



USMP
UNIVERSIDAD DE
SAN MARTIN DE PORRES

**FACULTAD DE
INGENIERÍA Y ARQUITECTURA**

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA ELECTRÓNICO
PARA CLASIFICAR PATRONES DE EBRIEDAD APLICANDO
REDES NEURONALES CON MÉTODOS DE RETRO
PROPAGACIÓN**

**TESIS
PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO**

PRESENTADA POR:

**ALBERCA MUNIVE, LUIS ENRICO
GIBAJA CUSI, EDWIN OMAR**

LIMA - PERÚ

2015

Agradecemos a Dios por esta tesis y se la dedicamos a nuestros padres, hijos y familiares. Gracias por ser la fuente de inspiración para alcanzar nuestros sueños.

ÍNDICE

	Pág.
RESUMEN	xv
ABSTRACT	xvii
INTRODUCCIÓN	xix
CAPÍTULO I. PLANTEAMIENTO DEL PROBLEMA	1
1.1 Situación Problemática	1
1.2 Objetivos	7
1.3 Justificación	8
1.4 Limitaciones	8
1.5 Viabilidad	9
CAPÍTULO II. MARCO TEÓRICO	10
2.1 Niveles de Alcoholemia	10
2.2 Formas de Eliminación del Alcohol	12
2.3 Métodos para Medir el Alcohol en la Sangre	14
2.4 Multas y Sanciones de Tránsito	17
2.5 Antecedentes en la Medición de Patrones de Ebriedad	18

2.6 Redes Neuronales	22
2.7 Filtros Activos	28
2.8 Teorema de Muestreo de Nyquist – Shannon	39
CAPÍTULO III. METODOLOGÍA	40
3.1 Módulos Utilizados	40
3.2 Etapa de Preprocesamiento	53
3.3 Etapa de Procesamiento	70
3.4 Etapa de Postprocesamiento	107
CONCLUSIONES	161
RECOMENDACIONES	163
FUENTES DE INFORMACIÓN	164
ANEXOS	¡Error! Marcador no definido.

ÍNDICE DE TABLAS

	Pág.
Tabla 1. Casusas más Frecuentes de Accidentes de Tránsito en el Año 2013	4
Tabla 2. Efectos del Alcohol a Diferentes Concentraciones	11
Tabla 3. Polinomios de Butterworth en forma factorizada.	38
Tabla 4. Coeficientes para Filtros de Butterworth	59
Tabla 5. Tabla Resumida de Conversión Análoga – Digital	69
Tabla 6. Medición del BAC detectado por un Alcoholímetro Calibrado VS BAC detectado por el sensor MQ3	169
Tabla 7. Cálculo de Sumatorias Requeridas para Determinar el Coeficiente de Correlación de Pearson	172
Tabla 8. Glosario	295

ÍNDICE DE FIGURAS

	Pág.
Fig. 1 Accidentes de Tránsito en el Año 2013	3
Fig. 2 Faltas de Tránsito M01 y M02	17
Fig. 3 Faltas de Tránsito M37, M38 y M39	18
Fig. 4 Sistema de Ignición de Bloqueo Vehicular	19
Fig. 5 Sistema de Reconocimiento Facial	20
Fig. 6 Sistema de Bloqueo Vehicular por Espectroscopia	21
Fig. 7 Expresión Matemática de una Neurona Artificial	23
Fig. 8 Estructura Típica de Red Neuronal Artificial	24
Fig. 9 Topología Monocapa de Red Neuronal	25
Fig. 10 Topología Multicapa de Red Neuronal	26
Fig. 11 Topología Recursiva de Red Neuronal	26
Fig. 12 Red de dos Puertas - Filtro Activo.	28
Fig. 13 Respuesta en Frecuencia de un Filtro Pasa Bajos	29
Fig. 14 Respuesta en Frecuencia de un Filtro Pasa Altos	30
Fig. 15 Respuesta en Frecuencia de un Filtro Pasabanda	31
Fig. 16 Respuesta en Frecuencia de un Filtro Rechaza Banda	33
Fig. 17 Respuesta del Filtro Butterworth	34
Fig. 18 Respuesta del Filtro Chebyshev	35
Fig. 19 Sensor de Etanol MQ3	41

Fig. 20 Sensor de Etanol MQ3 Abierto y Distribución de Pines	42
Fig. 21 Imagen Transversal del Sensor MQ3	42
Fig. 22 Bobina del Tubo Cerámico Caliente	43
Fig. 23 Tubo Cerámico Como Semiconductor	44
Fig. 24 Partículas de Etanol Haciendo Contacto con el Electrodo	44
Fig. 25 Circuito de Calibración del Sensor MQ3	45
Fig. 26 Características de Sensibilidad del Sensor MQ3	46
Fig. 27 Módulo del Sensor MQ3	47
Fig. 28 Módulo del Sensor MQ3 - Vista posterior y Distribución de Pines	48
Fig. 29 Características Principales de ATMEGA328P	50
Fig. 30 Módulo Bluetooth HC-05	52
Fig. 31 Pasabajos SK de 2° Orden	54
Fig. 32 Ventana de Diseño para un Filtro Pasabajos	56
Fig. 33 Topología Sallen-Key Pasa Bajos Butterworth de Orden 4 con Ganancia Unitaria	63
Fig. 34 Topología Sallen-Key Pasa Bajos Butterworth de Orden 4 con Ganancia Unitaria Diseñada para la Etapa de Preprocesamiento	64
Fig. 35 Respuesta en Frecuencia del Filtro Diseñado a la Frecuencia de Corte de 7.5KHz	65
Fig. 36 Respuesta en Frecuencia del Filtro Diseñado a la Frecuencia de Esquina de la Banda de Paso – Aproximadamente $f_{PB}=6.410KHz$	66
Fig. 37 Respuesta en Frecuencia del Filtro Diseñado a la Frecuencia de Esquina de la Banda de Paso - Aproximadamente $f_{SB}=8KHz$	67
Fig. 38 Comparación entre la Señal de Entrada y Salida del Filtro	68
Fig. 39 Arquitectura Perceptron Multicapa	72
Fig. 40 Matrices de Pesos y Umbrales de la capa "C"	73
Fig. 41 Activación de la Neurona "i" de la Capa "C"	73
Fig. 42 Gráfico de la Función Sigmoide	75
Fig. 43 Gráfico de la Función Tangente Hiperbólica	76
Fig. 44 Gráfico en el Hiperplano de la Adaptación de Pesos y Errores en la Red Neuronal	79

Fig. 45 Gráfico Representativo de la Última Capa de una Red Multicapa	80
Fig. 46 Gráfico Representativo de la Penúltima Capa de una Red Multicapa	82
Fig. 47 Activación de la Neurona “J”	85
Fig. 48 Buena Capacidad de Generalización	91
Fig. 49 Mala Capacidad de Generalización	91
Fig. 50 Buen Nivel de Generalización - Error vs Ciclos	92
Fig. 51 Mal Nivel de Generalización – Error vs Ciclos	92
Fig. 52 Topología de La Red Neuronal Diseñada	97
Fig. 53 Función de Error Medio	103
Fig. 54 Matriz de Pesos de la Capa de Entrada	104
Fig. 55 Matriz de Pesos de la Capa Oculta	105
Fig. 56 Matriz de Umbrales de la Capa Oculta	105
Fig. 57 Matriz de Umbrales de la Capa de Salida	106
Fig. 58 Conexión entre ATMEGA328P y el Módulo Bluetooth	107
Fig. 59 ATMEGA328P - Importación de la Librería “math.h”	108
Fig. 60 ATMEGA328P - Importación de la Librería “LiquidCrystal.h” y Configuración del LCD 16x2	108
Fig. 61 ATMEGA328P - Definición de las Variables Globales	109
Fig. 62 ATMEGA328P - Función “Setup ()”	110
Fig. 63 ATMEGA328P - Inicio de la Función “loop ()”	111
Fig. 64 Aplicación Android - Menú Principal	112
Fig. 65 ATMEGA328P - Recepción de Datos desde la Aplicación en Android	113
Fig. 66 ATMEGA328P - Inicio de la Función “medir_alcohol ()”	114
Fig. 67 Conexión entre el Transistor NPN PN2222A y el Sensor MQ3	114
Fig. 68 ATMEGA328P - Código de Calentamiento del Sensor MQ3	115
Fig. 69 Aplicación Android – Mensaje de Calentamiento del Sensor MQ3	116
Fig. 70 ATMEGA328P - Calentamiento y Cálculo del Voltaje Máximo del Sensor MQ3	117
Fig. 71 Aplicación Android – Mensaje de Medición del BAC	118
Fig. 72 ATMEGA328P - Normalización del Voltaje Máximo Calculado del Sensor MQ3	119

Fig. 73 ATMEGA328P - Función “map_double ()”	119
Fig. 74 ATMEGA328P - Función “Cuantificar_Current_BAC ()”	120
Fig. 75 ATMEGA328P - Ejecución de la Función “BAC_to_a1 ()” con la Variable “Cuant_Current_BAC” como Parámetro de Entrada	120
Fig. 76 ATMEGA328P - Función “BAC_to_a1 ()”	121
Fig. 77 ATMEGA328P - Función “int2bin ()”	122
Fig. 78 ATMEGA328P - Cálculo del Vector de Activación “a1”	122
Fig. 79 ATMEGA328P - Ejecución de la Función “red_neuronal ()”	123
Fig. 80 ATMEGA328P - Declaración de la Matriz de Pesos “W1”	124
Fig. 81 ATMEGA328P - Declaración de la Matriz de Pesos “W2”	125
Fig. 82 ATMEGA328P - Declaración de la Matriz de Umbrales “U2”	125
Fig. 83 ATMEGA328P - Declaración de la Matriz de Umbrales “U3”	125
Fig. 84 ATMEGA328P - Declaración del Vector de Activaciones de la Capa Oculta “a2”	126
Fig. 85 ATMEGA328P - Declaración del Vector de Salida “y”	126
Fig. 86 ATMEGA328P - Declaración de Variables para el Cálculo del Producto Matricial “a1xW1”	127
Fig. 87 ATMEGA328P - Cálculo del Producto Matricial “a1xW1”	128
Fig. 88 ATMEGA328P - Cálculo de “a1xW1 + U2”	129
Fig. 89 ATMEGA328P - Cálculo del Vector de Activaciones de la Capa Oculta “a2”	129
Fig. 90 ATMEGA328P - Función “sigmoidal ()”	130
Fig. 91 ATMEGA328P - Declaración de Variables para Calcular el Producto Matricial “a2xW2”	131
Fig. 92 ATMEGA328P - Cálculo del Producto Matricial “a2xW2”	132
Fig. 93 ATMEGA328P - Cálculo de “a2xW2 + U3”	133
Fig. 94 ATMEGA328P - Cálculo del Vector de Salida “y”	133
Fig. 95 ATMEGA328P - Clasificación del Nivel de Alcholemla	134
Fig. 96 ATMEGA328P - Clasificación del Tipo de Multa	135
Fig. 97 ATMEGA328P - Envío de la Clasificación del Nivel de Alcholemla y del Tipo de Multa a la Aplicación Android	136

Fig. 98 Aplicación Android - Resultados de Clasificación de la Red Neuronal	137
Fig. 99 Aplicación Android - Diseño	139
Fig. 100 Aplicación Android - Declaración de Variables Globales	140
Fig. 101 Aplicación Android - Cargado de Direcciones y Nombres de Dispositivos Bluetooth Conocidos en la Lista Desplegable "ListPicker1"	141
Fig. 102 Aplicación Android - Conexión con el Módulo Bluetooth HC-05	141
Fig. 103 Aplicación Android - Evento de Cambio de Estado del "CheckBox1" - X1: Tipo de Transporte - Particular=0	143
Fig. 104 Aplicación Android - Evento de Cambio de Estado del "CheckBox2" - X1: Tipo de Transporte - Público=1	145
Fig. 105 Aplicación Android - Evento de Cambio de Estado del "CheckBox3" - X2: ¿El Conductor Participó en un Accidente de Tránsito? - Si=1	146
Fig. 106 Aplicación Android - Evento de Cambio de Estado del "CheckBox4" - X2: ¿El Conductor Participó en un Accidente de Tránsito? - No=0	148
Fig. 107 Aplicación Android - Evento de Cambio de Estado del "CheckBox5" - X11: ¿El Accidente Dejó Muertos o Heridos? - Si=1	149
Fig. 108 Aplicación Android - Evento de Cambio de Estado del "CheckBox6" - X11: ¿El Accidente Dejó Muertos o Heridos? - No=0	150
Fig. 109 Aplicación Android - Evento Pulsar Botón "INICIAR(Button1)"	151
Fig. 110 Aplicación Android - Función "Disable_Options"	152
Fig. 111 Aplicación Android - Evento Disparar Timer "Calentamiento_de_Sensor_MQ3"	153
Fig. 112 Aplicación Android - Evento Disparar Timer "Medición_de_BAC"	154
Fig. 113 Aplicación Android - Evento Disparar Timer "Bluetooth_Capture_String" - Recepción de Datos del Módulo Bluetooth del Microcontrolador	155
Fig. 114 Aplicación Android - Clasificación "Todo en Orden"	156
Fig. 115 Aplicación Android - Función "Restart_Values"	157
Fig. 116 Aplicación Android - Clasificación de Multas y Sanciones de Tipo 1	158
Fig. 117 Aplicación Android - Clasificación de Multas y Sanciones de Tipo 2	158
Fig. 118 Aplicación Android - Clasificación de Multas y Sanciones de Tipo 3	159
Fig. 119 Aplicación Android - Clasificación de Multas y Sanciones de Tipo 4	159

Fig. 120 Aplicación Android - Clasificación de Multas y Sanciones de Tipo 5	160
Fig. 121 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Principal	175
Fig. 122 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Izquierda – Parte 1	176
Fig. 123 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Izquierda – Parte 2	177
Fig. 124 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Izquierda – Parte 3	178
Fig. 125 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Izquierda – Parte 4	179
Fig. 126 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Izquierda – Parte 5	180
Fig. 127 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Izquierda – Parte 6	181
Fig. 128 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Derecha – Parte 1	182
Fig. 129 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Derecha – Parte 2	183
Fig. 130 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Derecha – Parte 3	184
Fig. 131 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Derecha – Parte 4	185
Fig. 132 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Derecha – Parte 5	186
Fig. 133 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Derecha – Parte 6	187
Fig. 134 Código en Matlab del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal – Parte 1	188
Fig. 135 Código en Matlab del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal – Parte 2	189

Fig. 136 Código en Matlab del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal – Parte 3	190
Fig. 137 Código en Matlab del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal – Parte 4	191
Fig. 138 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Principal – Parte 1	270
Fig. 139 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Principal – Parte 2	271
Fig. 140 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Principal – Parte 3	271
Fig. 141 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Principal – Parte 4	272
Fig. 142 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Principal – Parte 5	272
Fig. 143 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Principal – Parte 6	273
Fig. 144 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 1	274
Fig. 145 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 2	275
Fig. 146 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 3	276
Fig. 147 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 4	277
Fig. 148 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 4	278
Fig. 149 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 5	279
Fig. 150 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 6	280

Fig. 151 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Central – Parte 1	281
Fig. 152 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Central – Parte 2	282
Fig. 153 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Derecha – Parte 1	283
Fig. 154 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Derecha – Parte 2	284
Fig. 155 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Derecha – Parte 3	285
Fig. 156 Código en Matlab del Algoritmo de Retropropagación – Parte 1	286
Fig. 157 Código en Matlab del Algoritmo de Retropropagación – Parte 2	287
Fig. 158 Código en Matlab del Algoritmo de Retropropagación – Parte 3	288
Fig. 159 Código en Matlab del Algoritmo de Retropropagación – Parte 4	289
Fig. 160 Código en Matlab del Algoritmo de Retropropagación – Parte 5	290
Fig. 161 Código en Matlab del Algoritmo de Retropropagación – Parte 6	291
Fig. 162 Diagrama de Flujo General de la Aplicación	292
Fig. 163 Diagrama Circuitual Completo	293
Fig. 164 PCB Completo	294

RESUMEN

Esta tesis tiene por objetivo presentar una alternativa innovadora, viable, fácil de utilizar y portable para clasificar patrones de ebriedad, la justificación de esta investigación está en la alta tasa de accidentes que se presentan en el país a consecuencia de conductores ebrios.

Hoy en el mercado peruano existen equipos de medición de nivel de etanol en la sangre de una persona, sin embargo, el costo de estos equipos tiene un valor aproximado de 4 o 5 veces mayor al equipo que se emplea en este proyecto; adicionalmente, estos no realizan una clasificación del nivel de etanol en la sangre de la persona para que sea tomada como referencia por un agente policial.

En esta tesis se diseña un sistema electrónico para clasificar patrones de ebriedad usando redes neuronales con el método de retropropagación, el cual permite obtener resultados rápidos y efectivos del grado de ebriedad de un conductor. Para las simulaciones y comparaciones se emplea un equipo comercial de medición de etanol en la sangre humana y la implementación del prototipo de tesis consiste en un sensor MQ3 que está implementado en un tarjeta electrónica que va convirtiendo los diferentes voltajes producidos en la

entrada de este sensor en señales digitales que son filtradas por un filtro de cuarto orden Butterworth para garantizar la reconstrucción exacta de la señal periódica inducida continua en banda base a partir de sus muestras, y luego se clasifica el nivel de etanol inducido.

Palabras claves: retropropagación, clasificación, patrones de ebriedad, MQ3, Butterworth.

ABSTRACT

This thesis aims to present an innovative, viable, easy to use and portable alternative to classify drinking patterns; the rationale for this investigation is the high rate of accidents that occur in the country as a result of drunk drivers.

Today in the Peruvian market there are equipment for measuring the level of ethanol in the blood of a person, however, the cost of these equipment has an approximate value of 4 or 5 times higher than the equipment used in this project; additionally, they do not perform a classification of level of ethanol in the blood of a person to be taken as a reference by a police officer.

In this thesis an electronic system is designed to classify drinking patterns using neural networks with the backpropagation method, which allows obtaining fast and effective results of the degree of drunkenness of a driver. For simulations and comparisons, a commercial equipment for measuring ethanol in human blood is used and the thesis prototype implementation consists of an MQ3 sensor that is implemented in an electronic card that turn different voltages produced at the entrance of this sensor into digital signals that are filtered by a fourth order Butterworth filter to ensure the exact reconstruction of the periodic

continuous induced signal in base band from its samples, and then the level of induced ethanol is classified.

Keywords: backpropagation, classification, drunken patterns, MQ3, Butterworth.

INTRODUCCIÓN

En la presente tesis se diseñó un sistema electrónico para la clasificación de patrones de ebriedad utilizando redes neuronales con el método de retropropagación, para lo cual se utilizó un sensor de etanol MQ3 como entrada para el circuito implementado y un módulo bluetooth para la publicación de los resultados de la clasificación en una aplicación móvil.

En el Capítulo 1, se explica la problemática o necesidad a la cual responde la elaboración de esta tesis, el origen del proyecto, los objetivos generales y los objetivos específicos.

En el Capítulo 2, se muestran los antecedentes nacionales e internacionales relacionados a la medición del nivel de alcohol en seres humanos, se muestra el estado del arte global de la tecnología y el marco teórico utilizado, el cual se compone principalmente de la teoría y diseño de filtros activos y de redes neuronales artificiales

En el Capítulo 3, se describe la metodología utilizada para el desarrollo de la presente investigación, los módulos utilizados, la etapa de preprocesamiento la cual consta del diseño un filtro pasabajos Butterworth de orden 4 con topología Sallen-Key con ganancia unitaria para poder cumplir con el teorema de muestreo de Nyquist y filtrar las frecuencias superiores a 7Khz en banda base

de la señal analógica del sensor MQ3, la etapa de procesamiento donde se diseñó la topología de la red neuronal y se calcularon los valores de las matrices de pesos y umbrales a través del algoritmo de retropropagación desarrollado en Matlab y finalmente, la etapa de postprocesamiento donde se desarrolló e implementó el algoritmo de clasificación en un microcontrolador ATMEGA328P y la publicación de los resultados en una aplicación móvil para Android.

En los anexos, se muestran los resultados experimentales, comparaciones del sistema electrónico desarrollado en contraste con un alcoholímetro de referencia, los diagramas de flujo de los programas desarrollados, la matriz de entrada y salida de la red neuronal, el diagrama circuital, el PCB y el glosario de términos.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Situación Problemática

Actualmente el consumo del alcohol tiene un alto índice de accidentes automovilísticos, de acuerdo a las estadísticas proporcionadas por el Ministerio de Transportes y Comunicaciones y por la policía de tránsito, como consecuencia hay un alto índice de pérdida de vidas, familias desamparadas, niños huérfanos, siendo este problema uno de los más grandes peligros que afrontan casi todas las ciudades del mundo y de manera especial los países de América del Sur donde no se capacita a la ciudadanía en los conocimientos básicos de una cultura vial.

“En el Perú el número de personas fallecidas por accidentes de tránsito se incrementaron en un 25% entre enero y marzo del 2014” (El Comercio, 2014).

Así mismo se puede citar los países de Chile, Colombia, Argentina, etc. donde los índices de muertes por esta problemática son significativos.

En esta investigación se ha confrontado de una manera analítica los resultados de la problemática que se obtienen anualmente en el país y que año a año va incrementándose sin que se puedan tomar medidas y políticas de decisión basadas en el uso de la tecnología científica para luchar contra esta realidad que ocasiona graves problemas a la colectividad del país, como se muestra en la siguiente información obtenida:

El exceso de velocidad y el alcohol es la mezcla fatal que, cada año, sigue cobrando miles de vidas en las pistas del país. En este año han ocurrido más de 76 mil 700 accidentes de tránsito, los cuales dejaron 2,430 muertos y 44,277 heridos.

Así lo informó el Consejo Nacional de Seguridad Vial del Ministerio de Transportes y Comunicaciones (MTC), al detallar que el 74% de hechos fatales se produjeron en vías ubicadas en zonas urbanas y el 26% ocurrieron en carreteras nacionales y regionales.

Al respecto, el general PNP® Enrique Medri, director del mencionado consejo nacional, remarcó que el 82% de los siniestros fatales fue por fallas humanas.

Así, entre las causas más frecuentes figuran el exceso de velocidad (34%), la imprudencia del conductor (28%), conducir en estado de ebriedad (12%) y la imprudencia de los peatones (8%). (Perú 21, 2013)

Con la información del artículo citado anteriormente, se elaboró el siguiente cuadro:

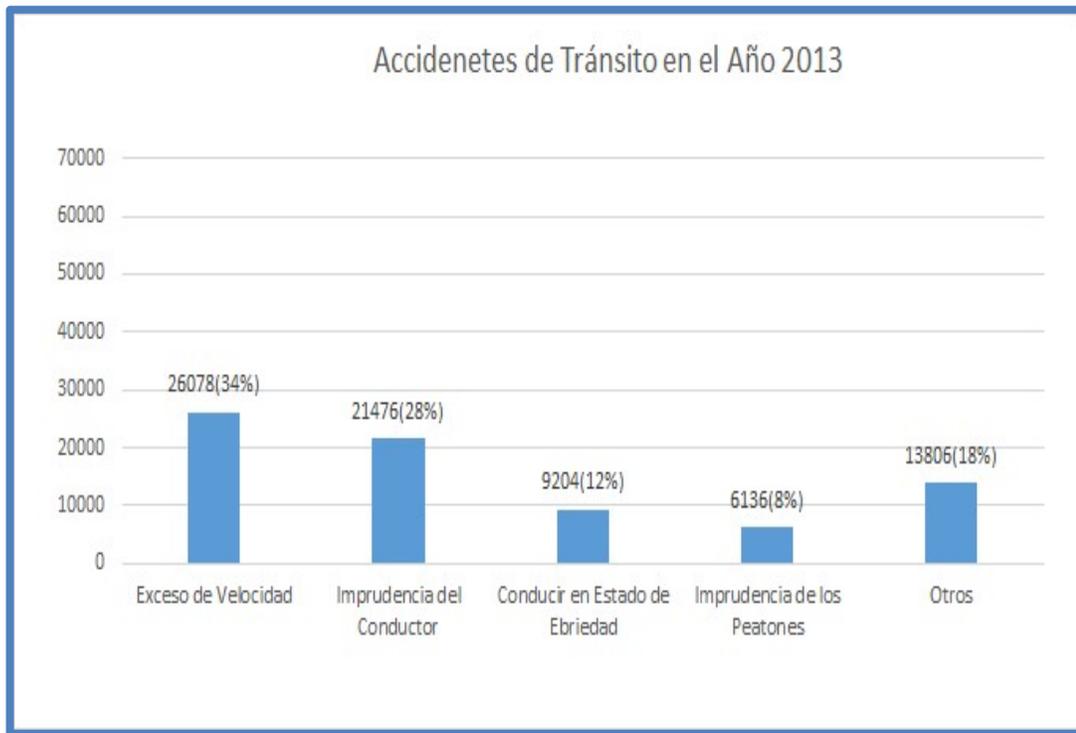


Fig. 1 Accidentes de Tránsito en el Año 2013

Elaboración: los autores

Fuente: Perú 21, 2013

De lo cual se obtuvieron las siguientes estadísticas:

Tabla 1. Casusas más Frecuentes de Accidentes de Tránsito en el Año 2013

Causas de Accidentes de Tránsito	Frecuencia	% Relativo
Exceso de Velocidad	26078	34.00%
Imprudencia del Conductor	21476	28.00%
Conducir en Estado de Ebriedad	9204	12.00%
Imprudencia de los Peatones	6136	8.00%
Otros	13806	18.00%
TOTAL	76700	100.00%

Elaboración: los autores

Se puede observar según el cuadro que el total de accidentes de tránsito ocurridos el año pasado por conductores en estado de ebriedad es de 9204 accidentes lo cual representaron el 12% del total, lo cual es una cifra significativa de muertes por esta causa.

En cuanto a cifras más cercanas al presente año, se tiene la siguiente información:

En los primeros tres meses del año, unas 550 personas han fallecido o quedado heridas de gravedad en diferentes accidentes de tránsito ocurridos en todo el país, informó el presidente de la Asociación de Víctimas de Accidentes de Tránsito, Carlos Villegas. Sostuvo que, si la tendencia de accidentes por mes se mantiene, no es difícil estimar que a diciembre se superará la cifra de fallecidos contabilizados a finales del 2013, cuando hubo un total de 3,590 muertes. (Perú 21, 2014)

Para finalizar, en cuanto a las cifras de accidentes de tránsito provocados por la imprudencia y ebriedad del conductor se tiene:

El exceso de velocidad ocasionó 25 mil 142 sucesos lamentables frente a los 23 mil 032 acontecidos entre enero y septiembre del año pasado, según informó el Organismo Peruano de Consumidores y Usuarios (Opecu). El 87% de los accidentes de tránsito a nivel nacional se debe principalmente a la imprudencia, exceso de velocidad y ebriedad del conductor, así informó el Organismo Peruano de Consumidores y Usuarios (Opecu) que instó a las autoridades competentes a intensificar su fiscalización en las celebraciones de Año Nuevo. (Publimetro, 2013)

Por los artículos anteriores se puede notar que una de las causas principales para un alto índice de accidentes de tránsito y por ende pérdida de vidas humanas en los últimos años, es debido a los conductores en estado etílico, por lo cual en la presente investigación se dará una solución que permita detectar y clasificar los patrones de ebriedad, de manera más efectiva que los métodos por alcoholímetro utilizados en la actualidad en el Perú.

1.1.1 Definición del Problema

A continuación, se describe el problema principal y los problemas secundarios:

1.1.1.1 Problema Principal

¿Cómo diseñar e implementar un sistema electrónico que permita clasificar patrones de ebriedad?

1.1.1.2 Problemas Secundarios

- ¿Cómo medir la concentración de alcohol en la sangre mediante un sensor electrónico?
- ¿Cómo y bajo que especificaciones realizar la conversión análoga-digital de la señal obtenida del sensor utilizado?
- ¿Cómo modelar matemáticamente, diseñar e implementar los filtros apropiados en la etapa de preprocesamiento?
- ¿Cómo diseñar la arquitectura de la red neuronal para la clasificación de patrones de ebriedad?
- ¿Cómo implementar un algoritmo propio de retropropagación para la correcta clasificación de patrones de ebriedad en la red neuronal?
- ¿Cómo implementar un algoritmo propio para la publicación de los resultados de clasificación en una aplicación móvil para sistema operativo Android?
- ¿Cómo medir la robustez de la solución propuesta?

1.2 Objetivos

A continuación, se describe el objetivo principal y los objetivos específicos:

1.2.1 Objetivo General

El objetivo de esta tesis es diseñar e implementar un sistema electrónico para clasificar patrones de ebriedad a través de redes neuronales de múltiple capa con algoritmo de descenso de gradiente o retro-propagación.

Por medio de una aplicación en Android se mostrarán los resultados de clasificación del proceso neuronal.

1.2.2 Objetivos Específicos

- Medir la concentración de alcohol en la sangre mediante un sensor electrónico
- Definir cómo y bajo que especificaciones realizar la conversión análogo-digital de la señal obtenida del sensor utilizado.
- Modelamiento matemático, diseño e implementación de filtros en la etapa de preprocesamiento para la obtención de data limpia.
- Diseño de la arquitectura de la red neuronal para la clasificación de patrones de ebriedad.
- Implementar un algoritmo propio de retropropagación para la clasificación de patrones de ebriedad en la red neuronal.

- Implementar un algoritmo propio para la publicación de los resultados de clasificación en una aplicación móvil para sistema operativo Android
- Comparación y validación de resultados obtenidos con un alcoholímetro comercial de referencia, para medir la robustez de la solución propuesta.

1.3 Justificación

El desarrollo de la presente tesis se justifica porque tiene un impacto e importancia social; es innovador porque en el Perú no se cuenta con un sistema electrónico que permita clasificar los patrones de ebriedad de manera portátil.

1.4 Limitaciones

Las limitaciones para la presente investigación son las siguientes:

a) Limitación Espacial

Lima Metropolitana.

b) Limitación Social

Población en general.

c) Limitación Temporal

La presente tesis abarca el periodo comprendido entre el mes de enero del 2013 a diciembre del 2014.

1.5 Viabilidad

La viabilidad para llevar a cabo la presente investigación, se detalla a continuación:

a) Viabilidad Técnica

Técnicamente la investigación es totalmente viable porque cuenta con módulos y dispositivos electrónicos que son muy utilizados y que se pueden adquirir fácilmente en el mercado.

b) Viabilidad Económica

En el aspecto económico el proyecto es viable en su totalidad ya que los dispositivos electrónicos requeridos tienen un bajo costo en el mercado.

c) Viabilidad Social

La investigación aporta una gran ayuda, al brindar una solución innovadora para clasificar los patrones de ebriedad, abriendo un nuevo camino para una detección más óptima de dichos patrones y por ende dando la posibilidad de reducir los accidentes de tránsito, así como también es un aporte a la industria electrónica del Perú.

d) Viabilidad Operativa

Operativamente la investigación es completamente viable, ya que se utilizó diseños electrónicos y algoritmos propios; así como plataformas de desarrollo de software libre como el APPInventor2.

e) Alternativas

Las alternativas que se utilizaron para la adaptación del sistema en la colectividad de vehículos privados serán la difusión y conocimiento a través de las redes sociales.

