tabla 7.** Resultados de los errores en las Iteraciones en una matriz, cuyo mapa tiene dimensión de 12 x 12

\*Tiempo de cálculo: 6 días



Tomando en cuenta los valores que se presentan en las tablas 6, 7 y 8, podemos concluir que para un mapa de mayor dimensión su error puede converger con mayor rapidez sin embargo, acorde a la teoría de las redes neuronales, se recomienda que no se realice un mapa mayor a 12x12, ya que puede causar la no clasificación correcta de los patrones, pues existirían demasiados patrones que no agrupan a más de 1 formulario. Por otro lado, tampoco se recomienda un mapa con dimensiones muy pequeño, es decir

menor a 4x4, ya que puede causar el conglomerado de patrones en una sola neurona. Tomando en cuenta que en un mapa de dimensión de 4x4 equivale a 16 neuronas y en un mapa de dimensión de 12x12 equivale a 144 neuronas.

Dado estas diferentes pruebas realizadas podemos realizar la recomendación de utilizar un mapa de 8x8, recordando que si se desea un resultado más rápido, se podría recomendar la creación de un mapa de 6x6.

## **2. Determinación del tamaño de la muestra a elegir para entrenamiento.**

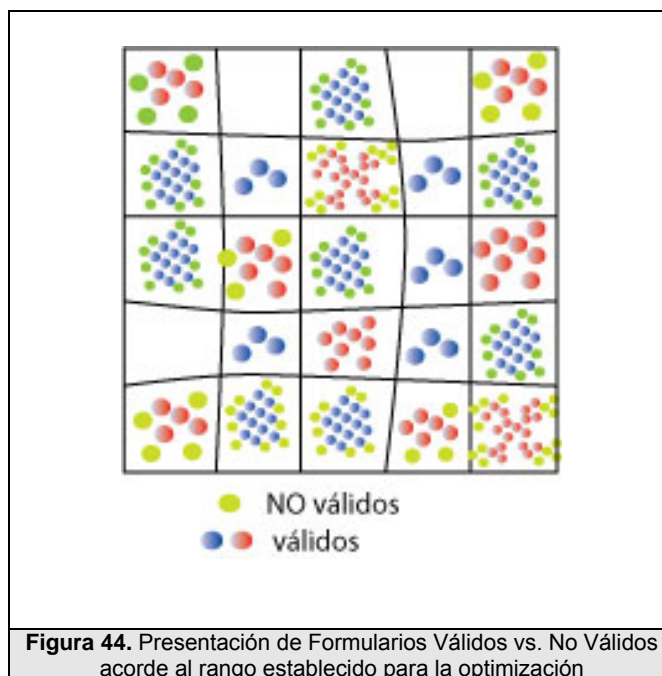
En este punto podemos mencionar que una muestra de un 75% de la población total sería adecuada, sin embargo para trabajos de gran escala, podríamos utilizar un rango entre 500 a 10.000 patrones para entrenamiento, esta recomendación es basada en los diferentes textos que relatan el mismo.

## **3. Determinación de los formularios válidos y no válidos usando el método neuronal.**

Luego de una exhausta investigación acerca del uso de redes neuronales para la clasificación de formularios, y más aún para nuestro caso en donde no se cuenta con el conocimiento de la salida de los resultados, los cuales

son los patrones (formularios) válidos y no válidos, se ha aplicado el método de redes neuronales utilizando el algoritmo de Kohonen.

Para lo cual este algoritmo nos entrega una perfecta clusterización de los resultados de una manera precisa, sin necesidad de conocer cuáles son los patrones que son válidos o no válidos. Sin embargo para cumplir el objetivo principal de discriminar los formularios válidos y no válidos, se ha aplicado una optimización de los formularios que han sido previamente clusterizados.



Usando como referencia teoría estadística, en la cual menciona que un discriminador no debería discriminar más del 10% de los formularios evaluados, se ha tomado como radio para la optimización de los datos un



90% de los formularios más cercanos al centroide que se forma en cada una de las neuronas que conforman el mapa de kohonen, según se muestra en la figura 44; Sin embargo esta medida puede ser ajustada por el experto estadístico conocedor de los comportamientos de las encuestas aplicadas a la evaluación docente.

Una vez identificados los valores que se encuentran fuera de este radio, se procederá a etiquetar como formularios no válidos, y formularios válidos a los que se encuentren dentro de este radio.

#### **4. Determinación de los formularios válidos y no válidos usando otros algoritmos.**

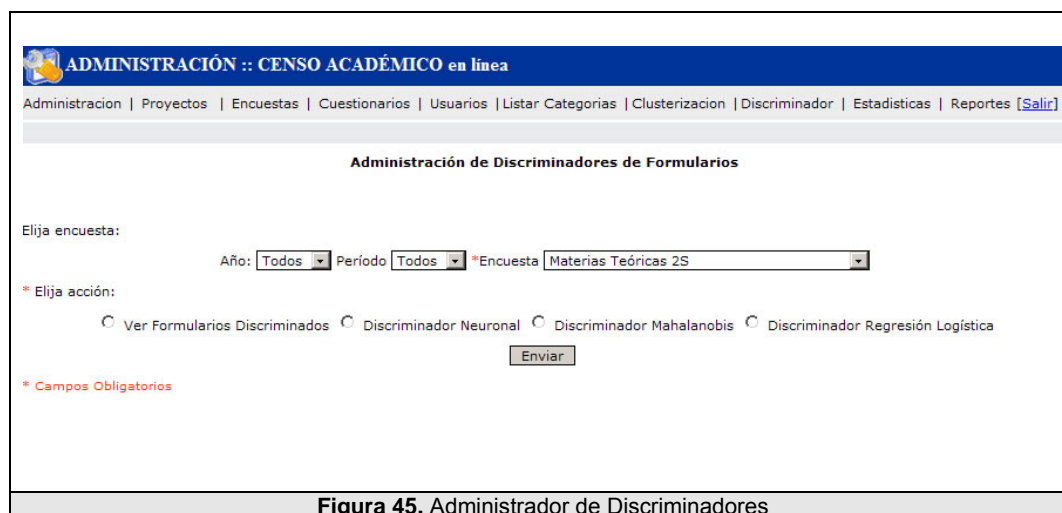
Como medida de minimización de error en la determinación de los formularios válidos y no válidos, se han aplicado otros dos algoritmos para la discriminación, los cuales utilizan la distancia euclidiana y la distancia de mahalanobis.

Estas dos técnicas estadísticas son muy utilizadas actualmente ya que permite un resultado más rápido, sin embargo su debilidad se encuentra en la exactitud de discriminación de los datos.

## 5.2 Reportes de entrega del sistema.

Este sistema entrega todos sus reportes bajo Web, siendo los reportes principales:

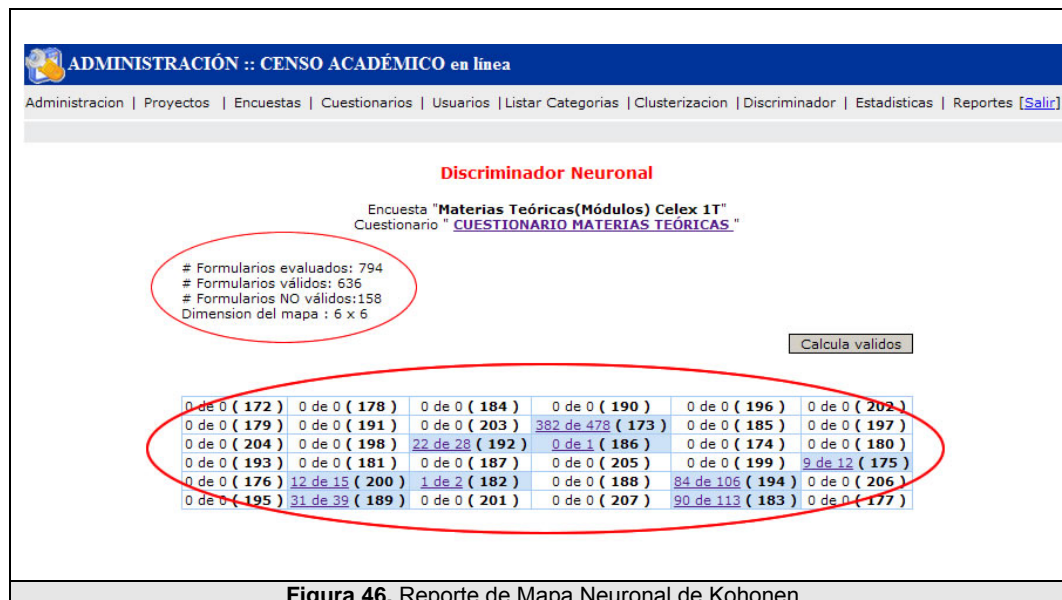
**Administración de Discriminadores.-** en donde el usuario podrá elegir la encuesta que desee con la finalidad de poder realizar las siguientes actividades: a) Ver formularios Discriminados, b) Ver Discriminador neurona, c) Ver discriminador Mahalanobis d) Ver discriminador con regresión logística. Ver Fig. 45



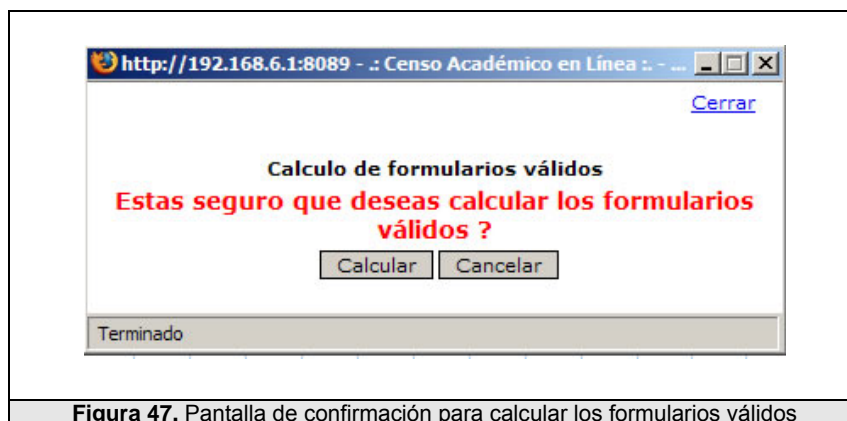
The screenshot shows a web application interface for 'ADMINISTRACIÓN :: CENSO ACADÉMICO en línea'. The navigation menu includes: Administración | Proyectos | Encuestas | Cuestionarios | Usuarios | Listar Categorías | Clusterización | Discriminador | Estadísticas | Reportes [Salir]. The main heading is 'Administración de Discriminadores de Formularios'. Below this, there are several form fields: 'Elija encuesta:' with a dropdown menu set to 'Materias Teóricas 2S'; 'Año:' with a dropdown set to 'Todos'; 'Período:' with a dropdown set to 'Todos'; and '\* Encuesta:' with a dropdown set to 'Materias Teóricas 2S'. Underneath, there is a section for '\* Elija acción:' with four radio button options: 'Ver Formularios Discriminados', 'Discriminador Neuronal', 'Discriminador Mahalanobis', and 'Discriminador Regresión Logística'. An 'Enviar' button is located below these options. A red note at the bottom left states '\* Campos Obligatorios'.

**Figura 45.** Administrador de Discriminadores

Una vez elegido la encuesta a la cual ha sido previamente aplicado el algoritmo de Kohonen en el módulo de java, podemos entregar el reporte que contiene el resumen del mapa neuronal.



Sin embargo para efectos de conocer cuáles son los formularios válidos y no válidos, procedemos a presionar en el botón “Calcula válidos”, para la cual nos aparecerá una pantalla para confirmar el cálculo a realizar.



Una vez realizado el cálculo podremos apreciar los resultados de los formularios válidos y no válidos, por cada una de las neuronas. Recordando que la optimización de los formularios se la realiza a todos los formularios que se encuentran más cercanos a su neurona respectiva.

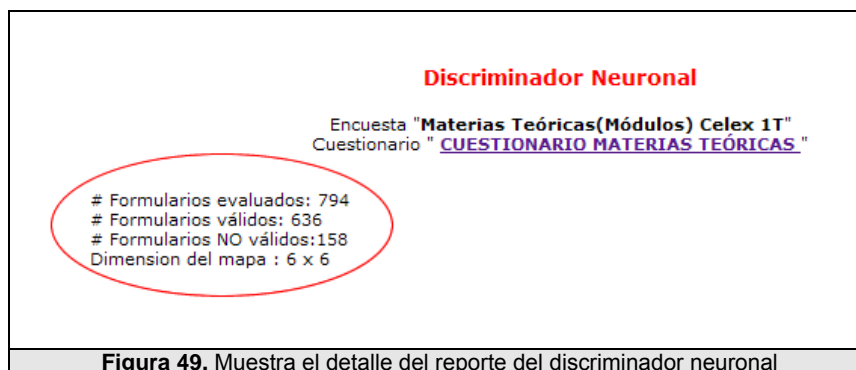
**Formularios de la Neurona # 192**

**Encuesta " Materias Teóricas(Módulos) Celex IT "**  
**Cuestionario " CUESTIONARIO MATERIAS TEÓRICAS "**

id_patron	distancia	Preg #1	Preg #2	Preg #3	Preg #4	Preg #5	Preg #6	Preg #7	Preg #8	Preg #9	Preg #10	Preg #11	Preg #12	Preg #13	Preg #14	Preg #15	Preg #16	Preg #17	Preg #18	Preg #19	Preg #20	Preg #21	Preg #22	Preg #23	Preg #24	Preg #25	Preg #26	Preg #27	Preg #28	Preg #29	Preg #30	Preg #31	Preg #32	Preg #33	Preg #34	
40991	0.2033	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3		
49099	0.6026	3	3	3	3	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3		
49006	0.9080	3	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	4	3	3	3	5	4	4	4	3	4	5	4			
48856	0.9895	3	3	3	3	3	4	3	3	3	3	3	3	4	3	3	4	3	3	3	3	3	3	3	2	4	4	4	5	3	3	4	4	3		
48994	0.9915	3	3	3	3	3	4	4	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	4	4	4	4	3	3	2	4	3	5	2		
49280	1.0005	2	3	3	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	3	4	3	3	3	4	3	5	3		
49507	1.0586	3	2	3	3	3	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	3	4	3	4	4	3	3		
49390	1.1250	2	3	2	3	2	3	3	3	3	3	3	2	3	2	2	2	2	2	2	2	2	3	3	3	2	2	2	3	2	2	2	3	3	2	
49355	1.1748	3	2	3	3	3	4	3	4	3	4	2	3	3	3	3	3	3	3	3	4	3	4	4	3	3	4	4	3	3	4	2	5	5	4	
49160	1.1798	3	2	2	2	2	3	3	3	3	4	2	2	2	3	2	3	2	2	3	3	4	2	3	4	4	3	3	3	3	3	2	3	4	2	
49167	1.1896	3	2	2	3	2	3	3	3	3	4	3	3	3	3	3	2	3	4	3	4	3	4	3	3	3	4	2	2	2	2	1	3	4	3	
49270	1.2606	3	3	3	3	3	3	4	4	1	5	3	3	3	3	3	3	3	3	2	2	2	3	2	2	3	2	2	3	3	4	3	4	3	3	
48921	1.3687	3	3	2	3	2	5	4	3	2	4	1	2	2	2	3	3	4	4	3	4	3	3	3	2	2	3	2	2	2	2	2	3	4	4	
49562	1.4236	3	2	3	2	2	4	3	3	2	3	2	3	2	2	2	2	2	3	2	2	3	3	3	2	2	4	3	3	3	1	3	5	5	2	
49679	1.4771	3	3	3	2	2	4	2	3	3	2	2	2	3	4	3	3	3	3	3	3	3	3	3	2	3	1	3	3	3	5	3	4	4	3	
49318	1.4992	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	2	2	2	2	2	4	1	4	3	
49015	1.5382	3	3	3	2	3	4	4	4	2	3	3	2	3	3	3	2	2	2	3	2	3	5	5	4	4	3	3	4	2	5	5	2	5	2	
49218	1.7089	3	3	3	2	2	3	5	5	5	4	3	3	3	2	3	3	3	3	2	3	4	4	4	4	3	3	3	3	2	3	4	5	3	5	
49261	1.7119	3	3	3	3	3	5	3	3	1	4	1	3	4	3	3	3	3	4	2	1	3	3	3	4	3	4	4	3	4	2	1	5	5	2	
49070	1.9720	3	3	3	3	3	4	3	2	2	4	1	2	4	3	4	3	4	4	4	3	4	2	5	5	4	4	4	4	4	3	4	4	3	3	
49023	2.2034	3	3	3	2	2	2	2	1	3	2	2	2	2	2	2	2	2	2	2	3	3	2	4	3	5	3	1	5	4	1	4	5	2	2	
49530	2.3150	3	3	3	3	4	4	4	3	2	3	2	1	2	2	4	1	2	5	1	1	4	3	4	2	2	3	4	4	4	4	4	4	4	5	2
49262	2.3157	3	2	3	3	2	5	4	2	4	3	2	4	3	4	2	1	3	2	4	1	3	2	3	2	3	3	5	3	2	3	3	3	2	3	
49352	2.3356	2	3	3	2	3	3	3	3	3	3	3	1	2	2	3	3	3	3	3	4	2	4	4	4	5	5	5	5	4	4	2	4	3	3	
48852	2.4347	2	3	3	3	2	2	2	3	2	3	3	3	3	4	2	3	3	4	3	4	1	3	1	1	3	3	2	2	1	1	3	5	2	2	
49267	2.0791	2	3	3	2	3	3	2	2	2	4	2	2	2	3	2	2	3	2	3	2	4	4	4	1	1	1	2	1	4	1	1	3	3	3	
49410	3.7077	3	3	3	3	2	1	1	1	1	4	4	4	3	4	3	4	3	4	3	1	1	2	3	2	2	2	2	4	3	2	5	5	5	5	
49751	5.9835	3	3	3	3	3	5	5	5	5	5	1	1	1	1	1	1	1	1	1	1	1	1	5	1	5	5	5	1	1	1	1	1	1	5	1

Figura 48. Reporte de Resultados Clasificados

Logrando el resumen de los formularios evaluados, válidos y no válidos.



Por otro lado como mencionamos anteriormente, estos resultados podría tener un sesgo mayor ya que este último paso es bajo la consideración de un experto estadístico. Es así que se incluye el reporte de varios métodos de discriminación para realizar la comparación y la obtención de los formularios válidos y no válidos.

**Lista de Formularios Válidos / NO Válidos**

Encuesta "**Materias Teóricas(Módulos) Celex 1T**"  
Cuestionario "**CUESTIONARIO MATERIAS TEÓRICAS**"

# Total Formularios generados: 1042  
# Total Formularios evaluados: 794  
# Total Formularios válidos: 794

Neuronal = 636 válidos de 794  
Regresion= 0 válidos de 794  
Mahalanobis = 338 válidos de 794  
Euclidiana = 771 válidos de 794

ID Formulario	Neuronal	Regresión	Mahalanobis	Euclidiana
49895	sí	--	sí	sí
49894	sí	--	sí	sí
49893	sí	--	XXX	sí
49892	XXX	--	XXX	sí
49891	sí	--	XXX	sí
49890	sí	--	XXX	sí
49889	sí	--	XXX	sí
49888	XXX	--	sí	sí
49887	sí	--	XXX	sí
49886	sí	--	sí	XXX

**Figura 50.** Listado de Formularios Válidos vs. No Válidos

### 5.3 Mapa de la red neuronal.

Una vez determinado la dimensión del mapa, podemos mencionar que mapa de la red neuronal es el equivalente a la colección de neuronas, y cada neurona contiene un vector de pesos. Este vector de pesos es generado aleatoriamente en la fase de entrenamiento y luego del mismo sus pesos son corregidos.

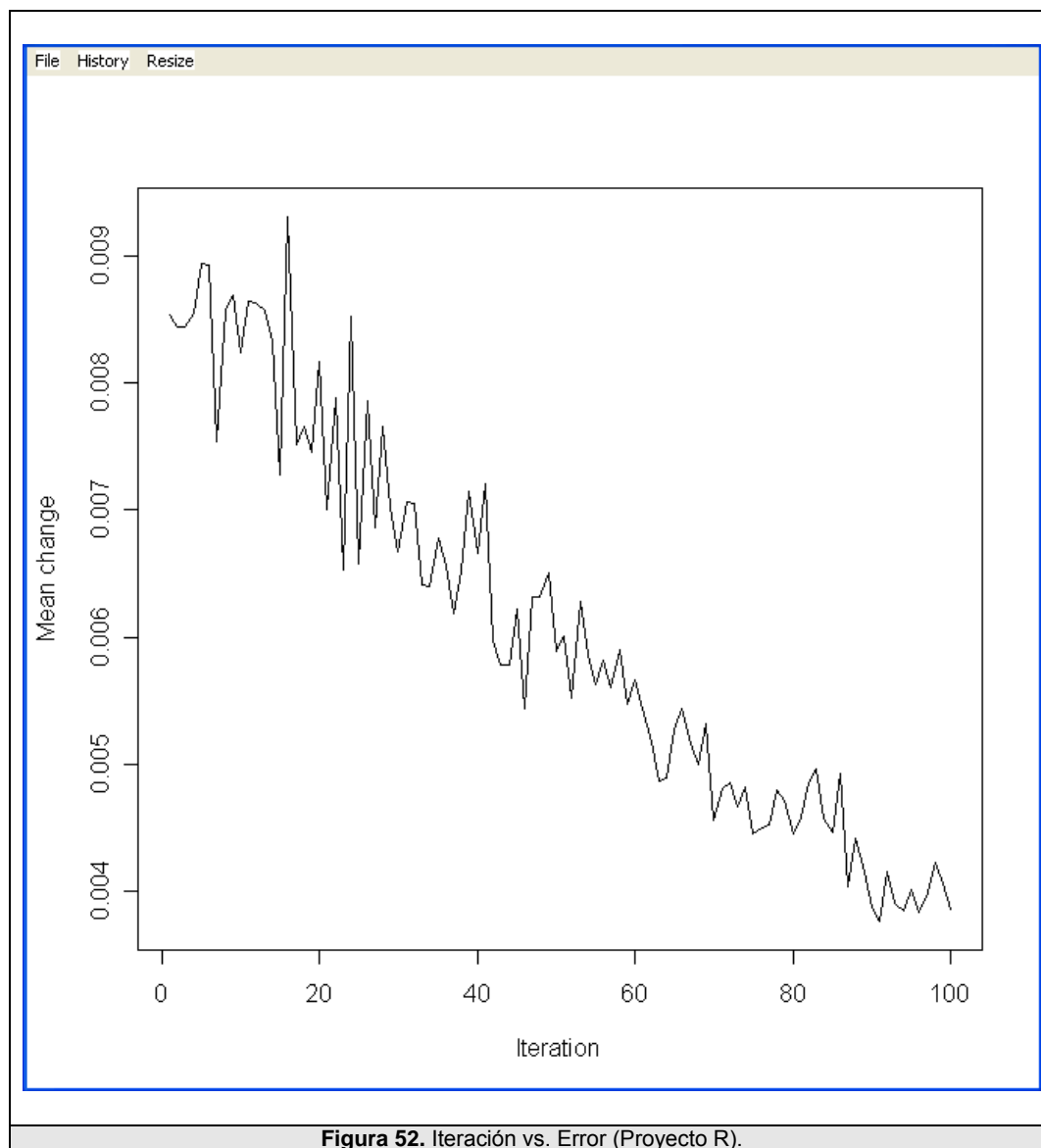
0 de 0 ( 172 )	0 de 0 ( 178 )	0 de 0 ( 184 )	0 de 0 ( 190 )	0 de 0 ( 196 )	0 de 0 ( 202 )
0 de 0 ( 179 )	0 de 0 ( 191 )	0 de 0 ( 203 )	382 de 478 ( 173 )	0 de 0 ( 185 )	0 de 0 ( 197 )
0 de 0 ( 204 )	0 de 0 ( 198 )	22 de 28 ( 192 )	0 de 1 ( 186 )	0 de 0 ( 174 )	0 de 0 ( 180 )
0 de 0 ( 193 )	0 de 0 ( 181 )	0 de 0 ( 187 )	0 de 0 ( 205 )	0 de 0 ( 199 )	9 de 12 ( 175 )
0 de 0 ( 176 )	12 de 15 ( 200 )	1 de 2 ( 182 )	0 de 0 ( 188 )	84 de 106 ( 194 )	0 de 0 ( 206 )
0 de 0 ( 195 )	31 de 39 ( 189 )	0 de 0 ( 201 )	0 de 0 ( 207 )	90 de 113 ( 183 )	0 de 0 ( 177 )

**Figura 51.** Mapa de la Red Neuronal.

### 5.4 Descripción del Mapa.

En cada sector del mapa podemos mencionar que pueden existir neuronas que no contienen formularios, o que pueden contener la mayoría de los formularios, es así que pueden aparecer etiquetas como “382 de 478 (173)”, representando a neurona con id igual a 173 y que contiene a 382 formularios válidos de los 478 formularios que han sido clusterizado en esta sección del mapa.

Una vez que le hacemos clic a las neuronas que contienen formularios, podremos apreciar todos los patrones (formularios) que contiene la misma.



## CONCLUSIONES Y RECOMENDACIONES

- Tener en cuenta que a mayor número de patrones de entrenamiento dará como resultado un mayor tiempo para el proceso.
- Algo que cabe tomar en cuenta y de mucha relevancia es el tipo de proceso que se ejecuta con Kohonen, este proceso no es predictivo, no es un análisis con probabilidades, ni proyecciones, este proceso es más real, los datos que se ingresan son valores que se encuentran dentro de la base, después del proceso de evaluación, por lo tanto son resultados que demuestran lo que se ha realizado de manera real.
- La aplicación de redes neuronales puede entregar una mayor certeza sobre la clasificación de formularios.
- Determina matemáticamente resultados que en muchas hipótesis no se han logrado determinar su certeza.
- Permite enfocar de manera más precisa los diferentes comportamientos sobre la evaluación docente.
- El período de entrenamiento de una red puede tomar varios días para finalizar el mismo.



- Obtener un discriminador para cualquier tipo de encuesta en donde todas sus preguntas utilizan escala Likert como alternativas.
- Obtener un porcentaje de error relativamente bajo, en comparación a otros métodos.
- Entrega información concreta sobre datos abstractos.
- Ayudar a determinar los formularios válidos y no válidos en los diferentes comportamientos que pueda obtener durante una encuesta.
- Permite entregar resultados concretas a teorías inciertas como por ejemplo:
  - Si un estudiante que pierde una materia, contesta el formulario de diferente manera a los demás estudiantes.

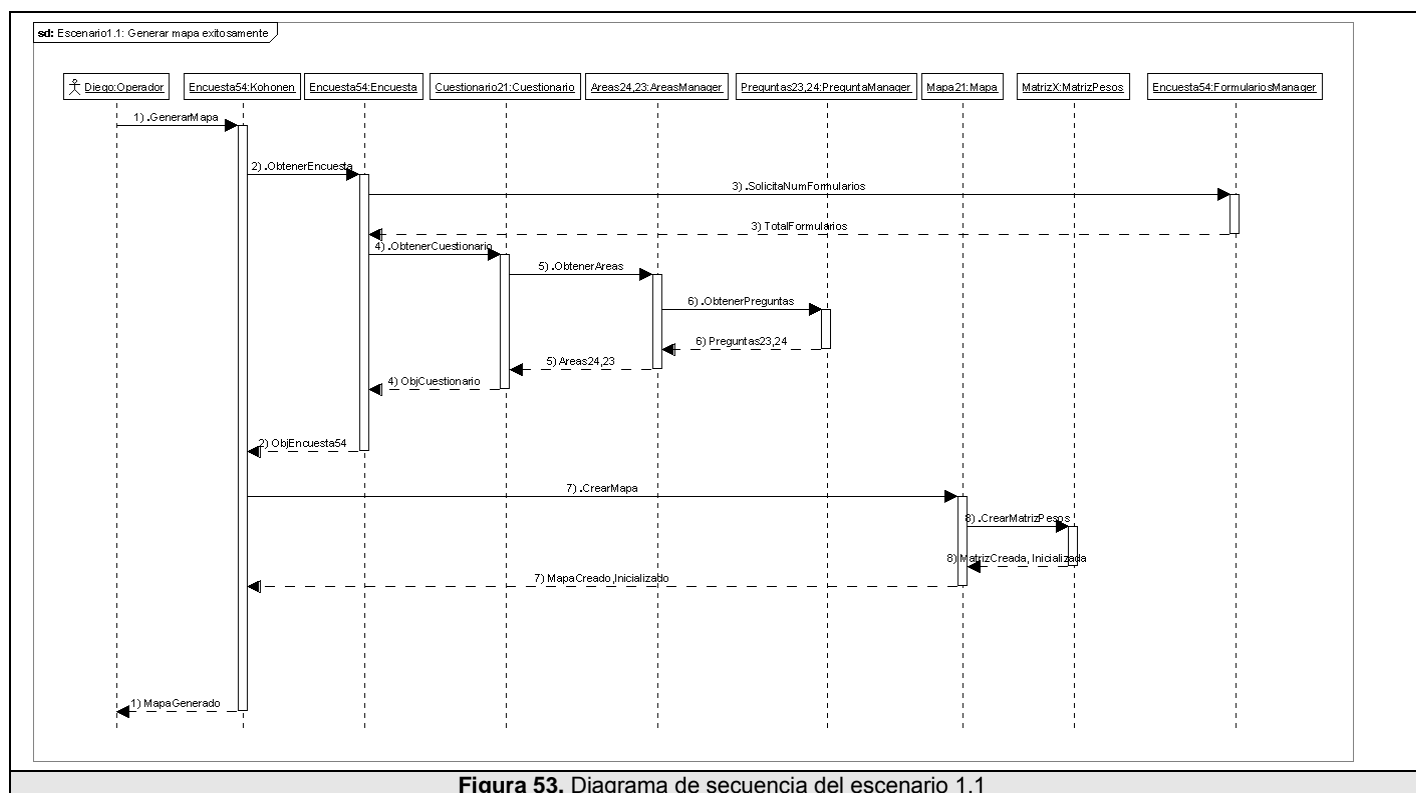
## POSIBLES INNOVACIONES

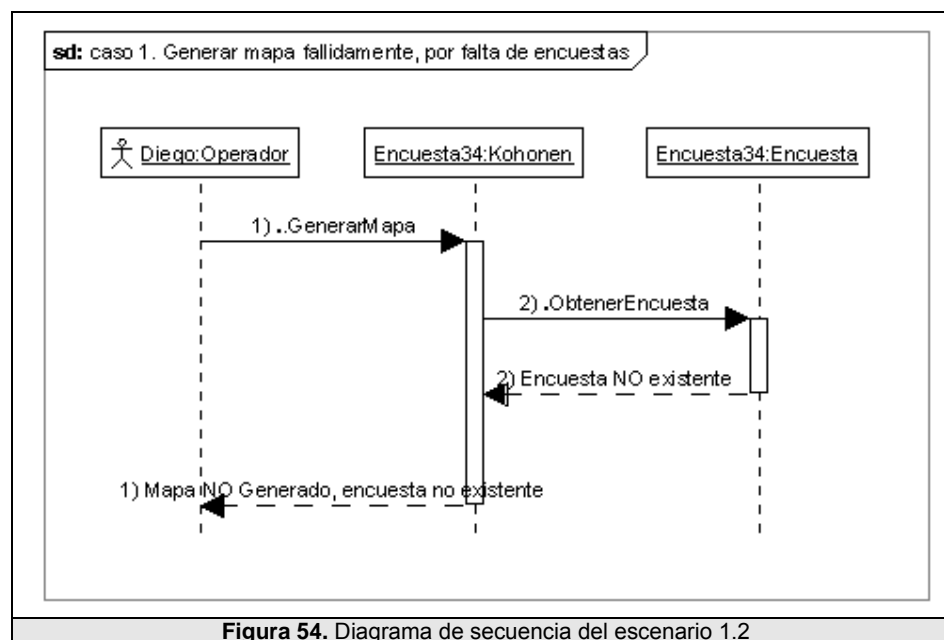
- Aplicar entrenamiento utilizando tecnologías GRID, para el funcionamiento más rápido del mismo.
- Determinar la comparación de los resultados vs entrevistas con las evaluadoras para determinar una mejor cercanía a la realidad, con respecto a la discriminación de formularios.
- Realizar otra hipótesis como la realización de la discriminación de formularios por facultad, o por paralelo. Este último punto se lo puede tomar como una mejora, además que puede optimizar la selección de formularios validos vs. No validos.

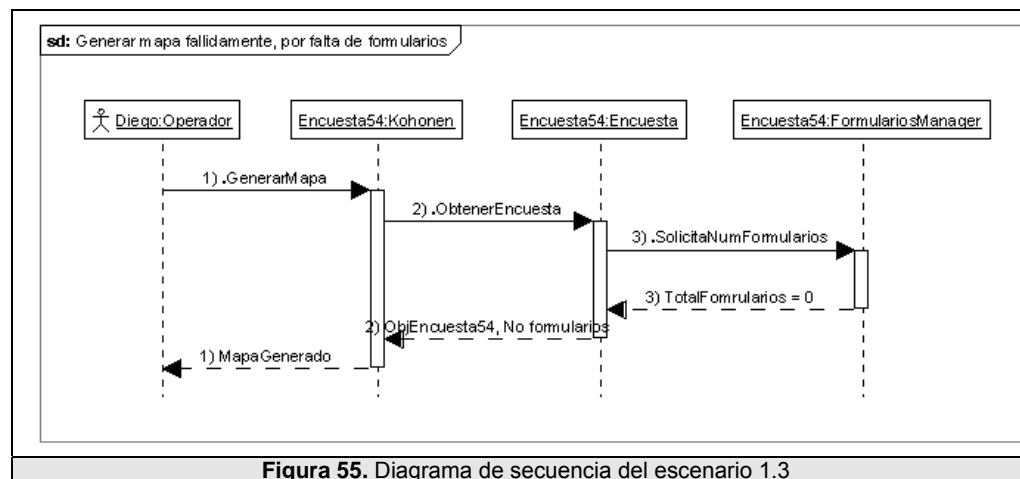
## ANEXOS

### Anexo 1: Diagramas de Interacción de Objetos del Sistema

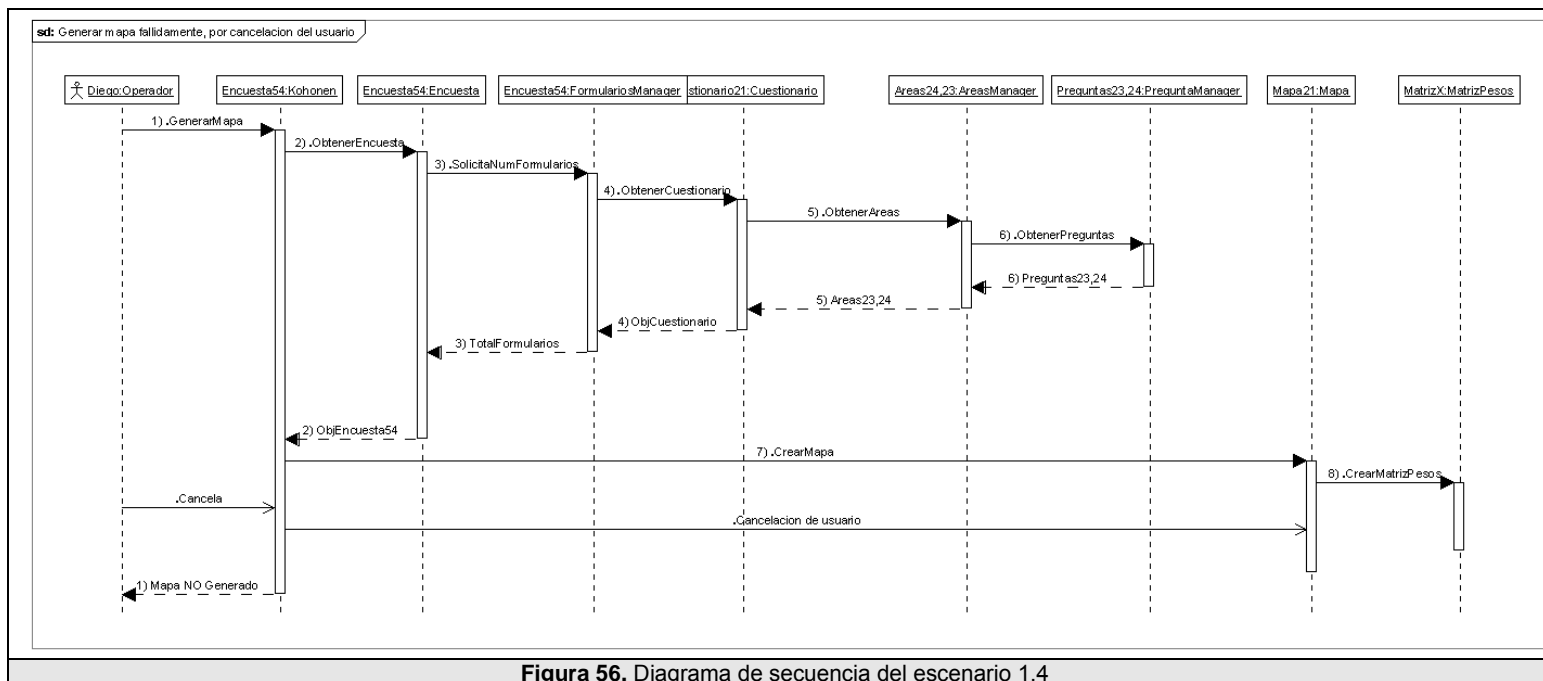
#### Escenario 1.1: Generar mapa exitosamente