CAPÍTULO II

MARCO TEÓRICO

2.1 Niveles de Alcoholemia

Cómo se vio en el primer capítulo, una de las causas más frecuentes de accidentes de tránsito es conducir con exceso de alcohol en la sangre.

El consumo de alcohol es un factor causal en más de 200 enfermedades y trastornos. Está asociado con el riesgo de desarrollar problemas de salud tales como trastornos mentales y comportamentales, incluido el alcoholismo, importantes enfermedades no transmisibles tales como la cirrosis hepática, algunos tipos de cáncer y enfermedades cardiovasculares, así como traumatismos derivados de la violencia y los accidentes de tránsito. (OMS, 2018)

En la presente investigación es de importancia definir la clasificación de los síntomas y signos que presenta un individuo ante los diferentes niveles de alcoholemia, para poder entrenar la red neuronal con dichos valores.

A continuación, se muestra una tabla de los efectos del alcohol a diferentes concentraciones, dicha tabla se tomará como referencia para la clasificación del nivel de alcoholemia que proporcionará la red neuronal.

Tabla 2. Efectos del Alcohol a Diferentes Concentraciones

Concentración (gr/L)	Estado clínico	Síntomas y signos
0.00 a 0.29	Sobrio	No presenta síntomas aparentes
0.30 a 0.99	Euforia	Disminución de inhibiciones, sociabilidad, autoconfianza, brusquedad, enlentecimiento de reacciones, ataxia.
1.00 a 1.49	Excitación y/o Embriaguez	Inestabilidad emocional, disminución de atención, juicio, control y percepciones sensoriales, sobrevaloración de capacidades propias.
1.50 a 1.99	Confusión	Trastorno de memoria y comprensión, desorientación, descoordinación muscular y somnolencia.
2.00 a 2.99	Estupor	Disminución de la conciencia, apatía, inercia, déficits motores.
3.00 a 3.99	Intoxicación Severa	Inconciencia, anestesia, disminución de reflejos.
>=4.00	Coma y/o posible muerte	Dificultades cardíacas y respiratorias, hipotermia, hipoglucemia, convulsiones y parálisis respiratoria.

Elaboración: los autores

Fuente: Lorenzo Fernández, Ladero, Leza Cerro, & Lizasoain Hernández, 2009, págs. 335-351

2.2 Formas de Eliminación del Alcohol

A continuación, se describen las diferentes formas de la eliminación del alcohol en el cuerpo humano, dicha teoría fue obtenida de la investigación del Magister en Toxicología César Augusto Canales Martínez de la Universidad Nacional Mayor de San Marcos.

La eliminación de alcohol etílico sin cambios en su estructura es aproximadamente un 10% del alcohol etílico absorbido y se va a eliminar por el aliento, la saliva, las heces, orina, sudor y la leche. Como la mayor parte de alcohol etílico absorbido se oxida, la eliminación es pulmonar (50 – 60%), entero hepático (25 – 30%), renal (5 – 7%) y el resto se elimina en pequeñas cantidades en sudor, lágrimas jugo gástrico, saliva y leche materna. (Canales Martínez, 2011, p.14)

a) Eliminación Pulmonar

La información que se tiene acerca de la eliminación del alcohol por vía pulmonar es la siguiente:

Esta vía de excreción, posible gracias a la volatilidad del alcohol, sigue un proceso inverso al de la absorción. Desde el punto de vista analítico y judicial es de gran importancia, pues los métodos de análisis incruentos (no sangrientos) se basan en este principio: Determinación de alcohol en el aire espirado. Se ha calculado que el alcohol presente en 2,000 mL de aire espirado equivale al que hay en 1 mL de sangre. (Canales Martínez, 2011, p.14)

b) Eliminación Urinaria

Con respecto a la eliminación del alcohol por vía urinaria, se tiene la siguiente información:

El alcohol difunde a través del glomérulo y no sufre proceso de reabsorción tubular. La concentración de alcohol en la orina dependerá de la alcoholemia, pero esta cambia continuamente y la de la orina no lo hace, la relación alcoholemia/alcoholuria no es 1, sino inferior. La concentración de alcohol en la vejiga de la orina reflejará la existente en la sangre durante el periodo medio del tiempo, pero teniendo en cuenta que su riqueza en agua es mayor y no se degrada, el cociente alcoholuria/alcoholemia puede variar según los autores teniendo una media de 1,3 a 1,44 cifra que está adoptada por la legislación inglesa para hacer la equiparación alcoholuria/alcoholemia. (Canales Martínez, 2011, p.15)

c) Eliminación por la Saliva

Con respecto a la eliminación del alcohol por la saliva, se sabe que “El alcohol se elimina por la saliva, aunque en cantidad secretada por esta vía es ínfima; con todo, dado el volumen de secreción salival, tiene el mismo interés analítico que la orina” (Canales Martínez, 2011, p.15).

d) Eliminación por la Leche

La eliminación del alcohol por la, leche se da de la siguiente manera “El alcohol se elimina asimismo por esta secreción, lo que debe ser tenido en cuenta por las madres lactantes” (Canales Martínez, 2011, p.15).

2.3 Métodos para Medir el Alcohol en la Sangre

Los sistemas de análisis comúnmente utilizados son en sangre, en orina, en saliva y en aliento:

2.3.1 Detección de Alcohol en Sangre

Este se considera el método más exacto para medir el grado de alcohol en la sangre, consiste en llevar la muestra de sangre del individuo a un laboratorio y que se determine los gramos de alcohol por litro de sangre de dicha muestra.

De acuerdo con Andrade Spatz (2009) “Está ampliamente demostrada la relación directa entre la concentración de alcohol en la sangre y el grado en que las reacciones y las decisiones se ven afectadas” (p.50). El mismo autor sostiene acerca de la concentración de alcohol en la sangre:

El BAC (concentración de alcohol en sangre) se considera la medida habitual para medir los niveles en una persona que se encuentra bajo la influencia del alcohol. Se mide la concentración en la sangre y se estima la concentración dentro de las células, asumiendo que se llega a una concentración de equilibrio en la interface sangre/célula. Es el examen legal más exacto que existe actualmente, pero presenta inconvenientes: el procedimiento es demasiado agresivo y caro, no es inmediato, requiere personal especializado y el traslado a un centro médico de análisis. Además, la muestra puede contaminarse en el proceso de extracción, transporte o almacenamiento. (p.50)

2.3.2 Detección de Alcohol en Orina

De acuerdo con Andrade Spatz (2009) "El test de orina indica la presencia de alcohol en el organismo, pero no indica el estado actual de la persona ni el contenido exacto de alcohol en la sangre". (p.51)

2.3.3 Análisis de Alcohol en la Saliva

Aunque se cree que puede existir una relación entre la concentración de alcohol en la sangre y la concentración de alcohol en la saliva, todavía no se ha podido encontrar la tecnología ni la reacción química que lo demuestre con exactitud y fiabilidad. (Andrade Spatz, 2009, p.52).

2.3.4 Análisis de Alcohol en el Aliento

Este método es el más utilizado actualmente en el Perú, consiste en soplar en un dispositivo el cuál mediante el cálculo de las partículas de etanol en el aliento exhalado determinará la concentración de alcohol en la sangre, ya que es muy sabido que existe una relación directa entre ambos.

Para determinar el efecto que puede tener el alcohol ingerido sobre la capacidad de conducir de una persona (que depende de la concentración de etanol en el cerebro), se mide la concentración de etanol en el aire exhalado. La concentración de etanol en el aire exhalado está en equilibrio con la que se encuentra en la sangre y ésta, a su vez, está en equilibrio con la que se presenta en el cerebro. (Andrade Spatz, 2009, p.52)

De acuerdo con (Andrade Spatz, 2009, p.52-53) el análisis de alcohol en el aliento presenta las siguientes ventajas:

- No es una prueba invasiva.
- Es más fácil, seguro y rápido obtener una muestra del aliento de una persona que una muestra de sangre o de orina.
- El resultado se obtiene de forma inmediata, a diferencia del tiempo que presenta un análisis de sangre o de orina.
- Es más económico tomar una muestra de aliento, y la probabilidad de alterar la muestra es nula.

La presente investigación optará por medir el etanol a través del aliento, por las ventajas expuestas anteriormente, para lo cual usaremos una nariz electrónica MQ3-Etanol Sensor.

2.4 Multas y Sanciones de Tránsito

Las faltas relacionadas con accidentes de tránsito y el manejar con exceso de alcohol en la sangre, de acuerdo con la tabla de infracciones del reglamento nacional de tránsito, obtenido de la página de la SAT (Servicio de Administración Tributaria de Lima) son M01, M02, M37, M38 y M39, cuyo detalle se muestra en las siguientes figuras:

REGLAMENTO NACIONAL DE TRÁNSITO
Decreto Supremo N° 016-2009-MTC. Texto Único Ordenado del Reglamento Nacional de Tránsito – Código de Tránsito y modificatorias.

Falta	Infracción	Calificación	Monto S/	Con descuento (*)	Sanción	Puntos	Medida Preventiva	Solidario
M01	Conducir con presencia de alcohol en la sangre en proporción mayor a lo previsto en el Código Penal, o bajo los efectos de estupefacientes, narcóticos y/o alucinógenos comprobado con el examen respectivo o por negarse al mismo y que haya participado en un accidente de tránsito.	Muy Grave	4,150.00	4,150.00	Multa y cancelación de la licencia de conducir e inhabilitación definitiva para obtener licencia	0	Internamiento del Vehículo y Retención de la Licencia	Propietario
M02	Conducir con presencia de alcohol en la sangre en proporción mayor a lo previsto en el Código Penal, bajo los efectos de estupefacientes, narcóticos y/o alucinógenos comprobada con el examen respectivo o por negarse al mismo.	Muy Grave	2,075.00	2,075.00	Multa y suspensión de la licencia de conducir por tres (3) años	0	Internamiento del Vehículo y Retención de la Licencia	Propietario

Fig. 2 Faltas de Tránsito M01 y M02

Fuente: Tabla de Infracciones del Reglamento Nacional de Tránsito – SAT.
(Decreto Supremo N° 016-2009-MTC, 2009)

Código	Descripción	Gravedad	Multa	Penalización	Penalización	Afectados	
M37	Conducir y ocasionar un accidente de tránsito con daños personales inobservando las normas de tránsito dispuestas en el presente Reglamento.	Muy Grave	0.00	0.00	Suspensión de la licencia de conducir por un (1) año	50 Internamiento del Vehículo y Retención de la Licencia	Propietario
M38	Conducir un vehículo para el servicio de transporte público y ocasionar un accidente de tránsito con daños personales inobservando las normas de tránsito dispuestas por el presente Reglamento.	Muy Grave	0.00	0.00	Suspensión de la licencia de conducir por tres (3) años	0 Internamiento del Vehículo y Retención de la Licencia	Propietario
M39	Conducir y ocasionar un accidente de tránsito con lesiones graves o muerte inobservando las normas de tránsito dispuestas en el presente Reglamento.	Muy Grave	0.00	0.00	Cancelación e inhabilitación definitiva del conductor para obtener una licencia de conducir	0 Internamiento del Vehículo y Retención de la Licencia	Propietario

Fig. 3 Faltas de Tránsito M37, M38 y M39

Fuente: Tabla de Infracciones del Reglamento Nacional de Tránsito – SAT.
(Decreto Supremo N° 016-2009-MTC, 2009)

2.5 Antecedentes en la Medición de Patrones de Ebriedad

A continuación, se describirán algunos antecedentes de otras soluciones para medir patrones de ebriedad.

2.5.1 Dispositivo de Ignición

Una de los mecanismos más conocidos que se utilizan para impedir que conductores en estado etílico puedan manejar un auto son los llamados dispositivos de ignición o dispositivos de interbloqueo de ignición.

Un interbloqueo de ignición es un alcoholímetro que va instalado en tu vehículo para medir los niveles de alcohol en tu aliento. Está conectado directamente al sistema de ignición de tu vehículo y requiere que soples en la boquilla adjunta para verificar la Concentración de Alcohol en el Aliento (BrAC, por sus siglas en inglés). Si tu concentración de alcohol excede un nivel legal determinado, el dispositivo temporalmente previene que tu vehículo encienda. (Draeger Inc., 2015)



Fig. 4 Sistema de Ignición de Bloqueo Vehicular

Fuente: Draeger Inc., 2015

2.5.2 Reconocimiento Facial

El Automotive Research & Testing Center de Taiwan, desarrolló el 2010 un sistema que utiliza técnicas de procesamiento de imágenes y señales inerciales para detectar conductores ebrios, distracciones del conductor y fatiga del conductor. “Integrando una calculadora de concentración de alcohol en el aliento y una técnica de reconocimiento facial, el sistema provee un mecanismo elegante para evitar conductores ebrios” (Automotive Research & Testing Center Corporation, 2010).

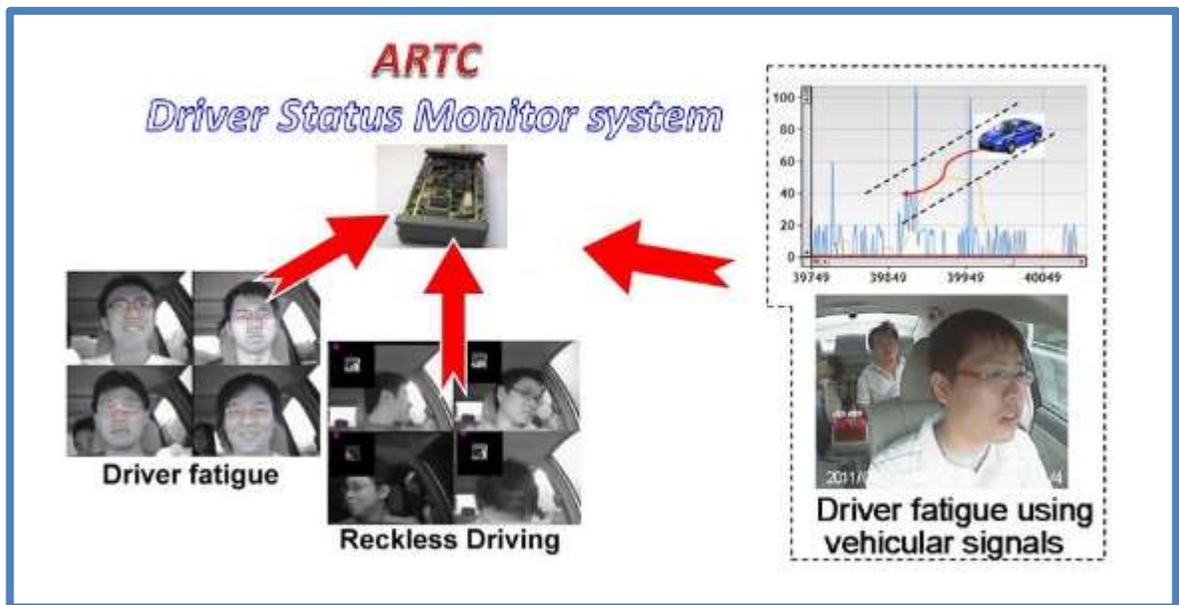


Fig. 5 Sistema de Reconocimiento Facial

Fuente: Automotive Research & Testing Center Corporation, 2010

2.5.3 Sistema de Detección de Alcohol Usando Espectroscopia

Un dispositivo, llamado “Drive Alcohol Detection System for Safety (DADSS)”. Está diseñado para inmovilizar el automóvil de una persona en estado etílico. Este nuevo sistema está basado en dos métodos: En la espectroscopia de tejidos, ya que puede detectar los niveles de alcohol en la sangre al colocar un dedo en una pantalla táctil, no muy diferente de un escáner de huellas dactilares. El otro método es como un soplador-analizador, sin embargo, no es muy similar a los sistemas actuales que requieren una a soplar en un tubo. En lugar de ello, el sistema funciona utilizando la espectroscopia distante; pudiendo detectar moléculas específicas asociadas con el alcohol en el aliento. El sensor de detección de esas partículas funciona a través de infrarrojos. Básicamente, el sistema será capaz de detectar la respiración de los conductores sin necesidad de soplar hacia ella. (DADSS.org, 2015)



Fig. 6 Sistema de Bloqueo Vehicular por Espectroscopia

Fuente: DADSS.org, 2015

2.6 Redes Neuronales

En la actualidad uno de los métodos más utilizados para la clasificación de patrones es el uso de redes neuronales compuesto por neuronas y capas las cuales tienen asociados valores de pesos y umbrales, los cuales irán modificándose en cada ciclo de entrenamiento hasta minimizar el error de clasificación lo máximo posible.

2.6.1 Definición

Las redes de neuronas artificiales (denominadas habitualmente como *RNA* o en inglés como "*ANN*") son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas que colaboran entre sí para producir un estímulo de salida. (Guillen, 2015)

Matemáticamente el comportamiento de una neurona artificial se expresa como la sumatoria del producto de múltiples entradas con sus respectivos pesos asociados a esta neurona, seguidamente esta sumatoria es evaluada en una función de activación no lineal para así obtener el valor salida de la neurona artificial como se muestra en la figura 7.

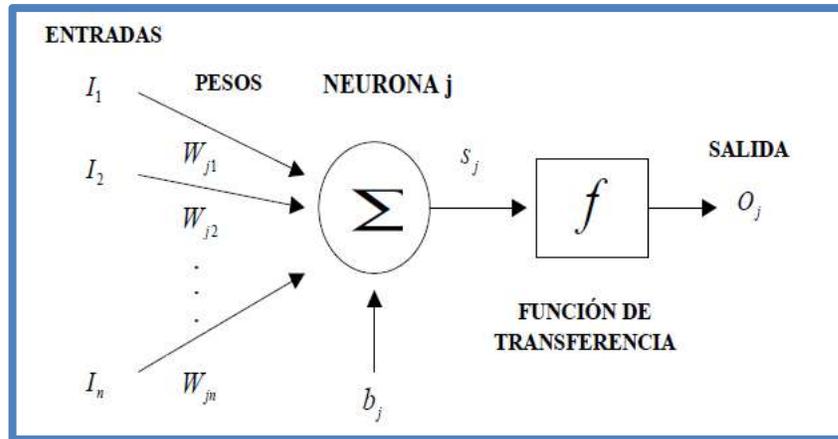


Fig. 7 Expresión Matemática de una Neurona Artificial

Fuente: Sánchez Castellanos y Meseas Teruel, 2009

Por tanto la expresión matemática de una neurona artificial será la siguiente:

$$O_j = f(w_{j1}I_1 + \dots + w_{jn}I_n + b_j) \quad (1)$$

Donde I_n es el valor de la n-ésima entrada (input) de la neurona artificial j ; w_{jn} es el peso (weights) asociado a la conexión entre la n-ésima entrada I_n y la neurona artificial j ; b_j es el valor umbral (threshold) asociado a la neurona j ; S_j es la sumatoria del producto de cada una de las entradas I con sus respectivos pesos W más el umbral b_j todos ellos asociados con la neurona artificial j y finalmente, O_j es la salida (output) de la neurona artificial el cuál es el resultado de evaluar S_j en la función de transferencia f . A continuación, se muestra un ejemplo de una función de activación para una neurona artificial:

$$S(u) = \begin{cases} 1, & u \geq 0 \\ 0, & u < 0 \end{cases} \quad (2)$$

2.6.2 Estructura

“La estructura de red depende de cómo se vaya a enfocar el problema a resolver, es el conjunto de una serie de neuronas que interconectan sus axones con las dendritas de las siguientes” (Sánchez Castellanos y Meseas Teruel, 2009). Un ejemplo lo podemos ver en la figura 8.

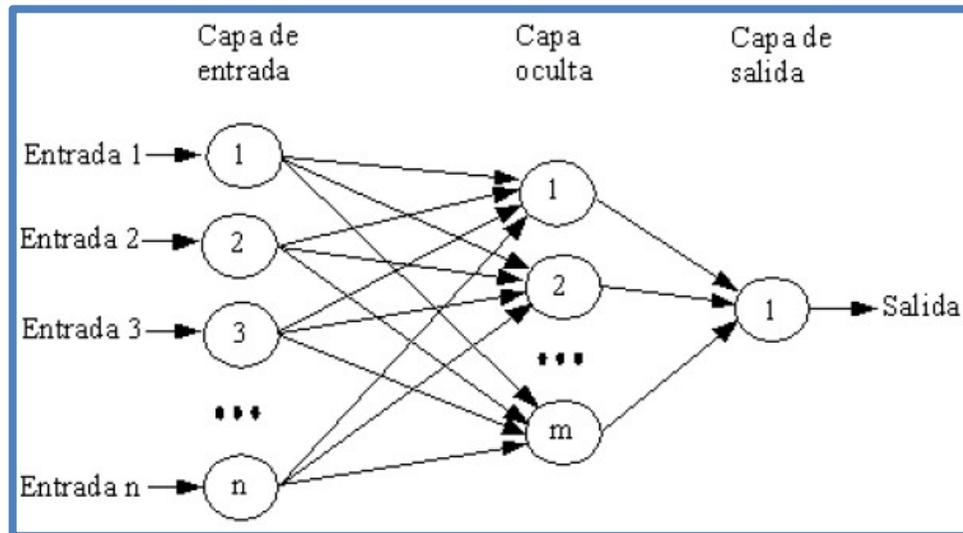


Fig. 8 Estructura Típica de Red Neuronal Artificial

Fuente: Sánchez Castellanos y Meseas Teruel, 2009

2.6.3 Topología de Red Neuronal Artificial

Según indican los autores (Sánchez Castellanos y Meseas Teruel, 2009) las redes neuronales artificiales presentan principalmente tres tipos de topologías las cuales se describen a continuación:

a) Mono capa

Es un tipo de RN con propagación hacia adelante o cíclica la cual no contiene ninguna capa oculta.

b) Multicapa

Es un tipo de red neuronal con propagación hacia adelante la cual tiene una o más capas ocultas.

c) Recurrente

Son aquellas que presentan al menos un lazo cerrado para una entrada de activación de una neurona.

A continuación, se visualiza las tres topologías de la Redes Neuronales Artificiales:

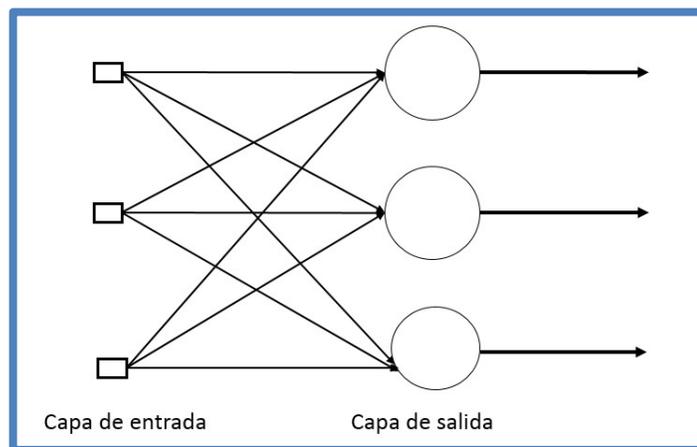


Fig. 9 Topología Monocapa de Red Neuronal

Fuente: Sánchez Castellanos y Meseas Teruel, 2009

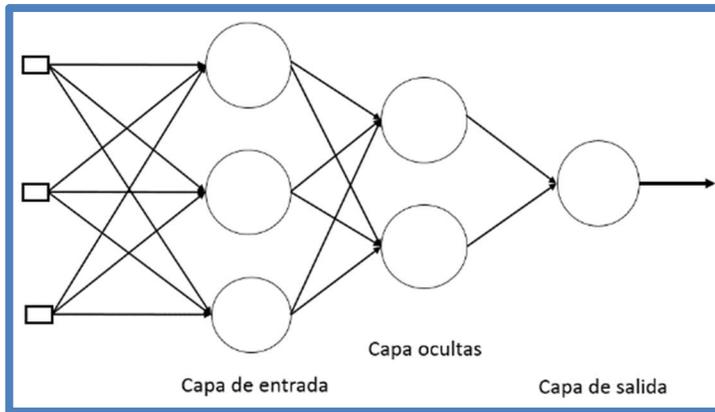


Fig. 10 Topología Multicapa de Red Neuronal

Fuente: Sánchez Castellanos y Meseas Teruel, 2009

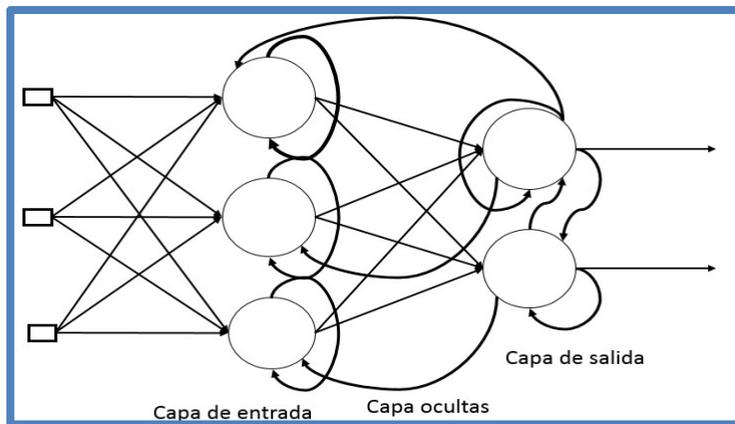


Fig. 11 Topología Recursiva de Red Neuronal

Fuente: Sánchez Castellanos y Meseas Teruel, 2009

2.6.4 Ventajas que Aportan las Redes Neuronales Artificiales

Según indican los autores (Sánchez Castellanos y Meseas Teruel, 2009), las ventajas que aportan las RNA son las siguientes:

a) Aprendizaje Adaptativo

Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.

b) Autoorganización

Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.

c) Tolerancia a Fallos

La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.

d) Operación en Tiempo Real

Los cálculos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.

e) Fácil Inserción dentro de la Tecnología Existente

Se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.

2.7 Filtros Activos

La teoría de filtros activos se desarrolló en base a la investigación sobre filtros activos del Ing. Juan Ignacio Huircán Quilaqueo de la Universidad de La Frontera de Chile. Este apartado se dividirá en tres conceptos: tipos de filtros y especificaciones de respuesta en frecuencia, funciones prototipos y determinación de las funciones de transferencia.

2.7.1 Tipos de Filtros y Especificaciones de Respuesta en Frecuencia

“Los filtros se pueden representar mediante la función de transferencia $H(s)$, la cual se expresa en términos de su ganancia o atenuación” (Huircán Quilaqueo, 2014). Así se tiene:

$$H(s) = \frac{V_o(s)}{V_i(s)} \quad (3)$$

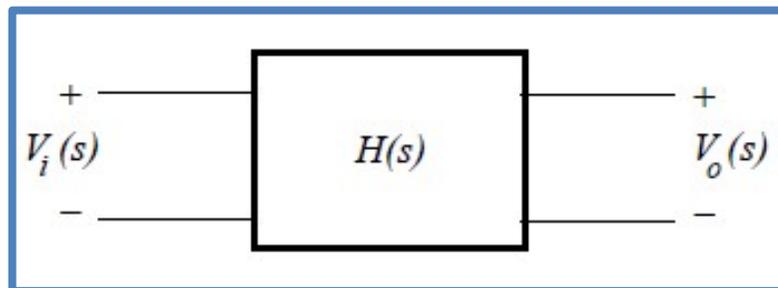


Fig. 12 Red de dos Puertas - Filtro Activo.

Fuente: Huircán Quilaqueo, 2014

Donde $V_i(s)$ es la entrada de filtro y $V_o(s)$, la salida. La transmisión del filtro se encuentra evaluando $H(s)|_{s=j\omega}$, así en términos de magnitud y fase se tiene:

$$H(j\omega) = |H(j\omega)|e^{j\phi(\omega)} \quad (4)$$

El espectro de la señal de salida sera obtenido por:

$$|V_o(j\omega)| = |H(j\omega)| |V_i(j\omega)| \quad (5)$$

De acuerdo al criterio de selección de frecuencia de paso o de rechazo, existen cuatro tipos de filtros.

2.7.1.1 Pasa Bajos (Low Pass Filter)

Son aquellos que tienen ganancia a frecuencias menores que la frecuencia de corte ω_c . Así, la banda de paso está dada para $0 < \omega < \omega_c$, donde ω_c se expresa en [rad/seg] o Hertz y corresponde a la frecuencia en la cual la ganancia es dividida por $\sqrt{2}$ (cae en -3dB). La ganancia disminuye a medida que se supera a dicha frecuencia. (Huircán Quilaqueo, 2014). De acuerdo a la Fig. 13, esta zona se conoce como banda de rechazo.

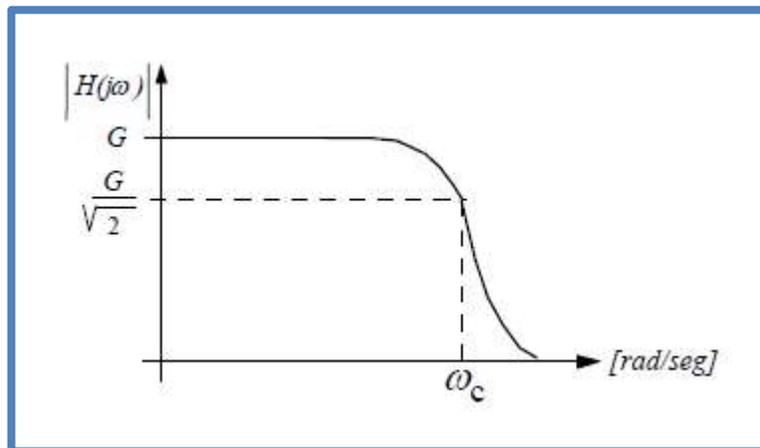


Fig. 13 Respuesta en Frecuencia de un Filtro Pasa Bajos

Fuente: Huircán Quilaqueo, 2014

La función de transferencia para un filtro pasabajos de orden n de ganancia G es:

$$H(s) = \frac{Gb_0}{s^n + b_{n-1}s^{n-1} + \dots + b_0} \quad (6)$$

2.7.1.2 Pasa Altos (High Pass Filter)

“Permite el paso de frecuencias superiores a ω_c ” (Huircán Quilaqueo, 2014). Su función de transferencia se puede obtener a partir de la ecuación 6 reemplazando $s = 1/s$, es decir:

$$H(s) = \frac{Gb_0}{s^n + b_{n-1}s^{n-1} + \dots + b_0} \Big|_{s = \frac{1}{s}} \quad (7)$$

Luego se tiene:

$$H(s) = \frac{Gs^n}{s^n + a_{n-1}s^{n-1} + \dots + a_0} \quad (8)$$

Donde $b_n = 1$; $a_{n-1} = \frac{b_i}{b_0}$; $i = 1, 2, \dots, n$. La respuesta en frecuencia se indica en la Fig. 14.