**Escenario 1.2: Generar mapa fallidamente, por falta de encuestas**

**Escenario 1.3: Generar mapa fallidamente, por falta de formularios**

### Escenario 1.4: Generar mapa fallidamente, por cancelación del usuario



### Escenario 2.1: Entrenar red neuronal exitosamente

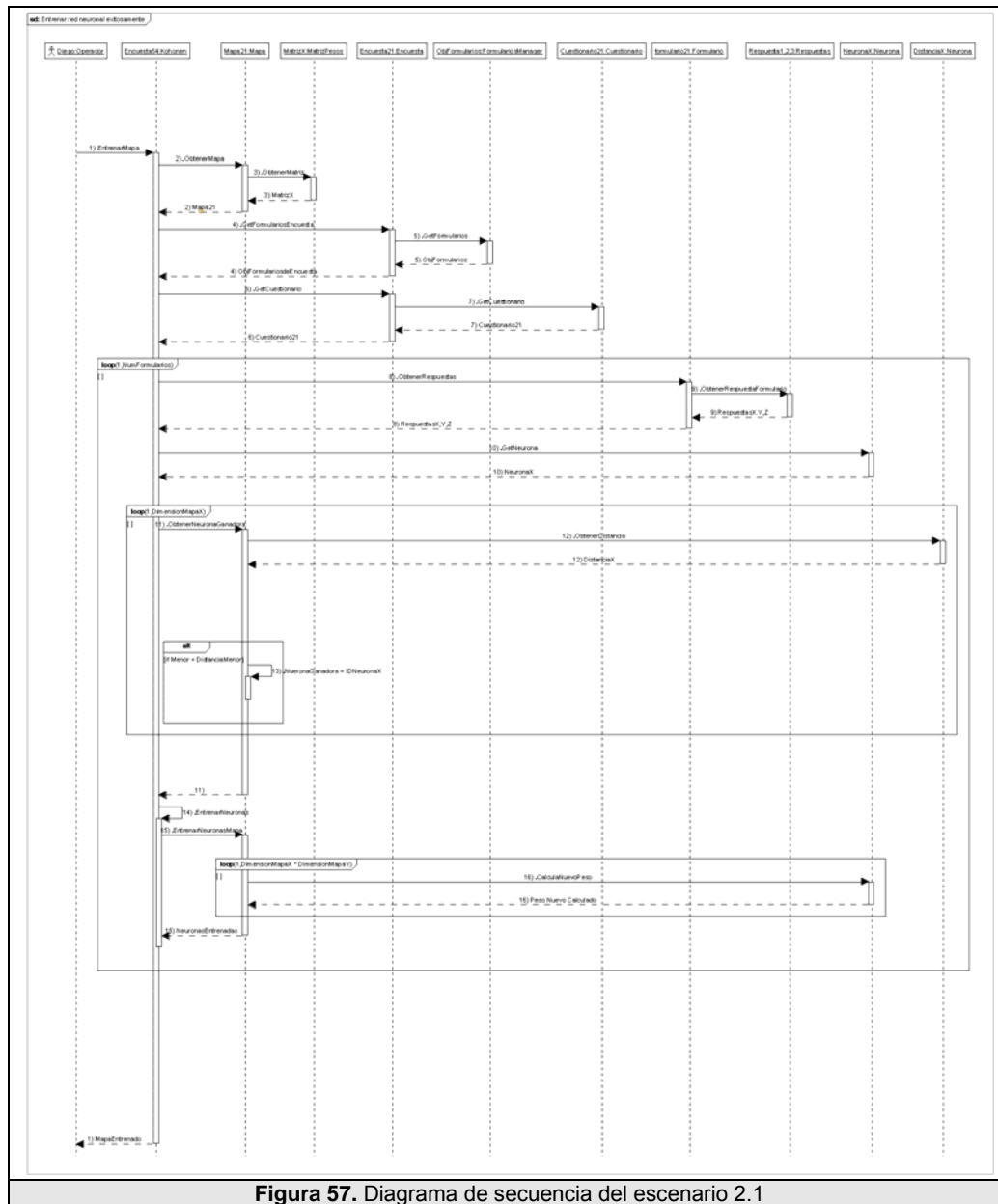
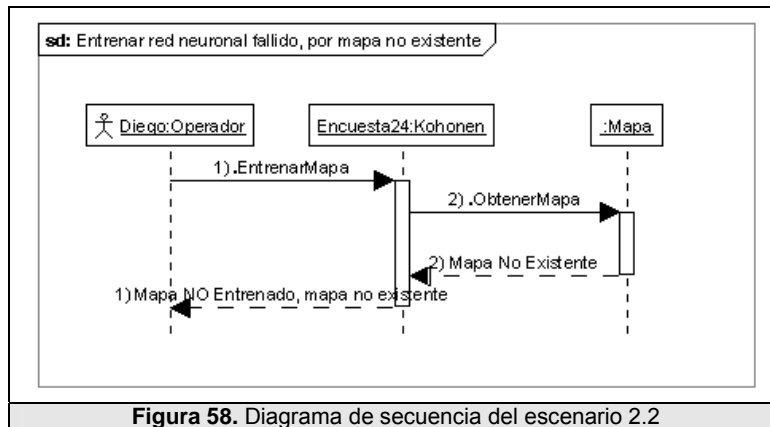
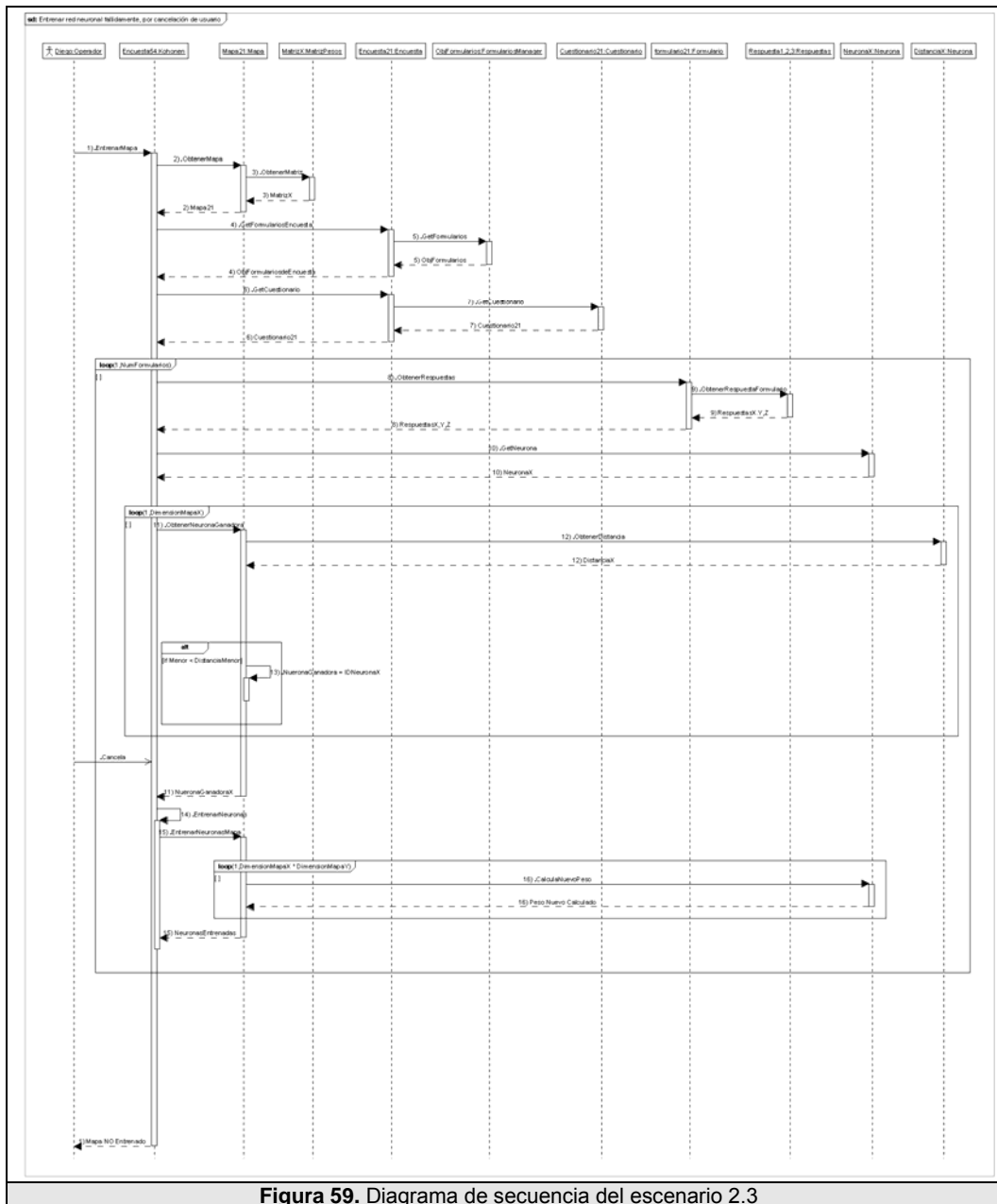


Figura 57. Diagrama de secuencia del escenario 2.1

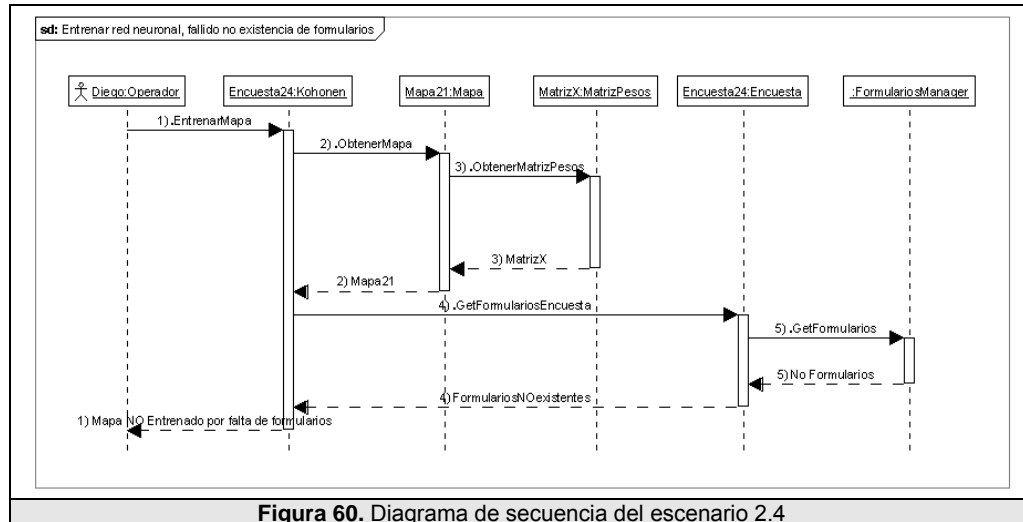
**Escenario 2.2: Entrenar red neuronal fallido por mapa no creado**



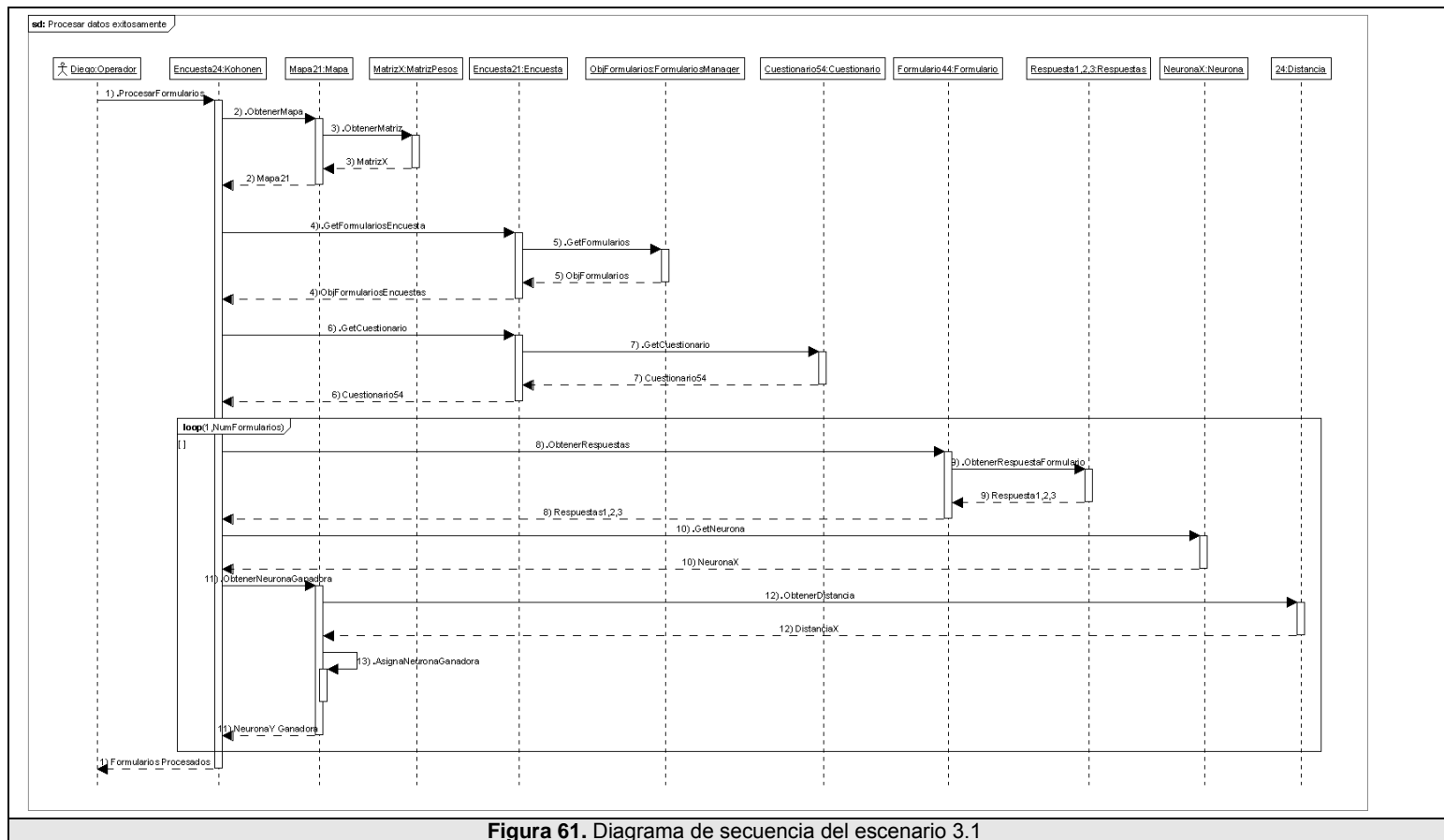
### Escenario 2.3: Entrenar red neuronal fallido por cancelación de usuario



## Escenario 2.4: Entrenar red neuronal fallido por la no existencia de formularios



### Escenario 3.1: Procesar datos exitosamente



### Escenario 3.2: Procesar datos fallidamente por cancelación de usuario

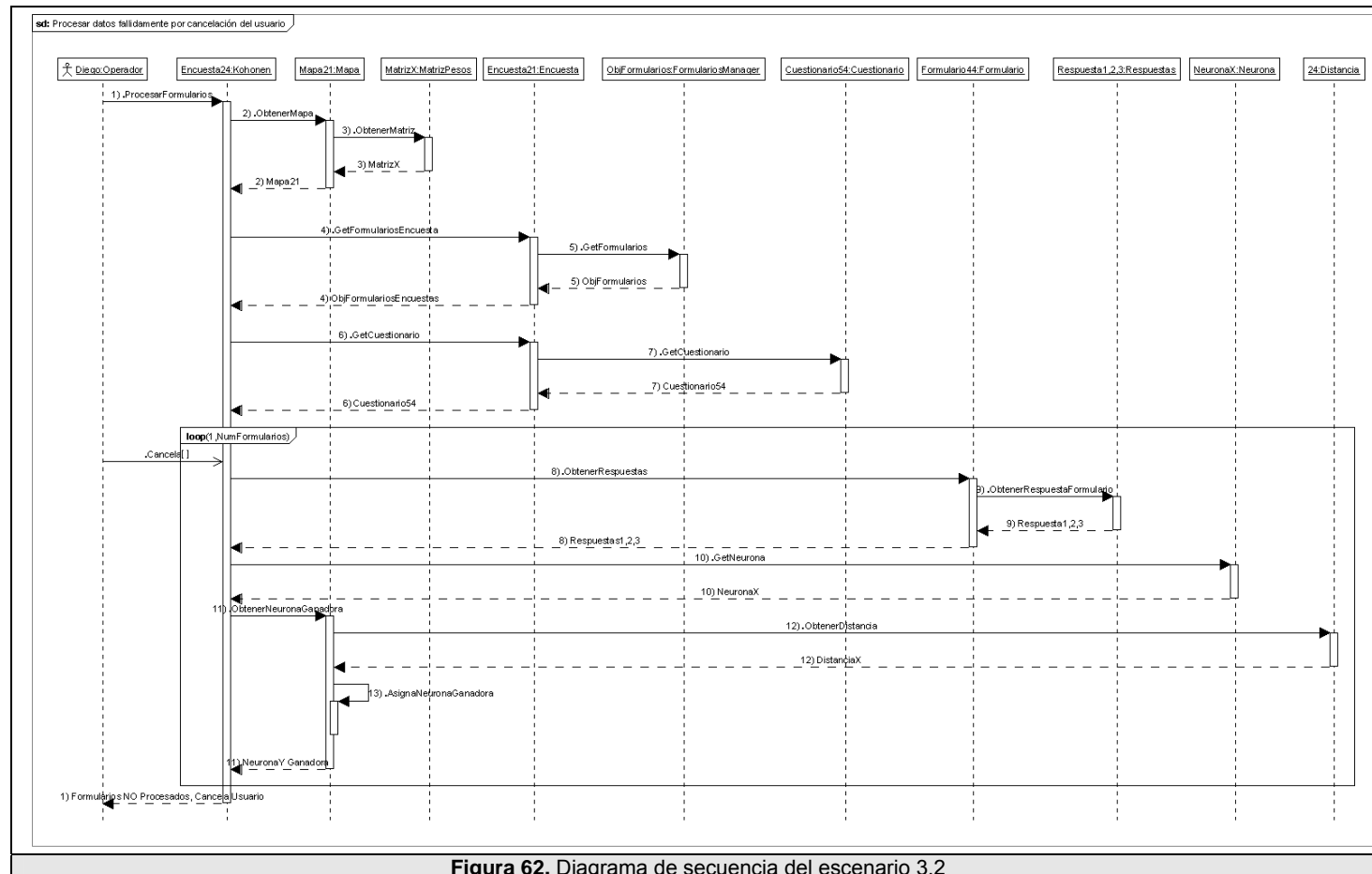
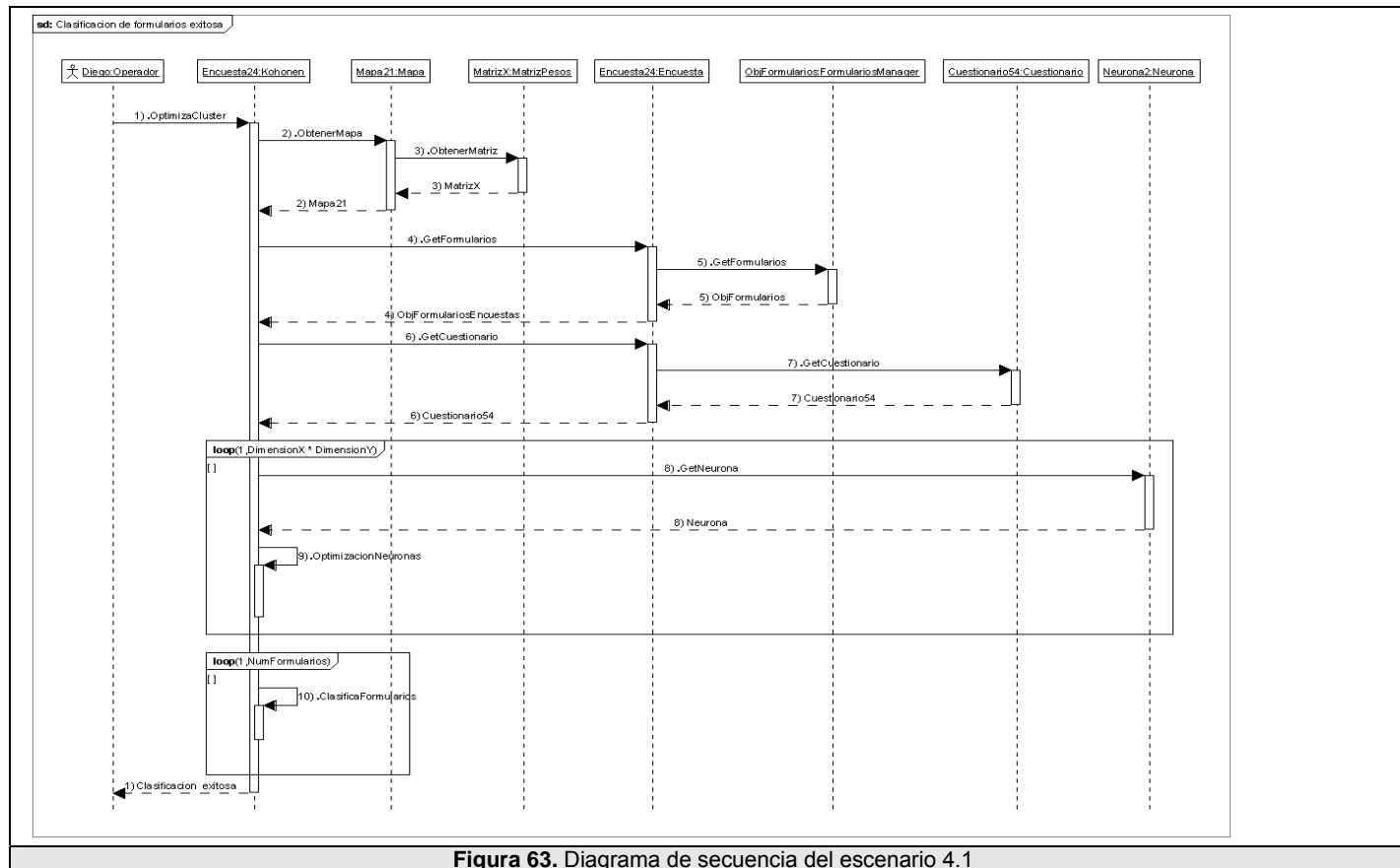
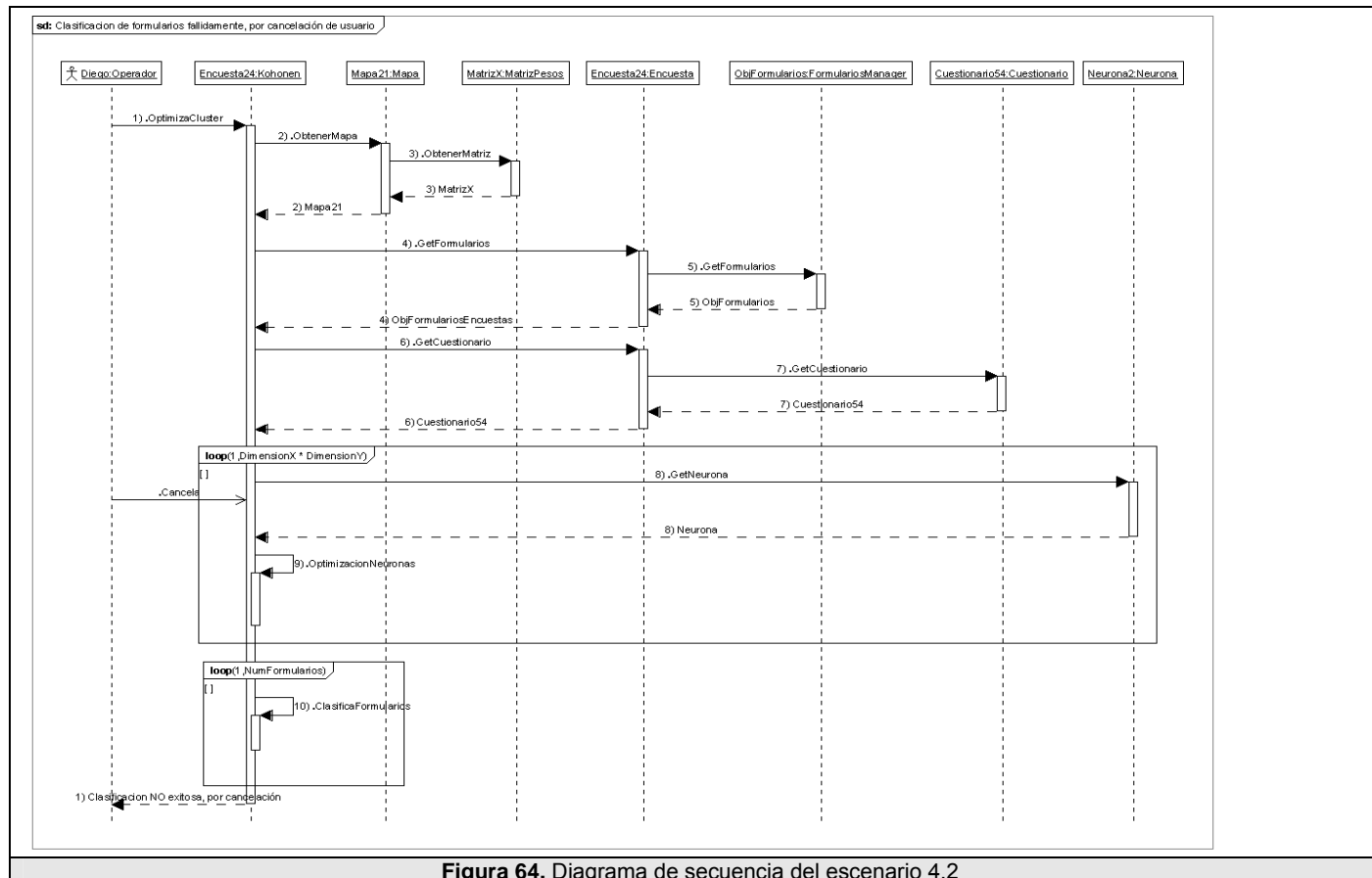


Figura 62. Diagrama de secuencia del escenario 3.2

### Escenario 4.1: Clasificación de los formularios



### Escenario 4.2: Clasificación de los formularios fallida por cancelación del usuario



\*Desarrollado utilizando UML 2.0 (20).