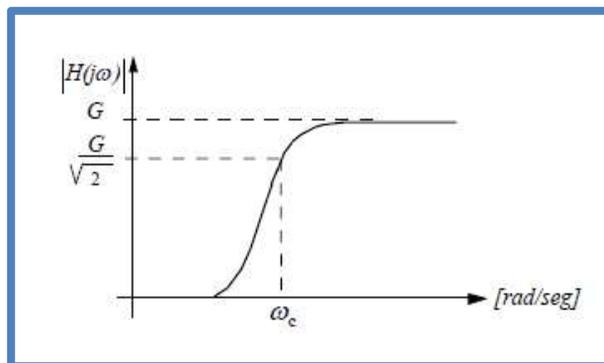


Fig. 14 Respuesta en Frecuencia de un Filtro Pasa Altos

Fuente: Huircán Quilaqueo, 2014

2.7.1.3 Pasabanda (Band Pass Filter):

“Este filtro deja pasar las frecuencias que se encuentran dentro una de banda B (expresado en rad/seg o en Hertz), centrada en ω_0 , atenuando todas las otras frecuencias” (Huircán Quilaqueo, 2014). La Fig. 15 muestra la respuesta en frecuencia del filtro centrado en ω_0 cuyo ancho de banda está definido por:

$$B = \omega_2 - \omega_1 \quad (9)$$

La función de transferencia se obtiene de la ecuación 6 como:

$$H(s) = \frac{Gb_0}{s^n + b_{n-1}s^{n-1} + \dots + b_0} \Big|_{s = \frac{s^2 + \omega_0^2}{B_s}} \quad (10)$$

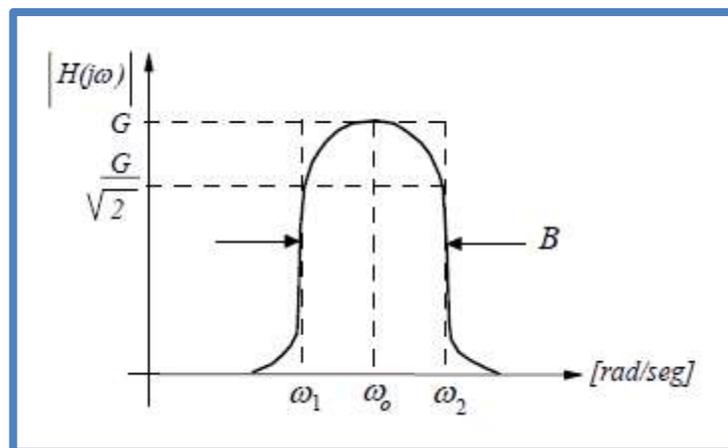


Fig. 15 Respuesta en Frecuencia de un Filtro Pasabanda

Fuente: Huircán Quilaqueo, 2014

Así, el orden del filtro pasabanda se incrementará al doble. Se define el factor de calidad Q, el cual mide la selectividad del filtro (un Q muy alto indica que el filtro es muy selectivo con banda de paso muy pequeña) como:

$$Q = \frac{w_0}{B} = \frac{f_0}{B} \quad (11)$$

La ganancia será la amplitud de la función de transferencia a la frecuencia w_0 , la cual corresponde a la media geométrica, pues está en escala logarítmica:

$$w_0 = \sqrt{w_2 w_1} \quad (12)$$

2.7.1.4 Rechazabanda (Band Reject Filter):

“También llamado elimina banda y para algunas situaciones filtro Notch, deja pasar todas las frecuencias excepto una única banda” (Huircán Quilaqueo, 2014). La banda de rechazo está definida por B, como se indica en la Fig. 16.

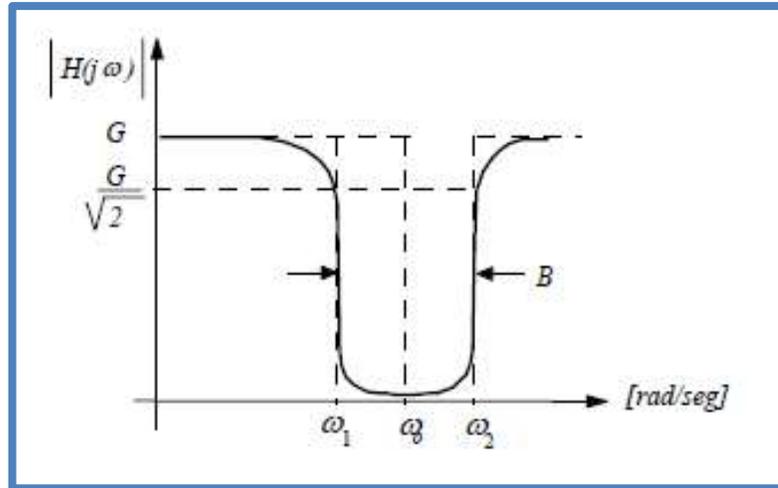


Fig. 16 Respuesta en Frecuencia de un Filtro Rechaza Banda

Fuente: Huircán Quilaqueo, 2014

La función de transferencia se obtiene haciendo la sustitución indicada en la ecuación 6 donde las características de banda son iguales al filtro pasabanda:

$$H(s) = \frac{Gb_0}{s^n + b_{n-1}s^{n-1} + \dots + b_0} \Bigg|_{s = \frac{B_s}{s^2 + \omega_0^2}} \quad (13)$$

2.7.2 Funciones Prototipos

2.7.2.1 Butterworth

“Este filtro tiene una respuesta plana en la banda de paso (llamada máximamente plana), a expensas de la respuesta en la región transición, la cual es de 20 dB/Década por polo” (Huircán Quilaqueo, 2014). El módulo de la respuesta en frecuencia del filtro pasabajos, para ganancia G , y frecuencia de corte ω_c está dado en la ecuación 14:

$$|H(j\omega)| = \frac{G}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}} \quad (14)$$

Donde $n = 1, 2, \dots, k$. es el orden. La Fig. 17 indica respuestas de este filtro para distintos n .

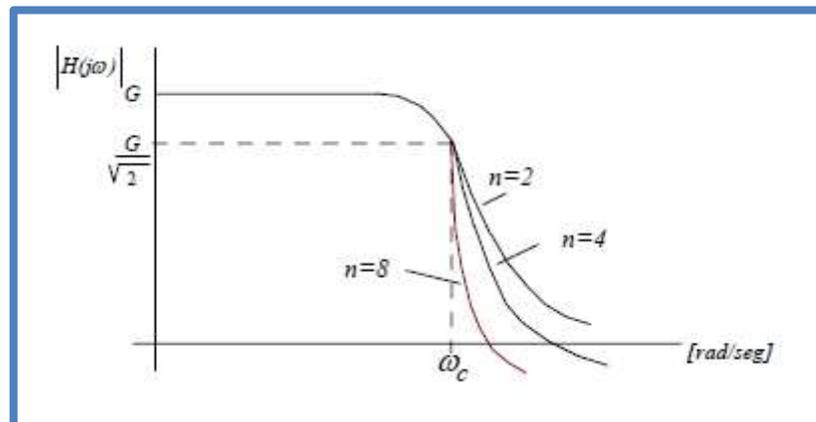


Fig. 17 Respuesta del Filtro Butterworth

Fuente: Huircán Quilaqueo, 2014

2.7.2.2 Chebyshev

Este filtro tiene una ondulación (ripple) en la banda de paso. Mientras mayor es el orden, mayor es la pendiente en la región de transición, pero mayor es el ripple y el número de ondulaciones en la banda de paso. (Huircán Quilaqueo, 2014) El módulo de la función de transferencia está dado por la ecuación 15, donde K_1 y ϵ , son valores constantes y $C_n(x)$ es el polinomio Chebyshev (en primera aproximación) de grado n :

$$|H(j\omega)| = \frac{K_1}{\sqrt{1 + \epsilon^2 C_n^2\left(\frac{\omega}{\omega_c}\right)}} \quad (15)$$

La Fig. 18 muestra la respuesta en frecuencia para diferentes n :

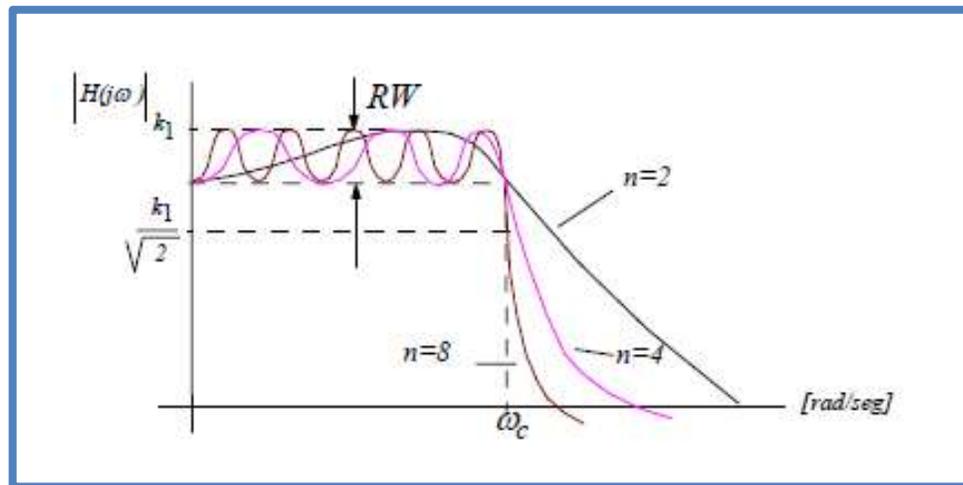


Fig. 18 Respuesta del Filtro Chebyshev

Fuente: Huircán Quilaqueo, 2014

De acuerdo al marco teórico estudiado hasta aquí, se tomará como función prototipo a Butterworth, debido a la respuesta máximamente plana que tiene en la banda de paso, ya que no se desea alterar la señal de entrada en la etapa de pre-procesamiento, solo se desea realizar el filtrado a las frecuencias correspondientes, por tanto a continuación se pasará a explicar sus propiedades.

2.7.2.3 Propiedades del Filtro Butterworth

“Una respuesta máximamente plana tiene muchas derivadas que son cero en el origen, $\omega = 0$ ” (Huiracán Quilaqueo, 2014). Para ganancia unitaria y una frecuencia $\omega = \omega_c$, se tiene que:

$$|H(j\omega)| \cong \frac{1}{\sqrt{2}} \quad (16)$$

También llamada -3 [dB]. Para $\omega \gg \omega_c$, se tiene que:

$$|H(j\omega)| \cong \frac{1}{\omega^n} \quad (17)$$

O también:

$$|H(j\omega)|_{dB} \cong 20 \log\left(\frac{1}{\omega^n}\right) = -20n \log(\omega) \quad (18)$$

Así, la variación será de $-20n$ [dB] por década, donde n es el orden del filtro.

2.7.3 Determinación de las Funciones de Transferencia

A continuación se explicará las ecuaciones necesarias para obtener la función de transferencia del filtro Butterworth en base a la teoría desarrollada por (Huircán Quilaqueo, 2014) .

Sea la ecuación 14 con $G = 1$ y $\omega_c = 1$, haciendo $s = j\omega$, entonces $\omega = s/j$, reemplazando se tiene:

$$|H(j\omega)|^2 = \frac{1}{1 + \left(\frac{s}{j}\right)^{2n}} = \frac{1}{1 + (-1)^n s^{2n}} \quad (19)$$

Los polos de la función de transferencia se obtienen para:

$$(-1)^n s^{2n} = -1 \quad (20)$$

Pero como:

$$e^{j(2k-1)\pi} = -1 \quad (21)$$

Entonces:

$$(-1)^n s^{2n} = e^{j(2k-1)\pi} \quad (22)$$

Así, los polos para $k = 1, 2, \dots, n$, estarán dados por:

$$s_k = e^{\frac{j(2k+n-1)\pi}{2n}} \quad (23)$$

Los polinomios que se obtienen, son de la forma:

$$H(s) = \frac{1}{(s-s_1)(s-s_2)\dots(s-s_n)} \quad (24)$$

Para $n = 2$ se tiene que:

$$s_1 = e^{j\frac{3\pi}{4}} \quad (25)$$

$$s_2 = e^{j\frac{5\pi}{4}} \quad (26)$$

Luego:

$$H(s) = \frac{1}{(s - e^{j\frac{3\pi}{4}})(s - e^{j\frac{5\pi}{4}})} = \frac{1}{s^2 + \sqrt{2}s + 1} \quad (27)$$

El denominador de la ecuación 27 corresponde a un polinomio de Butterworth indicado en la Tabla 3 que se presenta a continuación.

Tabla 3. Polinomios de Butterworth en forma factorizada.

n	Polinomios de Butterworth
1	$s+1$
2	$s^2 + \sqrt{2}s + 1$
3	$(s^2 + s + 1)(s + 1)$
4	$(s^2 + 0.7653s + 1)(s^2 + 1.8477s + 1)$
5	$(s^2 + 0.6180s + 1)(s^2 + 1.6180s + 1)(s + 1)$
6	$(s^2 + 0.5176s + 1)(s^2 + \sqrt{2}s + 1)(s^2 + 1.9318s + 1)$

Elaboración: los autores

De la misma forma son determinados los polinomios para orden superior.

2.8 Teorema de Muestreo de Nyquist – Shannon

El teorema de muestreo de Nyquist - Shannon es fundamental para la reconstrucción de señales en base a sus muestras. “Una forma de onda muestreada contiene toda la información sin distorsión, cuando la tasa de muestreo excede el doble de la frecuencia más alta contenida por la forma de onda” (Lavry, 2014, p.1). Si la frecuencia más alta contenida en una señal analógica es:

$$F_{\max}=B \quad (28)$$

y la señal se muestrea a una tasa:

$$F_s > 2F_{\max} \equiv 2B \quad (29)$$

Entonces dicha señal se puede recuperar completamente y sin distorsiones en base a cada una de sus muestras. “No cumplir los requerimientos causará distorsiones, llamadas distorsiones de aliasing” (Lavry, 2014, p.8). Portanto será necesario implementar un filtro pasabajos con frecuencia de corte igual a la mitad de la frecuencia de muestreo, para evitar problemas de aliasing.

CAPÍTULO III

METODOLOGÍA

3.1 Módulos Utilizados

A continuación, se describen todos los módulos utilizados para el desarrollo de la presente investigación.

3.1.1 Módulo de Sensor de Etanol MQ3

Según Park (2008) el sensor MQ3 tiene las siguientes características:

- Este sensor MQ3, es capaz de detectar etanol en el aire, es uno de los sensores de gas más sencillos de utilizar, ya que trabaja casi de la misma manera que los otros sensores de gas.



Fig. 19 Sensor de Etanol MQ3

Fuente: Park, 2008

- Está compuesto interiormente por un tubo cerámico de alumina (Al_2O_3), cubierto por dióxido de estaño (SnO_2), interiormente el tubo cerámico tiene una bobina de calentamiento. Todos estos elementos conforman el sistema de calentamiento del sensor, el cual sirve para detectar la cantidad de etanol en el aire.
- Se puede apreciar en la siguiente figura que 4 de los 6 pines del sensor están conectados al tubo cerámico y 2 a la bobina de calentamiento

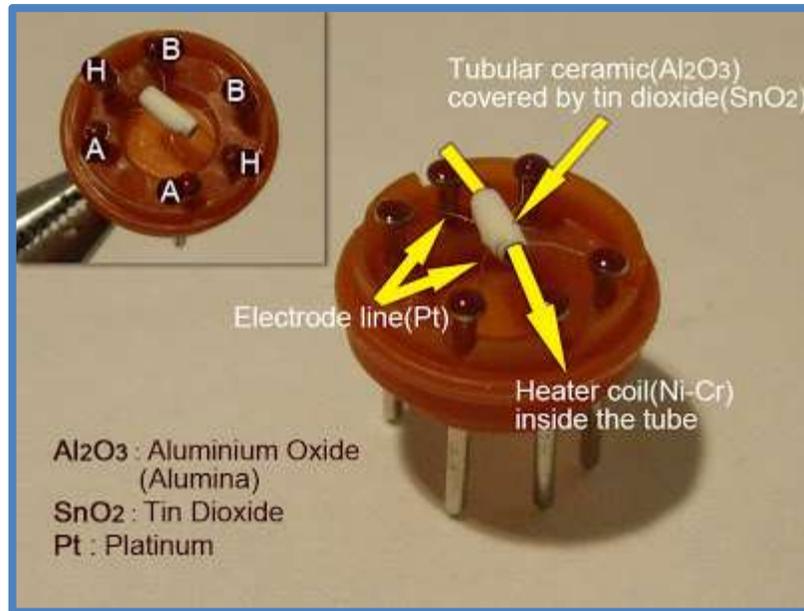


Fig. 20 Sensor de Etanol MQ3 Abierto y Distribución de Pines

Fuente: Park, 2008

- En la siguiente imagen, se puede observar una imagen de un corte transversal al tubo cerámico

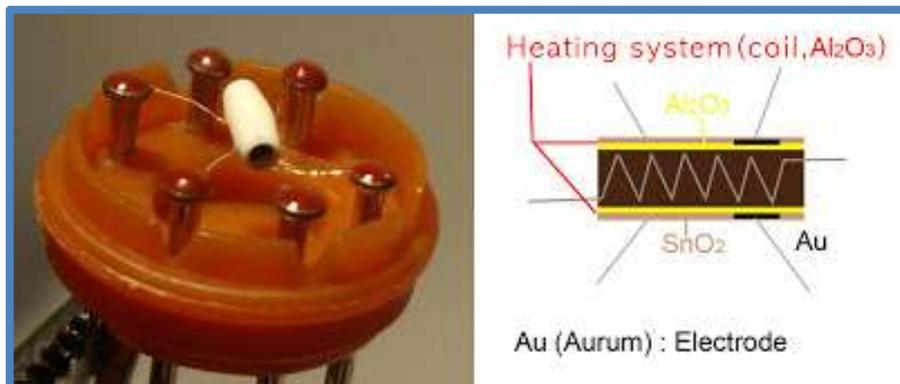


Fig. 21 Imagen Transversal del Sensor MQ3

Fuente: Park, 2008

- En la Fig. 21, se puede observar el tubo de alumina (color amarillo) cubierto con dióxido de estaño (color marrón claro), entre ellos se aprecia un electrodo de oro (color negro).
- Entonces cuando la bobina se calienta, el tubo cerámico de dióxido de estaño se convierte en un semiconductor, de modo que hay más electrones móviles, lo cual significa que está preparado para producir más flujo de corriente.

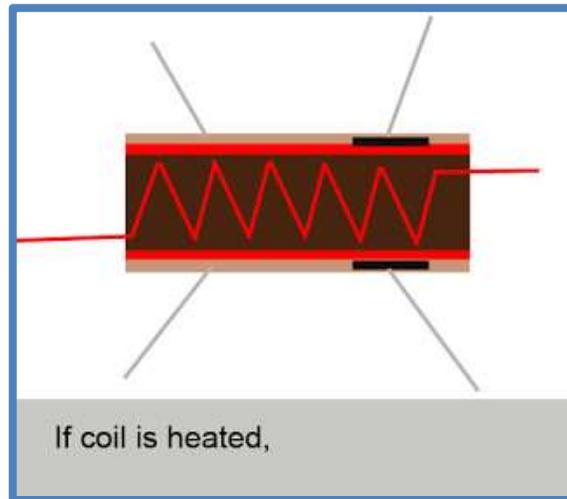


Fig. 22 Bobina del Tubo Cerámico Caliente

Fuente: Park, 2008

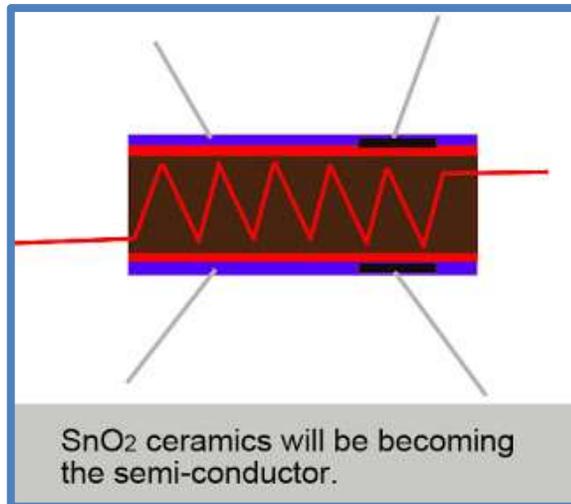


Fig. 23 Tubo Cerámico Como Semiconductor

Fuente: Park, 2008

- Entonces, cuando las moléculas de alcohol chocan con el electrodo que esta entre la alumina y el dióxido de estaño, el etanol arde como ácido acético y entonces más corriente es producida. Por tanto, mientras más moléculas de alcohol haiga, más corriente será obtenida.

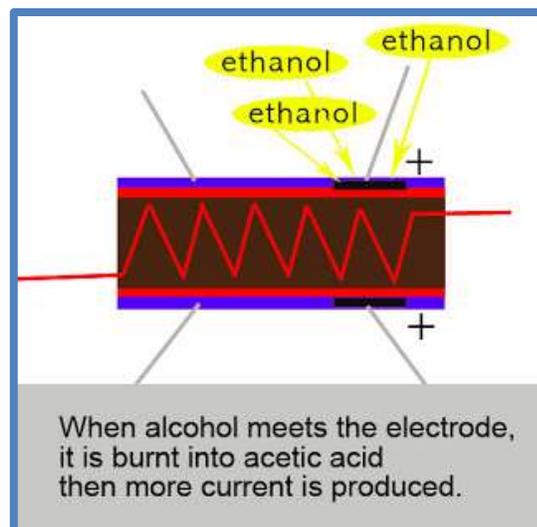


Fig. 24 Partículas de Etanol Haciendo Contacto con el Electrodo

Fuente: Park, 2008

Por tanto, es de esta forma cómo funciona el sensor de etanol MQ3, hay una relación directa entre la cantidad de etanol que hace contacto con el electrodo y la corriente producida por el sensor.

La Fig. 25. muestra la conexión de pines para la calibración del sensor, según el fabricante:

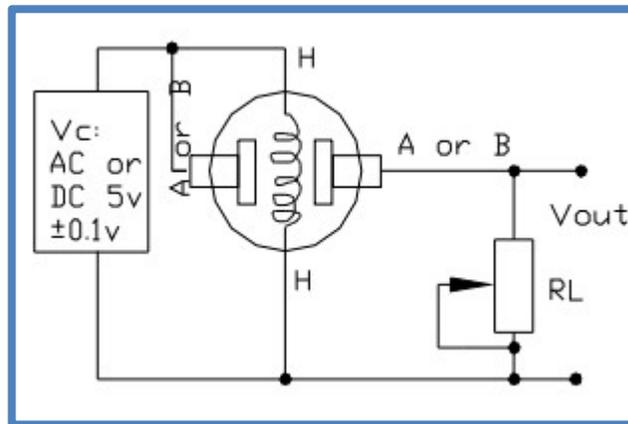


Fig. 25 Circuito de Calibración del Sensor MQ3

Fuente: Hanwei Eletronics Co.,Ltd., 2014

La Fig. 26 muestra las características de sensibilidad del sensor MQ3 para diferentes tipos de gases:

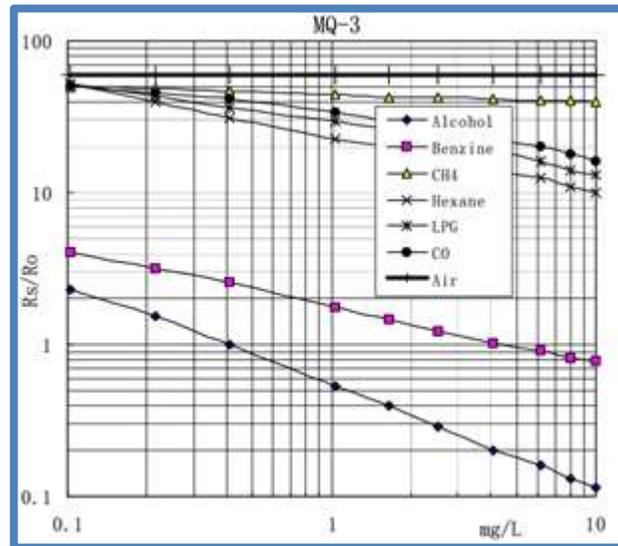


Fig. 26 Características de Sensibilidad del Sensor MQ3

Fuente: Hanwei Eletronics Co.,Ltd., 2014

Dónde:

- R_s : Resistencia del sensor a diferentes niveles de concentración con distintos gases
- R_o : Resistencia del sensor a 0.4mg/L de alcohol en aire limpio.
- La resistencia de carga R_L utilizada para el experimento fue de 200K Ω .
- La Fig. 26, fue elaborada bajo las siguientes condiciones climatológicas: Temperatura: 20°C; Humedad: 65% y Concentración de O_2 : 21%.

El fabricante indica en el datasheet del sensor que para calibrar correctamente el MQ3 debemos utilizar una resistencia de carga en el circuito de la Fig. 25 de $200\text{K}\Omega$ o en un rango de $100\text{K}\Omega$ a $470\text{K}\Omega$ y medir la resistencia del sensor en el punto de alarma deseado, en este caso el fabricante recomienda a 0.4mg/L de etanol.

Sin embargo, llevar a cabo esto sin las condiciones ambientales, ni los instrumentos de precisión necesarios de un laboratorio no es viable, por lo que se optó por adquirir un módulo el cual incluye al sensor MQ3 calibrado y que permite fácilmente con el ajuste de una resistencia variable, establecer el punto de alarma.

El módulo adquirido se muestra en la Fig. 27:



Fig. 27 Módulo del Sensor MQ3

Fuente: Electronilab, 2015

En la Fig. 28 se muestra una imagen de la parte posterior del módulo con la distribución de pines:

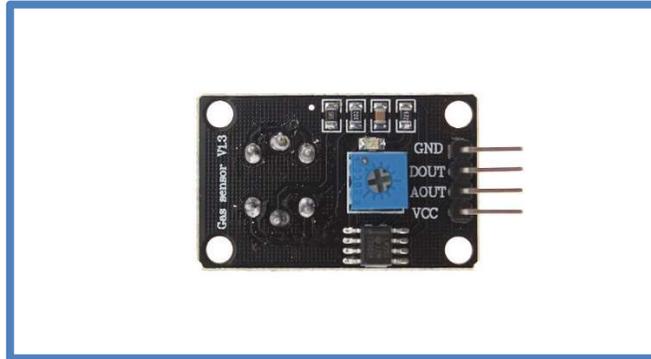


Fig. 28 Módulo del Sensor MQ3 - Vista posterior y Distribución de Pines

Fuente: Electronilab, 2015

El módulo consta de 4 pines Vcc: Alimentación; GND: Tierra; DOUT: Salida digital y AOUT: Salida Analógica.

Para el desarrollo del proyecto se utilizó la salida analógica para medir la cantidad de etanol exhalada por el usuario y posteriormente llevar la data a la etapa de pre-procesamiento para el muestreo y cuantificación.

3.1.1.1 Normalización del Voltaje de Salida del Módulo del Sensor de Etanol MQ3

Se estableció una relación entre el voltaje entregado por el sensor MQ3 y la concentración de alcohol en la sangre BAC, esto se logró utilizando la siguiente ecuación de normalización:

$$X' = \frac{(X - X_{\min})(b - a)}{(X_{\max} - X_{\min})} + a \quad (30)$$

Donde X es el valor original de la variable que se quiere normalizar; X_{\min} es el valor mínimo que puede tener la variable X; X_{\max} es el valor máximo que puede tener la variable X; a es el valor mínimo del intervalo al cual se requiere normalizar; b es el valor máximo del intervalo al cual se requiere normalizar y X' es el valor normalizado de la variable X.

Para la presente investigación se normalizó el voltaje digital obtenido del sensor MQ3 (cuyo valor a 10 bits fue limitado a un rango entre 160 y 1023, ya que el sensor en reposo presenta variaciones en su voltaje entre 0 y 159; por lo cual no se tomaron en cuenta para el cálculo del BAC) a un intervalo entre 0.00 y 1.99, el cual representa el BAC.

Reemplazando estos valores en la ecuación 30 se tiene:

$$BAC = \frac{(\text{VoltMQ3} - 160)(1.99 - 0.00)}{(1023 - 160)} + 0.00 \quad (31)$$

Para la validación del BAC medido y la contrastación se realizó un experimento tomando 30 muestras de BAC con el alcoholímetro calibrado de referencia y con el sensor MQ3, Los datos de este experimento y el cálculo del coeficiente de correlación de Pearson se encuentran en el Anexo 1.

3.1.2 Microprocesador ATMEGA328P

El microprocesador ATMEGA328P, es muy usado en el mercado por su bajo costo y alta eficiencia, diferenciándose de otras marcas por ejecutar una instrucción por cada ciclo de reloj (a diferencia de Microchip que ejecuta una instrucción por cada 4 ciclos de reloj), esto quiere decir que se puede explotar al cien por ciento la velocidad del reloj que se utilice, lo cual lo hace ideal para programar el algoritmo neuronal del proyecto. En la siguiente figura mostramos sus características principales:

- Operating Voltage:
 - 1.8 - 5.5V
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 4MHz@1.8 - 5.5V, 0 - 10MHz@2.7 - 5.5.V, 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
 - Active Mode: 0.2mA
 - Power-down Mode: 0.1µA
 - Power-save Mode: 0.75µA (Including 32kHz RTC)

Fig. 29 Características Principales de ATMEGA328P

Fuente: Atmel Corporation, 2014

Como se puede apreciar en la Fig. 29 el microprocesador soporta grandes velocidades (hasta 20MHz) a una corriente de funcionamiento baja de 0.2mA, además presenta robustez ante los cambios

de temperatura (-40°C – 85°C), lo cual lo hace un microprocesador ideal para el proyecto a implementar.

3.1.2.1 Características Principales del Conversor Análogo-Digital

De acuerdo al datasheet completo del microprocesador ATMEGA328P (Atmel Corporation, 2014, p.237), las principales características del convertidor análogo-digital son:

- Resolución máxima de 10 bits
- Frecuencia de muestreo 15Kmps a máxima resolución
- Rango de voltaje de entrada al ADC 0-Vcc(5V)
- Voltaje de referencia seleccionable

Antes de ingresar la señal analógica del sensor MQ3 al convertidor análogo-digital para muestrearla y cuantificarla, fue necesario aplicar el teorema de Nyquist ecuación 29, estudiado en el marco teórico, a fin de que la etapa de muestreo no presente problema alguno.

En este caso $F_s=15\text{Kmps}$, entonces:

$$15\text{Kmps} > 2F_{\max} \quad (32)$$

Finalmente:

$$F_{\max} < 7.5\text{KHZ} \quad (33)$$

Por tanto, para poder satisfacer este requerimiento se necesita del diseño de un filtro pasa bajos con frecuencia de corte 7.5KHz, el cual será diseñado e implementado en la etapa de pre-procesamiento.

3.1.3 Módulo Bluetooth HC-05

Es un módulo al cual se puede tener fácil acceso en el mercado y a bajo precio, su función será la de permitir la conexión entre el circuito neuronal diseñado y el celular Android, en la cual se publicarán los resultados de la clasificación.



Fig. 30 Módulo Bluetooth HC-05

Fuente: Amazon.com, Inc., 2015

En la Fig. 30, se observa la distribución de pines del módulo bluetooth, para el presente proyecto se utilizarán los pines de alimentación (Vcc y GND), además de los pines de TXD y RXD, los cuales irán conectados a los pines de transmisión serial del microcontrolador. Este punto se explicará con más detalle en la etapa de post-procesamiento.

3.2 Etapa de Preprocesamiento

A continuación, se describe como se diseñó e implementó la etapa de preprocesamiento en la presente investigación.

3.2.1 Filtro Butterworth Pasa Bajos

Tal y como se determinó en la ecuación 6, se requiere en la etapa de preprocesamiento un filtro pasa bajos, con frecuencia de corte de 7.5KHz, por tanto, se explicará el diseño e implementación del filtro.

3.2.2 Implementación con Amplificadores Operacionales

En cuanto a la implementación de filtros con amplificadores operacionales, se tiene la siguiente información:

Las funciones de transferencia de 2° orden se implementan con Amplificadores Operacionales, Resistores y Capacitores, como lo indica Hilburn y Johnson (1975), Sedra y Smith (1998). Existen dos circuitos clásicos, el VCVS o Sallen-Key (SK) y el de Múltiple Realimentación (MFB), también llamada Rauch (Dede y Espi, 1983). (Huircán Quilaqueo, 2014, p.6)

3.2.3 Circuito Pasa Bajos

La Fig. 31 muestra el circuito SK, donde la ecuación 34 representa la función de transferencia.

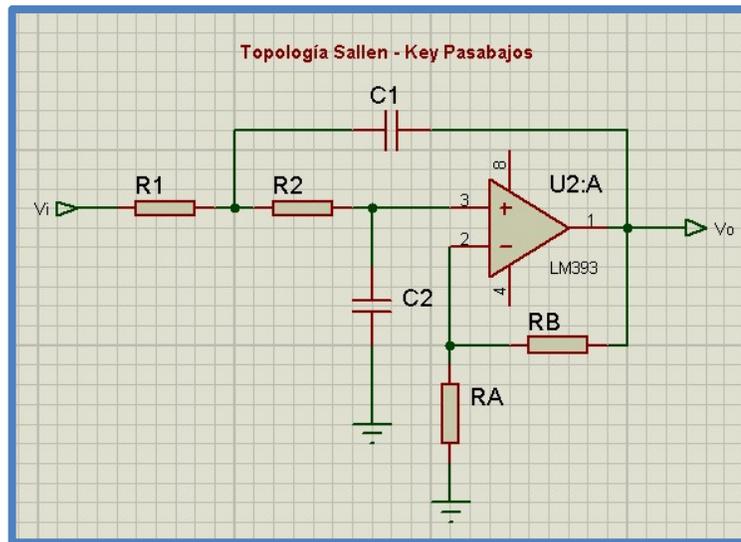


Fig. 31 Pasabajos SK de 2° Orden

Elaboración: los autores

Fuente: Software Proteus 8 Professional

Para el filtro SK se tiene:

$$H(s) = \frac{\left(\frac{R_B}{R_A} + 1\right) \frac{1}{R_1 R_2 C_1 C_2}}{s^2 + \left\{ \frac{1}{C_1} \left(\frac{1}{R_1} + \frac{1}{R_2} \right) - \frac{1}{R_2 C_2} \frac{R_B}{R_A} \right\} s + \frac{1}{R_1 R_2 C_1 C_2}} \quad (34)$$

Donde:

$$G = \frac{R_B}{R_A} + 1 \quad (35)$$

$$\omega_c^2 = \frac{1}{R_1 R_2 C_1 C_2} \quad (36)$$

En la presente de investigación es de interés aplicar un método que permita diseñar un filtro pasa bajos con ganancia unitaria, ya que no se desea que la señal de entrada del sensor de etanol varíe, solamente se requiere filtrar las componentes a una frecuencia mayor a 7.5KHz.

Por tanto, el método de diseño que mejor se ajusta a los requerimientos es el de Savant y Roden.

3.2.4 Método de Diseño Propuesto por Savant y Roden

En cuanto al método de diseño propuesto por Savant y Roden, se tiene la siguiente información:

Savant (1992) usa circuitos SK de 2° orden (pasa bajos y pasa altos) con ganancia unitaria, frecuencia de corte $1/r/s$ y $R_1 = R_2 = R = 1[\Omega]$, para el filtro pasa bajos ($C_1 = C_2 = C = 1[F]$, para el filtro pasa alto). Además, una tabla con coeficientes C_i/C o R_i/R , asociados a las respuestas de Butterworth y Chebyshev. (Huircán Quilaqueo, 2014, p.9)

En la presente investigación es de interés aplicar el método relacionado al filtro pasa bajos en la etapa de pre-procesamiento hacia el conversor A/D del microcontrolador; a continuación, se detallan los pasos del método utilizado.

3.2.5 Determinación del Orden del Filtro

Para determinar el orden del filtro a utilizar, el autor Huircán Quilaqueo (2014) indica el siguiente procedimiento:

Sea la ventana de diseño de la Fig. 32, se propone el uso de WPB, WSB, APB y ASB, para filtros Butterworth.

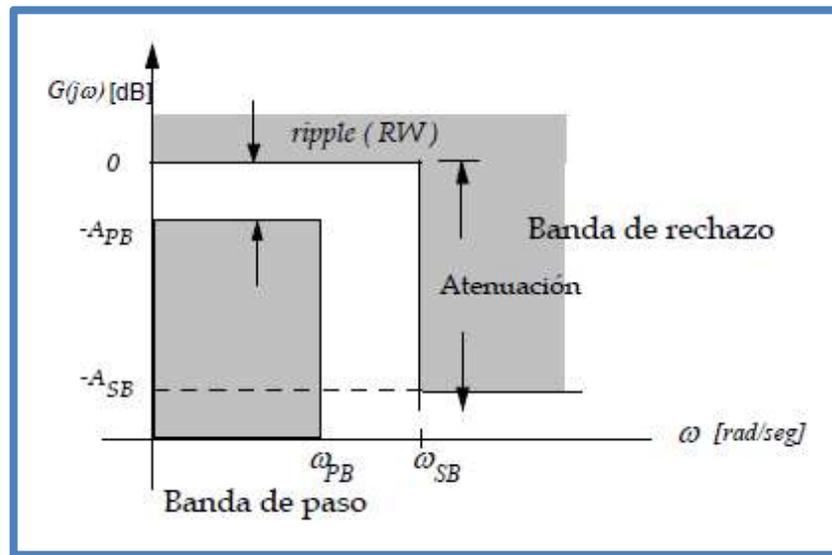


Fig. 32 Ventana de Diseño para un Filtro Pasabajos

Fuente: Huircán Quilaqueo, 2014

Dónde:

- APB: Máxima atenuación en la banda de paso en dB.
- RW: Ripple de la banda de paso o ancho de Ripple.
- ASB: Mínima atenuación de la banda de rechazo en dB.
- WPB: Frecuencia de esquina de la banda de paso en [r/s].
- WSB: Frecuencia de esquina de la banda de rechazo en [r/s].

La banda comprendida entre WPB y WSB se le conoce como “Banda de Transición” la cual empieza en un máximo de atenuación en la banda de paso (APB) y termina en un mínimo de atenuación en la banda de rechazo (ASB).

En el caso de filtros pasa bajos Butterworth no existirá Ripple, ya que tiene una respuesta máximamente plana cerca de $W=0$.

Entonces se tiene que:

$$nb = \frac{\log\left(\frac{e1}{e2}\right)}{\log\left(\frac{fPB}{fSB}\right)} \quad (37)$$

Dónde:

$$e1 = \sqrt{10^{0.1APB} - 1} \quad (38)$$

$$e2 = \sqrt{10^{0.1ASB} - 1} \quad (39)$$

$$fPB = \frac{WPB}{2\pi} \quad (40)$$

$$fPS = \frac{WPS}{2\pi} \quad (41)$$

Y nb es el orden del filtro a implementar.

3.2.6 Coeficientes para Filtros Butterworth

De acuerdo al autor Huircán Quilaqueo (2014) el procedimiento para determinar los coeficientes para filtros Butterworth es el siguiente:

Sea la función de transferencia SK de la ecuación 34, con $R = R_1 = R_2 = 1[\Omega]$, $W_c=1[r/s]$ y $G = \frac{R_B}{R_A} + 1 = 1$ (ganancia unitaria), entonces $\frac{R_B}{R_A} = 0$. Se tiene:

$$H(s) = \frac{1}{\frac{1}{C_1 C_2} s^2 + \frac{2}{C_1} s + \frac{1}{C_1 C_2}} = \frac{1}{C_1 C_2 s^2 + 2 C_2 s + 1} \quad (42)$$

Para $n=2$, $(s^2 + \sqrt{2}s + 1)$, igualando coeficientes:

$$b_2 = 1 = C_1 C_2 \quad (43)$$

$$b_1 = \sqrt{2} = 2 C_2 \quad (44)$$

De esta forma:

$$C_2 = \frac{b_1}{2} \quad (45)$$

$$C_1 = \frac{b_2}{C_2} \quad (46)$$

Así:

$$C_2 = \frac{b_1}{2} = \frac{\sqrt{2}}{2} \cong 0.7071 \quad (47)$$

$$C_1 = \frac{b_2}{C_2} = \frac{1}{0.7071} \cong 1.4142 \quad (45)$$

Para $n=4$ (polinomio de orden 4 de Butterworth), se tiene $(s^2+0.7653s+1)(s^2+1.8477s+1)$

Se calcularán dos pares de coeficientes, para el primer y segundo polinomio.

Así se tiene que $C_2 = \frac{0.7653}{2} \cong 0.3825$ y $C_1 = \frac{1}{C_2} = \frac{1}{0.3825} \cong 2.613$ para la etapa 2, luego para la etapa 1 $C_2 = \frac{1.8477}{2} \cong 0.9241$ y $C_1 = \frac{1}{C_2} = \frac{1}{0.9241} \cong 1.082$

Considerando el polinomio de orden 6 y 8 se puede construir la tabla 4.

Tabla 4. Coeficientes para Filtros de Butterworth

	Butterworth	
# polos	C1=C R1=R	C2=C R1=R
2 etapa 1	1.414	0.7071
4 etapa 1 etapa 2	1.082	0.9241
	2.613	0.3825
6 etapa 1 etapa 2 etapa 3	1.035	0.9660
	1.414	0.7071
	3.863	0.288
8 etapa 1 etapa 2 etapa 3 etapa 4	1.020	0.9809
	1.202	0.8313
	1.800	0.5557
	5.125	0.1950

Elaboración: los autores

Si la función es pasa alto, los coeficientes serán idénticos, esto tomando $C = C_1 = C_2 = 1[F]$. Para esta situación, los coeficientes son R_i/R .

3.2.7 Ecuaciones de Diseño del Filtro

De acuerdo con el autor Huircán Quilaqueo (2014) el procedimiento para determinar las ecuaciones de diseño del filtro es el siguiente:

Dado el orden n , se obtienen los coeficientes C_1/C y C_2/C respectivos. Como la tabla 4 está construida para frecuencia $1[r/s]$ y $R = 1[\Omega]$, entonces para $W_c \neq 1[r/s]$ y $R \neq 1[\Omega]$, se debe escalar la frecuencia. Así se hace $s=W_cR$, se reemplaza en la ecuación 42 y se igualan coeficientes, luego:

$$b_2 = R^2 W_c^2 C_1 C_2 \quad (46)$$

$$b_1 = 2R(W_c)C_2 \quad (47)$$

Entonces de la ecuación 47 se tiene:

$$C_2 = \frac{b_1}{2RW_c} = \frac{b_1}{2} \frac{1}{RW_c} \quad (48)$$

De la ecuación 46 se tiene:

$$C_1 = \frac{b_2}{C_2 R^2 W_c^2} = \frac{b_2}{C_2} \frac{1}{R^2 W_c^2} \quad (49)$$

Reemplazando la ecuación 48 en la ecuación 49:

$$C1 = \frac{\frac{b2}{1}}{\frac{b1}{2RWc}} \cdot \frac{1}{R^2Wc^2} = \frac{2(RWc)b2}{b1} \frac{1}{R^2Wc^2} = 2 \frac{b2}{b1} \frac{1}{RWc} \quad (50)$$

Para el polinomio de Butterworth de $n = 2$ ($s^2 + \sqrt{2}s + 1$), se tiene de ecuación 51 y ecuación 52:

$$C2 = \frac{\sqrt{2}}{2} \frac{1}{RWc} = 0.7071 \frac{1}{RWc} \quad (51)$$

$$C1 = 2 \frac{1}{\sqrt{2}} \frac{1}{RWc} = 1.414 \frac{1}{RWc} \quad (52)$$

Lo mismo ocurrirá para $n = 4$, se mantienen los coeficientes de la tabla 4, pero aparece un factor en función de R y Wc . Finalmente, los capacitores para $f_c \neq 0$ y $R \neq 1[\Omega]$ estarán dados por:

$$Cn = Ci \frac{1}{RWc} = Ci \frac{1}{2\pi f_c R} \quad (53)$$

Dónde: “ n ” es el orden del filtro e “ i ” son los coeficientes 1 y 2 de cada etapa del filtro a implementar.

3.2.8 Procedimiento de Diseño del Filtro

De acuerdo con el autor Huircán Quilaqueo (2014) el procedimiento para el diseño del filtro es el siguiente:

Para una frecuencia de corte dada f_c y un $R \neq 1[\Omega]$:

- Se determina el orden del filtro n
- Se escogen de la tabla los coeficientes C_i/C o R_i/R
- Con cada coeficiente se calcula C_n o R_n usando la ecuación 53

3.2.9 Diseño de Filtro Butterworth Pasabajos

Aplicando el método expuesto, primeramente, se determinará el orden del filtro necesario que cumpla con las especificaciones de diseño:

- Máxima atenuación en la banda de paso $APB=1.3\text{dB}$
- Mínima atenuación en la banda de rechazo $ASB=3\text{dB}$
- Frecuencia de esquina de la banda de paso $f_{PB}=7\text{KHz}$
- Frecuencia de esquina de la banda de rechazo $f_{SB}=8\text{KHz}$
- Frecuencia de corte $f_c = 7.5\text{KHz}$
- Resistencia $R = 10\text{K}[\Omega]$

Se determina e_1 y e_2 para calcular el orden del filtro:

$$e_1 = \sqrt{10^{0.1APB} - 1} = \sqrt{10^{0.1(1.3)} - 1} = 0.590730803829 \quad (54)$$

$$e_2 = \sqrt{10^{0.1ASB} - 1} = \sqrt{10^{0.1(3)} - 1} = 0.997628345111 \quad (55)$$

De la ecuación 37 se tiene:

$$nb = \frac{\log\left(\frac{e1}{e2}\right)}{\log\left(\frac{fPB}{fSB}\right)} = \frac{\log\left(\frac{0.590730803829}{0.997628345111}\right)}{\log\left(\frac{7(10^3)}{8(10^3)}\right)} \cong 3.9243 \quad (56)$$

Por lo tanto, el filtro que se usará para satisfacer las exigencias del diseño será de 4° orden, como se aprecia en la Fig. 33.

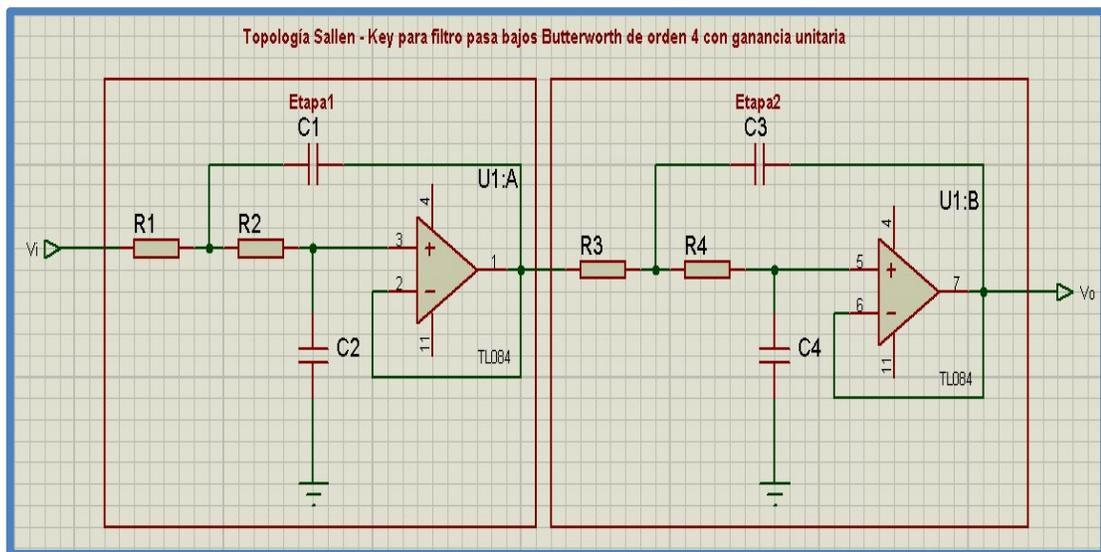


Fig. 33 Topología Sallen-Key Pasa Bajos Butterworth de Orden 4 con Ganancia Unitaria

Elaboración: los autores

Fuente: Software Proteus 8 Professional

Para $n = 4$, tenemos 2 etapas como indica la tabla 4, entonces aplicando ecuación 56, para cada etapa:

Etapas:

$$C1 = 1.082 \frac{1}{2\pi \cdot 7.5(10^3) \cdot 10(10^3)} = 2296 \text{pF} \cong 2200 \text{pF} \quad (57)$$

$$C2=0.9241 \frac{1}{2\pi 7.5(10^3)10(10^3)} = 1961\text{pF} \cong 2000\text{pF} \quad (58)$$

Etapa2:

$$C1=2.613 \frac{1}{2\pi 7.5(10^3)10(10^3)} = 5545\text{pF} \cong 5500\text{pF} \quad (59)$$

$$C2=0.3825 \frac{1}{2\pi 7.5(10^3)10(10^3)} \cong 812\text{pF} \cong 800\text{pF} \quad (60)$$

Por tanto, el circuito de la Fig. 34, quedaría de la siguiente manera:

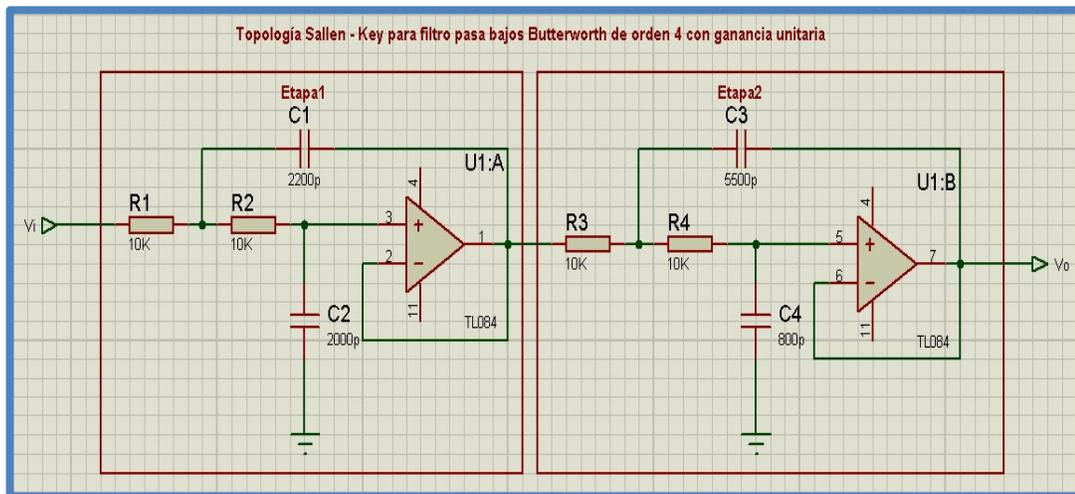


Fig. 34 Topología Sallen-Key Pasa Bajos Butterworth de Orden 4 con Ganancia Unitaria Diseñada para la Etapa de Preprocesamiento

Elaboración: los autores

Fuente: Software Proteus 8 Professional

Con la ayuda del simulador Proteus, se obtuvo la respuesta en frecuencia del filtro diseñado desde 10Hz-1MHz, la cual se muestra a continuación:

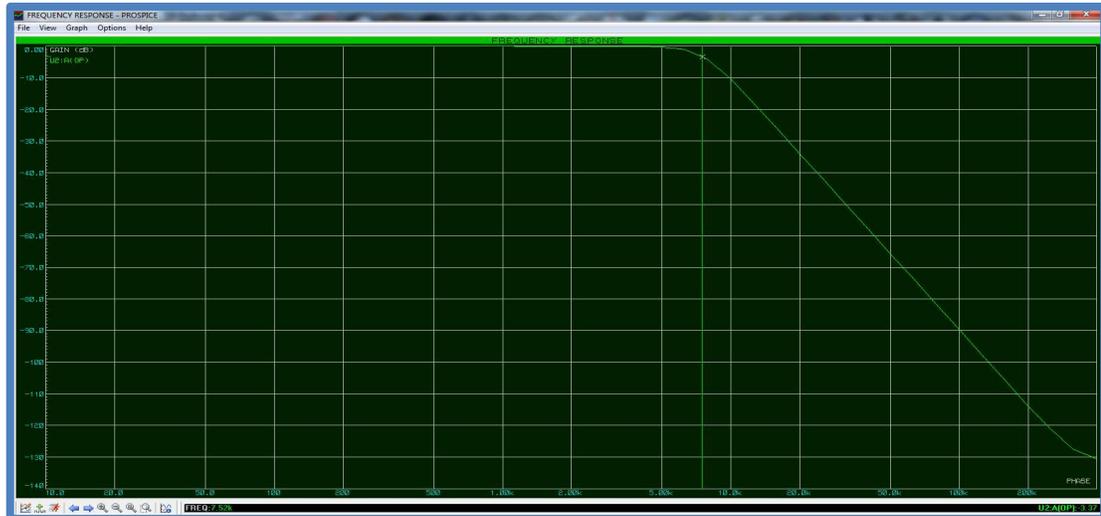


Fig. 35 Respuesta en Frecuencia del Filtro Diseñado a la Frecuencia de Corte de 7.5KHz

Elaboración: los autores

Fuente: Software Proteus 8 Professional

Se observa de la Fig. 35, que aproximadamente a la frecuencia de corte 7.5KHz hay una pérdida de -3.37dB, lo cual indica que la señal se ha reducido aproximadamente a $2.5119(10^{-40})$ de la señal original, lo cual indica una buena tasa de pérdida a la frecuencia de corte, la cual como mínimo debería ser de -3dB.

Para el análisis de la banda de transición 7KHz-8KHz, usaremos las siguientes gráficas obtenidas de la respuesta en frecuencia del filtro:

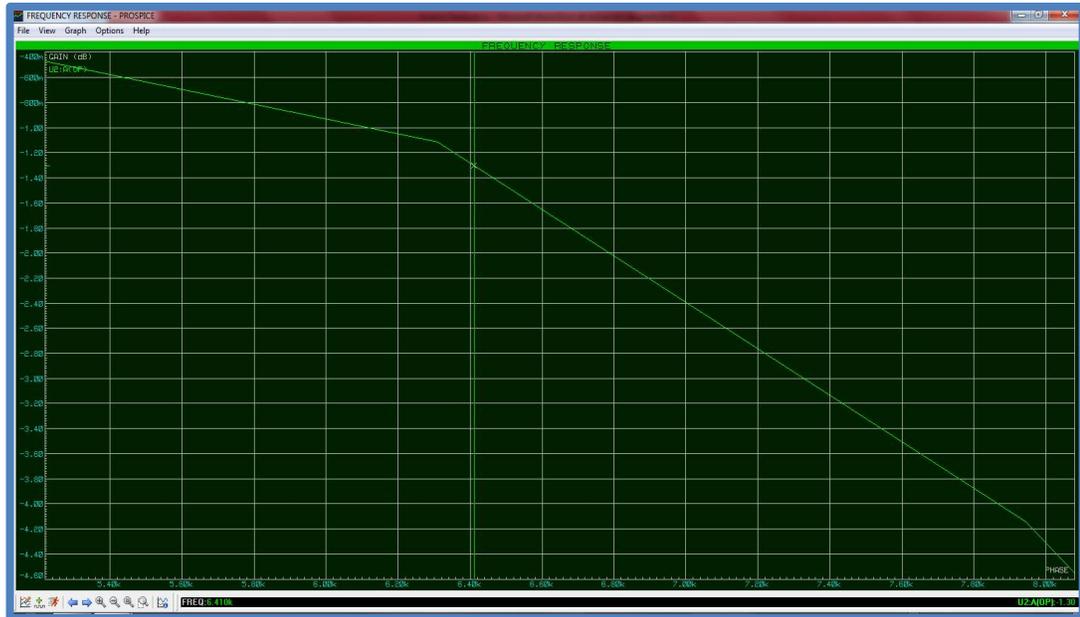


Fig. 36 Respuesta en Frecuencia del Filtro Diseñado a la Frecuencia de Esquina de la Banda de Paso – Aproximadamente $f_{PB}=6.410\text{KHz}$

Elaboración: los autores

Fuente: Software Proteus 8 Professional

Como se observa en la Fig. 36, a la frecuencia de esquina de la banda de paso 6.410KHz, hay una pérdida de la señal de -1.3dB, lo cual fue diseñado para una frecuencia de 7KHz, esto es debido a que se utilizan en la implementación valores de capacitores aproximados (comerciales) y no los valores exactos.

En la Fig.37, se observa que la tasa de pérdidas es de -4.32dB, este valor indica que el filtro es lo suficientemente selectivo para nuestra investigación, ya que se propuso que ASB=3dB como mínimo.

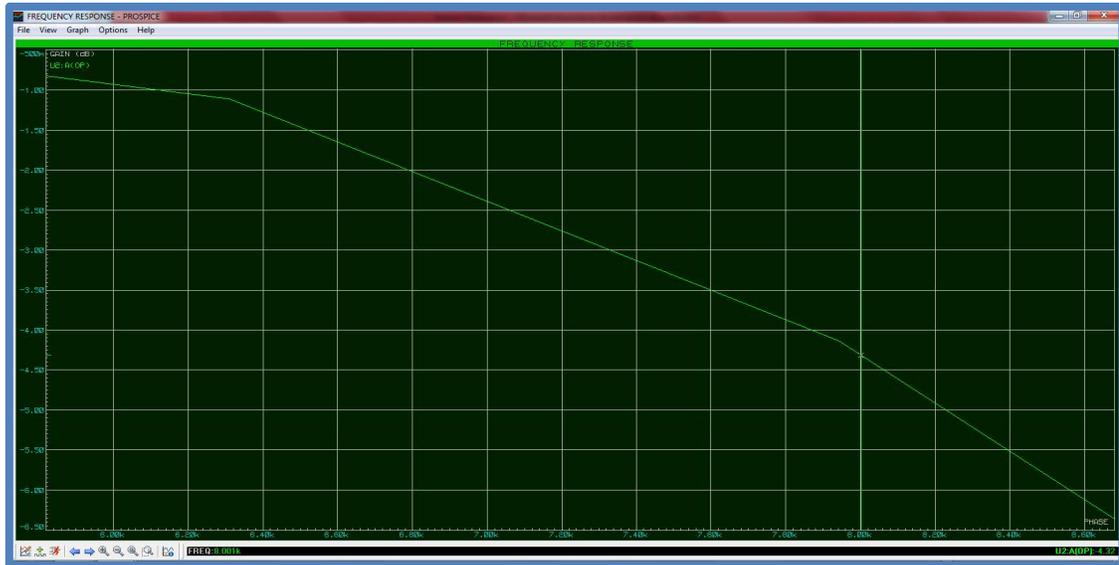


Fig. 37 Respuesta en Frecuencia del Filtro Diseñado a la Frecuencia de Esquina de la Banda de Paso - Aproximadamente $f_{SB}=8\text{KHz}$

Elaboración: los autores

Fuente: Software Proteus 8 Professional

Finalmente, se observa que en la banda de paso la reducción de la señal es mínima, en la Fig. 38 se muestra una comparación entre la señal de entrada y salida del filtro con una diferencia mínima de aproximadamente 60mV.

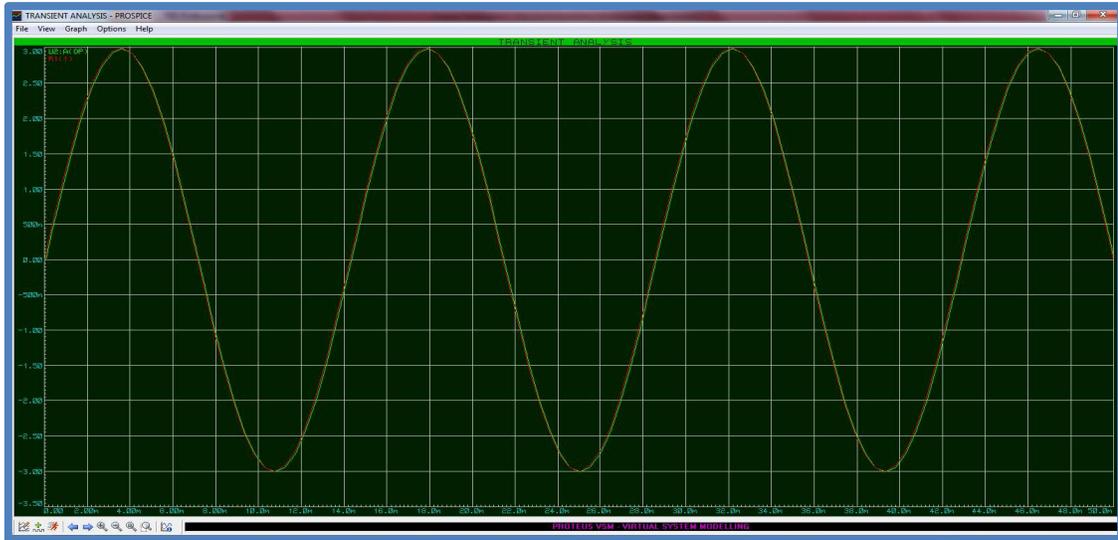


Fig. 38 Comparación entre la Señal de Entrada y Salida del Filtro

Elaboración: los autores

Fuente: Software Proteus 8 Professional

Con el filtro pasa bajos implementado ahora se pasa a explicar el proceso de cuantificación del conversor análogo-digital del micro procesador ATMEGA328.

3.2.10 Convertidor Análogo-Digital

Se tienen 1024(10 bits) intervalos de voltaje desde 0V hasta un máximo de 5V entregados por el sensor de etanol, por tanto, la cantidad de voltaje en cada intervalo será la siguiente:

$$V_{\text{intervalo}} = \frac{5V}{1024} \cong 4.8828\text{mV} \quad (61)$$

Por lo tanto, de la ecuación 64 se puede construir la siguiente tabla de conversión análogo-digital:

Tabla 5. Tabla Resumida de Conversión Análoga – Digital

Intervalo	Voltaje Análogo	Representación Digital
1	0V – 4.8828mV	0
2	4.8828mV – 9.7656mV	1
3	9.7657mV – 14.6484mV	2
4	14.6485mV – 19.5312mV	3
5	19.5313mV – 24.414mV	4
...
267	1.298828125V - 1.3037109375V	266
...
390	1.8994140625V - 1.904296875V	389
...
635	3.095703125V - 3.1005859375V	634
...
840	4.0966796875V - 4.1015625V	839
...
1024	4.9951171875V - 5V	1023

Elaboración: los autores

3.3 Etapa de Procesamiento

A continuación, se detalla la etapa de procesamiento diseñada para la presente investigación la cual consta de una topología de perceptrón multicapa.