## Anexo 2: Diccionario de Datos del modelo de la base de datos

<b>ENCUESTA</b>		
Estructura que mantiene la definición de una encuesta.		
<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
id_encuesta	Referencia de la encuesta, clave primaria.	INTEGER
id_proyecto	Clave foránea de la tabla proyecto	INTEGER
id_cuestionario	Clave foránea de la tabla cuestionario	INTEGER
fecha_inicio	Fecha de inicio de la evaluación	DATE
fecha_fin	Fecha final de evaluación	DATE
mensaje	Mensaje para los evaluadores	TEXT
nota	Nota para los evaluadores	TEXT
anio	Año académico de la evaluación	CHAR(4)
termino	Período educativo, semestral, bimestral, trimestral	CHAR(2)
nombre_encuesta	Nombre de la encuesta	TEXT
estado	Estado de la encuesta, puede ser bloqueado o no bloqueado. Bloqueada cuando un formulario de la encuesta ya ha sido evaluado	BIT(1)
fecha_extendida	Lapso extendido para evaluar	DATE
fecha_publicacion	Fecha de publicación de resultados	DATE
propietario	Usuario administrador, creador de la encuesta	CHAR(15)
lista_acceso	Lista de usuarios que tienen acceso a esta encuesta, clave foránea de la tabla lista_acceso	INTEGER

**Tabla 8.** Tabla Encuesta

<b>FORMULARIO</b>		
Estructura que mantiene el formulario de la encuesta, determinada por el tipo de cuestionario que se selecciona para la evaluación.		
<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
id_formulario	Clave primaria, referencia del formulario.	SERIAL
id_encuesta	Clave foránea de la tabla encuesta	INTEGER
fecha_inicio_evaluacion	Fecha de inicio del período de evaluación	DATE
estado	Evaluable o no evaluado	INTEGER DEFAULT 0
fecha_fin_evaluacion	Fecha de finalización del período de evaluación	DATE,
fecha_extendida_evaluacion	Lapso extendido para evaluar	DATE,
fecha_evaluado	Fecha de realización de evaluación de este formulario	TIMESTAMP(0)
tiempo_evaluando	Tiempo de evaluación del encuestado	INTEGER
propietario	Evaluador del formulario	VARCHAR(15),
id_grupo	Grupo de evaluación al que pertenece el evaluador	INTEGER
anulado	Si el evaluador no se encuentra destinado a evaluar este formulario.	BOOLEAN DEFAULT false,
valido_neuronal	Si el módulo neuronal lo presenta como válido o no válido	BOOLEAN
mahalanobis	Si el módulo de mahalanobis lo presenta como válido o no válido	BOOLEAN
regresion	Si el módulo de regresión lo presenta como válido o no válido	BOOLEAN

**Tabla 9.** Tabla Formulario



<b>NEURONAL CLUSTER KOHONEN</b>		
Estructura del clúster de la red de Kohonen		
<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
id_patron	Clave primaria, id formulario	SERIAL
id_neurona	Identificador de la neurona del mapa de Kohonen.	INTEGER
id_mapa	Identificador del mapa de la encuesta que se emplea en Kohonen.	INTEGER
distancia_euclidiana	Distancia euclidiana que cada neurona tendra en el mapa al momento de ingresar un formulario.	INTEGER

**Tabla 10.** Tabla Neuronal Clúster Kohonen.

<b>NEURONAL MAPA</b>		
Estructura del mapa que se crea para el proceso de red de Kohonen.		
<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
id_mapa	Clave primaria, id de mapa de encuesta.	SERIAL,
id_encuesta	Clave foranea de la tabla encuesta.	INTEGER
tamano_y	Dimensión en x del mapa	INTEGER
tamano_x	Dimensión en y del mapa	INTEGER
num_neuronas	Numero de neuronas del mapa	INTEGER
clasificado	Clasificado o no clasificado	BIT(1) DEFAULT B'0':"bit"
entrenado	Entrenado o no entrenado	BIT(1) DEFAULT B'0':"bit"
num_form_total	Total de Formularios de encuesta	INTEGER
num_form_selec	Total de formularios seleccionados para el entrenamiento.	BIGINT

**Tabla 11.** Tabla Neuronal Mapa

<b>NEURONAL NEURONAS</b>		
Estructura que permite guardar una relación entre las neuronas y su mapa.		
<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
id_neurona	Identificador de neurona	SERIAL
id_mapa	Identificador de mapa	INTEGER
pos_x	Posición x en el mapa de la neurona	INTEGER
pos_y	Posición y en el mapa de la neurona	INTEGER
factor_cercania	Factor de cercanía entre neuronas.	INTEGER
distancia_neurona_ganadora	Distancia contra la neurona vencedora.	INTEGER
estado	Vencedora o no vencedora	BIGINT

**Tabla 12.** Tabla Neuronal Neuronas

<b>NEURONAL VECTOR</b>		
Estructura que permite guardar una relación entre la neurona y su vector de pesos.		
<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
id_vector	Id del vector de pesos.	SERIAL
id_neurona	Neurona asociada al vector de pesos.	INTEGER
id_pregunta	Item del vector relacionado con el id de pregunta de un cuestionario.	INTEGER
peso	Valor de peso de la neurona.	INTEGER

**Tabla 13.** Tabla Neuronal Vector

## **Anexo 3: Instalación del sistema.**

### **Introducción**

Esta guía proporcionará al usuario una breve descripción de los requerimientos necesarios para el correcto funcionamiento del sistema seguido por un instructivo con los pasos para instalar los mismos bajo Windows.

Es necesario seguir estrictamente cada uno de los pasos que se señalan a continuación con la finalidad de no cometer errores bizarros.

### **Programas a utilizar**

Entre los programas que se utilizaran para el funcionamiento de este sistema podemos mencionar:

- PostgreSQL 8.1.

Url: <http://www.postgresql.org/>

Licencia: BSD

- Java 1.5.0.2 SE

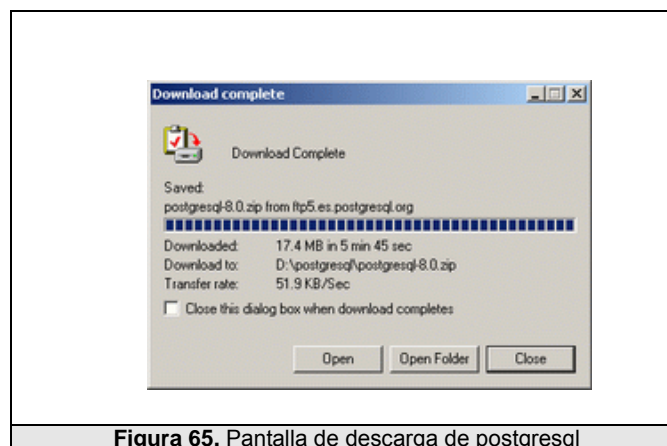
Url: <http://java.sun.com/>

Licencia: GPL

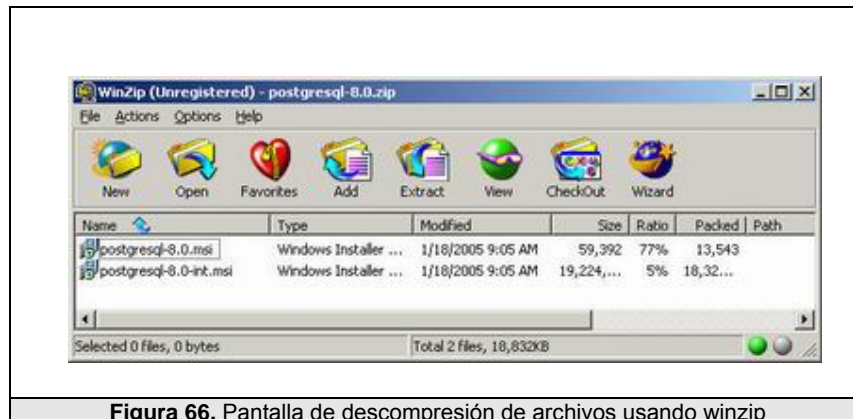
- Eclipse 3.1  
Url: <http://www.eclipse.org/>  
Licencia: GPL
- Winzip (opcional)  
Url: <http://www.winzip.com/>  
Licencia: Free evaluation
- IntelliJ IDEA 6.0 (opcional)  
Url: <http://www.jetbrains.com/idea/>  
Licencia: Comercial.
- Php.
- Apache.

## Instalación de Postgresql

1. Descargamos el paquete instalador de la página oficial de postgresql.



2. Descomprimos los ficheros con el instalador y hacemos doble clic.



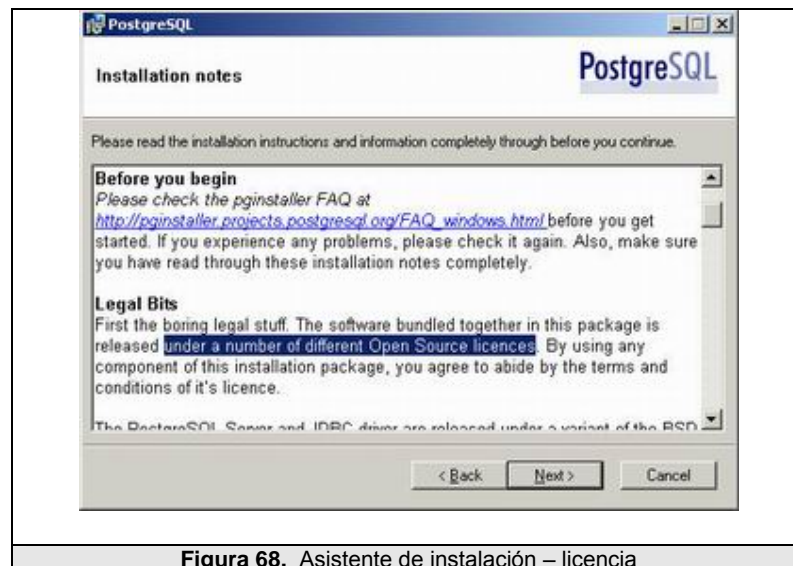
**Figura 66.** Pantalla de descompresión de archivos usando winzip

- Nos aparece el asistente y usaremos el idioma inglés.

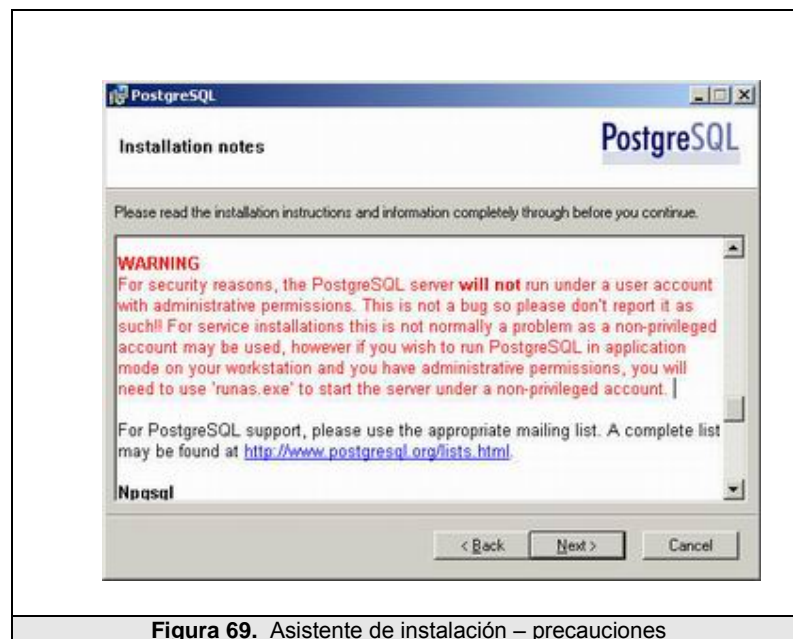


**Figura 67.** Asistente de instalación – selecciona idioma

- Leeremos las instrucciones sobre el modelo de licencias

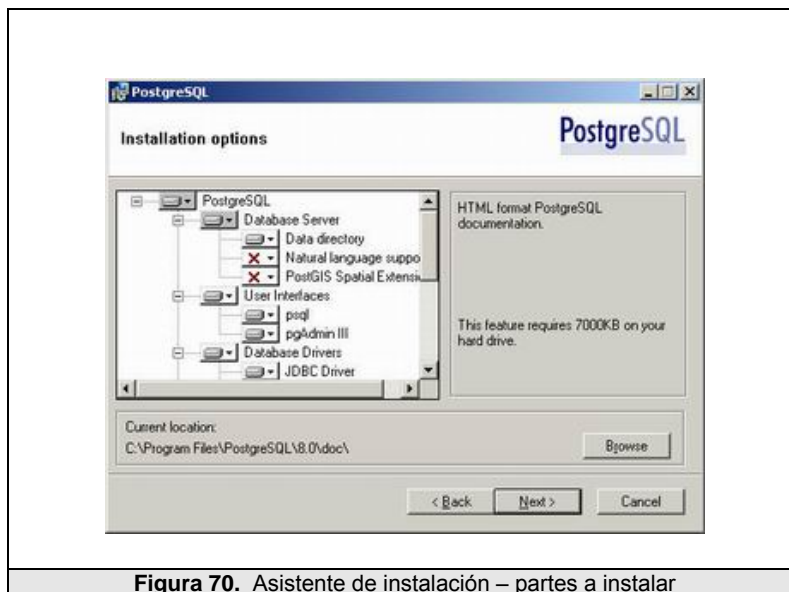


5. Y precauciones para que no corramos el servidor como administrador



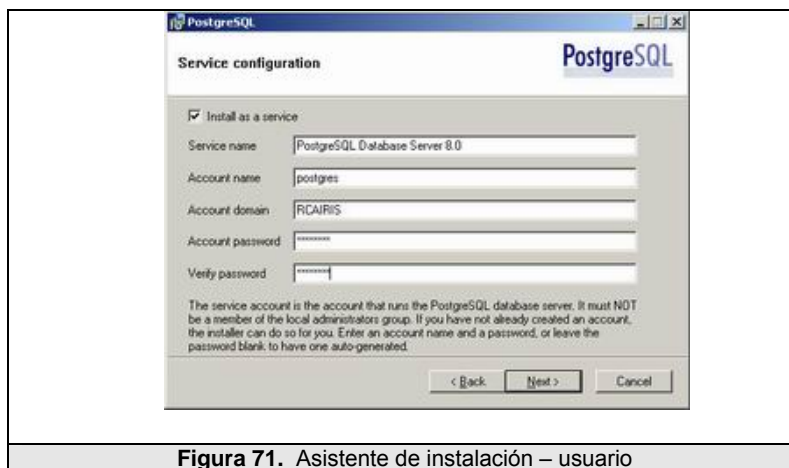
6. Elegimos las partes a instalar





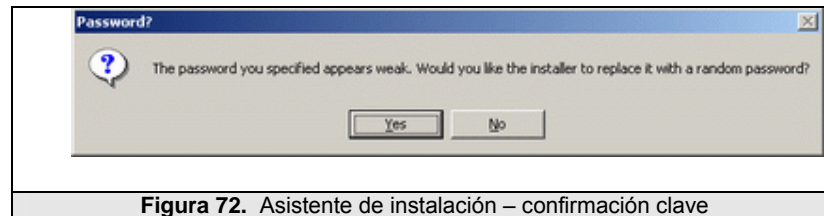
**Figura 70.** Asistente de instalación – partes a instalar

7. Elegimos instalar la base de datos como un servicio y establecemos un usuario y contraseña. Si no existe se creará. Si existe, no debe estar en el grupo de administradores.



**Figura 71.** Asistente de instalación – usuario

8. Si la contraseña es muy fácil de encontrar, nos sugiere una aleatoria. No hay problema porque es la que se pone en el administrador de servicios y no la tenemos ni que conocer



**Figura 72.** Asistente de instalación – confirmación clave

9. Definimos parámetros de configuración (sobre todo el super-usuario y contraseña).



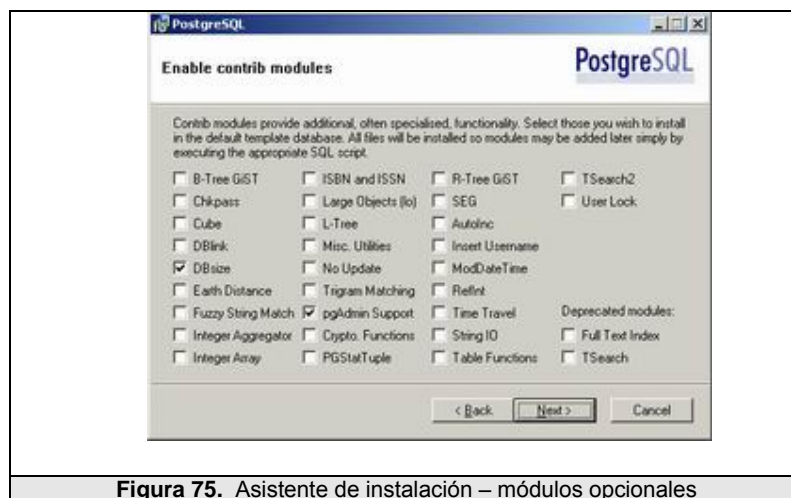
**Figura 73.** Asistente de instalación – confirmación clave

10. Elegimos el nombre usado como lenguaje de Script para los procedimientos almacenados.



**Figura 74.** Asistente de instalación – elije el lenguaje

11. Seleccionamos los módulos opcionales (se recomienda dejar la configuración por defecto).



**Figura 75.** Asistente de instalación – módulos opcionales

## Instalación de Apache

En esta guía se explicará como instalar el servidor Apache en Windows:

1. Procedemos a descargar la versión más actualizada del instalador Win32 binario para Apache2, (*apache\_2.0.55-win32-x86-no\_ssl.msi*), desde la página oficial de apache (<http://httpd.apache.org/>).

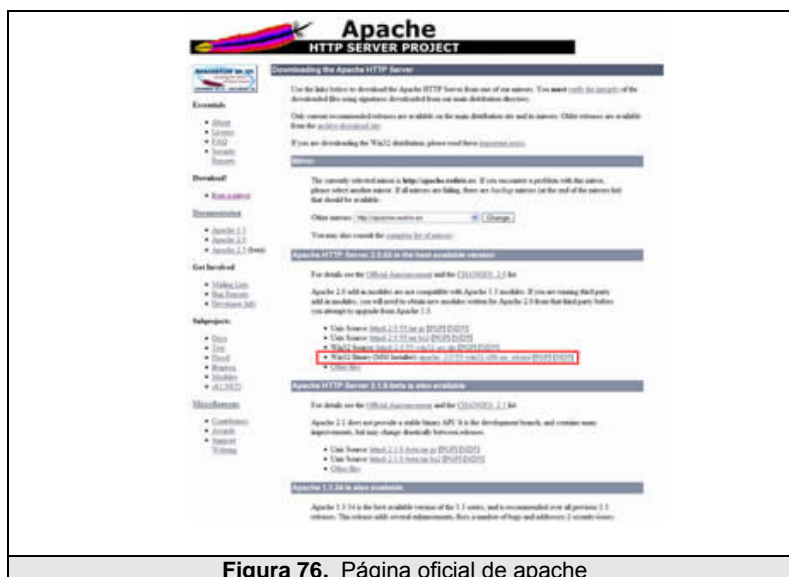


Figura 76. Página oficial de apache

2. Una vez descargado el archivo, hacemos doble clic para ejecutarlo.



Figura 77. Icono instalador de apache

- Veremos una ventana como la siguiente. Hacemos clic en **Next** para empezar a instalar Apache.