3.3.1 Perceptron Multicapa (MLP)

La implementación de Red Neuronal en la tesis utilizo la topología del Perceptron Multicapa debido a que presenta ventajas sobre la clasificación de patrones y es un aproximador universal.

A continuación, se describen algunas características del perceptron multicapa según Galván León y Valls Ferrán (2015):

- Minsky y Papert (1969) demostraron que el perceptron simple y Adaline no pueden resolver problemas no lineales (como XOR).
- La combinación de varios perceptrones podría resolver ciertos problemas no lineales, pero no existía un mecanismo automático para adaptar los pesos de las capas ocultas.
- Rumelhart y otros autores (1986) presentaron la Regla Delta Generalizada para adaptar los pesos propagando los errores hacia atrás (retropropagación), para múltiples capas y funciones de activación no lineales.

- Se demostró que el Perceptrón Multicapa (MLP) es un APROXIMADOR UNIVERSAL.
- Un MLP puede aproximar relaciones no lineales entre datos de entrada y de salida.
- Es una de las arquitecturas más utilizadas en la resolución de problemas reales por ser un aproximador universal y por su fácil uso y aplicabilidad.
- Se ha aplicado con éxito en reconocimiento de voz, reconocimiento de imágenes, conducción de vehículos, diagnósticos médicos, etc.

Según los autores Galván León & Valls Ferrán (2015) los MLP son adecuados cuando:

- Las instancias se representan mediante muchos pares atributo-valor. Estos valores pueden estar correlados o ser independientes.
- La salida puede ser discreta, real o un vector de valores discretos o reales.
- Los ejemplos de entrenamiento pueden tener errores. Los MLP son muy robustos al ruido.
- Se permiten largos tiempos de entrenamiento.
- Se necesitan respuestas muy rápidas ante nuevas instancias.
- No es necesario entender o interpretar lo que ha aprendido el MLP.

La figura 39 expresa la estructura de una red neuronal de topología multicapa, la cual presenta tres tipos de capas: una de entrada, una oculta y una de salida, las cuales operaran de forma paralelela para adaptar los pesos y errores en dos sentidos:

- En sentido hacia adelante: Su objetivo es activar todas las entradas de las capas hasta la salida para poder obtener datos para corregir los pesos y errores que presentaron en la salida al comparar con la salida requerida.
- En sentido hacia atrás: Su objetivo es retropropagar los pesos y salidas con los valores obtenidos en el anterior sentido y poder adaptar los pesos y errores con nuevos valores que permitieron corregir la salida obtenida y aproximar lo más cercano a la salida requerida por la clasificación que se esté usando.

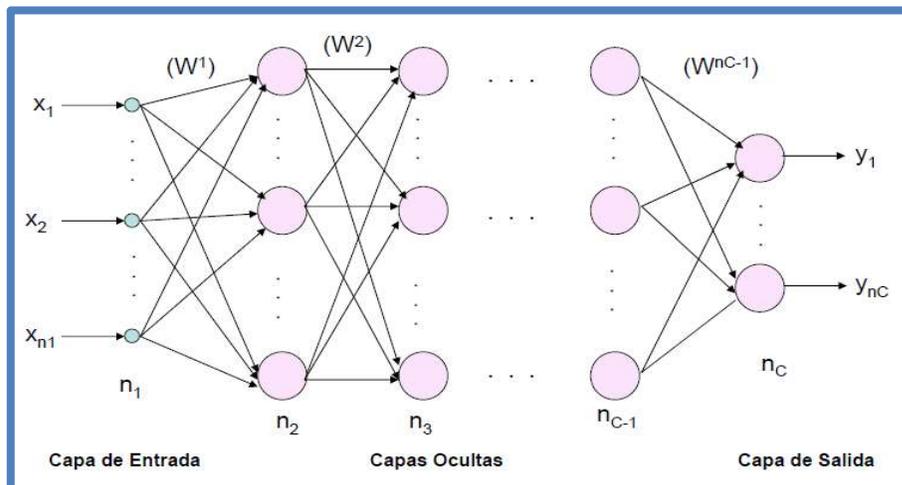


Fig. 39 Arquitectura Perceptron Multicapa

Fuente: Galván León y Valls Ferrán, 2015

3.3.1.1 Notación Matemática de la Propagación de los Patrones de Entrada en la Red Neuronal Multicapa

Los autores (Galván León y Valls Ferrán, 2015) afirman que las notaciones matemáticas de la figura 39 son las siguientes:

- C capas, n_c neuronas en la capa $C = 1, 2, \dots, C$
- Fig. 40, matriz de pesos de la capa C a la capa (c+1). W^C y el vector de umbrales de la capa U^C :

$$W^c = (w_{ij}^c) = \begin{pmatrix} w_{11}^c & w_{12}^c & \dots & w_{1n_{c+1}}^c \\ w_{21}^c & w_{22}^c & \dots & w_{2n_{c+1}}^c \\ \dots & \dots & \dots & \dots \\ w_{n_c1}^c & w_{n_c2}^c & \dots & w_{n_cn_{c+1}}^c \end{pmatrix} \quad U^c = (u_i^c) = \begin{pmatrix} u_1^c \\ u_2^c \\ \dots \\ u_{n_c}^c \end{pmatrix}$$

Fig. 40 Matrices de Pesos y Umbrales de la capa “C”

Fuente: Galván León y Valls Ferrán, 2015

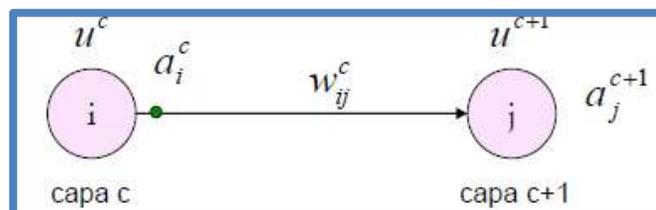


Fig. 41 Activación de la Neurona “i” de la Capa “C”

Fuente: Galván León y Valls Ferrán, 2015

Activación de las neuronas de entrada:

$$a_i^1 = X_i \text{ para } i=1,2,\dots, n_1 \quad (62)$$

Dónde:

$$X=(x_1+x_2+\dots+x_{n_1}) \quad (63)$$

representa el vector de entrada.

Activación de las neuronas de la capa oculta:

$$a_i^1 = f \left(\sum_{j=1}^{n_1-1} w_{ji}^{c-1} \cdot a_j^{c-1} + u_i^c \right) \quad (64)$$

para $i=1,2,\dots, n_c$ y $C=2,3,\dots, C-1$

Dónde:

a_j^{c-1} : Son las activaciones de las neuronas de la capa C-1, f: Es la función de activación.

Activación de las neuronas de la capa de salida:

$$y_i = a_i^C = f \left(\sum_{j=1}^{n_c-1} w_{ji}^{c-1} \cdot a_j^{c-1} + u_i^c \right) \quad (65)$$

para $i=1,2,\dots, n_c$

Dónde:

$y=(y_1+y_2+\dots+y_{n_c})$ representa el vector de salida de la red.

3.3.2 Función de Activación:

Dependiendo del problema que se requiera resolver se pueden utilizar diferentes funciones de activación, las funciones de activación más utilizadas son:

a) Función Sigmoide:

$$f_1(x) = \frac{1}{1 + e^{-x}} \quad (66)$$

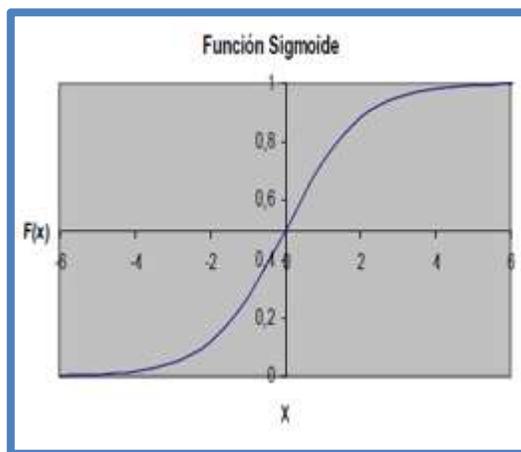


Fig. 42 Gráfico de la Función Sigmoide

Fuente: Galván León y Valls Ferrán, 2015

b) Tangente Hiperbólica

$$f_2(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (67)$$

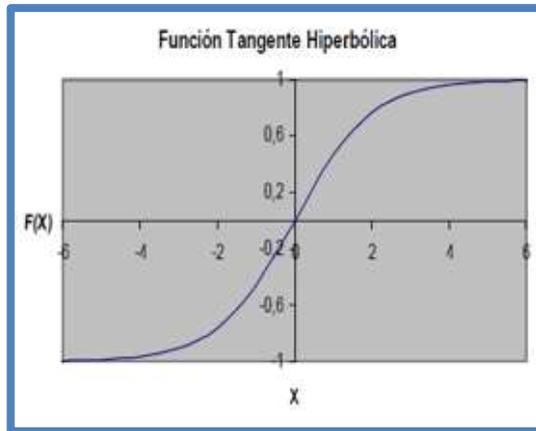


Fig. 43 Gráfico de la Función Tangente Hiperbólica

Fuente: Galván León y Valls Ferrán, 2015

Según los autores Galván León y Valls Ferrán (2015) estas funciones de transferencia presentan las siguientes características:

- Ambas son crecientes con dos niveles de saturación.
- Normalmente la función de activación es común a todas las neuronas.
- La elige el diseñador según el valor de activación que se desee $[-1,1]$ o $[0,1]$.
- Ambas funciones están relacionadas:

$$f_2(x) = 2f_1(x) - 1 \quad (68)$$

- El perceptron multicapa define, a través de sus conexiones y neuronas, una función continua no lineal del espacio R^{n1} en el espacio R^{nc}

$$Y = F(X, W) \quad (69)$$

3.3.2.2 Aprendizaje

Cómo indican los autores Galván León y Valls Ferrán (2015), el aprendizaje del Perceptrón Multicapa es un proceso iterativo supervisado: modificación paulatina de los parámetros de la red (pesos y umbrales) hasta que la salida de la red sea lo más próxima posible a la salida deseada o esperada para cada patrón de entrenamiento.

Dado: Conjunto de patrones.

- Vector de entrada: $x(n)=(x_1, x_2, x_3, \dots, x_n)$
- Vector de salida deseada: $s(n)$

Encontrar: Pesos W y umbrales U tales que:

$$s(n) \approx y(n) \quad \forall \text{ patrón } 'n' \leftrightarrow |s(n)-y(n)| \approx 0 \quad \forall 'n' \leftrightarrow \text{Minimizar } E = \sum |s(n)-y(n)| \quad (70)$$

El aprendizaje es equivalente a minimizar la función error E dada por:

$$E = \frac{1}{N} \sum_{n=1}^N e(n) \quad (71)$$

Error medio de “N” patrones

Dado que:

$$e(n) = \frac{1}{2} \sum_{i=1}^{n_c} (s_i(n) - y_i(n))^2 \quad (72)$$

Donde: s_i es la salida deseada; y_i es la salida obtenida

3.3.3 Regla Delta Generalizada o Algoritmo de Retropropagación

La regla delta generalizada o algoritmo de retropropagación (backpropagation en inglés) es una técnica que permite la disminución rápida en la función de error de la clasificación de patrones con los cuales se está entrenando la red neuronal, de esta manera el error de clasificación se reduce con cada ciclo de entrenamiento. Maza Rivera (2011) afirma:

Un algoritmo muy conocido que mejora el del Perceptron y permite un aprendizaje más rápido y campo de aplicación es el propuesto por Widrow y Hoff en 1960, denominado regla delta del mínimo error cuadrado (LMS Error: Least Mean Squared Error), también conocida como regla de Widrow - Hoff, que se aplicó en las redes desarrolladas por los mismos autores, conocidas como Adaline (Adaptive Linear Element), con una única neurona de salida y Madaline (Multiple Adaline), con varias neuronas de salida. Widrow y Hoff definieron una función que permitía cuantificar el error global cometido en cualquier momento durante el proceso de entrenamiento de la red, lo cual es importante, ya que cuanto más información se tenga sobre el error cometido, más rápido se puede aprender. (p.72)

3.3.3.1 Notación Matemática de la Regla Delta Generalizada

La forma en la que la regla delta generalizada calcula la variación de los pesos de la red neuronal es aplicando la gradiente negativa de la función de error cuadrático medio con respecto a los pesos de la red neuronal, multiplicado por la tasa de aprendizaje α :

$$\Delta w = -\alpha \frac{\partial e(n)}{\partial w} \quad (73)$$

Luego esta variación Δw es utilizada para la actualizar los pesos para el siguiente ciclo de entrenamiento:

$$w(n) = w(n-1) - \alpha \frac{\partial e(n)}{\partial w} \quad (74)$$

De esta manera se logra reducir el error de la salida de la red neuronal con respecto a los pesos en cada ciclo de entrenamiento.

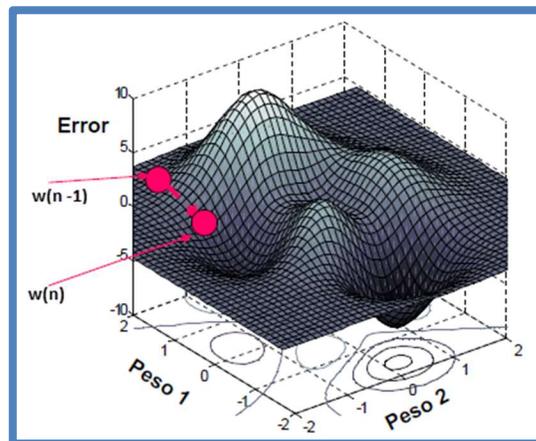


Fig. 44 Gráfico en el Hiperplano de la Adaptación de Pesos y Errores en la Red Neuronal

Fuente: Galván León y Valls Ferrán, 2015

Las siguientes notaciones mostraron matemáticamente como obtener los pesos de la última capa, penúltima capa y la retro propagando los errores para corregirlos y dar una correcta aproximación entre la salida obtenida y la salida requerida. Estas notaciones y demostraciones matemáticas desde el subcapítulo 3.3.2.2 hasta el 3.3.2.6, son propiedad intelectual de Galván León y Valls Ferrán (2015)

3.3.3.2 Pesos de la Última Capa

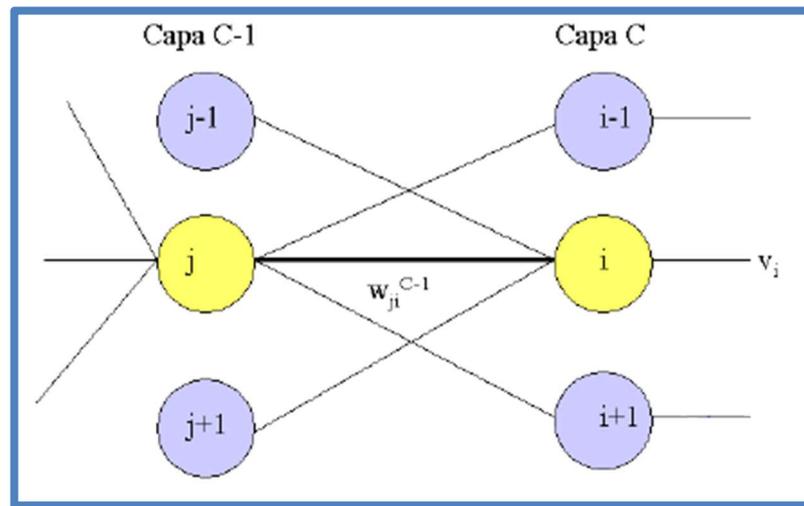


Fig. 45 Gráfico Representativo de la Última Capa de una Red Multicapa

Fuente: Galván León y Valls Ferrán, 2015

Dado que:

$$w_{ji}^{C-1}(n) = w_{ji}^{C-1}(n-1) - \alpha \frac{\partial e(n)}{\partial w_{ji}^{C-1}} \quad (75)$$

Dónde:

$$\frac{\partial e(n)}{\partial w_{ji}^{C-1}} = \frac{\partial e(n)}{\partial y_i(n)} \frac{\partial y_i(n)}{\partial w_{ji}^{C-1}} \quad (76)$$

El peso w_{ji}^{C-1} sólo afecta a y_i , el resto de salidas no depende de él.

$$e(n) = \frac{1}{2} \sum_{i=1}^{n_c} (s_i(n) - y_i(n))^2 = \frac{1}{2} (s_1(n) - y_1(n))^2 + \dots + \frac{1}{2} (s_i(n) - y_i(n))^2 + \dots \quad (77)$$

$$\frac{\partial e(n)}{\partial y_i(n)} = - (s_i(n) - y_i(n)) \quad (78)$$

Por lo tanto, quedaría la ecuación 76 como:

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} = - (s_i(n) - y_i(n)) \cdot \frac{\partial y_i(n)}{\partial w_{ji}^{c-1}} \quad (79)$$

Ahora calcular:

$$\frac{\partial y_i(n)}{\partial w_{ji}^{c-1}} \quad (80)$$

La salida y_i es:

$$y_i = a_i^c = f \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} \cdot a_j^{c-1} + u_i^c \right) \quad (81)$$

Por lo tanto la derivada de y_i :

$$\frac{\partial y_i(n)}{\partial w_{ji}^{c-1}} = f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} \cdot a_j^{c-1} + u_i^c \right) a_j^{c-1} \quad (82)$$

Derivando el error sería:

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} = - (s_i(n) - y_i(n)) \cdot f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} \cdot a_j^{c-1} + u_i^c \right) a_j^{c-1} \quad (83)$$

Se define el término δ asociado a la neurona i de la capa C:

$$\delta_i^c = (s_i(n) - y_i(n)) \cdot f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} \cdot a_j^{c-1} + u_i^c \right) \quad (84)$$

Reemplazando la ecuación 84 en la ecuación 83 quedara:

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} = -\delta_i^c \cdot a_j^{c-1} \quad (85)$$

Por lo tanto, el nuevo peso será:

$$w_{ji}^{c-1}(n) = w_{ji}^{c-1}(n-1) + \alpha \delta_i^c \cdot a_j^{c-1}(n) \quad (86)$$

Lo mismo ocurre con los umbrales de la última capa:

$$u_i^c(n) = u_i^c(n-1) + \alpha \delta_i^c(n) \quad (87)$$

Para modificar el peso w_{ji}^{c-1} basta considerar:

- La activación de la neurona origen j
- El término δ_i asociado a la neurona destino i

El término δ_i contiene el error medido en la neurona i de salida y la derivada de la activación de dicha neurona.

3.3.3.3 Pesos de la Penúltima Capa

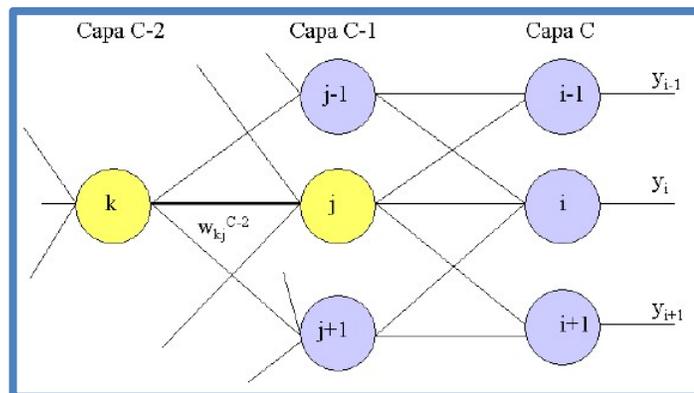


Fig. 46 Gráfico Representativo de la Penúltima Capa de una Red Multicapa

Fuente: Galván León y Valls Ferrán, 2015

3.3.3.4 Pesos de la Capa C-2 a la C-1

$$w_{kj}^{c-2}(n) = w_{kj}^{c-2}(n-1) - \alpha \frac{\partial e(n)}{\partial w_{kj}^{c-2}} \quad (88)$$

Ahora este peso afecta a todas las salidas.

$$e(n) = \frac{1}{2} \sum_{i=1}^{n_c} (s_i(n) - y_i(n))^2 = \frac{1}{2} (s_1(n) - y_1(n))^2 + \dots + \frac{1}{2} (s_i(n) - y_i(n))^2 + \dots \quad (89)$$

Todas las salidas y dependen del peso w_{kj}

Por lo tanto, la derivada del error será:

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = - \sum_{i=1}^{n_c} (s_i(n) - y_i(n)) \cdot \frac{\partial y_i(n)}{\partial w_{kj}^{c-2}} \quad (90)$$

Suma de las derivadas de cada una de las salidas de la red ahora hay que calcular la derivada de y_i respecto al peso w_{kj}^{c-2}

Como y_i es:

$$y_i = a_i^c = f \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} \cdot a_j^{c-1} + u_i^c \right) \quad (91)$$

Ahora la variable w_{kj}^{c-2} esta dentro del termino a_j

$$\frac{\partial y_i(n)}{\partial w_{kj}^{c-2}} = f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} \cdot a_j^{c-1} + u_i^c \right) \cdot w_{ji}^{c-1} \cdot \frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}} \quad (92)$$

Se sustituye la ecuación 92 en la ecuación 90:

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = - \sum_{i=1}^{n_c} (s_i(n) - y_i(n)) \cdot f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} \cdot a_j^{c-1} + u_i^c \right) \cdot w_{ji}^{c-1} \cdot \frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}} \quad (93)$$

De acuerdo a la definición de δ_i^c descrito en la ecuación 84, quedaría de la siguiente forma el error:

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = - \sum_{i=1}^{n_c} \delta_i^c \cdot w_{ji}^{c-1} \cdot \frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}} \quad (94)$$

La derivada de la activación de la neurona **j** será:

$$\frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}} = f' \left(\sum_{k=1}^{n_{c-2}} w_{ki}^{c-2} \cdot a_k^{c-2} + u_j^{c-1} \right) \cdot a_k^{c-2} \quad (95)$$

El único termino que depende de w_{kj} es el que sale de k

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = - \sum_{i=1}^{n_c} \delta_i^c \cdot w_{ji}^{c-1} \cdot f' \left(\sum_{k=1}^{n_{c-2}} w_{kj}^{c-2} \cdot a_k^{c-2} + u_j^{c-1} \right) \cdot a_k^{c-2} \quad (96)$$

Se define el término δ asociado a la neurona **j** de la capa C-1

$$\delta_j^{c-1} = \sum_{i=1}^{n_c} \delta_i^c \cdot w_{ji}^{c-1} \cdot f' \left(\sum_{k=1}^{n_{c-2}} w_{kj}^{c-2} \cdot a_k^{c-2} + u_j^{c-1} \right) \quad (97)$$

Suma de los términos δ de la siguiente capa multiplicados por los correspondientes pesos

Sustituyendo la ecuación 97 en la ecuación 96 y de acuerdo con la definición de δ asociado a la neurona **j** de la capa C-1:

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = -\delta_j^{c-1} \cdot a_k^{c-2} \quad (98)$$

Por lo tanto, el nuevo peso será:

$$w_{kj}^{c-2}(n) = w_{kj}^{c-2}(n-1) + \alpha \delta_j^{c-1}(n) \cdot a_k^{c-2}(n) \quad (99)$$

Lo mismo ocurre con los umbrales de la penúltima capa:

$$u_j^{c-1}(n) = u_j^{c-1}(n-1) + \alpha \delta_j^{c-1}(n) \quad (100)$$

Para modificar el peso w_{kj}^{c-2} basta considerar:

- La activación de la neurona origen **K**
- El término δ_i asociado a la neurona destino **J**

Generalizando la obtención de los pesos para cualquier capa C a la capa C+1 (C=1, 2...C-2).

$$w_{kj}^c(n) = w_{kj}^c(n-1) + \alpha \delta_j^{c+1}(n) \cdot a_k^c(n) \quad (101)$$

$$u_j^{c+1}(n) = u_j^{c+1}(n-1) + \alpha \delta_j^{c+1}(n) \quad (102)$$

Dónde $k=1, 2, \dots, n_c$; $j=1, 2, \dots, n_{c+1}$; $c=1, 2, \dots, C-2$

Basta tener en cuenta la activación de la que parte la conexión y el término δ de la neurona a la que llega la conexión.

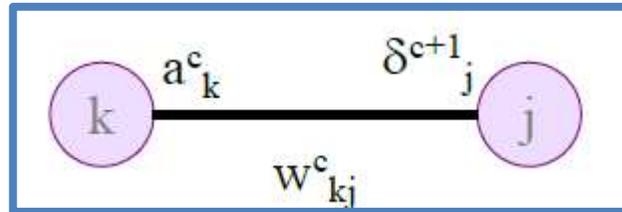


Fig. 47 Activación de la Neurona "J"

Fuente: (Galván León y Valls Ferrán, 2015)

El término δ de una neurona j se calcula utilizando la derivada de su función de activación y la suma de los términos δ de las neuronas de la siguiente capa.

$$\delta_j^{c+1} = \sum_{i=1}^{n_{c+2}} \delta_i^{c+2} \cdot w_{ji}^{c+1} \cdot f' \left(\sum_{k=1}^{n_c} w_{kj}^c \cdot a_k^c + u_j^{c+1} \right) \quad (103)$$

3.3.3.5 Derivada de la Función de Activación

Función sigmoideal

$$f_1(x) = \frac{1}{1 + e^{-x}} \quad (104)$$

Su derivada:

$$f'_1(x) = \frac{-1}{(1 + e^{-x})^2} \cdot (-e^{-x}) \quad (105)$$

$$f'_1(x) = \frac{1}{1 + e^{-x}} \cdot \frac{e^{-x}}{1 + e^{-x}} = \frac{1}{1 + e^{-x}} \cdot \left(\frac{1 + e^{-x} - 1}{1 + e^{-x}} \right) = \frac{1}{1 + e^{-x}} \cdot \left(\frac{1 + e^{-x}}{1 + e^{-x}} - \left(\frac{1}{1 + e^{-x}} \right) \right) \quad (106)$$

Donde finalmente, queda:

$$f'_1(x) = f_1(x) (1 - f_1(x)) \quad (107)$$

De la ecuación 84 simplificamos la notación, suprimimos la referencia al patrón n, (n):

$$\delta_i^c = (s_i - y_i) \cdot f' \left(\sum_{j=1}^{n_{c-1}} w_{ij}^{c-1} \cdot a_j^{c-1} + u_i^c \right) \quad (108)$$

Para la última capa resulta:

$$\delta_i^c = (s_i - y_i) \cdot y_i (1 - y_i) \quad (109)$$

Para el resto de las capas de la ecuación 103 resulta:

$$\delta_j^{c+1} = \sum_{i=1}^{n_{c+2}} \delta_i^{c+2} \cdot w_{ji}^{c+1} \cdot a_j^{c+1} (1 - a_j^{c+1}) \quad (110)$$

3.3.3.6 Resumen

- Cada neurona de salida distribuye hacia atrás su error (valor δ) a todas las neuronas ocultas que se conectan a ella ponderado por el valor del peso de la conexión.
- Así, cada neurona oculta recibe un cierto error (δ) de cada neurona de salida, siendo la suma el valor δ de la neurona oculta.
- Estos errores se van propagando hacia atrás, llegando a la primera capa.

Para la última capa:

$$w_{ji}^{c-1}(n) = w_{ji}^{c-1}(n-1) + \alpha \delta_i^c(n) \cdot a_j^{c-1}(n) \quad (111)$$

$$u_i^c(n) = u_i^c(n-1) + \alpha \delta_i^c(n) \quad (112)$$

Dónde de la ecuación 109:

$$\delta_i^c = (s_i - y_i) \cdot y_i (1 - y_i) \quad (113)$$

Para el resto de capas:

$$w_{kj}^c(n) = w_{kj}^c(n-1) + \alpha \delta_j^{c+1}(n) \cdot a_k^c(n) \quad (114)$$

$$u_j^{c+1}(n) = u_j^{c+1}(n-1) + \alpha \delta_j^{c+1}(n) \quad (115)$$

Dónde de la ecuación 110:

$$\delta_j^{c+1} = \sum_{i=1}^{n_{c+2}} \delta_i^{c+2} \cdot w_{ji}^{c+1} \cdot a_j^{c+1} (1 - a_j^{c+1}) \quad (116)$$

3.3.3.7 Razón de Aprendizaje y Momento

Al utilizar la regla delta generalizada se puede tener una situación en la cual el algoritmo se quede estancado en un mínimo local, punto en el cual aún no se ha alcanzado el mínimo máximo de la función de error, lo cual provoca que la clasificación de la red neuronal no sea tan adecuada como se espera. Galván León y Valls Ferrán (2015) afirman: “El cambio en el peso es proporcional al gradiente del error, siendo α (razón o tasa de aprendizaje) la constante de proporcionalidad” (p.37). Los mismos autores afirman que:

- Si α es grande, el error puede oscilar alrededor del mínimo.
- Si α es pequeña, la convergencia es más lenta.

Los autores Galván León y Valls Ferrán (2015) afirman: “Se puede modificar la ley de aprendizaje añadiendo un término llamado momento” (p.37).

$$w(n)=w(n-1)-\alpha \frac{\partial e(n)}{\partial w} +n\Delta w(n-1) \quad (117)$$

Donde:

$$\Delta w(n-1)=w(n-1)-w(n-2) \quad (118)$$

“Esta nueva ley conserva las propiedades de la ley anterior, pero incorpora cierta inercia que puede evitar oscilaciones en el error” (Galván León y Valls Ferrán, 2015, p.37).

De acuerdo con la Universidad Tecnológica de Pereira (2000) la variación de la tasa de aprendizaje α debe estar en concordancia con las siguientes reglas heurísticas:

Si el error cuadrático de todos los parámetros del set de entrenamiento se incrementa en un porcentaje ζ típicamente entre 1% y 5%, después de la actualización de los pesos, esa actualización es descartada, la tasa de aprendizaje se multiplica por un factor $0 < \rho < 1$, y el coeficiente de momentum es fijado en cero.

Si el error cuadrático se decrementa después de la actualización de los pesos, esa actualización es aceptada y la tasa de aprendizaje es multiplicada por un factor $\eta > 1$. Si γ (coeficiente de momentum) había sido previamente puesto en cero, se retorna a su valor original.

Si el error cuadrático se incrementa en un valor menor a ζ , los pesos actualizados son aceptados, pero la tasa de aprendizaje y el coeficiente de momentum no son cambiados (p.110).

Estas reglas heurísticas se utilizarán para el entrenamiento de la red neuronal de la presente investigación.

3.3.3.8 Capacidad de Generalización

La capacidad de generalización de una red neuronal se refiere a capacidad que posee dicha red de clasificar correctamente patrones con los cuales no fue entrenada.

Los autores Galván León y Valls Ferrán (2015) sostienen lo siguiente acerca de la capacidad de generalización de una red neuronal

- No solo es importante saber si la red se adapta a los patrones de entrenamiento, sino que hay que conocer cómo se comportará ante patrones no utilizados en el entrenamiento.
- Hay que evaluar la capacidad de generalización con un conjunto de datos diferente al utilizado para el entrenamiento.
- A veces, es necesario exigir menor ajuste a los datos de entrenamiento para obtener mejor generalización.
- Para poder hacer esto, es necesario evaluar la capacidad de generalización a la vez que se realiza el entrenamiento, no sólo al final.
- Cada cierto número de ciclos de entrenamiento se pasa el conjunto de test o (validación) (sin ajustar pesos) para medir la capacidad de generalización.
- Así puede medirse tanto la evolución del error de entrenamiento como la de test (o validación).

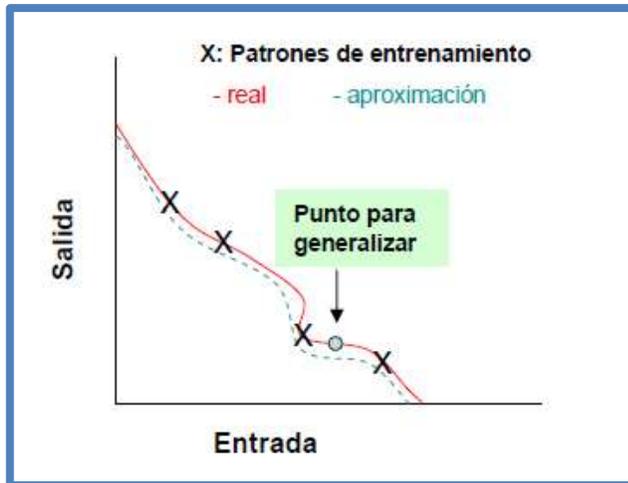


Fig. 48 Buena Capacidad de Generalización

Fuente: Galván León y Valls Ferrán, 2015

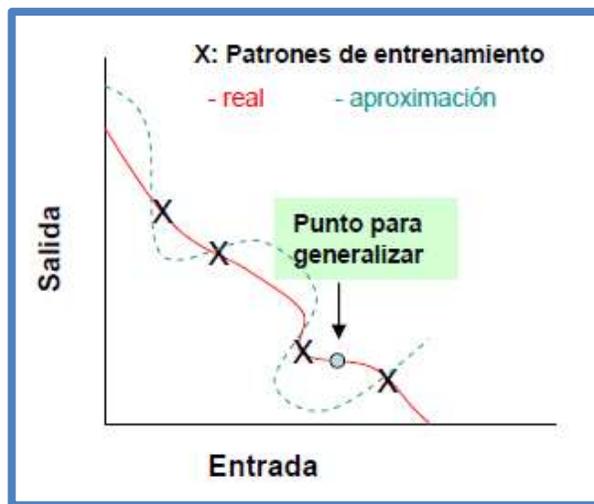


Fig. 49 Mala Capacidad de Generalización

Fuente: Galván León y Valls Ferrán, 2015

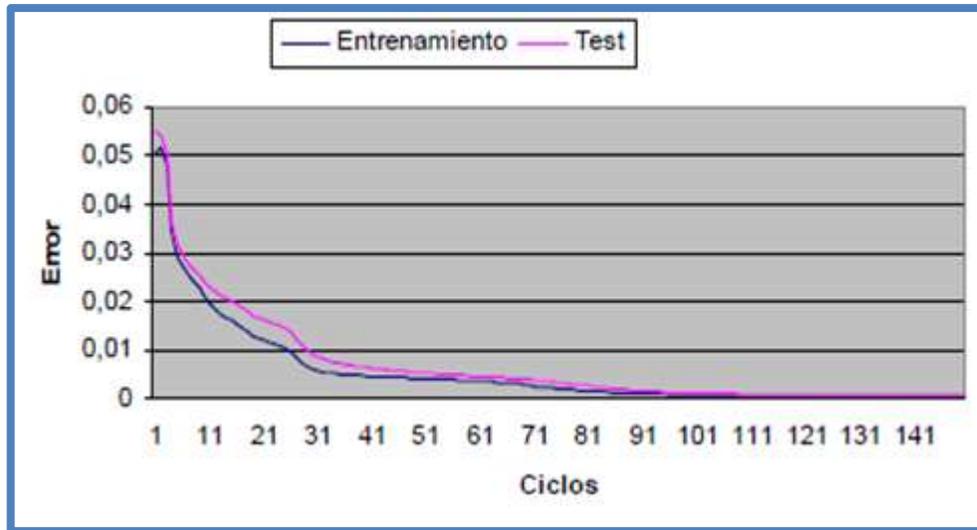


Fig. 50 Buen Nivel de Generalización - Error vs Ciclos

Fuente: Galván León y Valls Ferrán, 2015

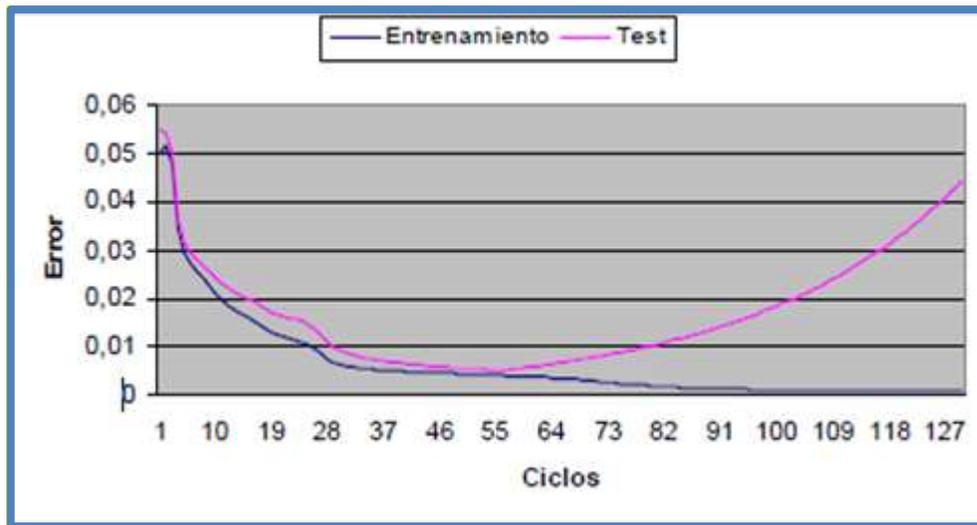


Fig. 51 Mal Nivel de Generalización – Error vs Ciclos

Fuente: Galván León y Valls Ferrán, 2015

3.3.3.9 Proceso de Aprendizaje en el MLP

Los autores Galván León y Valls Ferrán (2015), indican el proceso de aprendizaje del MLP en los siguientes siete pasos:

1. Se inicializan los pesos y umbrales de la red (valores aleatorios próximos a 0).
2. Se presenta un patrón (n) de entrenamiento, $(x(n), s(n))$, y se propaga hacia la salida, obteniéndose la respuesta de la red $y(n)$.
3. Se evalúa el error cuadrático, $e(n)$, cometido por la red para el patrón n .
4. Se aplica la regla delta generalizada para modificar los pesos y umbrales de la red neuronal.
 - 4.1. Se calculan los valores δ para todas las neuronas de la capa de salida.
 - 4.2. Se calculan los valores δ para el resto de las neuronas de la red, empezando desde la última capa oculta y retro propagando dichos valores hacia la capa de entrada.
 - 4.3. Se modifican pesos y umbrales.
5. Se repiten los pasos 2, 3 y 4 para todos los patrones de entrenamiento, completando así un ciclo de aprendizaje.
6. Se evalúa el error total E cometido por la red. Es el error de entrenamiento. Puede medirse también el error medio o el error acumulado para todos los patrones.
7. Se repiten los pasos 2, 3, 4, 5 y 6 hasta alcanzar un mínimo del error de entrenamiento, para lo cual se realizan m ciclos de aprendizaje. Puede utilizarse otro criterio de parada:
 - 7.1. El error de entrenamiento permanece estable.
 - 7.2. El error de validación permanece estable.
 - 7.3. El error de validación empieza a aumentar.

3.3.4 Diseño de la Arquitectura de Red Multicapa

Se tomó en cuenta nueve puntos principales para diseñar la arquitectura de la red neuronal para la clasificación de patrones de ebriedad. A continuación, se detallan estos puntos:

a) Target Error

Se seleccionó un valor de 10^{-4} para los resultados de clasificación de la red neuronal.

b) Función de Activación

Se implementó la función de activación sigmoideal en la red neuronal, la cual trabaja en el rango [0, 1], para la clasificación de patrones de ebriedad, para obtener salidas binarias.

c) Número de Neuronas de Entrada y de Salida

Se utilizó 11 neuronas en la capa de entrada las cuales representan los siguientes parámetros de entrada:

- X1: Tipo de transporte (Particular=0 Público=1).
- X2: ¿El conductor participó en un accidente de tránsito? (No=0 Si=1).
- X3-X10: BAC binario en 8 bits, medido por el sensor MQ3.
- X11: ¿El accidente dejó muertos o heridos? (No=0 Si=1; sólo podrá ser 1 si el conductor participó en un accidente de tránsito X2=1, Si X2=0 entonces X11=0 siempre).

Se utilizó 2 bits de salida (Y1 Y2) para representar los cuatro primeros niveles de embriaguez de la Tabla 2 del capítulo II. Esto porque el sensor MQ3 puede medir el BAC en el rango de 0.00 a 1.99 gr/L:

- Y1=0, Y2=0; Clasificación: Normal
- Y1=0, Y2=1; Clasificación: Euforia
- Y1=1, Y2=0; Clasificación: Excitación y/o embriaguez

- Y1=1, Y2=1; Clasificación: Confusión

Se utilizó 3 bits de salida (Y3 Y4 Y5) para representar la falta cometida por el conductor, de acuerdo con la tabla de infracciones del reglamento nacional de tránsito, obtenido de la página de la SAT (Servicio de Administración Tributaria de Lima) tal y como se muestra en las figuras 2 y 3 del capítulo 2:

- Y3=0, Y4=0, Y5=0; Falta: Ninguna, Todo en orden
- Y3=0, Y4=0, Y5=1; Falta: M37
- Y3=0, Y4=1, Y5=0; Falta: M38
- Y3=0, Y4=1, Y5=1; Falta: M02
- Y3=1, Y4=0, Y5=0; Falta: M01
- Y3=1, Y4=0, Y5=1; Falta: M39

d) Número de Neuronas en la Capa Oculta

Se utilizó la regla general “Rule of Thumbs” (Escuela Superior de Computo - Instituto Politécnico Nacional de México DF, 2009):

$$h = \frac{2(n+m)}{3} \quad (119)$$

Donde h: Número de neuronas de la capa oculta; n: Número de Entradas; m: Número de salidas

Siempre y cuando el número de neuronas en la capa oculta (h) no sea más de dos veces el número de entradas (n)

$$h < 2n \quad (120)$$

Para la arquitectura propuesta los valores son n=11 y m=5

Se calcula h (Número de neuronas de la capa oculta):

$$h = \frac{2(11+5)}{3} \approx 11 \quad (121)$$

También se comprueba que:

$$11 < 2(11) \quad (122)$$

Por tanto, el número de neuronas de la capa oculta que se escogió es 11

e) Tasa de Aprendizaje

La tasa de aprendizaje seleccionada para el presente proyecto fue 0.1

f) Factor de Amortiguación de la Tasa de Aprendizaje

El factor de amortiguación de la tasa de aprendizaje seleccionado para el presente proyecto fue 0.5

g) Factor de Incremento de la Tasa de Aprendizaje

El factor de incremento de la tasa de aprendizaje seleccionado para el presente proyecto fue 1.5

h) Porcentaje de Incremento del Error Cuadrático Medio

El porcentaje de incremento del error cuadrático medio seleccionado para el presente proyecto fue 3%

i) Coeficiente de Momentum

El coeficiente de momentum seleccionado para el presente proyecto fue 0.1

A continuación, se ilustra la topología de la red neuronal diseñada:

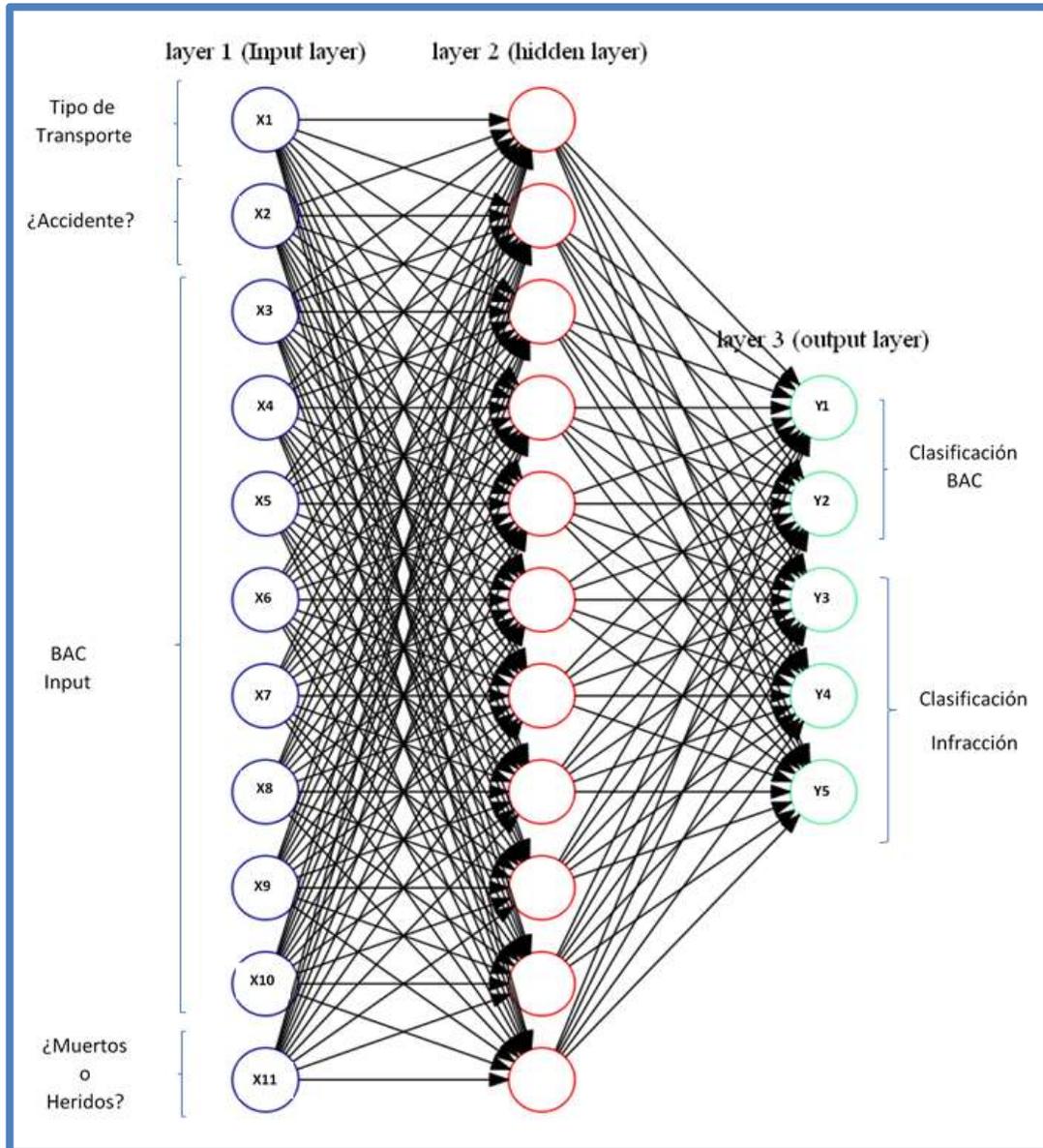


Fig. 52 Topología de La Red Neuronal Diseñada

Elaboración: los autores

Fuente: Software Graphviz

3.3.5 Vectores de Entrada y Salida

Para entrenar la red neuronal, se programó un código para generar las matrices de entrada y de salida esperada.

Para el caso de la matriz de entrada se generaron todas las combinaciones posibles (1200) entre las entradas X1-X11 Y para la matriz de salida su respectiva clasificación tanto del BAC (Y1 Y2) y de la sanción respectiva (Y3 Y4 Y5)

- El diagrama de flujo del programa para generar la matriz de datos de entrada y de salida deseada de la red neuronal se encuentra en el **ANEXO 2**
- El código en Matlab para generar la matriz de datos de entrada y de salida deseada de la red neuronal se encuentra en el **ANEXO 3**
- La matriz de datos de entrada de la red neuronal se almacenó en el archivo **inputs_BAC_Classification_VFINAL.mat - ANEXO 4**
- La matriz de datos de salida deseada se almacenó en el archivo **outputs_BAC_Classification_VFINAL.mat - ANEXO 5**

3.3.6 Algoritmo de Retropropagación

Para el algoritmo de retro-propagación se utilizó la función de activación sigmoideal, cuyo ajuste de pesos y umbrales está definido por las ecuaciones del 114 al 119.

A continuación, definiremos las matrices implicadas en el algoritmo neuronal:

Vector delta de la capa oculta:

$$\begin{aligned} & \text{Vector } d_2 \\ & [d_{(1,1)}^2 \ d_{(2,1)}^2 \ d_{(3,1)}^2 \ d_{(4,1)}^2 \ d_{(5,1)}^2 \ d_{(6,1)}^2 \ d_{(7,1)}^2 \ d_{(8,1)}^2 \ d_{(9,1)}^2 \ d_{(10,1)}^2 \ d_{(11,1)}^2] \end{aligned} \quad (123)$$

Vector delta de la capa de salida:

$$\begin{aligned} & \text{Vector } d_3 \\ & [d_{(1,1)}^3 \ d_{(2,1)}^3 \ d_{(3,1)}^3 \ d_{(4,1)}^3 \ d_{(5,1)}^3] \end{aligned} \quad (124)$$

Vector de activación de la capa de entrada:

$$\begin{aligned} & \text{Vector } a_1 \\ & [a_{(1,1)}^1 \ a_{(1,2)}^1 \ a_{(1,3)}^1 \ a_{(1,4)}^1 \ a_{(1,5)}^1 \ a_{(1,6)}^1 \ a_{(1,7)}^1 \ a_{(1,8)}^1 \ a_{(1,9)}^1 \ a_{(1,10)}^1 \ a_{(1,11)}^1] \end{aligned} \quad (125)$$

Vector de activación de la capa oculta:

$$\begin{aligned} & \text{Vector } a_2 \\ & [a_{(1,1)}^2 \ a_{(1,2)}^2 \ a_{(1,3)}^2 \ a_{(1,4)}^2 \ a_{(1,5)}^2 \ a_{(1,6)}^2 \ a_{(1,7)}^2 \ a_{(1,8)}^2 \ a_{(1,9)}^2 \ a_{(1,10)}^2 \ a_{(1,11)}^2] \end{aligned} \quad (126)$$

Vector de Salida:

$$\begin{aligned} & \text{Vector } y \\ & [y_{(1,1)} \ y_{(1,2)} \ y_{(1,3)} \ y_{(1,4)} \ y_{(1,5)}] \end{aligned} \quad (127)$$

Matriz de pesos de la capa de entrada:

Matriz W_1

$$\begin{aligned}
 &W_{(1,1)}^1 \ W_{(1,2)}^1 \ W_{(1,3)}^1 \ W_{(1,4)}^1 \ W_{(1,5)}^1 \ W_{(1,6)}^1 \ W_{(1,7)}^1 \ W_{(1,8)}^1 \ W_{(1,9)}^1 \ W_{(1,10)}^1 \ W_{(1,11)}^1; \\
 &W_{(2,1)}^1 \ W_{(2,2)}^1 \ W_{(2,3)}^1 \ W_{(2,4)}^1 \ W_{(2,5)}^1 \ W_{(2,6)}^1 \ W_{(2,7)}^1 \ W_{(2,8)}^1 \ W_{(2,9)}^1 \ W_{(2,10)}^1 \ W_{(2,11)}^1; \\
 &W_{(3,1)}^1 \ W_{(3,2)}^1 \ W_{(3,3)}^1 \ W_{(3,4)}^1 \ W_{(3,5)}^1 \ W_{(3,6)}^1 \ W_{(3,7)}^1 \ W_{(3,8)}^1 \ W_{(3,9)}^1 \ W_{(3,10)}^1 \ W_{(3,11)}^1; \\
 &W_{(4,1)}^1 \ W_{(4,2)}^1 \ W_{(4,3)}^1 \ W_{(4,4)}^1 \ W_{(4,5)}^1 \ W_{(4,6)}^1 \ W_{(4,7)}^1 \ W_{(4,8)}^1 \ W_{(4,9)}^1 \ W_{(4,10)}^1 \ W_{(4,11)}^1; \\
 &W_{(5,1)}^1 \ W_{(5,2)}^1 \ W_{(5,3)}^1 \ W_{(5,4)}^1 \ W_{(5,5)}^1 \ W_{(5,6)}^1 \ W_{(5,7)}^1 \ W_{(5,8)}^1 \ W_{(5,9)}^1 \ W_{(5,10)}^1 \ W_{(5,11)}^1; \\
 &W_{(6,1)}^1 \ W_{(6,2)}^1 \ W_{(6,3)}^1 \ W_{(6,4)}^1 \ W_{(6,5)}^1 \ W_{(6,6)}^1 \ W_{(6,7)}^1 \ W_{(6,8)}^1 \ W_{(6,9)}^1 \ W_{(6,10)}^1 \ W_{(6,11)}^1; \\
 &W_{(7,1)}^1 \ W_{(7,2)}^1 \ W_{(7,3)}^1 \ W_{(7,4)}^1 \ W_{(7,5)}^1 \ W_{(7,6)}^1 \ W_{(7,7)}^1 \ W_{(7,8)}^1 \ W_{(7,9)}^1 \ W_{(7,10)}^1 \ W_{(7,11)}^1; \\
 &W_{(8,1)}^1 \ W_{(8,2)}^1 \ W_{(8,3)}^1 \ W_{(8,4)}^1 \ W_{(8,5)}^1 \ W_{(8,6)}^1 \ W_{(8,7)}^1 \ W_{(8,8)}^1 \ W_{(8,9)}^1 \ W_{(8,10)}^1 \ W_{(8,11)}^1; \\
 &W_{(9,1)}^1 \ W_{(9,2)}^1 \ W_{(9,3)}^1 \ W_{(9,4)}^1 \ W_{(9,5)}^1 \ W_{(9,6)}^1 \ W_{(9,7)}^1 \ W_{(9,8)}^1 \ W_{(9,9)}^1 \ W_{(9,10)}^1 \ W_{(9,11)}^1; \\
 &W_{(10,1)}^1 \ W_{(10,2)}^1 \ W_{(10,3)}^1 \ W_{(10,4)}^1 \ W_{(10,5)}^1 \ W_{(10,6)}^1 \ W_{(10,7)}^1 \ W_{(10,8)}^1 \ W_{(10,9)}^1 \ W_{(10,10)}^1 \ W_{(10,11)}^1; \\
 &W_{(11,1)}^1 \ W_{(11,2)}^1 \ W_{(11,3)}^1 \ W_{(11,4)}^1 \ W_{(11,5)}^1 \ W_{(11,6)}^1 \ W_{(11,7)}^1 \ W_{(11,8)}^1 \ W_{(11,9)}^1 \ W_{(11,10)}^1 \ W_{(11,11)}^1
 \end{aligned} \tag{128}$$

Matriz de pesos de la oculta:

Matriz W_2

$$\begin{aligned}
 &W_{(1,1)}^2 \ W_{(1,2)}^2 \ W_{(1,3)}^2 \ W_{(1,4)}^2 \ W_{(1,5)}^2 \\
 &W_{(2,1)}^2 \ W_{(2,2)}^2 \ W_{(2,3)}^2 \ W_{(2,4)}^2 \ W_{(2,5)}^2 \\
 &W_{(3,1)}^2 \ W_{(3,2)}^2 \ W_{(3,3)}^2 \ W_{(3,4)}^2 \ W_{(3,5)}^2 \\
 &W_{(4,1)}^2 \ W_{(4,2)}^2 \ W_{(4,3)}^2 \ W_{(4,4)}^2 \ W_{(4,5)}^2 \\
 &W_{(5,1)}^2 \ W_{(5,2)}^2 \ W_{(5,3)}^2 \ W_{(5,4)}^2 \ W_{(5,5)}^2 \\
 &W_{(6,1)}^2 \ W_{(6,2)}^2 \ W_{(6,3)}^2 \ W_{(6,4)}^2 \ W_{(6,5)}^2 \\
 &W_{(7,1)}^2 \ W_{(7,2)}^2 \ W_{(7,3)}^2 \ W_{(7,4)}^2 \ W_{(7,5)}^2 \\
 &W_{(8,1)}^2 \ W_{(8,2)}^2 \ W_{(8,3)}^2 \ W_{(8,4)}^2 \ W_{(8,5)}^2 \\
 &W_{(9,1)}^2 \ W_{(9,2)}^2 \ W_{(9,3)}^2 \ W_{(9,4)}^2 \ W_{(9,5)}^2 \\
 &W_{(10,1)}^2 \ W_{(10,2)}^2 \ W_{(10,3)}^2 \ W_{(10,4)}^2 \ W_{(10,5)}^2 \\
 &W_{(11,1)}^2 \ W_{(11,2)}^2 \ W_{(11,3)}^2 \ W_{(11,4)}^2 \ W_{(11,5)}^2
 \end{aligned} \tag{129}$$

Matriz de umbrales de la capa oculta:

$$\begin{aligned} & \text{Vector } U_2 \\ & U_{(1,1)}^2 \ U_{(1,2)}^2 \ U_{(1,3)}^2 \ U_{(1,4)}^2 \ U_{(1,5)}^2 \ U_{(1,6)}^2 \ U_{(1,7)}^2 \ U_{(1,8)}^2 \ U_{(1,9)}^2 \ U_{(1,10)}^2 \ U_{(1,11)}^2 \end{aligned} \quad (130)$$

Matriz de umbrales de la capa de salida:

$$\begin{aligned} & \text{Vector } U_3 \\ & U_{(1,1)}^3 \ U_{(1,2)}^3 \ U_{(1,3)}^3 \ U_{(1,4)}^3 \ U_{(1,5)}^3 \end{aligned} \quad (131)$$

La activación de una neurona de la capa “c” es igual a la sumatoria de todos los pesos por las activaciones de la capa anterior con respecto a dicha neurona, más el umbral de dicha neurona, todo ello evaluado en la función de activación.

Si quisiéramos calcular a_1^2 , que es la activación de la neurona 1 de la capa 2, la cual está definida en la ecuación 129 como el elemento $a_{(1,1)}^2$ del vector a_2 , entonces de acuerdo con lo explicado en la ecuación 94 sería:

$$\begin{aligned} a_{(1,1)}^2 = & f(w_{(1,1)}^1 a_{(1,1)}^1 + w_{(2,1)}^1 a_{(1,2)}^1 + w_{(3,1)}^1 a_{(1,3)}^1 + w_{(4,1)}^1 a_{(1,4)}^1 \\ & + w_{(5,1)}^1 a_{(1,5)}^1 + w_{(6,1)}^1 a_{(1,6)}^1 + w_{(7,1)}^1 a_{(1,7)}^1 \\ & + w_{(8,1)}^1 a_{(1,8)}^1 + w_{(9,1)}^1 a_{(1,9)}^1 + w_{(10,1)}^1 a_{(1,10)}^1 \\ & + w_{(11,1)}^1 a_{(1,11)}^1 + U_{(1,1)}^2) \end{aligned} \quad (132)$$

Lo mismo sucedería para todas las activaciones de la red neuronal, por lo que podemos definir una relación matricial entre las ecuaciones 129, 128, 131, 133 y 130, 129, 132, 134, de la siguiente manera:

$$a_2 = f(a_1 \cdot W_1 + U_2) \quad (133)$$

$$y = f(a_2 \cdot W_2 + U_3) \quad (134)$$

Donde la “f” es la función de activación.

Para el caso de la actualización de pesos y umbrales, recordando las ecuaciones de actualización de pesos y umbrales 114, 115, 117 y 118, haciendo el mismo análisis que en el caso anterior, se puede establecer una relación entre las matrices de pesos y umbrales, de modo que se tiene:

$$W_1 = W_1 + \alpha \cdot d_2 \cdot (a_1)^T + \text{momentum} \cdot \Delta W_1 \quad (135)$$

$$U_2 = U_2 + \alpha \cdot (d_2)^T \quad (136)$$

$$W_2 = W_2 + \alpha \cdot d_3 \cdot (a_2)^T + \text{momentum} \cdot \Delta W_2 \quad (137)$$

$$U_3 = U_3 + \alpha \cdot (d_3)^T \quad (138)$$

Dónde α : Tasa de aprendizaje; **momentum**: Tasa de momentum que se utilizó para evitar caídas en mínimos locales; ΔW_1 : Diferencia de la matriz de pesos de la capa de entrada con su valor anterior $W_1(t) - W_1(t-1)$; ΔW_2 : Diferencia de la matriz de pesos de la capa de salida con su valor anterior $W_2(t) - W_2(t-1)$.

Habiendo expresado el cálculo de las activaciones de todas las neuronas de la red, la actualización de pesos y la de umbrales, se ha logrado aumentar la eficiencia computacional del algoritmo implementado.

- El diagrama de flujo del algoritmo de retropropagación se encuentra en el **ANEXO 6**
- El código en Matlab del algoritmo de retropropagación se encuentra en el **ANEXO 7**

3.3.7 Resultados

Se utilizó el algoritmo de retropropagación con todos los elementos explicados anteriormente, después de 10000 épocas de entrenamiento se obtuvimos los siguientes resultados:

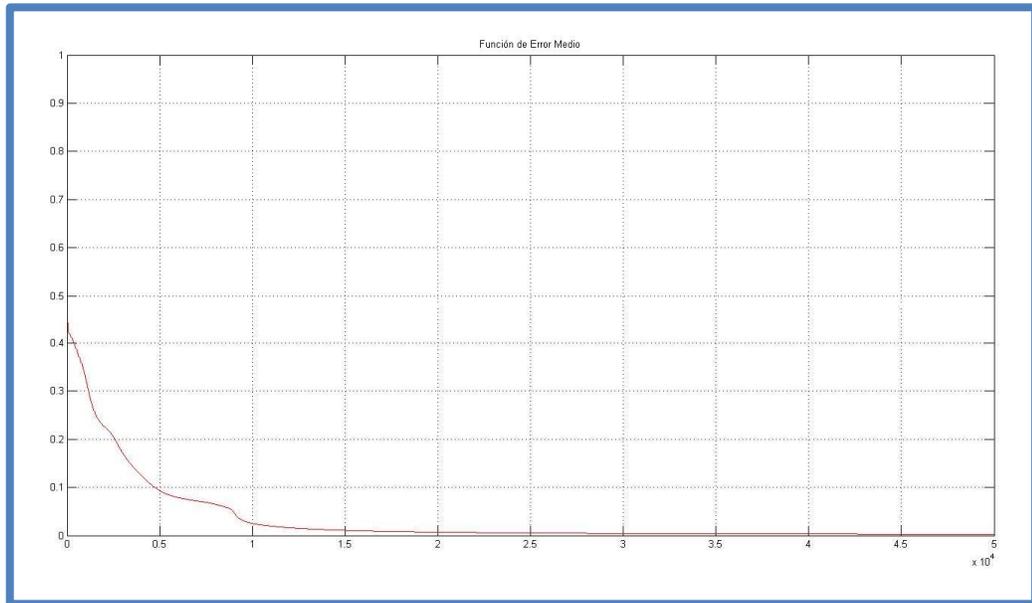


Fig. 53 Función de Error Medio

Elaboración: los autores

Fuente: Software Matlab

Se aprecia del resultado que el error medio después de los ciclos de entrenamiento es aceptable aprox. 10^{-4} como fue diseñado.