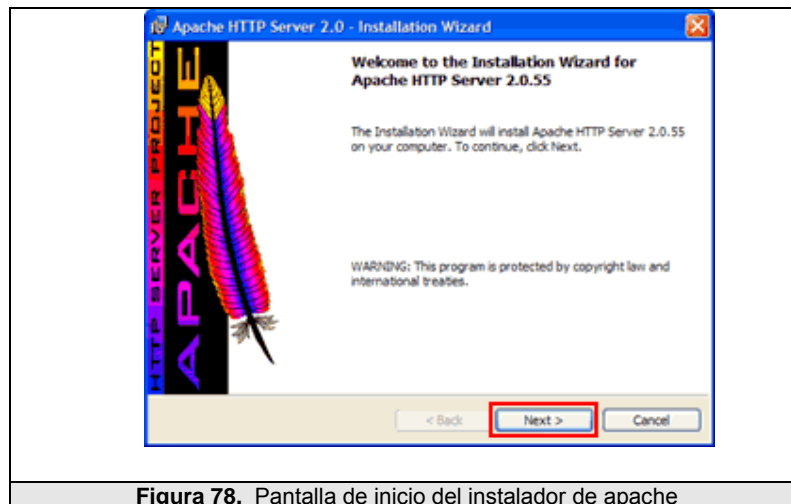


Figura 78. Pantalla de inicio del instalador de apache

- Aceptamos** los términos de la licencia y hacemos clic en el botón **Next**.

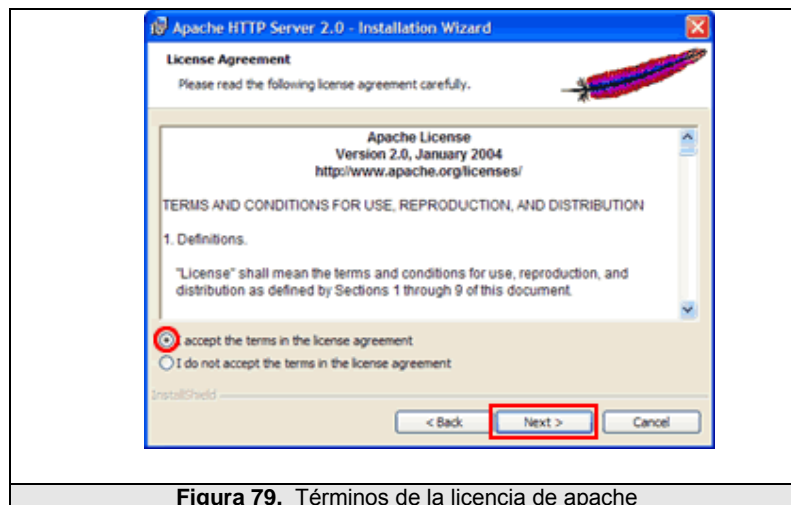
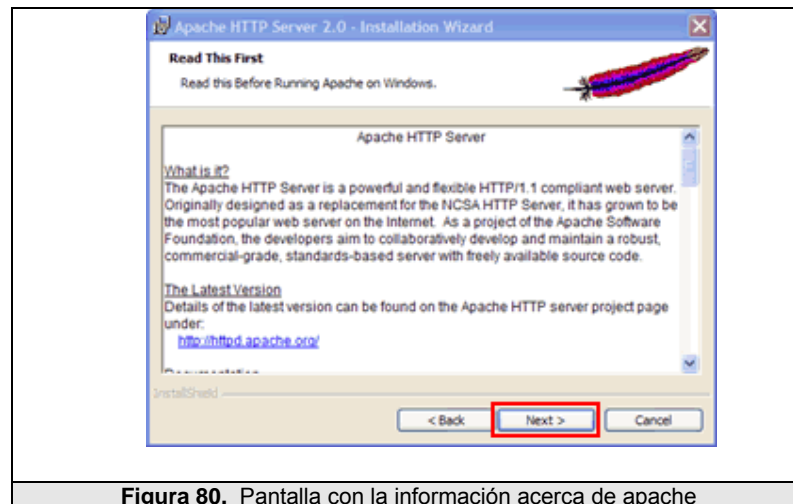


Figura 79. Términos de la licencia de apache

5. Aparecerá una ventana con información acerca de Apache, y hacemos clic sobre el botón **Next**



6. En la siguiente pantalla el instalador de Apache nos pide la información para el servidor. Como sólo queremos ejecutar Apache en nuestro ordenador (Apache también se puede hacer accesible desde Internet), ponemos **localhost** en **Network Domain** y en **Server Name**. Introducimos nuestra **dirección de email**, elegimos **for All Users**.
7. En un browser escribimos la dirección: <http://127.0.0.1/>, y debemos ver la página de inicio de apache.

## Instalación de PHP

A continuación se detalle los pasos para la instalación de php 4 bajo Windows:

1. Bajarse el instalador de PHP4 de la página oficial (<http://www.php.net>).
2. La instalación de PHP es muy sencilla, una vez bajado hay que descomprimirlo; una vez descomprimido en el "c:\\" obteniendo una carpeta que se llame "c:\php-4.2.3-Win32", y luego procedemos a cambiarle el nombre a "c:\php".
3. Procedemos a entrar en esa carpeta y copiar el archivo php.ini-dist a nuestro directorio de Windows (normalmente c:\windows) con el nombre php.ini; el archivo debe quedar así: c:\windows\php.ini

La instalación básica de PHP ya ha finalizado, ahora hay que darle soporte a Apache para PHP, y hacer algunas modificaciones en el archivo php.ini para que todo funcione correctamente, que son las siguientes:

1. Para que Apache soporte PHP tenemos que abrir el archivo httpd.conf (se encuentra en \$APACHE\_ROOT\conf) y añadir al final del mismo este texto:

```
ScriptAlias /php/ "c:/php/"
```

```
AddType application/x-httpd-php .php
```

```
Action application/x-httpd-php "/php/php.exe"
```

2. Procedemos a reiniciar apache para que soporte PHP.



## **Instalación de Java**

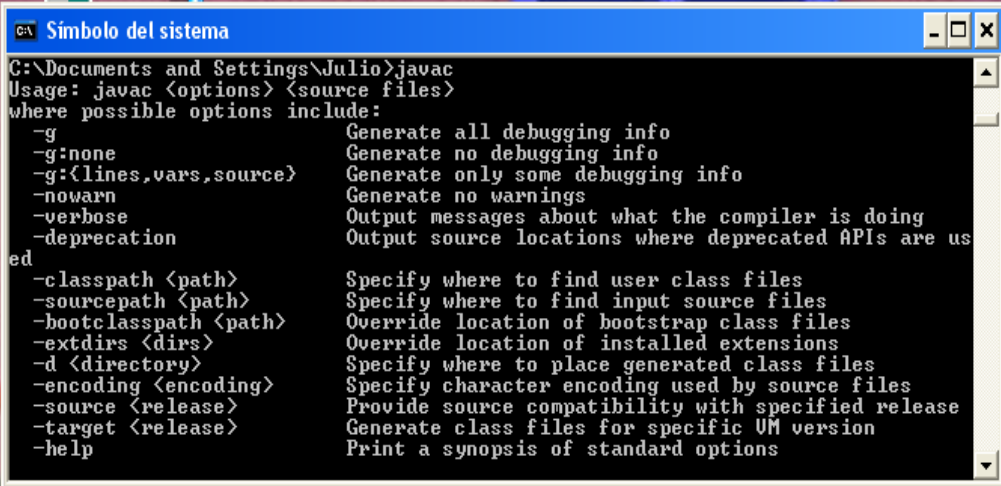
1. Se debe hacer doble clic en el programa de instalación del jdk: "jdk-1\_4\_2\_06-windows-i586-p.exe".
2. Cuando aparezca el cuadro de licencia, pulse "Yo acepto los términos del acuerdo de licencia " y luego "Siguiente".
3. En la siguiente pantalla, no modifique nada y pulse "Siguiente".
4. En la siguiente pantalla sobre los navegadores web, no modifique nada y pulse "Instalar".
5. Espere que se terminen de copiar todos los ficheros. Esto llevará unos minutos.
6. Cuando termine, pulse "Finalizar" y si pide reiniciar el ordenador, proceda a hacerlo.

## **Configuración del java**

Abra la Consola del Sistema, también llamada Símbolo del Sistema (o en inglés, "Console", "DOS Console", "Command prompt"). Habitualmente se encuentra en "Menú Inicio", Programas, Accesorios. Si no se encuentra allí, se lo puede abrir accediendo a "Menú Inicio", Ejecutar y escriba "cmd" (o si no mediante la palabra "command").

En el símbolo del sistema, escriba java

Si se obtiene un mensaje similar a éste:



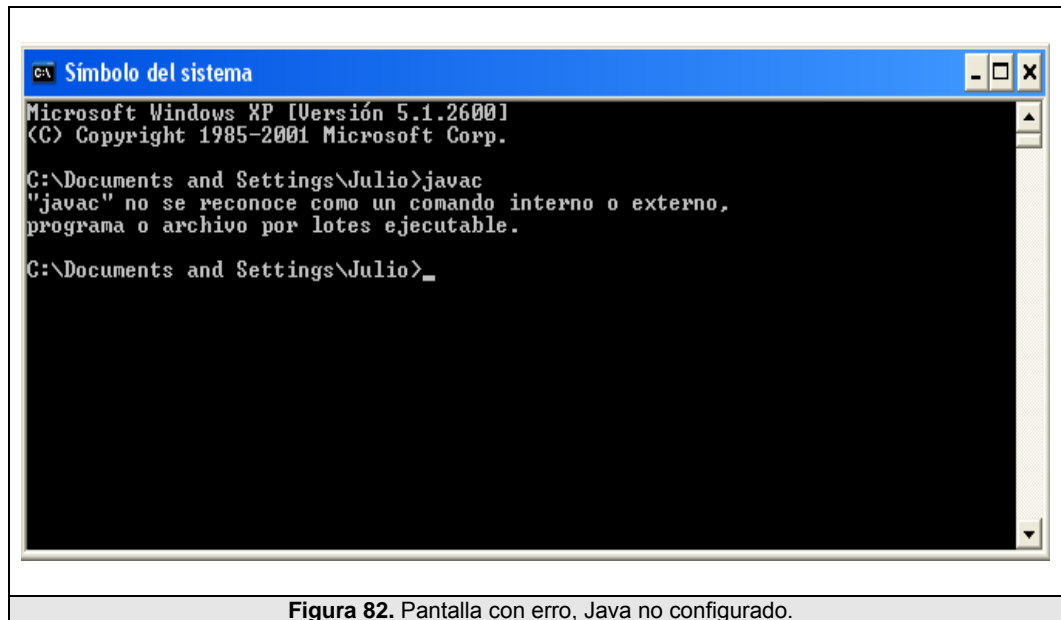
```

C:\Documents and Settings\Julio>javac
Usage: javac <options> <source files>
where possible options include:
-g                Generate all debugging info
-g:none          Generate no debugging info
-g:<lines,vars,source>  Generate only some debugging info
-nowarn         Generate no warnings
-verbose        Output messages about what the compiler is doing
-deprecation    Output source locations where deprecated APIs are used
-classpath <path>    Specify where to find user class files
-sourcepath <path>   Specify where to find input source files
-bootclasspath <path>  Override location of bootstrap class files
-extdirs <dirs>     Override location of installed extensions
-d <directory>     Specify where to place generated class files
-encoding <encoding> Specify character encoding used by source files
-source <release>   Provide source compatibility with specified release
-target <release>   Generate class files for specific VM version
-help            Print a synopsis of standard options
  
```

**Figura 81.** Pantalla confirmando Java configurado..

No hace falta que se siga con este paso porque ya se tiene configurado el Java. Siga con el paso siguiente.

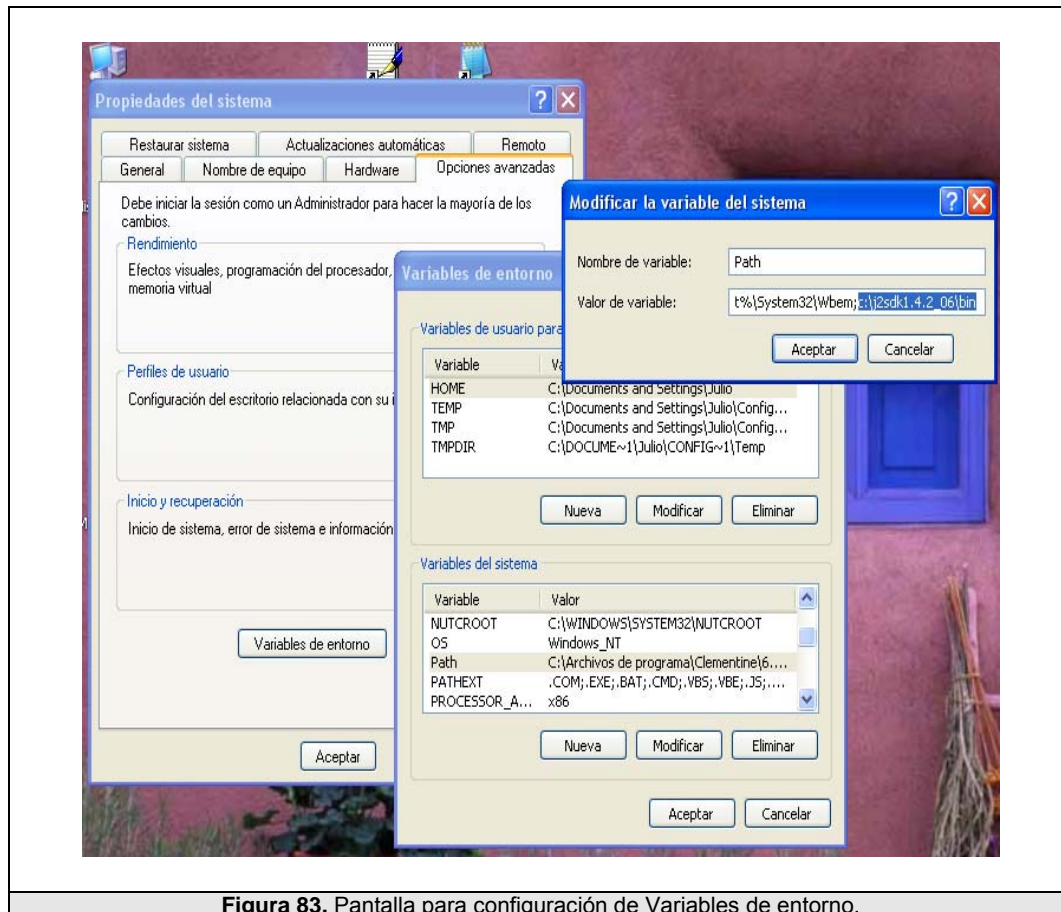
Sin embargo, si se obtiene un mensaje como éste:



Siga las siguientes instrucciones.

1. Pulse con el botón derecho del ratón sobre "Mi PC" y seleccione "Propiedades".
2. Seleccione la pestaña "Opciones avanzadas" y pulse en el botón inferior "Variables de entorno".
3. Tanto en el cuadro superior como el inferior, busque una variable que se llama "Path", haga clic sobre ella y pulse el botón "Modificar" (si aparece en los dos cuadros, haga lo siguiente dos veces).
4. En el cuadro de valor de variable, añada al final.

<lo que hubiere antes>;c:\j2sdk1.4.2\_06\bin y pulse "Aceptar".



**Figura 83.** Pantalla para configuración de Variables de entorno.

5. Busque otra vez en la lista de variables si existe "CLASSPATH". Si no existiese, no se preocupe, siga con el siguiente punto. Si existe, clic sobre ella y pulse el botón "Modificar" (si aparece en los dos cuadros, haga lo siguiente dos veces). En el cuadro de valor de variable, añada al final. Sin borrar nada.

<lo que hubiera antes>;c:\j2sdk1.4.2\_06\bin. y pulse "Aceptar".

6. Cierre el cuadro de diálogo de variables de entorno con "Aceptar" y otra vez "Aceptar" para "Propiedades del sistema".

Si se tenía abierta la consola del sistema, **ciérrela** para que se carguen los nuevos valores.

Abra la consola otra vez y compruebe que al escribir javac y java se obtienen los mensajes habituales de Java. Si sigue apareciendo el mensaje "javac" no se reconoce como un comando interno o externo, programa o archivo por lotes ejecutable revise que se ha seguido todos los pasos correctamente y vuelva a repetir el paso 3.

## **Anexo 4: Instalación del sistema en Linux.**

Para instalar el sistema bajo GNU Linux, necesitando los siguientes paquetes (RPM) de instalación:

- Servidor Web Apache: **httpd-2.0.49-4**
- Servidor de Base de Datos Postgresql: **postgresql-server-7.4.2-1**
- Cliente de Base de Datos Postgresql: **postgresql-7.4.2-1**
- Librerías de Postgresql: **postgresql-libs-7.4.2-1**
- Distribución del Paquete PHP: php-4.3.4-11
- Librerías para la comunicación de PHP con Postgresql: **php-pgsql-4.3.4-11**
- Java Runtime: j2re-1\_4\_2\_07-linux-i586
- Librerías de Java par la conexión con Postgresql: **postgresql-jdbc-7.4.2-1**

## Instalación y Configuración de los Paquetes

### Servidor Web Apache

#### Instalación:

Para instalar este paquete ejecute la siguiente línea:

```
$ rpm -iU httpd-2.0.49-4.rpm
```

Al realizar esto se instalará en servidor Web Apache, además se lo agregará como un servicio del sistema. Para iniciar el servidor Web debemos ejecutar la siguiente línea de código:

```
$ service httpd start
```

Para comprobar que se está corriendo el servidor, en un navegador Web se deberá escribir la **ip** del servidor en la barra de direcciones, por ejemplo si la ip del servidor es 127.0.0.1, deberíamos colocar **http://127.0.0.1**. Al hacer esto deberá mostrarse la pagina inicial del apache indicando que el servidor está instalado y corriendo.

#### Configuración:

Para configurar el servidor web debemos editar el archivo **httpd.conf** que se encuentra en `/etc/httpd/conf`, modificando la línea `DocumentRoot`, indicando la carpeta donde se guardarán las páginas Web del servidor; en nuestro caso

colocaremos todas las páginas Web en la carpeta /var/www/html, entonces debemos editar la línea de la siguiente manera:

```
DocumentRoot "/var/www/html"
```

Luego de esto modificamos la línea AddDefaultCharset UTF-8, la cual establece la codificación de los caracteres permitidos en las páginas web, colocando la codificación ISO-8859-1 en vez de UTF-8, entonces la línea quedaría de la siguiente manera:

```
AddDefaultCharset ISO-8859-1
```

### **Servidor de Base de Datos Postgresql**

#### **Instalación:**

Para instalar este paquete ejecute la siguiente línea:

```
$ rpm -iU postgresql-server-7.4.2-1.rpm
```

Además se deben instalar los siguientes paquetes:

- postgresql-libs-7.4.2-1
- postgresql-7.4.2-1

Para instalarlos debemos ejecutar las siguientes líneas, en este orden:



```
$ rpm -iU postgresql-libs-7.4.2-1
```

```
$ rpm -iU postgresql-7.4.2-1.rpm
```

La primera línea instalará las librerías del Postgresql y la segunda línea instalará los clientes de Postgresql.

### **Configuración:**

A continuación debemos configurar el servidor, para esto primero debemos especificar el lugar en disco donde se guardarán las bases de datos. Entonces crearemos la carpeta data en el directorio `/usr/share/pgsql/`, para crear esta carpeta hacemos lo siguiente:

Primero nos localizamos en el directorio `/usr/share/pgsql/`, ejecutando la siguiente línea:

```
$ cd /usr/share/pgsql/
```

Luego creamos la carpeta data ejecutando la siguiente línea:

```
$ mkdir data
```

Luego de esto debemos crear un usuario llamado **postgres**, el cual podrá conectarse a la base de datos, para esto ejecutamos la siguiente línea:

```
$ adduser postgres
```

Luego de esto debemos dar privilegios para que cualquier usuario pueda manejar el directorio data que creamos, para esto ejecutamos la siguiente línea:

```
$ chmod 777 /usr/share/pgsql/data/
```

Luego de esto nos logoneamos al servidor con el usuario postgres, e inicializamos el servidor de bases de datos indicando la carpeta en donde debe almacenar las bases de datos de postgresql, ejecutando la siguiente línea:

```
$ initdb -D /usr/share/postgresql/data
```

Después de esto debemos logearnos al servidor mediante una conexión ssh, usando el usuario postgresql. Luego de esto debemos inicializar el postmaster, el cual inicializará la base de datos. En esta parte debemos tomar en cuenta la cantidad de conexiones que al mismo tiempo podrá manejar la base de datos y el buffer de conexión (si usamos conexión persistente a la base), en nuestro caso usaremos un número de conexiones (N) igual a 64 y un tamaño de buffer de 128 (B). Además debemos especificar que permita conexiones remotas al servidor (i) y el puerto el cual usará (en nuestro caso 5432). Para esto ejecutamos la siguiente línea:

```
$ postmaster -D /usr/share/postgresql/data -i -p5432 -N64 -B128
```

### **Distribución del Paquete PHP**

#### **Instalación:**

Para instalar este paquete ejecute la siguiente línea:

```
$ rpm -iU php-4.3.4-11.rpm
```

Esta instalación, además de instalar el PHP en el sistema, se encargará de modificar el servidor apache para que pueda ejecutar scripts PHP.