Con la etapa de procesamiento culminada con éxito, ahora se pueden utilizar los pesos y umbrales obtenidos para la clasificación de los patrones de ebriedad con un error muy bajo.

A continuación, se muestran los pesos y umbrales obtenidos después del entrenamiento:

```
W1 =  
  
Columns 1 through 9  
  
 1.7491  -0.0658  -14.2000  -0.8163  -17.3728   2.3562  14.7824  -1.9312  -0.0563  
18.6949  -2.1051  -11.4640  -30.7804  16.3577  -9.9862  -31.2864   7.6944   6.1459  
14.3352  -12.5651  -19.0077  18.9232  -17.2265  27.1463   6.3839   3.4487  29.3517  
11.2254  -42.7427  -23.3566  20.7556  -16.5836   0.0506   8.2302  -3.4888  -8.1118  
 9.3199  -42.2215  -9.7718   7.2552  -1.8116   6.1908  19.6014  -10.9269   0.2720  
 5.5115  -20.7200  -1.0363   6.9622  -2.2581   6.6336  10.4988   8.1174  -11.6410  
-0.7607  -12.6999  -3.5315   0.0792   0.9668   5.6806   6.5160   8.5334  -11.8782  
-0.7352  -6.3339  -3.5324   3.9813   2.0529  -0.1856   3.9736   4.0320  -11.8288  
-0.0716  -6.3078  -0.8988   1.1324  -5.3795   3.3686   2.5133   4.2406   0.8505  
-0.1817  -0.0044  -0.8483   3.1335  -5.4090   0.9247  -0.0611   0.1412   1.1349  
14.7530   0.1668  -5.6864   8.8130  36.2334   1.0009  -0.7558  24.5322  -2.1810  
  
Columns 10 through 11  
  
-8.1501   0.1453  
 4.8060  -0.1450  
15.6937  26.6567  
-12.0119 12.0968  
-5.4416 11.4002  
-1.8890  2.6927  
-1.8579  2.3900  
-1.8319  2.3996  
-0.0118  0.1885  
 0.2499  0.1284  
-18.0609 0.0325
```

Fig. 54 Matriz de Pesos de la Capa de Entrada

Elaboración: los autores

Fuente: Software Matlab

```
W2 =  
  
    3.3804    1.7665   41.8126  -27.2756  -25.7663  
    2.0911  -48.0465   -8.0571   12.7599   -3.7571  
   -4.5191   -0.8054  -17.5353  -28.1974    3.1524  
    2.4086    1.4279  -16.8306   28.8371   26.0085  
   -0.6165    0.0693  -21.4054   -9.3073   48.5992  
    9.7664    3.6464   -2.4640   12.8597   11.4491  
   -2.1113   -0.3947  -15.4134   29.1400   27.0443  
    3.4163    0.6547   24.7099  -20.6136    8.5963  
  -16.0278   22.7316  -11.5902    2.7203   20.8856  
    6.4555    2.2599  -10.8712    4.4196  -21.4504  
   35.2082  -27.8670    2.0718  -14.0314    3.0656
```

Fig. 55 Matriz de Pesos de la Capa Oculta

Elaboración: los autores

Fuente: Software Matlab

```
U2 =  
  
Columns 1 through 9  
-28.4863   42.0329   24.0866  -17.3168   -3.1117  -17.0314  -33.3402  -24.9745    9.0003  
  
Columns 10 through 11  
10.1388  -23.2197
```

Fig. 56 Matriz de Umbrales de la Capa Oculta

Elaboración: los autores

Fuente: Software Matlab

```
U3 =  
-21.9859    11.1543    4.2817   -3.7834  -13.2514
```

Fig. 57 Matriz de Umbrales de la Capa de Salida

Elaboración: los autores

Fuente: Software Matlab

3.4 Etapa de Postprocesamiento

En esta etapa se explicó cómo se lleva a cabo el proceso de transmisión de la información desde la placa principal, al dispositivo Android

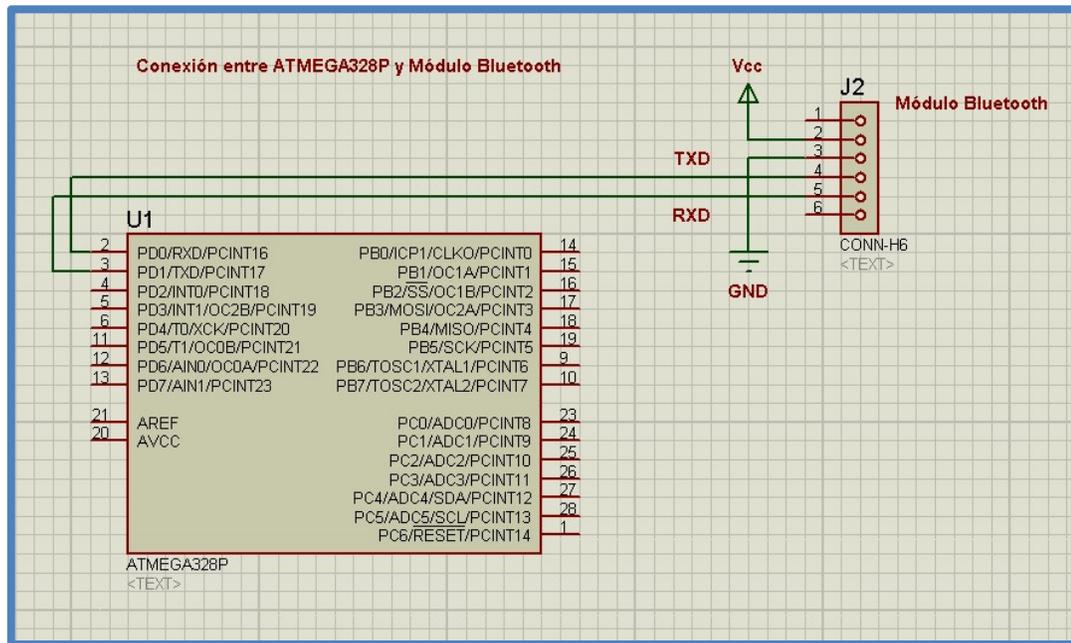


Fig. 58 Conexión entre ATMEGA328P y el Módulo Bluetooth

Elaboración: los autores

Fuente: Software Proteus 8 Professional

Como se puede observar en la Fig. 58 los pines TXD y RXD del módulo bluetooth, van conectados a los pines 2(RXD) y 3(TXD) del microcontrolador respectivamente para lograr la comunicación serial.

3.4.1 Código del Microcontrolador ATMEGA328P

El código del microcontrolador ATMEGA328P ha sido desarrollado en su totalidad por los autores de esta tesis. Para programar este microcontrolador se utilizó la **board Arduino UNO** y el **software de programación Arduino IDE** el cual está basado en **C++**. A continuación, se detalla el funcionamiento del código del microcontrolador ATMEGA328P en 28 etapas.

-Etapa 1:

Se incluyó la librería **math.h** para las operaciones matemáticas necesarias como la función **sigmoidal ()** y para compilar el algoritmo de clasificación de retropropagación:

```
//Se incluye la libreria para operaciones matemáticas
#include <math.h>
```

Fig. 59 ATMEGA328P - Importación de la Librería “math.h”

-Etapa 2:

También se incluyeron en la librería **LiquidCrystal.h** para configurar el **LCD 16x2** que se utilizó para mostrar mensajes auxiliares como mensajes de bienvenida, de calentamiento del sensor, etc.

```
//Se incluye la libreria para el LCD 16x2
#include <LiquidCrystal.h>

//Se configura la relación entre los pines del ATMEGA328
//y el LCD 16x2
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

Fig. 60 ATMEGA328P - Importación de la Librería “LiquidCrystal.h” y Configuración del LCD 16x2

-Etapa 3:

Se definen las siguientes variables globales:

```
//Se define el vector de
//activaciones de la capa de entrada de la red neuronal
double a1[1][11];

//Se define una variable para el pin analógico
//que se utilizará para leer el voltaje del sensor MQ3
int analogPin=5;

//Se define una variable para el pin digital que
//se utilizará para activar el timbre de medición
//del BAC
int PinTimbre=8;

//Se define una variable para el pin digital que
//se utilizará para activar la base del transistor
//y encender el sensor MQ3
int PinBase=10;

//Se define una variable para el pin digital que
//se utilizará para activar el LED de medición del BAC
int PinLed=13;

//Se define una variable para almacenar
//el valor cuantificado del BAC medido
long Cuant_Current_BAC=0;

//Se define una variable para almacenar
//el valor del BAC medido
double Current_BAC=0;

//Se define una variable para almacenar
//la cadena de clasificación que se
//enviará a la aplicación de Android
String Delivered_String="";
```

Fig. 61 ATMEGA328P - Definición de las Variables Globales

-Etapa 4:

Se ejecutó el setup de la **fuentes de referencia** del **convertidor A/D**, los **pines de salida** del microcontrolador ATMEGA328 y de la configuración del **LCD**:

```
void setup() {
  //Código de Setup:

  //Se utiliza como voltaje de referencia para el convertidor A/D,
  //la fuente de alimentación principal del circuito
  analogReference(EXTERNAL);

  //Se coloca el pin digital 1 como salida
  pinMode(PinTimbre, OUTPUT);

  //Se coloca el pin digital 10 como salida
  pinMode(PinBase, OUTPUT);

  //Se coloca el pin digital 13 como salida
  pinMode(PinLed, OUTPUT);

  //Se coloca el pin de activación del timbre en 0
  digitalWrite(PinTimbre, LOW);

  //Se coloca el pin de activación de la base del transistor en 0
  digitalWrite(PinBase, LOW);

  //Se coloca el pin de activación del LED en 0
  digitalWrite(PinLed, LOW);

  //Se inicia la transmisión serial a 9600 baudios
  Serial.begin(9600);

  //Se define que se utilizará un LCD de 16 columnas y 2 filas
  lcd.begin(16, 2);

  //Se limpia el LCD
  lcd.clear();
}
```

Fig. 62 ATMEGA328P - Función "Setup ()"

-Etapa 5:

Seguidamente se ejecutó el bucle infinito en donde repetidamente se muestra el mensaje en la pantalla del LCD 16x2 el mensaje de "Pulse INICIAR" el cual indico que se tiene que presionar el respectivo botón en la aplicación de Android para empezar con la medición y clasificación del BAC; y de la multa respectiva. Adicionalmente en la segunda fila del LCD se muestra el valor digital obtenido del sensor MQ3; y constantemente se lee el bus del puerto serial el cual está conectado al módulo bluetooth, para que al momento de recibir los 3 parámetros desde la aplicación de Android se inicie con la medición:

```
void loop() {  
  //Código que se ejecuta repetidamente:  
  
  //Se posiciona el cursor del LCD en la Columnna 0 y Fila 0  
  lcd.setCursor(0,0);  
  
  //Se muestra el mensaje de inicio de la primera fila  
  lcd.print(F("Pulse INICIAR"));  
  
  //Se posiciona el cursor del LCD en la Columnna 0 y Fila 1  
  lcd.setCursor(0,1);  
  
  //Se muestra el mensaje de inicio de la segunda fila  
  lcd.print("BAC: "+String(analogRead(analogPin))+ "  ");  
  
  //Se define una variable llamada "dato" en la cual se almacenará  
  //la data recibida por la transmisión serial  
  String dato = Serial.readString();  
}
```

Fig. 63 ATMEGA328P - Inicio de la Función "loop ()"

-Etapa 6:

Cuando se reciben las 3 variables (**X1**, **X2** y **X11**) a través de la transmisión bluetooth desde la aplicación de Android, estas variables se almacenan en los campos **0,1** y **10** del vector de activación **a1** de la red neuronal, posteriormente se ejecuta la función **medir_alcohol ()**:



Fig. 64 Aplicación Android - Menú Principal

Elaboración: los autores

Fuente: Software AppInventor2

```

//Si el tamaño del string recibido es igual a 3 caracteres,
//entonces se ejecutarán las funciones correspondientes
if(dato.length()==3){

    //Se almacena el primer caracter del string recibido -
    //Tipo de Transporte particular o público(Particular=0 Público=1) -
    //en el primer campo del vector "al"
    al[0][0]=String(dato.charAt(0)).toDouble();

    //Se almacena el segundo caracter del string recibido -
    //¿El conductor participó en un accidente de tránsito?(No=0 Si=1) -
    //en el segundo campo del vector "al"
    al[0][1]=String(dato.charAt(1)).toDouble();

    //Se almacena el tercer caracter del string recibido -
    //¿El accidente dejó muertos o heridos?(No=0 Si=1
    //| Sólo podrá ser 1 si el conductor participó en un
    //accidente de tránsito al[0][1]=1,
    //Si al[0][1]=0 entonces al[0][10]=0 siempre) -
    //en el onceavo campo del vector "al"
    al[0][10]=String(dato.charAt(2)).toDouble();

    //Se ejecuta la función para medir el BAC actual,
    //imprimir su valor y cuantificarlo y almacenar su valor
    //cuantificado en la variable Cuant_BAC
    medir_alcohol();
}

```

Fig. 65 ATMEGA328P - Recepción de Datos desde la Aplicación en Android

-Etapa 7:

Dentro de la función **medir_alcohol ()** primeramente se activa en pin que está conectado a la base del transistor tipo NPN PN2222A, llevándolo a saturación y conectando el pin GND del sensor **MQ3** a masa:

```
void medir_alcohol(){  
  
    //Se activa la base del transistor y se enciende el sensor MQ3  
    digitalWrite(PinBase, HIGH);  
  
}
```

Fig. 66 ATMEGA328P - Inicio de la Función “medir_alcohol ()”

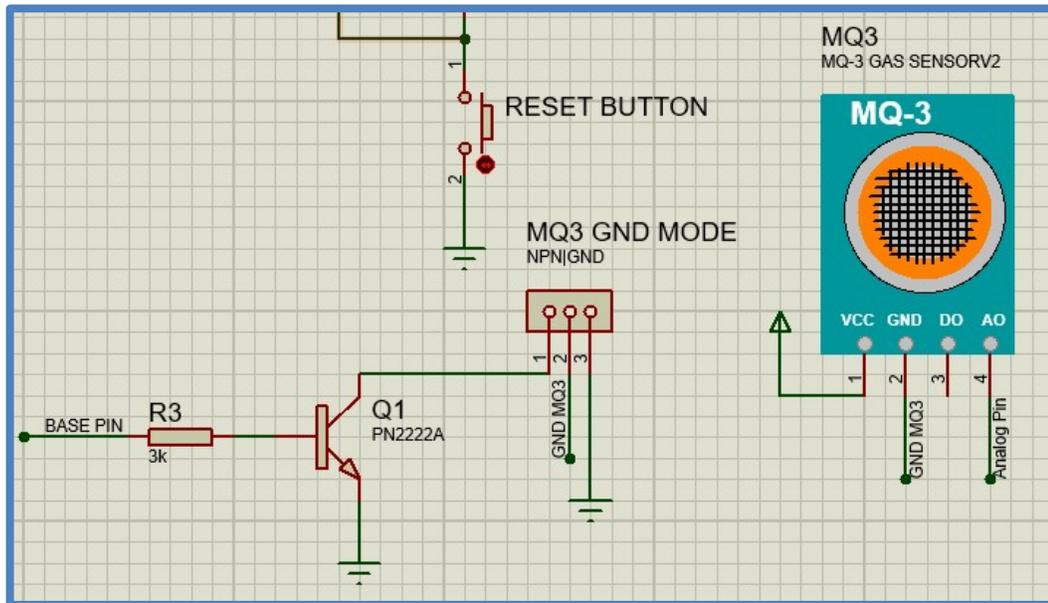


Fig. 67 Conexión entre el Transistor NPN PN2222A y el Sensor MQ3

Elaboración: los autores

Fuente: Software Proteus

-Etapa 8:

Al activarse el sensor **MQ3** se muestra el mensaje de **“Calentando...”** en el **LCD 16x2** y en la **aplicación de Android**; y el sensor se calienta durante 15 segundos, ya que este es el tiempo en el cual se calienta el alcoholímetro de referencia que se utilizó para calibrar el sensor **MQ3**.

```
//Se almacena el tiempo en milisegundos desde que
//se inició el programa en la variable "start_heat_time"
unsigned long start_heat_time = millis();

//Se calienta el sensor MQ3 durante 15 Segundos
while(millis()-start_heat_time<=15000){

    //Se limpia el LCD
    lcd.clear();

    //Se posiciona el cursor del LCD en la Columna 0 y Fila 0
    lcd.setCursor(0,0);

    //Se muestra mensaje de que
    //se esta calentando el sensor para la medición
    lcd.print(F("Calentando..."));

}
```

Fig. 68 ATMEGA328P - Código de Calentamiento del Sensor MQ3



Fig. 69 Aplicación Android – Mensaje de Calentamiento del Sensor MQ3

Elaboración: los autores

Fuente: Software AppInventor2

-Etapa 9:

Se activa el **pin del timbre** y el **pin del led de medición**; se muestra el mensaje de **“Comenzando medición”** en el **LCD 16x2** y en la **aplicación de Android**; se calcula el voltaje digital máximo del sensor MQ3 durante 5 segundos y se muestra el tiempo restante para que termine la medición:

```
//Se activa el timbre
digitalWrite(PinTimbre, HIGH);

//Se activa el LED
digitalWrite(PinLed, HIGH);

//Se limpia el LCD
lcd.clear();

//Se posiciona el cursor del LCD en la Columna 0 y Fila 0
lcd.setCursor(0,0);

//Se muestra mensaje de que se esta empezando con la medición
lcd.print(F("Comenzando"));

//Se posiciona el cursor del LCD en la Columna 0 y Fila 1
lcd.setCursor(0,1);

//Se muestra mensaje de que se esta empezando con la medición
lcd.print(F("Medición"));

//Se crea una variable para almacenar el voltaje máximo obtenido en la medición de etanol
int volt_max = 0;

//Se almacena el tiempo en milisegundos desde que se inició la medición en la variable "start_time"
unsigned long start_time = millis();

//La medición del BAC se realiza durante 5 segundos
while(millis()-start_time<=5000) {

    //Se calcula el voltaje máximo dentro del periodo de medición de etanol,
    //Se limita el valor digital del voltaje de salida del sensor MQ3 entre 160 y 1023
    if(volt_max<constrain(analogRead(analogPin),160,1023)){

        volt_max=constrain(analogRead(analogPin),160,1023);

    }

    //Se posiciona el cursor del LCD en la Columna 9 y Fila 1
    lcd.setCursor(9,1);

    //Se muestra el tiempo restante para que termine la medición
    lcd.print(String((6000-(millis()-start_time))/1000) + "... ");

}

}
```

Fig. 70 ATMEGA328P - Calentamiento y Cálculo del Voltaje Máximo del Sensor MQ3



Fig. 71 Aplicación Android – Mensaje de Medición del BAC

Elaboración: los autores

Fuente: Software AppInventor2

-Etapa 10:

Se desactiva el **pin del timbre**, el **pin del led de medición** y el **pin que está conectado a la base del transistor tipo NPN PN2222A**. Seguidamente se normaliza el valor del voltaje digital máximo obtenido a un valor entre 0.00 y 1.99 utilizando la función **map_double ()**, ingresando como parámetros de entrada el voltaje digital máximo obtenido, el valor mínimo del voltaje digital(160), el valor máximo del voltaje digital (1023), el valor mínimo de la normalización 0.00gr/L y el valor máximo de la normalización 1.99gr/L; esto es porque el sensor MQ3 tiene la capacidad de medir el BAC en un rango de 0.00 a 1.99 gr/L. Este valor normalizado se almacena en la variable **Current_BAC** y representa el BAC medido, este valor de BAC obtenido es utilizado como parámetro en la función **Cuantificar_Current_BAC ()**:

```
//Se desactiva el timbre
digitalWrite(PinTimbre, LOW);

//Se desactiva el LED
digitalWrite(PinLed, LOW);

//Se desactiva la base del transistor y se apaga el sensor MQ3
digitalWrite(PinBase, LOW);

//Se calcula el BAC actual, en base al voltaje máximo del sensor MQ3 obtenido en la medición
Current_BAC = map_double(volt_max, 160, 1023, 0.00, 1.99);

//Se cuantifica la medición del BAC obtenido y se almacena en la variable global Cuant_BAC
Cuantificar_Current_BAC(Current_BAC);

}
//*****
```

Fig. 72 ATMEGA328P - Normalización del Voltaje Máximo Calculado del Sensor MQ3

```
//Función matemática para convertir el valor digital del sensor MQ3 al valor real de gr/L de Etanol
double map_double(long x, long in_min, long in_max, double out_min, double out_max){
    return ((x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min);
}
```

Fig. 73 ATMEGA328P - Función “map_double ()”

-Etapa 11:

En la función **Cuantificar_Current_BAC ()** se cuantifico el valor del BAC medido, multiplicándolo por 100, obteniendo un rango de posibles valores de 0 a 199 y almacenando este valor en la variable **Cuant_Current_BAC**:

```
//*****Se define función para cuantificar el BAC calculado
//y para posteriormente almacenarlo en el vector de entrada de la Red Neuronal*****
void Cuantificar_Current_BAC(double Val_Current_BAC){

    //Se cuantifica el valor de BAC medido a 2 decimales, mutliplicandolo por 100 y
    //almacenandolo en la variable "Cuant_Current_BAC"
    Cuant_Current_BAC = String(100*(String(Val_Current_BAC, 2).toDouble())).toInt();

}
//*****
```

Fig. 74 ATMEGA328P - Función "Cuantificar_Current_BAC ()"

-Etapa 12:

Se retornó al bucle principal del programa y se ejecutó la función **BAC_to_a1 ()** utilizando el valor del BAC cuantificado obtuvieron en el paso anterior, como parámetro de entrada para esta función:

```
//*****CLASIFICACION DEL BAC ACTUAL*****
//Se convierte el BAC actual a formato
//de 8 bits y se almacena en el vector
//de entrada de la red neuronal al
BAC_to_a1(Cuant_Current_BAC);
```

Fig. 75 ATMEGA328P - Ejecución de la Función "BAC_to_a1 ()" con la Variable "Cuant_Current_BAC" como Parámetro de Entrada

-Etapa 13:

En la función **BAC_to_a1 ()** se convierte el valor cuantificado del BAC a binario de 8 bits utilizando la función **int2bin ()** y el resultado de esta función se almacena en el vector **Str_Bin_BAC []**

```
//Función para convertir el valor cuantificado del BAC a digital y
//almacenarlo en el vector de entrada "al" de la red neuronal
void BAC_to_a1(long Val_Cuant_BAC){
    //Se define un vector tipo char para almacenar los
    //bits del valor binario del BAC obtenido
    char Str_Bin_BAC[9];

    //Se almacenan los bits del valor binario del BAC
    //obtenido en el vector "Str_Bin_BAC"
    int2bin(Val_Cuant_BAC).toCharArray(Str_Bin_BAC,9);
}
```

Fig. 76 ATMEGA328P - Función "BAC_to_a1 ()"

-Etapa 14:

En la función **int2bin ()** se utilizó un bucle “for” para obtener cada bit (desde el 7 hasta el 0) de la variable **Current_BAC** que representa el BAC medido, se concatenan estos bits en la variable tipo string llamada **Str_Binary** y finalmente, se retorna el valor de esta variable:

```
*****FUNCION PARA CONVERTIR DE ENTERO A BINARIO DE 8 DIGITOS*****
String int2bin(Long num){
    *****SE CREA UNA VARIABLE TIPO STRING
    *****PARA ALMACENAR LA CONVERSION DEL NUMERO EN BINARIO*****
    String Str_Binary="";

    *****SE CREA UN "FOR" PARA LEER LOS 8 BITS MAS SIGNIFICATIVOS DEL NUMERO INGRESADO
    for(int index_binary = 7; index_binary >= 0; index_binary--){

        ***** LEEN LOS BITS DEL NUMERO INGRESADO DESDE EL BIT "7"
        *****HASTA EL BIT "0" Y SE ALMACENAN EN LA VARIABLE TIPO STRING "Str_Binary"
        Str_Binary+=String(bitRead(num, index_binary));

    }

    *****SE RETORNA LA VARIABLE "Str_Binary" QUE CONTIENE EL VALOR EN BINARIO A 8 BITS DEL NUMERO INGRESADO
    return Str_Binary;
}
```

Fig. 77 ATMEGA328P - Función “int2bin ()”

-Etapa 15:

Volviendo a la función **BAC_to_a1 ()** se utilizó un bucle “for” para almacenar cada bit del valor digital del BAC (representado por el vector **Str_Bin_BAC []**) obtenido en el paso anterior en los campos del 2 (entrada X3) al 9 (entrada X10) del vector de activación **a1 []** de la red neuronal:

```
//Se define un "for" para almacenar los bits del BAC obtenido
//en el vector de activación "a1" de la capa de entrada de la Red Neuronal
for(int al_index = 2; al_index <= 9; al_index++){

    //Se almacenan los bits del BAC obtenido en el vector de activación "a1"
    //de la capa de entrada de la Red Neuronal
    al[0][al_index] = String(Str_Bin_BAC[al_index-2]).toDouble();

}
}
```

Fig. 78 ATMEGA328P - Cálculo del Vector de Activación “a1”

-Etapa 16:

Con el valor del vector de entrada de la red neuronal **a1** calculado en los pasos anteriores, se vuelven al bucle principal del programa y se ejecuta la función **red_neuronal ()**:

```
//Se evalua el BAC actual en la red neuronal para su clasificación
red_neuronal();
//*****
}
}
```

Fig. 79 ATMEGA328P - Ejecución de la Función “red_neuronal ()”

-Etapa 17:

En la función `red_neuronal ()` se declaran las matrices de pesos de la capa de entrada **W1** y de la capa oculta **W2**, el vector de umbrales de la capa oculta **U2** y de la capa de salida **U3**; también el vector de activación de la capa oculta **a2** y el vector de salida **Y** de la red neuronal:

```
void red_neuronal() {  
  
    //Se define la matriz de pesos calculada para la capa de entrada de la red neuronal  
    double W1[11][11]={  
    //Fila 1  
    {1.74913690858213,-0.0658418981191268,-14.2000253332531,-0.816302748793418,  
    -17.3727535498203,2.35617479140531,14.7824250964336,-1.93123405080623,-0.0562782279738559,  
    -8.15007502967464,0.145342312468199},  
    //Fila 2  
    {18.6948904274841,-2.10514113410531,-11.4639709097670,-30.7803826559885,16.3576881362944,  
    -9.98622202944466,-31.2863746725190,7.69444556780119,8.14593851712341,4.80599757607756,-0.144962434821750},  
    //Fila 3  
    {14.3352253750442,-12.5650564324997,-19.0076681095943,18.9231500309749,-17.2264720724324,27.1463276674896,  
    6.38385839311517,3.44869329961337,29.3517115539874,15.6936804262719,26.6566535458805},  
    //Fila 4  
    {11.2253581690288,-42.7426771480042,-23.3565587267739,20.7556082596608,-16.5835788301601,0.0505574162539678,  
    6.23020483755424,-3.48878284450550,-8.11177423217779,-12.0119057877426,12.0967515620177},  
    //Fila 5  
    {9.31989942026189,-42.2215083528123,-9.77181334975485,7.25524487674342,-1.81162706584863,6.19083747968745,  
    19.6013892042831,-10.9268572658038,0.271995457741977,-5.44162447507452,11.4002377803214},  
    //Fila 6  
    {5.51147153295554,-20.7200312423404,-1.03625147352042,6.96218297541085,-2.25808221202965,6.63362717544678,  
    10.4987568877878,8.11736434174346,-11.6410471725636,-1.88896123531232,2.69274342190035},  
    //Fila 7  
    {-0.760667575384792,-12.6998605186784,-3.53151160239596,0.0791682721673952,0.966761628716753,5.68060815127112,  
    6.51598621208924,8.53338207362539,-11.8781512199877,-1.85790732532949,2.38995751495902},  
    //Fila 8  
    {-0.735246770374863,-6.33392372656322,-3.53243023066927,3.98131397426032,2.05293950175344,-0.185622786442356,  
    3.97363661108733,4.03197354558336,-11.8288182310478,-1.83194540311635,2.39963059032860},  
    //Fila 9  
    {-0.0716109063149955,-6.30781478202002,-0.898792508519461,1.13235184756608,-5.37946072778094,3.36857581235069,  
    2.51328178628889,4.24060238010481,0.850537028714786,-0.0118326821376826,0.188548231185019},  
    //Fila 10  
    {-0.181749608890032,-0.00435598264719042,-0.848337679469073,3.13348965504008,-5.40903529451124,0.924672350469086,  
    -0.0610624664675141,0.141171552145513,1.13494533121757,0.249857520366973,0.128355905861376},  
    //Fila 11  
    {14.7530457457418,0.166803859595301,-5.68636210552962,8.81296887850136,36.2334370568580,1.00088854975602,  
    -0.755753448626941,24.5322374022536,-2.18099095835909,-18.0609002348550,0.0325065455258029}  
    };  
}
```

Fig. 80 ATMEGA328P - Declaración de la Matriz de Pesos "W1"

```

//Se define la matriz de pesos calculada para la capa oculta de la red neuronal
double W2[11][5]={
//Fila 1
{3.38044846174048,1.76646770288234,41.8126031493096,-27.2755717601654,-25.7663085475305},
//Fila 2
{2.09113110605541,-48.0465310285685,-8.05714116959637,12.7599091772015,-3.75714753597178},
//Fila 3
{-4.51911951698503,-0.805356346903796,-17.5353121391823,-28.1973940328807,3.15244145560016},
//Fila 4
{2.40859612047344,1.42786841101055,-16.8306340677033,28.8370922423072,26.0085178627017},
//Fila 5
{-0.616538833790136,0.0693055690341323,-21.4054180685093,-9.30730429698247,48.5991928564298},
//Fila 6
{9.76636518962427,3.64644285784334,-2.46401222658192,12.8596803854018,11.4491001389027},
//Fila 7
{-2.11126708902586,-0.394660858428084,-15.4134032790862,29.1400097222386,27.0442703493071},
//Fila 8
{3.41633902850686,0.654740023229059,24.7098984362937,-20.6135751315316,8.59634003567833},
//Fila 9
{-16.0278453920471,22.7316374383787,-11.5902218282388,2.72030678689043,20.8856438589853},
//Fila 10
{6.45549847287139,2.25994529118327,-10.8711730986633,4.41956942180112,-21.4503760856271},
//Fila 11
{35.2081874649028,-27.8669703365470,2.07175810382860,-14.0313515413991,3.06562744648362}
};

```

Fig. 81 ATMEGA328P - Declaración de la Matriz de Pesos “W2”

```

//Se define el vector de umbrales calculada para la capa oculta de la red neuronal
double U2[1][11]={
{-28.4863106473638,42.0328850286602,24.0866044939839,-17.3168171952947,-3.11169431002040,
-17.0314424151371,-33.3401605377440,-24.9744882366921,9.00027726136925,10.1388396600353,-23.2197074076147}
};

```

Fig. 82 ATMEGA328P - Declaración de la Matriz de Umbrales “U2”