#### **Configuración:**

Para que no se muestren los errores probables de alguna aplicación PHP, se debe manipular la línea `display_errors = On`, y se debe colocar lo siguiente:

```
display_errors = Off
```

### **Librerías para la comunicación de PHP con Postgresql**

#### **Instalación:**

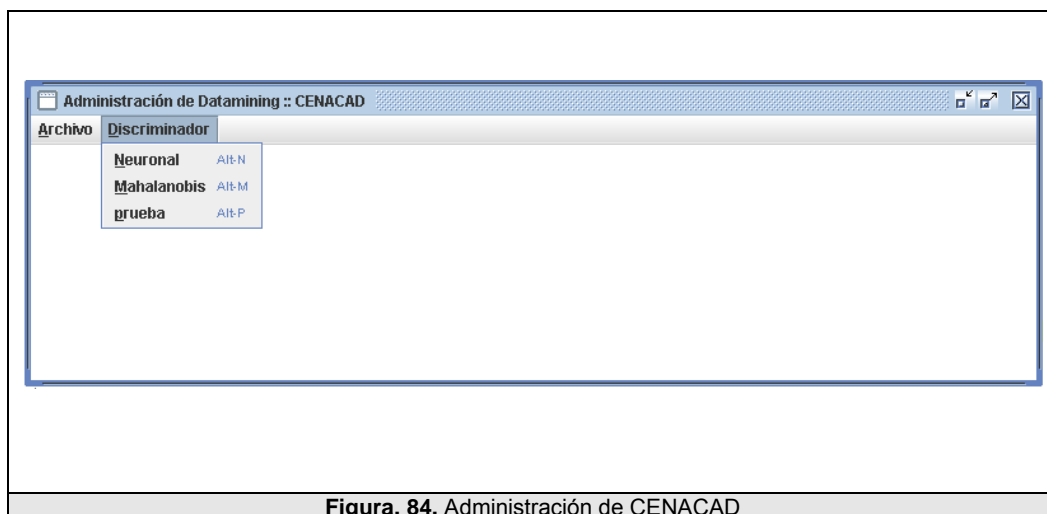
Para instalar este paquete ejecute la siguiente línea:

```
$ rpm -iU php-pgsql-4.3.4-11.rpm
```

## Anexo 5. Manual de Usuario

### Manual del usuario sobre el usuario del módulo Discriminador (JAVA)

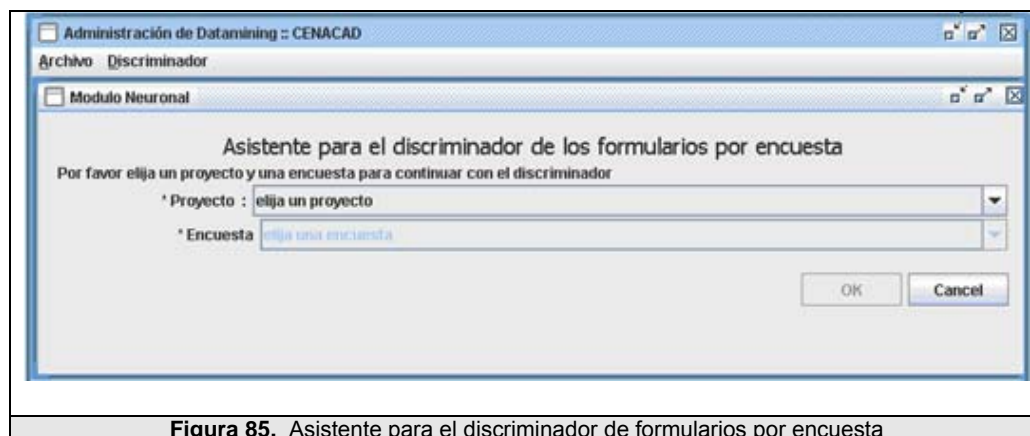
Al momento de arrancar con el uso del sistema, este presenta una interfaz en la cual se visualiza el menú de selección para el proceso la discriminación de formularios.



**Figura. 84.** Administración de CENACAD

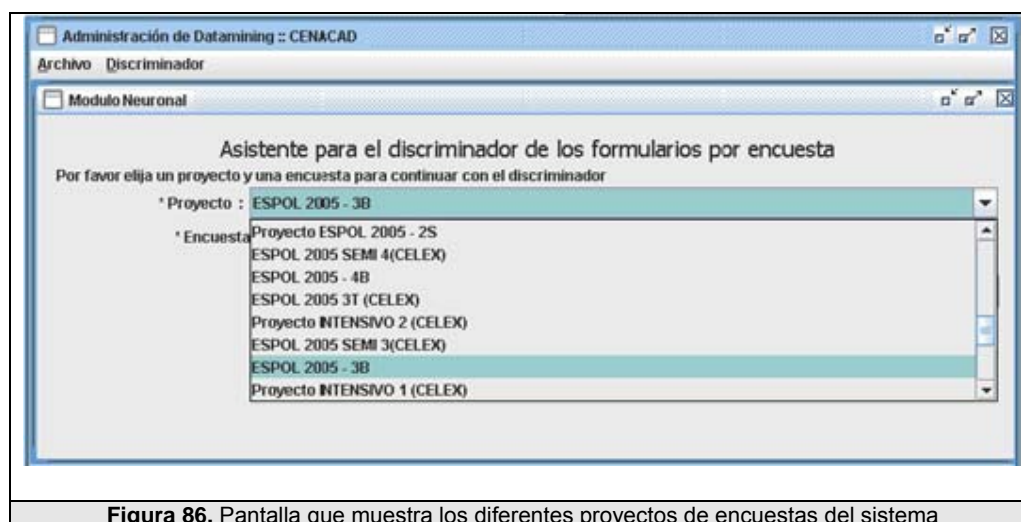
En este menú seleccionaremos “Discriminador → neuronal”, el mismo que se ha implementado usando las reglas u algoritmo de Kohonen, dentro de un modelo “no supervisado”, que se utiliza para procesos de clasificación.

Al momento de seleccionar esta opción se deberá ingresar el proyecto<sup>1</sup>.



**Figura 85.** Asistente para el discriminador de formularios por encuesta

Por ejemplo: Proyecto ESPOL 2005 – 3B, este proyecto indica mediante el nombre que es un proyecto realizado a nivel de toda ESPOL, en el año 2005, y dentro del 3er período Bimestral.



**Figura 86.** Pantalla que muestra los diferentes proyectos de encuestas del sistema

<sup>1</sup> Proyecto: Recolección de varias encuestas con la finalidad de unificar su reporte, que puede ser especificada por unidad, grupo y término.

Luego que se ha determinado el proyecto, es necesaria elegir la encuesta que contiene el proyecto para realizar el proceso de discriminación de formularios.

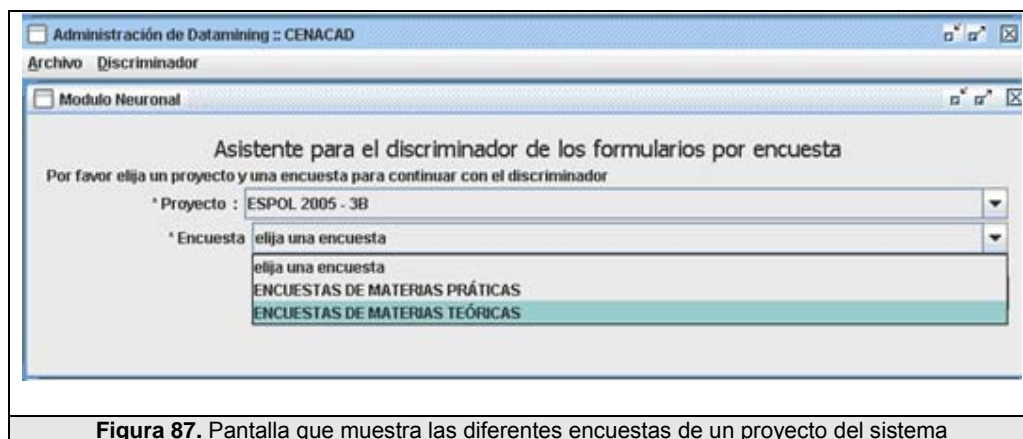
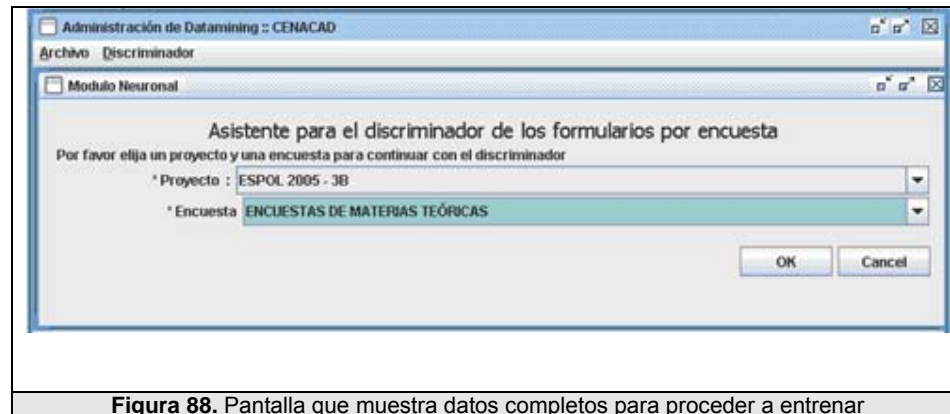


Figura 87. Pantalla que muestra las diferentes encuestas de un proyecto del sistema

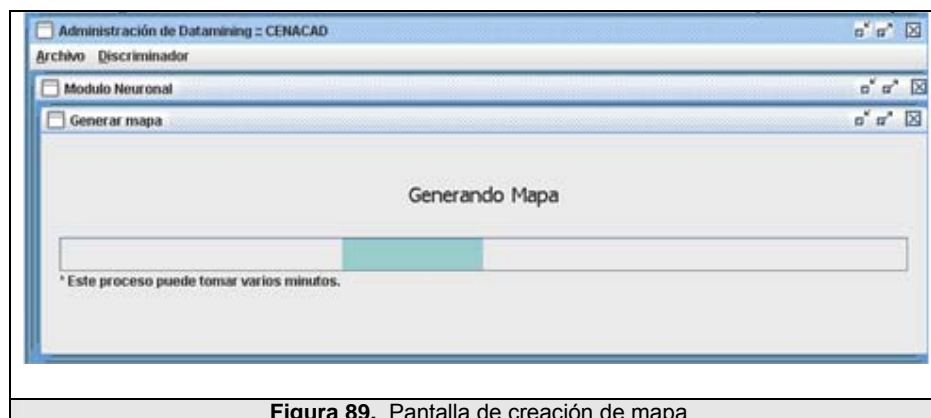
\* **NOTA:** Debemos recordar que el sistema nos permite realizar este proceso utilizando cualquier cuestionario, sin importar el número de preguntas que tenga el mismo. Sin embargo se recalca que las preguntas de este cuestionario deben ser del tipo “escala likert”.



**Figura 88.** Pantalla que muestra datos completos para proceder a entrenar

Con estos parámetros seleccionados, procedemos a presionar un clic en el botón "OK".

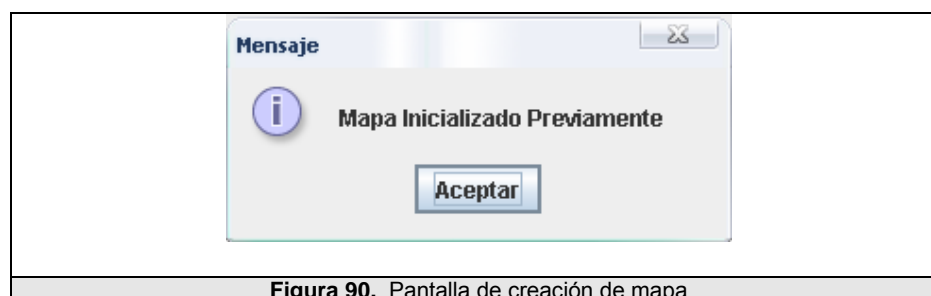
Al momento de completar los datos requeridos se procede con la creación del mapa con valores aleatorio entre 0 y 1, dando así el inicio a la fase de "generación de mapa", el mismo que tiene una dimensión de 8 x 8 neuronas (dimensión establecida bajo previa investigación), es decir una matriz de 64 neuronas, hay que tener en cuenta esto para al final del proceso notar que se presentaran un total de 64 patrones como resultados, es decir, un patrón por neurona.



**Figura 89.** Pantalla de creación de mapa

**\*NOTA:** este proceso puede durar entre 4 a 8 horas.

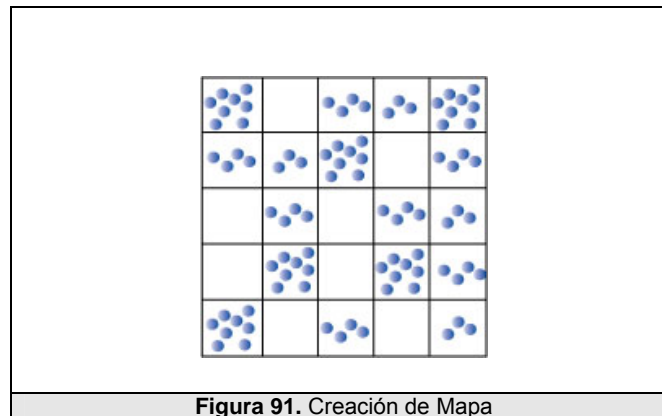
Una vez concluido la generación del mapa con sus vectores inicializados aleatoriamente aparecerá la pantalla que muestra la Figura 1.9.



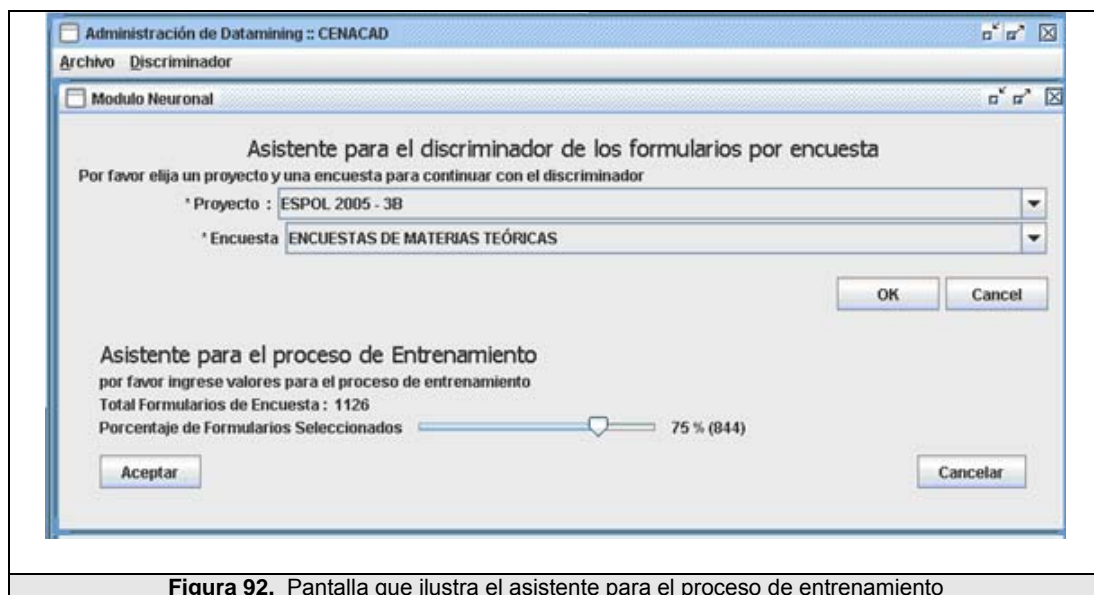
**Figura 90.** Pantalla de creación de mapa

Una vez realizado esta fase de inicialización podemos proceder a la siguiente fase de búsqueda de conglomerados, utilizando la matriz creada de dimensión de 8 por 8 (matriz neuronal), de los cuales cada casilla de la matriz consta de un vector de tamaño similar al número de preguntas que tenga el cuestionario de encuesta seleccionado.





Luego de esto se puede proceder a realizar la segunda fase del proceso neuronal, la misma que es la fase de entrenamiento, en donde el usuario tiene la facilidad de seleccionar el número de formularios a ingresar para tal efecto (tiene un porcentaje por defecto equivalente al 75% del total de formularios dentro de la encuesta).



**\*NOTA:** Este proceso puede durar entre 4 a 8 días.

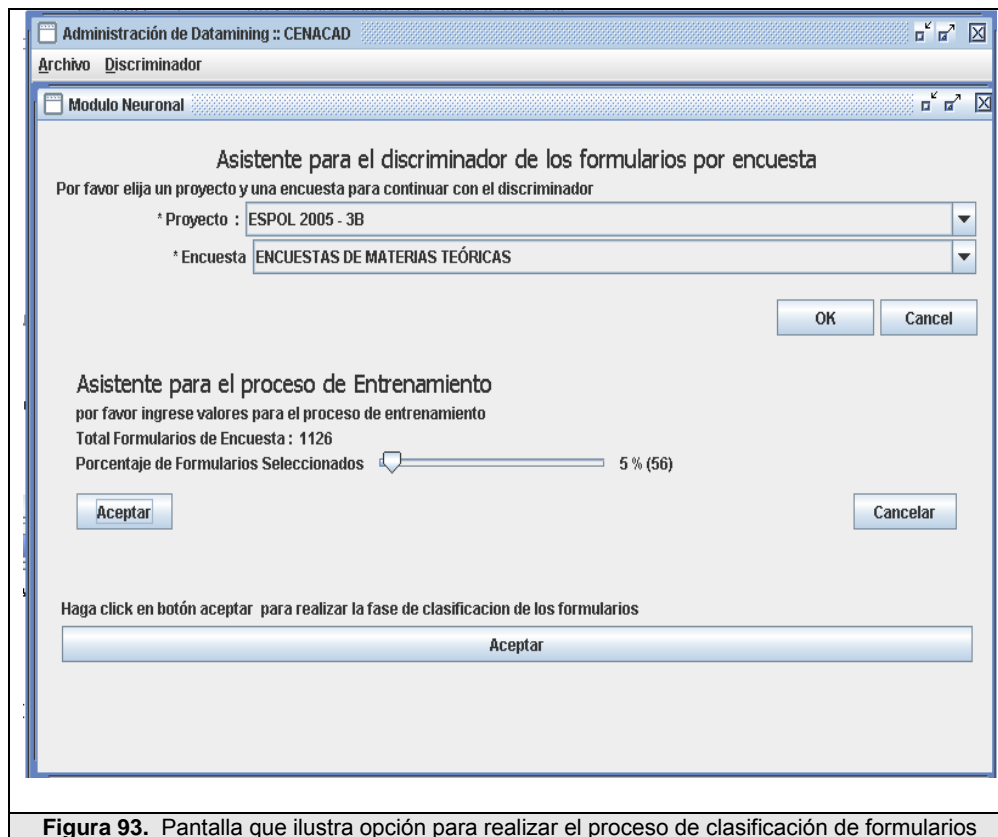
**\*NOTA:** Para efectos de prueba se recomienda seleccionar un porcentaje pequeño de formularios.

Una vez determinado el valor se procede a realizar el entrenamiento. Cuyo valor recomendado es del 75% para poblaciones menores a 500, y si es mayor al mismo se puede considerar entre 500 a 10.000 patrones<sup>2</sup> para su respectivo entrenamiento.

Luego del proceso de entrenamiento (el tiempo de esta fase depende mucho del valor o porcentaje que se ingresa para entrenar), se procede a realizar la fase de clusterización o categorización, en esta fase los patrones que se han determinado en el proceso de entrenamiento cumplen una función muy importante, ya que se debe a ellos la realización de esta fase, el uso de los patrones neuronales hallados durante el proceso de entrenamiento y que el sistema los maneja de manera interna, permite asociar los formularios de la encuesta y categorizarlos.

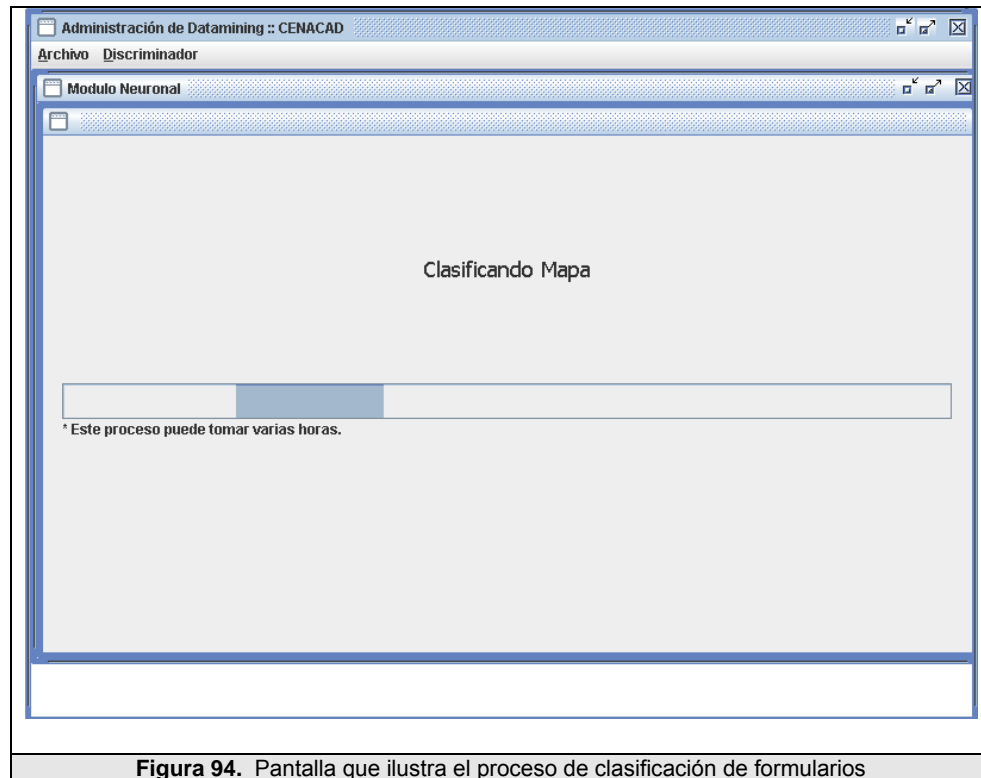
---

<sup>2</sup> Patrones: equivalente a decir formularios



**\*NOTA:** este proceso puede durar entre 15 minutos a 2 horas.

**\*NOTA:** El proceso de clasificación se da inicio y puede variar el tiempo que demore debido al número de formularios que existan dentro de la encuesta seleccionada.



**Figura 94.** Pantalla que ilustra el proceso de clasificación de formularios

Luego de haber finalizado el proceso de clasificación de los formularios de una encuesta, procedemos a ingresar al módulo de Administración o Visualización de los resultados que se ha desarrollado en un módulo Web. (Ver anexo 3-1).

## ANEXO 5-1

### Administración de Módulo de Discriminadores (WEB)

En este módulo el usuario podrá elegir la encuesta que desee con la finalidad de poder realizar las siguientes actividades: a) Ver formularios Discriminados, b) Ver Discriminador neurona, c) Ver discriminador Mahalanobis d) Ver discriminador con regresión logística.

ADMINISTRACIÓN :: CENSO ACADÉMICO en línea

Administración | Proyectos | Encuestas | Cuestionarios | Usuarios | Listar Categorías | Clusterización | Discriminador | Estadísticas | Reportes [\[Salir\]](#)

**Administración de Discriminadores de Formularios**

Elija encuesta:

Año:  Período:  \*Encuesta:

\* Elija acción:

Ver Formularios Discriminados  Discriminador Neuronal  Discriminador Mahalanobis  Discriminador Regresión Logística

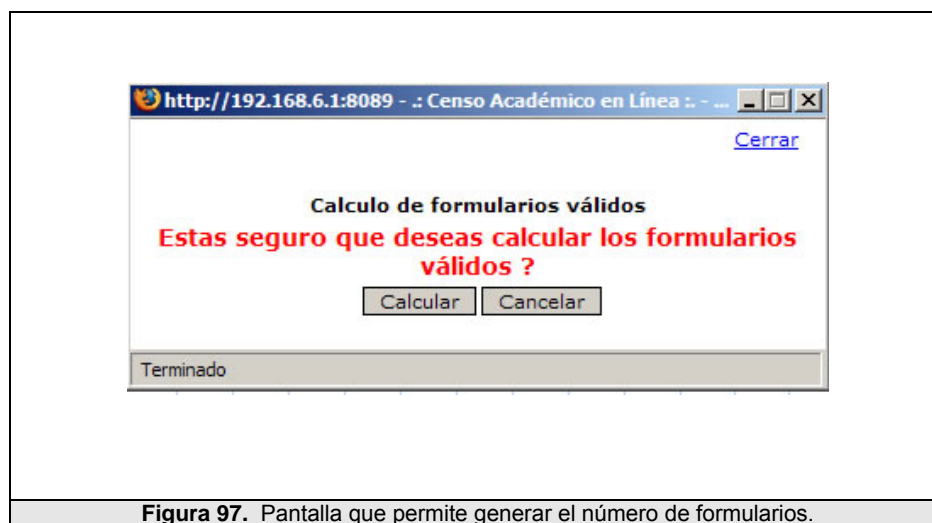
\* Campos Obligatorios

**Figura 95.** Pantalla que ilustra el menú de administración del Discriminador de Formularios

Una vez elegido la encuesta a la cual ha sido previamente aplicado el algoritmo de Kohonen en el módulo de java, podemos entregar el reporte que contiene el resumen del mapa neuronal.



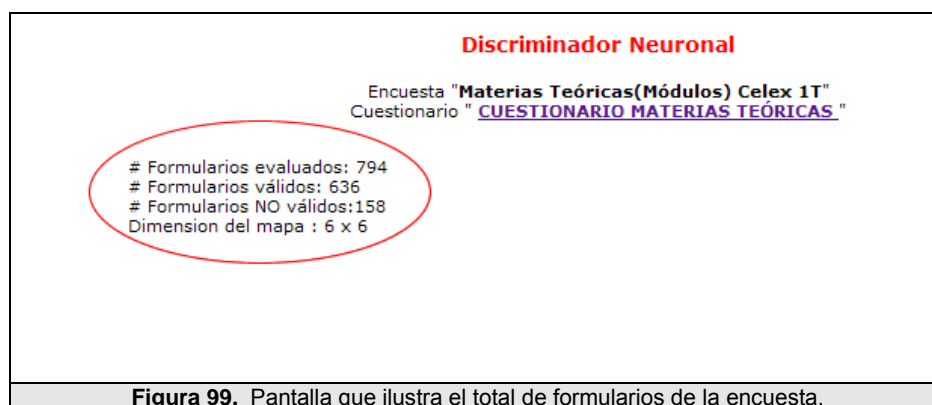
Sin embargo para efectos de conocer cuáles son los formularios válidos y no válidos, procedemos a presionar en el botón “Calcula válidos”, para la cual nos aparecerá una pantalla para confirmar el cálculo a realizar.



Una vez realizado el cálculo podremos apreciar los resultados de los formularios válidos y no válidos, por cada una de las neuronas. Recordando que la optimización de los formularios se realiza a todos los formularios que se encuentran más cercanos a su neurona respectiva.

Formularios de la Neurona # 192																																				
Encuesta " Materias Teóricas(Módulos) Celex IT "																																				
Cuestionario " CUESTIONARIO MATERIAS TEÓRICAS "																																				
id_patron	distancia	Preg #1	Preg #2	Preg #3	Preg #4	Preg #5	Preg #6	Preg #7	Preg #8	Preg #9	Preg #10	Preg #11	Preg #12	Preg #13	Preg #14	Preg #15	Preg #16	Preg #17	Preg #18	Preg #19	Preg #20	Preg #21	Preg #22	Preg #23	Preg #24	Preg #25	Preg #26	Preg #27	Preg #28	Preg #29	Preg #30	Preg #31	Preg #32	Preg #33	Preg #34	
40991	0.2033	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3		
40909	0.6026	3	3	3	3	3	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
49006	0.9000	3	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
48856	0.9895	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
48994	0.9915	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
49280	1.0005	2	3	3	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
49507	1.0586	3	2	3	3	3	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
49390	1.1250	2	3	2	3	2	3	3	3	3	3	3	2	3	2	2	2	2	2	2	2	2	2	3	3	3	2	2	2	3	2	2	2	3	3	
49355	1.1748	3	2	3	3	3	4	3	4	3	4	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
49160	1.1798	3	2	2	2	2	3	3	3	3	4	2	2	2	3	2	2	2	2	2	3	3	4	2	3	4	4	3	3	3	3	3	3	3	3	
49167	1.1896	3	2	2	3	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
49270	1.2606	3	3	3	3	3	4	4	1	5	3	3	3	3	3	3	3	3	3	3	2	2	2	3	2	2	3	3	2	3	3	4	3	4	3	
49021	1.3587	3	3	2	3	2	5	4	3	2	4	1	2	2	2	3	3	3	4	4	3	4	3	3	3	3	2	2	3	2	2	2	2	3	4	
49562	1.4236	3	2	3	2	2	4	3	3	2	3	2	3	2	2	2	2	2	3	2	2	3	3	3	2	2	4	3	3	1	3	5	5	2	2	
49679	1.4771	3	3	3	2	2	4	2	3	3	2	2	2	3	4	3	3	3	3	3	3	3	3	3	3	2	3	1	3	3	5	3	4	4	3	
49318	1.4592	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
49015	1.5382	3	3	3	2	3	4	4	4	2	3	3	2	3	3	3	3	2	2	2	2	2	2	3	5	5	4	4	3	3	4	2	5	5	2	
49218	1.7080	3	3	3	2	2	3	5	5	5	4	4	3	3	3	2	3	3	3	3	2	3	4	4	4	4	3	3	3	3	2	3	4	5	3	
49261	1.7119	3	3	3	3	3	5	3	3	1	4	1	3	4	3	3	3	3	4	2	1	3	3	3	3	4	3	4	3	2	1	5	5	2	2	
49070	1.9720	3	3	3	3	3	4	3	2	2	4	1	2	4	3	4	3	4	4	3	4	2	5	4	3	4	4	4	3	4	4	3	4	4	3	
49023	2.2024	3	3	3	2	2	2	2	1	3	2	2	2	2	2	1	2	2	2	2	2	2	3	2	4	3	5	3	1	5	4	1	4	5	2	
49530	2.3150	3	3	3	3	4	4	4	3	2	3	2	1	2	2	4	1	2	5	1	1	4	3	4	2	2	3	4	4	4	4	4	4	5	2	
49262	2.3157	3	2	3	3	2	5	4	2	4	3	2	4	3	4	2	1	3	2	4	1	3	2	3	2	3	3	5	3	2	3	3	3	2	3	
49352	2.3356	2	3	3	2	3	3	3	3	3	3	3	1	2	2	3	3	3	3	3	3	3	4	2	4	4	4	5	5	5	4	4	2	4	3	
48852	2.4347	2	3	3	3	2	2	2	3	2	3	3	3	3	3	4	2	3	3	4	3	4	1	3	1	1	3	3	2	2	1	1	3	5	2	
49267	2.0791	2	3	3	2	3	3	2	2	4	2	2	2	3	2	2	3	2	3	2	4	4	4	1	1	1	2	1	4	1	1	3	3	3	3	
49410	3.7077	3	3	3	3	2	1	1	1	1	4	4	4	3	4	3	4	3	4	3	1	1	2	3	2	2	2	4	3	2	5	5	5	5	5	
49751	5.9835	3	3	3	3	3	5	5	5	5	5	1	1	1	1	1	1	1	1	1	1	1	1	5	1	5	5	5	5	1	1	1	1	1	5	1

Figura 98. Pantalla que conocer los patrones o formularios válidos y no válidos



**\*Formularios generados:** Total de formularios que fue generado por el sistema, para cada estudiante en cada paralelo que ha tomado.

**\*Formularios evaluados:** Total de formularios que han sido evaluados por los estudiantes que se encuentra dentro del número de formularios generados

**\*Formularios válidos:** total de formularios que cumplen con más de dos algoritmos planteados en el proyecto.

Por otro lado como mencionamos anteriormente, estos resultados podría tener un sesgo mayor ya que este último paso es bajo la consideración de un



experto estadístico. Es así que se incluye el reporte de varios métodos de discriminación para realizar la comparación y la obtención de los formularios válidos y no válidos. (Ver Figura 100)

<b>Lista de Formularios Válidos / NO Válidos</b>				
Encuesta " <b>Materias Teóricas(Módulos) Celex 1T</b> "				
Cuestionario " <b><u>CUESTIONARIO MATERIAS TEÓRICAS</u></b> "				
# Total Formularios generados: 1042			Neuronal = 636 válidos de 794	
# Total Formularios evaluados: 794			Regresión= 0 válidos de 794	
# Total Formularios válidos: 794			Mahalanobis = 338 válidos de 794	
			Euclidiana = 771 válidos de 794	
ID Formulario	Neuronal	Regresión	Mahalanobis	Euclidiana
49895	sí	--	sí	sí
49894	sí	--	sí	sí
49893	sí	--	XXX	sí
49892	XXX	--	XXX	sí
49891	sí	--	XXX	sí
49890	sí	--	XXX	sí
49889	sí	--	XXX	sí
49888	XXX	--	sí	sí
49887	sí	--	XXX	sí
49886	sí	--	sí	XXX

**Figura 100.** Pantalla que ilustra si un formulario cumple con los algoritmos de clasificación.

**Anexo 6: Manual del programador.****Clase: ClasificaGUI.java**

Descripción: Clase controla la clasificación de los formularios válidos y no válidos acorde a las especificaciones del usuario.

**Clase: ClusterKohonenBean.java**

Descripción: Clase maneja la instancia del objeto clúster.

**Clase: ClusterKohonenComparator.java**

Descripción: Clase maneja una instancia del objeto clúster.

**Clase: ClusterKohonenListener.java**

Descripción: Clase maneja los eventos del manager de los objetos clúster.

**Clase: ClusterKohonenManager.java**

Descripción: Clase maneja todos los objetos clúster

**Clase: CreateMapaGUI.java**

Descripción: Clase que crea la interface gráfica de usuario para seleccionar el proyecto y la encuesta de los cuales se generará la red neuronal.

**Clase: EntrenamientoGUI.java**

Descripción: Clase que controla la interface para realizar el entrenamiento de los mapas.

**Clase: GenerandoMapaGUI.java**

Descripción: Clase que muestra gráficamente el avance de la generación de la red neuronal para el proyecto y la encuesta seleccionados.

**KohonenFeatureMapBean.java**

Descripción: Clase que controla una instancia del objeto mapa.

**KohonenFeatureMapManager.java**

Descripción: Clase que controla todos los objetos “mapa”.

**NeuronBean.java**

Descripción: Clase que controla una instancia del objeto neurona.

**NeuronListener.java**

Descripción: Clase que controla los eventos de los objetos de manager de las neuronas.

**NeuronManager.java**

Descripción: Clase que controla todos los objetos neuronas.

**VectorBean.java**

Descripción: Clase que controla una instancia del objeto vector.

**VectorListener.java**

Descripción: Clase que controla los eventos de la clase vector.

**VectorManager.java**

Descripción: Clase que controla todos los objetos vector.

**WeightMatrix.java**

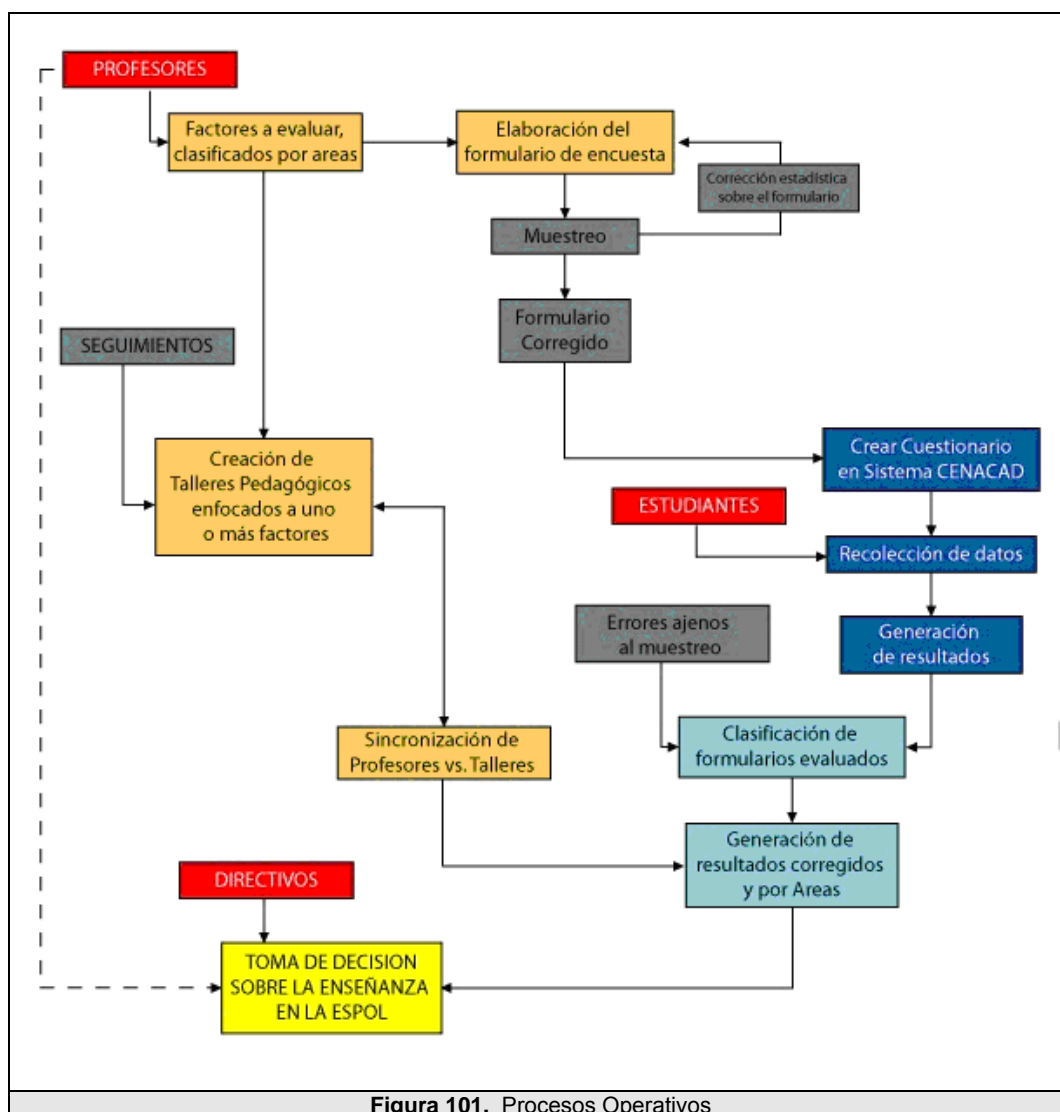
Descripción: Clase que controla la matriz de pesos del mapa de Kohonen.

**Anexo 7: Requerimientos.**

- Procesador 3.0 GHZ. o superior.
- Computador con memoria RAM 512 MB
- PIV 1.8 o mayor
- Disco 20 GB
- Servidor Web Apache: **httpd-2.0.49-4**
- Servidor de Base de Datos Postgresql: **postgresql-server-7.4.2-1**
- Cliente de Base de Datos Postgresql: **postgresql-7.4.2-1**
- Librerías de Postgresql: **postgresql-libs-7.4.2-1**
- Distribución del Paquete PHP: php-4.3.4-11
- Librerías para la comunicación de PHP con Postgresql: **php-pgsql-4.3.4-11**
- Java Runtime: j2re-1\_4\_2\_07-linux-i586
- Librerías de Java par la conexión con Postgresql: **postgresql-jdbc-7.4.2-1**

### Anexo 8: Manual operacional.

El manual operacional involucra los pasos antes, durante y después del proceso de clasificación neuronal, es una visión general de la problemática generada y la forma en la que se cierra el ciclo con la solución del proceso clasificatorio.



En el gráfico adjunto se visualizan los procesos que se generan antes del proceso de clasificación (color celeste), en ellos se emplea análisis estadístico, pedagógico, realización de encuestas, los mismos que van siendo ejecutados con la ayuda de profesionales involucrados cada uno en su respectiva área de conocimiento.

En este manual se presenta en detalle los recursos que son necesarios para realizar el proceso de discriminación de formularios, un proceso automático que depende de procesos operativos. Tales como:

### **1. Como se desarrolla el cuestionario de encuesta (Proceso Administrativo)**

La necesidad de medir las falencias, habilidades, características en general que un docente presenta en la realización de su cátedra determina la creación de herramientas para determinar el grado de acierto o desacierto en el uso de las técnicas empleadas por el docente, Estas técnicas se deben medir o evaluar con la respuesta de los estudiantes.

El uso de cuestionarios de evaluación en el cual se detallan, mediante el uso de preguntas las características o factores a evaluar a los docentes, se ha generalizado hoy en día para la evaluación de docentes en los respectivos cursos que dictan.

Se considera uno de los procesos operativos la realización de un cuestionario de evaluación, en el cual interviene un personal dedicado (pedagógico, psicológico) para determinar las áreas de evaluación, se propone a discusión los temas a evaluar y analizar la metodología para hacerlo, además basándose en factores a medir se determina el cuestionario más óptimo.

Esto está fuera del propósito resolutivo de esta tesis pero cabe mencionar como se va generando paso a paso el uso de la evaluación hasta llegar al modelo de clasificación.

Hemos notado la creación de un cuestionario de encuesta no sistematizado, se necesita determinar la efectividad de dicho cuestionario para medir los factores que se hayan planteado, para este análisis interviene el personal estadístico, el mismo que con el uso de herramientas diseñadas para tal misión determina si es apto o no el cuestionario para ser aplicado a una encuesta. (Este proceso necesita ser retroalimentado por procesos de prueba de evaluación).

Luego de determinar el cuestionario idóneo, se decide realizar la encuesta en un lapso de tiempo definido por el área administrativa ó en el momento que se desee por la misma. Existen tipos de cuestionarios que se aplican de



acuerdo a la naturaleza del curso ó materia, es decir, cuestionarios que evalúan materias técnica-practica (laboratorios), y materias completamente teóricas.

Este es el método operativo realizado para evaluar a un docente.

## **2. Como es el proceso de evaluación (Proceso Técnico-Operativo)**

En el paso anterior se conoció el proceso de la creación de un cuestionario mediante la existencia de características o factores de evaluación y determinándose el adecuado, luego de este paso el cuestionario creado genera un proceso evaluativo el mismo que puede ser realizado de manera manual (empleando personal de campo) o mediante un proceso sistemático (empleando un sistema evaluador).

Para el proceso de la tesis se ha empleado el uso del sistema evaluador docente CENACAD. Este sistema suplanta el proceso operacional de campo de manera óptima, y permite el ingreso del modelo de cuestionario de evaluación realizado por el área administrativa, mediante la creación de un proyecto evaluativo que se lo relaciona a unidades académicas o grupos dentro de alguna unidad académica. Esto evita la creación excesiva de documentos físicos (hojas) de evaluación, y asocia la población real que debe ser evaluada.

Luego de haber asociado el cuestionario a la población de evaluación respectiva se procede a abrir el proceso de evaluación dentro del período asignado.

El modo de evaluación es el siguiente: Los estudiantes acceden vía Web a la dirección [www.cenacad.espol.edu.ec](http://www.cenacad.espol.edu.ec), el estudiante ingresa usuario y clave proporcionado por la institución educativa, y luego con la aprobación del acceso se le presentan los cuestionarios de evaluación asignados por el número de materias en las que el estudiante se encuentra registrado en el período actual (semestre, bimestre, trimestre, etc.). Esto es realizado a cada uno de los estudiantes que han sido seleccionados dentro del proyecto de evaluación, obteniendo de esta manera un proceso más rápido, datos reales (directos del estudiante).

En el proceso nace una problemática que se genera al momento de la evaluación por parte del estudiante, el sistema permite la evaluación como se lo hiciera de manera manual con una mayor eficiencia en la recolección de datos, sin embargo existen errores que se encuentran fuera del alcance del proceso de creación del proyecto y al proceso de evaluación, es un problema causado por el comportamiento del estudiante, es decir, un estudiante puede evaluar al apuro, de manera negativa en contra del docente por varias

situaciones, afectando de esta manera los resultados reales que se miden con el cuestionario. Esto no es fácil controlar pues se encuentra dentro del estudiante al momento en el que realiza la encuesta. Se conoce que no existe encuesta exacta es por ello la existencia del grado de incertidumbre de una evaluación, pero existen técnicas que ayudan a reducir el rango de valores que se alejan del proceso real de evaluación.

Mediante el uso de una de estas técnicas expuesta en esta tesis, se logra reducir este número para generar valores más reales, reduciendo así los errores al momento de calificar la cátedra de un docente. La técnica empleada es automática y ha sido expuesta en los capítulos de este documento es conocida como “búsqueda de conglomerados” ó “clustering”, su funcionamiento se basa en la creación de grupos empleando patrones asociados a los mismos, estos patrones se obtienen de los resultados de una encuesta. Esto se conoce como clasificación de cuestionarios evaluados.

### **3. Como clasificar los cuestionarios evaluados (Proceso Operativo-Automático)**

Luego del proceso evaluativo se ejecuta el proceso clasificatorio para la optimización de datos. Para realizar el proceso clasificatorio se necesita una evaluación con el período cerrado, el proceso empleado de Redes Neuronales como se ha mencionado es un proceso Off-line, es decir, el

proceso se utiliza luego de la recolección de datos para dar inicio a la etapa de entrenamiento, por lo tanto la discriminación se debe realizar después del proceso de evaluación, esto da a notar que no es un proceso predictivo sino más real por la manipulación de datos reales.

Cuando se cierra el período de evaluación, se debe chequear cuantos formularios tenemos para discriminar.

El proceso clasificatorio depende de parámetros. Debe determinarse un número de formularios a formar parte de la capa de entrenamiento de la red, un número que represente una muestra lo suficientemente completa para evitar los clusters con errores de aprendizaje.

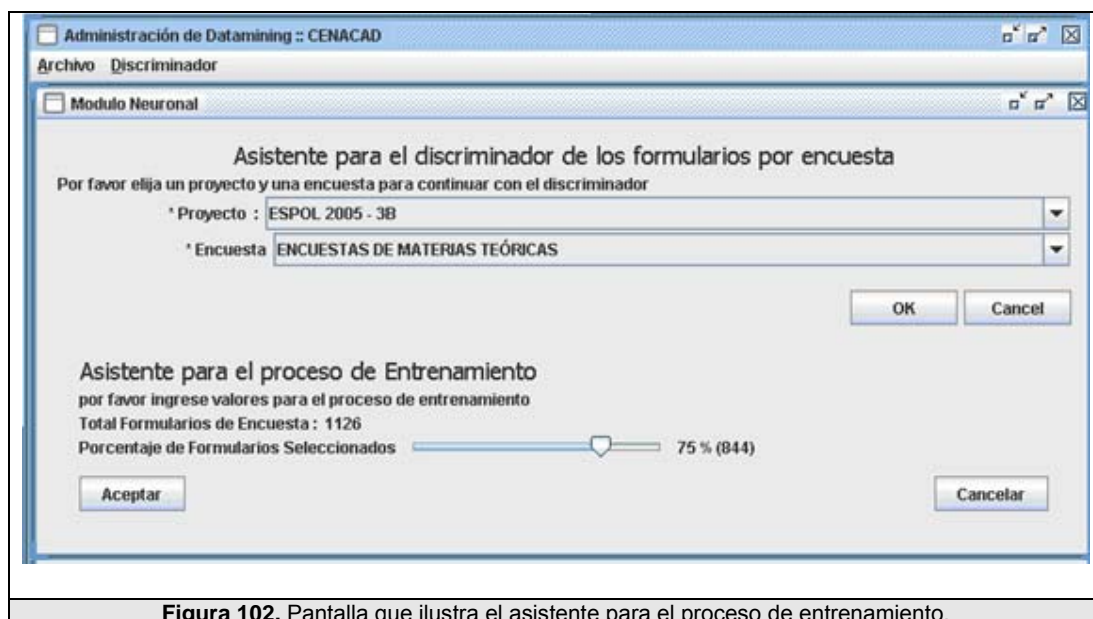
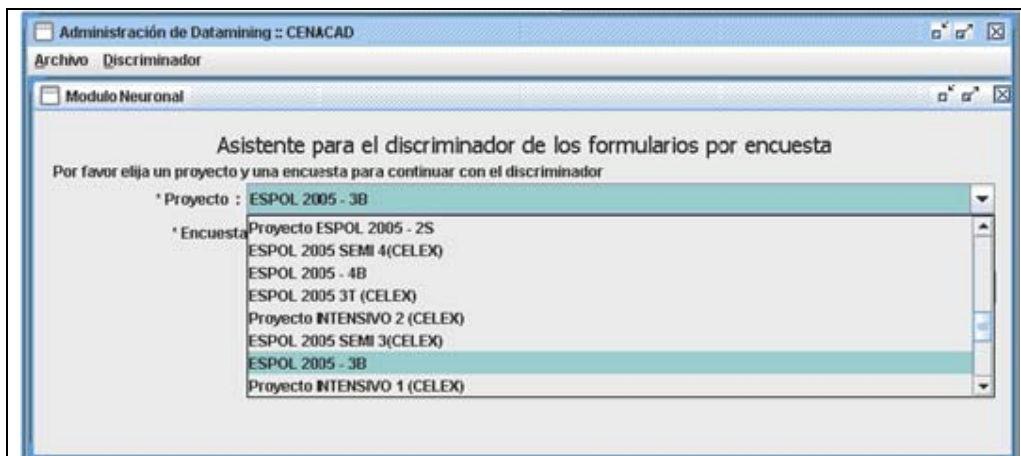


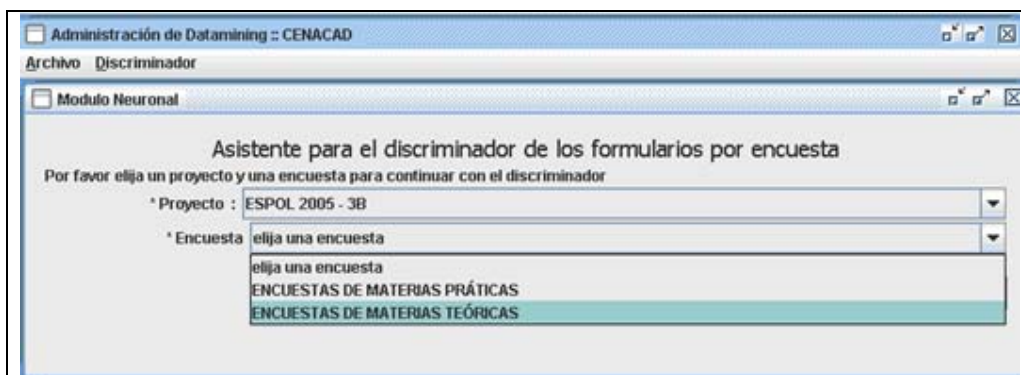
Figura 102. Pantalla que ilustra el asistente para el proceso de entrenamiento.

Por tal motivo se recomienda un 75% del total de formularios evaluados para una población con un tamaño menor a 500 pero si la población es grande es recomendable un rango entre 500 a 10000 formularios para el entrenamiento.



**Figura 103.** Pantalla que muestra los diferentes proyectos de encuestas del sistema

Se observa con la elección del proyecto, y la encuesta de evaluación el número de formularios que se han registrado como evaluados en el sistema,



**Figura 104.** Pantalla que muestra las diferentes encuestas de un proyecto del sistema

Además la elección de encuesta asocia al mapa el cuestionario de evaluación empleado durante el proceso evaluativo, y que determina con ello el mapa de características del proceso de Kohonen. Este mapa se interpreta como la relación estudiante-profesor creada mediante las respuestas del estudiante al cuestionario evaluador del docente.

Cada conglomerado se lo puede visualizar como el comportamiento que presentan grupos de estudiantes hacia los docentes. (Estudiantes en desacuerdo con el tipo de enseñanza del docente, Estudiantes en acuerdo con la enseñanza brindada, Estudiantes en parcial acuerdo, Estudiantes que no reflejan los resultados de la mayoría de grupo). En fin es una relación donde las características expuestas a medir a los docentes se las visualiza de forma global con la personalidad de los estudiantes.

El funcionamiento del algoritmo se lo ha expuesto durante el desarrollo de este documento en el capítulo 2, resumiendo en este anexo se basa en hallar las relaciones existentes en los cuestionarios de evaluación, encontrando de esta manera patrones que resuman las respuestas de los estudiantes que han evaluado con seriedad, honestidad, dejando a un lado los datos no acordes a la evaluación por motivos ajenos a la misma.

El proceso resuelve agrupar cuestionarios cercanos entre si, permitiendo observar una uniformidad en las respuestas, ya sean estos con valores bajos o altos para la calificación del docente presentada en varios grupos, la misma que se debe a la diversidad de personalidad que existe en la población que interviene en la evaluación, además se reconoce así los cuestionarios no acorde a las respuestas de la mayoría, se determina un grupo de 8x8 por pruebas realizadas anteriormente y por procesos que se han empleado en otros estudios similares, con estos grupos no se decide la existencia final de los grupos podrían existir una mayor cantidad de ellos, cabe recordar que el método se emplea para hallar dichos grupos en la fase de entrenamiento, se observan las diferencias que existen entre uno y otro, y se determina así cuales son los grupos de patrones del proceso de clasificación.

El proceso de entrenamiento con la encuesta seleccionada a discriminar junto con los formularios evaluados tiene un tiempo considerable de espera, esto depende del mapa construido además del tamaño de la muestra de entrenamiento ingresada.

Luego del proceso de entrenamiento y ya con los patrones determinados se procede a clasificar los cuestionarios evaluados,

Los patrones asociados a las neuronas del mapa de características permite determinar los conglomerados que se encuentran dentro de los cuestionarios evaluados por los estudiantes, estos conglomerados unidos a un factor discriminante que determina el tope máximo de diferencia entre el patrón y el cuestionario evaluado por cada neurona generan los grupos de cuestionarios clasificados en válidos o no válidos, de esta manera se evita que exista dentro de cada conglomerado valores todavía lejanos al patrón relacionado, y se halla los cuestionarios no válidos al grupo que están asociados.

Con el mapa definido y con ayuda de otros métodos que se han involucrado para clasificar, se toma las respectivas decisiones en los resultados presentados en reportes, considerando los resultados de cuestionarios validos y no validos, estas decisiones definen el valor real de ciertos parámetros para nuevos procesos en la calificación de docentes. Con esto se cierra un nuevo ciclo ó permite comenzar otros con la retroalimentación de la información generada por el proceso neuronal. Y que pueden intervenir como parte de una calificación adecuada al docente, dependiendo de esta manera la elección del área administrativa en la toma de decisiones.



**Anexo 9: Limitaciones en el uso del Sistema.**

- El tiempo de procesamiento es excesivo, ya que se trabaja durante días para la fase de entrenamiento.
- El cuestionario que se aplica tiene que ser con preguntas que tienen alternativas con escala likert.

## REFERENCIAS

1. **Palacios Burgos, Francisco José.** Redes Neuronales con GNU/Linux. *Herramientas en GNU/Linux para estudiantes universitarios*. [En línea] 12 de Agosto de 2003. [Citado el: 18 de Diciembre de 2006.] [http://es.tldp.org/Presentaciones/200304curso-glisa/redes\\_neuronales/curso-glisa-redes\\_neuronales-html/](http://es.tldp.org/Presentaciones/200304curso-glisa/redes_neuronales/curso-glisa-redes_neuronales-html/).
2. **Wikipedia, Enciclopedia.** Wikipedia. [En línea] 23 de Agosto de 2006. [Citado el: 18 de Diciembre de 2006.] <http://es.wikipedia.org/wiki/Aprendizaje>.
3. **WordReference.com.** WordReference.com Español. *WordReference.com Español*. [En línea] 01 de Enero de 2005. [Citado el: 01 de Enero de 2007.] <http://www.wordreference.com/definicion/generalizar>.
4. **Hilera, José R y Martínez, Víctor J.** *Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones*. [ed.] ALFAOMEGA S.A. Madrid : RA-MA, 2000.
5. **Wikipedia.** Wikipedia. *Wikipedia*. [En línea] 2 de Diciembre de 2005. [Citado el: 23 de Agosto de 2006.] [http://en.wikipedia.org/wiki/Supervised\\_learning](http://en.wikipedia.org/wiki/Supervised_learning).
6. **Zuluaga A., Camilo M., Acosta B., María Isabel y Salazar I., Harold.** Tutorial de Redes Neuronales. *Tutorial de Redes Neuronales*. [En línea] 30 de Enero de 2000. [Citado el: 30 de Abril de 2005.] <http://ohm.utp.edu.co/neuronales/>.
7. **Wikipedia.** Adaline. *Adaline*. [En línea] 15 de Agosto de 2006. [Citado el: 12 de Diciembre de 2006.] <http://es.wikipedia.org/wiki/Adaline>.
8. **CSCI, Consejo Superior de Investigaciones Científicas.** Características de las Redes Neuronales Artificiales. *Características de las Redes Neuronales Artificiales*. [En línea] 14 de Enero de 2000. [Citado el: 25 de Agosto de 2005.] [http://www.geocities.com/siliconvalley/campus/9756/Tutorial/RNA\\_caracteristicas.htm](http://www.geocities.com/siliconvalley/campus/9756/Tutorial/RNA_caracteristicas.htm).
9. **Babel, Ediciones Torre de.** CONEXIONISMO - PORTAL DE FILOSOFÍA Y PSICOLOGÍA EN INTERNET. *ELEMENTOS DE LAS ARQUITECTURAS CONEXIONISTAS*. [En línea] Madrid, 20 de Enero de 1992. [Citado el: 15 de Agosto de 2005.] <http://www.e-torredebabel.com/Psicologia/Conexionismo/Conexionismo-MecanismosdeProcesamiento.htm>.
10. **Wikipedia.** Ivan Pavlov. *Ivan Pavlov*. [En línea] 25 de Enero de 2000. [Citado el: 20 de Agosto de 2005.] [http://es.wikipedia.org/wiki/Ivan\\_Pavlov](http://es.wikipedia.org/wiki/Ivan_Pavlov).
11. —. Condicionamiento instrumental. *B.F. Skinner*. [En línea] 20 de Enero de 2000. [Citado el: 23 de Agosto de 2005.] [http://es.wikipedia.org/wiki/Condicionamiento\\_instrumental](http://es.wikipedia.org/wiki/Condicionamiento_instrumental).
12. —. ART. *ART*. [En línea] 20 de Enero de 2000. [Citado el: 20 de Diciembre de 2005.] [http://en.wikipedia.org/wiki/Adaptive\\_resonance\\_theory](http://en.wikipedia.org/wiki/Adaptive_resonance_theory).

13. —. Frank Rosenblatt. *Frank Rosenblatt*. [En línea] 12 de Enero de 2004. [Citado el: 20 de Agosto de 2006.] [http://en.wikipedia.org/wiki/Frank\\_Rosenblatt](http://en.wikipedia.org/wiki/Frank_Rosenblatt).
14. **Palmer, A, Montaña, J J y Jiménez, R.** Tutorial sobre Redes Neuronales Artificiales: Los Mapas Autoorganizados de Kohonen. *Psicologia.com*. [En línea] 1 de Enero de 2002. [Citado el: 15 de Enero de 2006.] <http://www.bibliopsiquis.com/psicologiacom/vol6num1/3301/#entren32>.
15. **Kuhn, Thomas Samuel.** *La Estructura de las Revoluciones Científicas*. México : s.n., 1986. pág. 138.
16. **Estévez Valencia, Ph.D Valencia y Achermann Undurraga, Cristóbal.** VISUALIZACIÓN DE DATOS MEDIANTE EL MAPA AUTO-ORGANIZATIVO DE KOHONEN Y ANÁLISIS DE COMPONENTES. *VISUALIZACIÓN DE DATOS MEDIANTE EL MAPA AUTO-ORGANIZATIVO DE KOHONEN Y ANÁLISIS DE COMPONENTES*. [En línea] 25 de Abril de 1998. [Citado el: 20 de Noviembre de 2006.] <http://www2.ing.puc.cl/~iing/ed431/anales/anal02.htm>.
17. **Real Academia, Española.** Real Academia Española. *Real Academia Española*. [En línea] 12 de Diciembre de 2000. [Citado el: 20 de Diciembre de 2006.] <http://buscon.rae.es/draeI/SrvltGUIBusUsual?LEMA=clase>.
18. **Wikipedia.** Mahalanobis. *Mahalanobis*. [En línea] 15 de Enero de 2006. [Citado el: 10 de Agosto de 2006.] [http://es.wikipedia.org/wiki/Distancia\\_de\\_Mahalanobis](http://es.wikipedia.org/wiki/Distancia_de_Mahalanobis).
19. —. Regresión Lineal. [En línea] 01 de 12 de 2004. [Citado el: 15 de Agosto de 2006.] [http://es.wikipedia.org/wiki/Regresi%C3%B3n\\_lineal](http://es.wikipedia.org/wiki/Regresi%C3%B3n_lineal).
20. —. Unified Modeling Language. *UML 2*. [En línea] 12 de Noviembre de 2004. [Citado el: 15 de Octubre de 2006.] [http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language).