```

//Se define el vector de umbrales calculada para la capa de salida de la red neuronal
double U3[1][5]={
{-21.9858574459554,11.1542958195944,4.28167089770932,-3.78342937815825,-13.2514004650561}
};

```

Fig. 83 ATMEGA328P - Declaración de la Matriz de Umbrales “U3”

```
//Se define el vector de activaciones de la capa oculta de la red neuronal  
double a2[1][11];
```

Fig. 84 ATMEGA328P - Declaración del Vector de Activaciones de la Capa Oculta "a2"

```
//Se define el vector de salida de la red neuronal  
double y[1][5];
```

Fig. 85 ATMEGA328P - Declaración del Vector de Salida "y"

-Etapa 18:

Seguidamente se procedió con el cálculo del vector de activación de la capa oculta **a2**, para empezar, se calcula el producto matricial **a1xW1**, para lo cual se declararon las variables descritas a continuación:

```

//*****CALCULO DEL VECTOR "a2"*****

//*****CALCULO DEL PRODUCTO MATRICIAL "a1*W1"*****
//Se crea una variable para almacenar el número
//de filas del vector "a1"
const int al_rows=1;

//Se crea una variable para almacenar el número
//de columnas de la matriz "W1"
const int W1_columns=sizeof(W1[0])/sizeof(W1[0][0]);

//Se crea una variable para almacenar el número
//de columnas del vector "a1"
//que es igual al número de filas de la matriz "W1"
int counter=sizeof(al[0])/sizeof(al[0][0]);

//Se define un vector para almacenar la multiplicación
//del vector "a1" con la matriz "W1"
double product_al_W1[al_rows][W1_columns];

//Se define una variable "suma" para almacenar la suma de
//la multiplicación de cada elemento de la fila actual
//del vector "a1" con cada elemento correspondiente
//de la columna actual de la matriz "W1"
double suma;
```

Fig. 86 ATMEGA328P - Declaración de Variables para el Cálculo del Producto Matricial "a1xW1"

-Etapa 19:

Se utilizó bucles **for** para multiplicar cada elemento de las filas del vector **a1** por cada elemento respectivo de las columnas del vector **W1** y sumándolos para obtener cada elemento del vector resultante **a1xW1** y almacenarlos en la variable **product_a1_W1**:

```
//Se recorren las filas del vector "a1" y las columnas de la matriz "W1"
//para llevar a cabo la multiplicación
for(int index_al_rows=0; index_al_rows < al_rows; index_al_rows++){

    for(int index_W1_columns=0; index_W1_columns < W1_columns; index_W1_columns++){
        //Se inicializa la variable "suma" con valor cero
        suma=0;

        //Se utiliza la variable "counter" para recorrer las columnas del vector "a1" y
        //las filas de la matriz "W1"
        for(int index_counter=0; index_counter < counter; index_counter++){
            //Se calcula la suma de la multiplicación de cada elemento de la fila actual del vector "a1"
            //con cada elemento correspondiente de la columna actual de la matriz "W1" y
            //se almacena el resultado en la variable "suma"
            suma=suma+al[index_al_rows][index_counter]*W1[index_counter][index_W1_columns];

        }

        //Se coloca el valor de la variable "suma" calculado anteriormente
        //en la posición respectiva del vector del producto "a1*W1" resultante
        product_al_W1[index_al_rows][index_W1_columns]=suma;

    }

}

//*****FIN DEL CALCULO DEL PRODUCTO MATRICIAL "a1*W1"*****
```

Fig. 87 ATMEGA328P - Cálculo del Producto Matricial "a1xW1"

-Etapa 20:

Se utilizó bucles **for** para recorrer las columnas de los vectores **product_a1_W1** y **U2** con el fin de sumar cada uno de sus elementos entre si y almacenar cada suma de estos elementos en la variable **sum_a1productW1_U2**:

```
*****CALCULO DE LA SUMA MATRICIAL "a1*W1+U2" Y EL VECTOR "a2"*****  
  
//Se define un vector para almacenar la suma del vector  
//"product_al_W1" con el vector "U2", tendrá las mismas dimensiones que el vector "product_al_W1"  
double sum_alproductW1_U2[al_rows][W1_columns];  
  
//Se recorren las filas del vector "al" y las columnas de la matriz  
//"W1" para llevar a cabo la suma de los vectores "product_al_W1" y "U2"  
for(int index_al_rows=0; index_al_rows < al_rows; index_al_rows++){  
    for(int index_W1_columns=0; index_W1_columns < W1_columns; index_W1_columns++){  
        //Se suma cada elemento del vector "product_al_W1" con cada elemento respectivo del vector "U2" y se almacena el resultado  
        //en el vector "sum_alproductW1_U2"  
        sum_alproductW1_U2[index_al_rows][index_W1_columns]=product_al_W1[index_al_rows][index_W1_columns]+U2[index_al_rows][index_W1_columns];  
    }  
}
```

Fig. 88 ATMEGA328P - Cálculo de "a1xW1 + U2"

-Etapa 21:

Seguidamente se utilizó la función **sigmoidal ()**, utilizaron como parámetro de entrada cada elemento del vector **sum_a1productW1_U2** y almacenando el resultado que obtuvieron de la evaluación de cada uno de estos elementos en dicha función en el vector de activación de la capa oculta **a2**:

```
//Se evalua el elemento actual del vector "sum_alproductW1_U2" en la función "sigmoidal" y el resultado  
//se almacena en la posición correspondiente del vector "a2"  
a2[index_al_rows][index_W1_columns]=sigmoidal(sum_alproductW1_U2[index_al_rows][index_W1_columns]);  
}  
}  
*****FIN DEL CALCULO DE LA SUMA MATRICIAL "a1*W1+U2" Y EL VECTOR "a2"*****  
  
*****FIN DEL CALCULO DEL VECTOR "a2"*****
```

Fig. 89 ATMEGA328P - Cálculo del Vector de Activaciones de la Capa Oculta
"a2"

```
double sigmoidal(double x){
    //Se define variable para almacenar el resultado
    double y;
    //Se evalua el valor de la variable "x"
    //en la función sigmoidal y se almacena el resultado
    //en la variable "y"
    y=1/(1+exp(-x));
    //Se devuelve el valor de la variable "y"
    return y;
}

//*****
```

Fig. 90 ATMEGA328P - Función "sigmoidal ()"

-Etapa 22:

Seguidamente se procedió con el cálculo del vector de salida de la red neuronal **y**, para empezar, se calcula el producto matricial **a2xW2**, para lo cual se declaran las variables descritas a continuación:

```

//*****CALCULO DEL VECTOR DE SALIDA "y"*****

//*****CALCULO DEL PRODUCTO MATRICIAL "a2*W2"*****
//Se crea una variable para almacenar el número de filas del vector "a2"
const int a2_rows=1;

//Se crea una variable para almacenar el número de columnas de la matriz "W2"
const int W2_columns=sizeof(W2[0])/sizeof(W2[0][0]);

//Se crea una variable para almacenar el número de
//columnas del vector "a2" que es igual al número de filas de la matriz "W2"
int counter2=sizeof(a2[0])/sizeof(a2[0][0]);

//Se define un vector para almacenar la multiplicación
//del vector "a2" con la matriz "W2"
double product_a2_W2[a2_rows][W2_columns];

//Se define una variable "suma2" para almacenar la
//suma de la multiplicación de cada elemento de la
//fila actual del vector "a2" con cada elemento
//correspondiente de la columna actual de la matriz "W2"
double suma2;
```

Fig. 91 ATMEGA328P - Declaración de Variables para Calcular el Producto Matricial "a2xW2"

-Etapa 23:

Se utilizan bucles **for** para multiplicar cada elemento de las filas del vector **a2** por cada elemento respectivo de las columnas del vector **W2** y sumándolos para obtener cada elemento del vector resultante **a2xW2** y almacenarlos en la variable **product_a2_W2**:

```
//Se recorren las filas del vector "a2" y las columnas de la matriz
//"W2" para llevar a cabo la multiplicación
for(int index_a2_rows=0; index_a2_rows < a2_rows; index_a2_rows++){
  for(int index_W2_columns=0; index_W2_columns < W2_columns; index_W2_columns++){
    //Se inicializa la variable "suma2" con valor cero
    suma2=0;
    //Se utiliza la variable "counter2" para recorrer las columnas del vector
    //"a2" y las filas de la matriz "W2"
    for(int index_counter2=0; index_counter2<counter2; index_counter2++){
      //Se calcula la suma de la multiplicación de cada elemento de la fila actual del vector
      //"a2" con cada elemento correspondiente de la columna actual de la matriz "W2" y se
      //almacena el resultado en la variable "suma2"
      suma2=suma2+a2[index_a2_rows][index_counter2]*W2[index_counter2][index_W2_columns];
    }
    //Se coloca el valor de la variable "suma2" calculado
    //anteriormente en la posición respectiva del vector del producto "a2*W2" resultante
    product_a2_W2[index_a2_rows][index_W2_columns]=suma2;
  }
}
//*****FIN DEL CALCULO DEL PRODUCTO MATRICIAL "a2*W2"*****
```

Fig. 92 ATMEGA328P - Cálculo del Producto Matricial "a2xW2"

-Etapa 24:

Se utilizó bucles **for** para recorrer las columnas de los vectores **product_a2_W2** y **U3** con el fin de sumar cada uno de sus elementos entre sí y almacenar cada suma de estos elementos en la variable **sum_a2productW2_U3**:

```
*****CALCULO DE LA SUMA MATRICIAL "a2*W2+U3" Y EL VECTOR DE SALIDA "y"*****  
  
//Se define un vector para almacenar la suma del  
//vector "product_a2_W2" con el vector "U3", tendrá  
//las mismas dimensiones que el vector "product_a2_W2"  
double sum_a2productW2_U3[a2_rows][W2_columns];  
  
//Se recorren las filas del vector "a2" y las columnas de la matriz  
//"W2" para llevar a cabo la suma de los vectores "product_a2_W2" y "U3"  
for(int index_a2_rows=0; index_a2_rows<a2_rows; index_a2_rows++){  
    for(int index_W2_columns=0; index_W2_columns<W2_columns; index_W2_columns++){  
        //Se suma cada elemento del vector "product_a2_W2" con cada elemento respectivo del vector "U3" y se almacena el resultado en el vector  
        //sum_a2productW2_U3"  
        sum_a2productW2_U3[index_a2_rows][index_W2_columns]=product_a2_W2[index_a2_rows][index_W2_columns]+U3[index_a2_rows][index_W2_columns];  
    }  
}
```

Fig. 93 ATMEGA328P - Cálculo de "a2xW2 + U3"

-Etapa 25:

Seguidamente se utilizó la función **sigmoidal ()**, utilizaron como parámetro de entrada cada elemento del vector **sum_a2productW2_U3** y almacenando el resultado que se obtuvieron de la evaluación de cada uno de estos elementos en dicha función en el vector de salida de la red neuronal **y**:

```
//Se evalúa el elemento actual del vector "sum_a2productW2_U3" en la función "sigmoidal" y el resultado redondeado  
//se almacena en la posición correspondiente del vector de salida "y"  
y[index_a2_rows][index_W2_columns]=round(sigmoidal(sum_a2productW2_U3[index_a2_rows][index_W2_columns]));  
  
}  
}  
*****FIN DEL CALCULO DE LA SUMA MATRICIAL "a2*W2+U3" Y EL VECTOR DE SALIDA "y"*****  
  
*****FIN DEL CALCULO DEL VECTOR DE SALIDA "y"*****
```

Fig. 94 ATMEGA328P - Cálculo del Vector de Salida "y"

-Etapa 26:

Se procedió con la clasificación del BAC Y1:0 Y2:0 **NORMAL**, Y1:0 Y2:1 **EUFORIA**, Y1:1 Y2:0 **EXCITACIÓN** y Y1:1 Y2:1 **CONFUSIÓN**. Esta clasificación se concatena con el **BAC** calculado y el valor resultante se almacena en la variable **Delivered_String**:

```

//*****SE EMPIEZA CON LA CLASIFICACIÓN DE LA SALIDA DE LA RED NEURONAL*****

//*****SI EL PRIMER BIT DE LA SALIDA ES 0 Y EL SEGUNDO BIT
//DE LA SALIDA ES 0 ENTONCES LA CLASIFICACION DEL BAC ACTUAL ES NORMAL*****
if(round(y[0][0] == 0) && round(y[0][1] == 0)){

    //Se concatena el valor del BAC calculado con un espacio
    //en blanco y la cadena "NORMAL"; y se almacena en la variable "Delivered_String"
    Delivered_String=String(Current_BAC)+" "+"NORMAL";

}

//*****SI EL PRIMER BIT DE LA SALIDA ES 0 Y EL SEGUNDO BIT
//DE LA SALIDA ES 1 ENTONCES LA CLASIFICACION DEL BAC ACTUAL ES EUFORIA*****
if(round(y[0][0] == 0) && round(y[0][1] == 1)){

    //Se concatena el valor del BAC calculado con un espacio
    //en blanco y la cadena "EUFORIA"; y se almacena en la variable "Delivered_String"
    Delivered_String=String(Current_BAC)+" "+"EUFORIA";

}

//*****SI EL PRIMER BIT DE LA SALIDA ES 1 Y EL SEGUNDO BIT
//DE LA SALIDA ES 0 ENTONCES LA CLASIFICACION DEL BAC ACTUAL ES EXCITACION Y/O EMBRIAGUEZ*****
if(round(y[0][0] == 1) && round(y[0][1] == 0)){

    //Se concatena el valor del BAC calculado con un espacio
    //en blanco y la cadena "EXCITACION"; y se almacena en la variable "Delivered_String"
    Delivered_String=String(Current_BAC)+" "+"EXCITACION";

}

//*****SI EL PRIMER BIT DE LA SALIDA ES 1 Y EL SEGUNDO BIT
//DE LA SALIDA ES 1 ENTONCES LA CLASIFICACION DEL BAC ACTUAL ES CONFUSION*****
if(round(y[0][0] == 1) && round(y[0][1] == 1)){

    //Se concatena el valor del BAC calculado con un espacio
    //en blanco y la cadena "CONFUSION"; y se almacena en la variable "Delivered_String"
    Delivered_String=String(Current_BAC)+" "+"CONFUSION";

}
}

```

Fig. 95 ATMEGA328P - Clasificación del Nivel de Alcholemia

-Etapa 27:

A continuación, se procede a clasificar el tipo de multa respectiva que le corresponde al conductor, en base a los parámetros de entrada que se ingresaron en la aplicación de Android **X1: Tipo de Transporte particular o público (Particular=0 Público=1), X2: ¿El conductor participó en un accidente de tránsito? (No=0 Si=1), X11: ¿El accidente dejó muertos o heridos? (No=0 Si=1 | Sólo podrá ser 1 si el conductor participó en un accidente de tránsito X2=1, Si X2=0 entonces X11=0 siempre)** y en base al **BAC** calculado; y el valor resultante de esta clasificación se concatena con la variable **Delivered_String**:

```

//*****SI EL TERCER BIT DE LA SALIDA ES 0 Y EL CUARTO BIT DE LA SALIDA ES 0 Y
//EL QUINTO BIT DE LA SALIDA ES 1 ENTONCES LAS MULTAS Y SANCIONES SON DEL TIPO 1*****
if(round(y[0][2] == 0) && round(y[0][3] == 0) && round(y[0][4] == 1)){

    //Se concatena el valor de la variable " Delivered_String" con la cadena "TIPO_1"
    Delivered_String=Delivered_String+" "+"TIPO_1";

}

//*****SI EL TERCER BIT DE LA SALIDA ES 0 Y EL CUARTO BIT DE LA SALIDA ES 1 Y
//EL QUINTO BIT DE LA SALIDA ES 0 ENTONCES LAS MULTAS Y SANCIONES SON DEL TIPO 2*****
if(round(y[0][2] == 0) && round(y[0][3] == 1) && round(y[0][4] == 0)){

    //Se concatena el valor de la variable " Delivered_String" con la cadena "TIPO_2"
    Delivered_String=Delivered_String+" "+"TIPO_2";

}

//*****SI EL TERCER BIT DE LA SALIDA ES 0 Y EL CUARTO BIT DE LA SALIDA ES 1 Y
//EL QUINTO BIT DE LA SALIDA ES 1 ENTONCES LAS MULTAS Y SANCIONES SON DEL TIPO 3*****
if(round(y[0][2] == 0) && round(y[0][3] == 1) && round(y[0][4] == 1)){

    //Se concatena el valor de la variable " Delivered_String" con la cadena "TIPO_3"
    Delivered_String=Delivered_String+" "+"TIPO_3";

}

//*****SI EL TERCER BIT DE LA SALIDA ES 1 Y EL CUARTO BIT DE LA SALIDA ES 0 Y
//EL QUINTO BIT DE LA SALIDA ES 0 ENTONCES LAS MULTAS Y SANCIONES SON DEL TIPO 4*****
if(round(y[0][2] == 1) && round(y[0][3] == 0) && round(y[0][4] == 0)){

    //Se concatena el valor de la variable " Delivered_String" con la cadena "TIPO_4"
    Delivered_String=Delivered_String+" "+"TIPO_4";

}

//*****SI EL TERCER BIT DE LA SALIDA ES 1 Y EL CUARTO BIT DE LA SALIDA ES 0 Y
//EL QUINTO BIT DE LA SALIDA ES 1 ENTONCES LAS MULTAS Y SANCIONES SON DEL TIPO 5*****
if(round(y[0][2] == 1) && round(y[0][3] == 0) && round(y[0][4] == 1)){

    //Se concatena el valor de la variable " Delivered_String" con la cadena "TIPO_5"
    Delivered_String=Delivered_String+" "+"TIPO_5";

}

```

Fig. 96 ATMEGA328P - Clasificación del Tipo de Multa

-Etapa 28:

Finalmente, se envió la variable **Delivered_String** a la aplicación de Android, para mostrar los resultados de la clasificación:

```
//Se envia la cadena contenida en la variable  
//"Delivered_String" a la aplicación de Android  
Serial.print(Delivered_String);  
  
}
```

Fig. 97 ATMEGA328P - Envío de la Clasificación del Nivel de Alcoholemia y del Tipo de Multa a la Aplicación Android

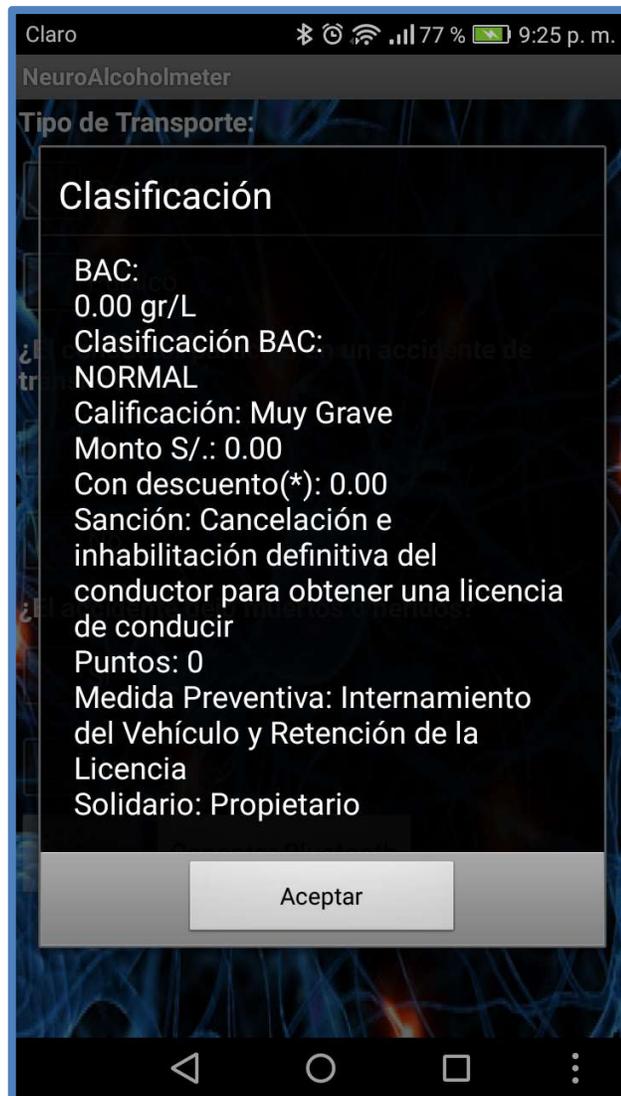


Fig. 98 Aplicación Android - Resultados de Clasificación de la Red Neuronal

3.4.2 Aplicación en Android

El código de la aplicación móvil ha sido desarrollado en su totalidad por los autores de esta tesis. Para programar la aplicación móvil se utilizó la herramienta APP Inventor2 desarrollada por el MIT. A continuación, se detalla el funcionamiento del código de la aplicación móvil en 15 etapas.

-Etapa 1:

El diseño de la aplicación Android consta de 3 timers **Bluetooth_Capture_String**, **Calentamiento_de_Sensor_MQ3** y **Medición_de_BAC**; los cuales se utilizaron respectivamente para capturar los strings enviados desde el módulo bluetooth del microcontrolador para la clasificación de alcoholemia y clasificación de la multa, para calentar el sensor MQ3 y para medir el BAC. Un **cliente bluetooth** para establecer la conexión entre la aplicación Android y el módulo bluetooth del microcontrolador; y un **notificador** para mostrar los mensajes de calentamiento, medición y clasificación en la aplicación Android.

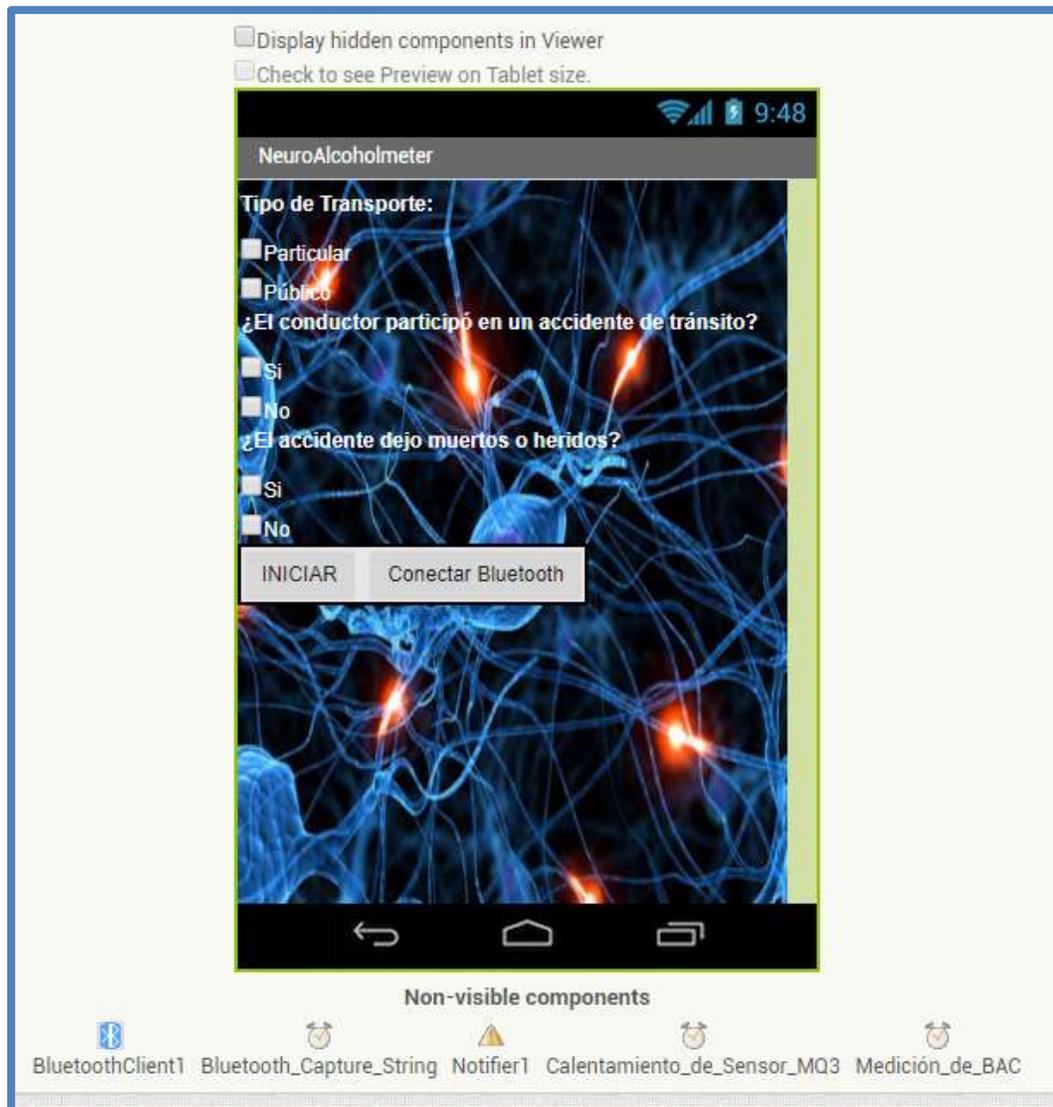


Fig. 99 Aplicación Android - Diseño

-Etapa 2:

En la aplicación Android se definen las siguientes variables globales **Tipo_Transporte** para almacenar la variable de entrada **X1: Tipo de transporte (Particular=0 Público=1)**, **Accidente** para almacenar la variable de entrada **X2: ¿El conductor participó en un accidente de tránsito? (No=0 Si=1)**, **Muertos_o_Heridos** para almacenar la variable de entrada **X11: ¿El accidente dejó muertos o heridos? (No=0 Si=1; sólo podrá ser 1 si el conductor participó en un accidente de tránsito X2=1, Si X2=0 entonces X11=0 siempre)**, **Recived_String** para almacenar la cadena de clasificación enviada desde el microcontrolador, **BAC** para almacenar el BAC medido, **BAC_Classification** para almacenar la clasificación del nivel de alcoholemia, **Multas_y_Sanciones** para almacenar la clasificación de la multa correspondiente, **calibration_seconds** para realizar la calibración del sensor MQ3 y **measurement_seconds** para realizar la medición del BAC.



Fig. 100 Aplicación Android - Declaración de Variables Globales

-Etapa 3:

Al iniciar la aplicación se debe conectar al módulo bluetooth HC-05 del microcontrolador utilizado la lista desplegable **Conectar Bluetooth (ListPicker1)** cuyo funcionamiento es el siguiente; antes de seleccionar algún elemento de la lista desplegable se cargan en esta lista todas la direcciones y nombres de los dispositivos bluetooth conocidos:

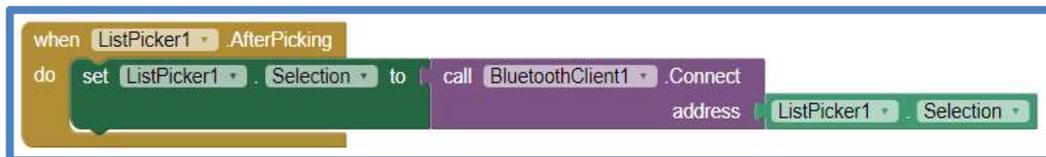


```
when ListPicker1 BeforePicking
do set ListPicker1 Elements to BluetoothClient1 AddressesAndNames
```

Fig. 101 Aplicación Android - Cargado de Direcciones y Nombres de Dispositivos Bluetooth Conocidos en la Lista Desplegable “ListPicker1”

-Etapa 4:

Después de seleccionar el dispositivo bluetooth HC-05, se procede a establecer la conexión con dicho dispositivo:



```
when ListPicker1 AfterPicking
do set ListPicker1 Selection to call BluetoothClient1 Connect
address ListPicker1 Selection
```

Fig. 102 Aplicación Android - Conexión con el Módulo Bluetooth HC-05

-Etapa 5:

Luego de establecer la conexión con el módulo bluetooth HC-05 se deben de seleccionar las variables de entrada para la clasificación por parte de la red neuronal en el microcontrolador **X1: Tipo de transporte (Particular=0 Público=1)**, **X2: ¿El conductor participó en un accidente de tránsito? (No=0 Si=1)** y **X11: ¿El accidente dejó muertos o heridos? (No=0 Si=1; sólo podrá ser 1 si el conductor participó en un accidente de tránsito X2=1, Si X2=0 entonces X11=0 siempre).**

Si se selecciona el **CheckBox1** lo cuál indica que el tipo de transporte intervenido es particular, entonces se establece el valor de la variable global **Tipo_Transporte** a cero, adicionalmente se desactiva el **CheckBox2** el cuál es utilizado para indicar que el transporte intervenido es público y se activan los checkboxes **CheckBox3** y **CheckBox4**; los cuales se utilizan para indicar si el conductor participó en un accidente de tránsito o no. En caso contrario si el **CheckBox1** no esta seleccionado, entonces se establece el valor de la variable global **Tipo_Transporte** a vacío, se activa el **CheckBox2** y se desactivan los checkboxes **CheckBox3** y **CheckBox4**.

Si el **CheckBox1** y el **CheckBox2** no están seleccionados, porque el usuario por error eligió el tipo de transporte equivocado y es necesario volver a establecer este valor desmarcando el checkbox respectivo, entonces se establecen el valor de las variables globales **Tipo_Transporte**, **Accidente** y **Muertos_o_Heridos** a un valor vacío; se deshabilitan y desmarcan los checkboxes **CheckBox3**, **CheckBox4**, **CheckBox5** y **CheckBox6**; y se deshabilita el botón **INICIAR (Button1)**.

A continuación, se muestra la función de cambio de estado del **CheckBox1**:

```
when CheckBox1 . Changed
do
  if CheckBox1 . Checked
  then
    set global Tipo_Transporte to "0"
    set CheckBox2 . Enabled to false
    set CheckBox3 . Enabled to true
    set CheckBox4 . Enabled to true
  else
    set global Tipo_Transporte to ""
    set CheckBox2 . Enabled to true
    set CheckBox3 . Enabled to false
    set CheckBox4 . Enabled to false
  if not CheckBox1 . Checked and not CheckBox2 . Checked
  then
    set global Tipo_Transporte to ""
    set global Accidente to ""
    set global Muertos_o_Heridos to ""
    set CheckBox3 . Enabled to false
    set CheckBox3 . Checked to false
    set CheckBox4 . Enabled to false
    set CheckBox4 . Checked to false
    set CheckBox5 . Enabled to false
    set CheckBox5 . Checked to false
    set CheckBox6 . Enabled to false
    set CheckBox6 . Checked to false
    set Button1 . Enabled to false
```

Fig. 103 Aplicación Android - Evento de Cambio de Estado del “CheckBox1” - X1: Tipo de Transporte - Particular=0

-Etapa 6:

De la misma manera si se selecciona el **CheckBox2** lo cuál indicó que el tipo de transporte intervenido es público, entonces se establece el valor de la variable global **Tipo_Transporte** a uno, adicionalmente se desactiva el **CheckBox1** el cuál es utilizado para indicar que el transporte intervenido es particular y se activan los checkboxes **CheckBox3** y **CheckBox4**; los cuales se utilizan para indicar si el conductor participó en un accidente de tránsito o no. En caso contrario si el **CheckBox2** no está seleccionado, entonces se establece el valor de la variable global **Tipo_Transporte** a vacío, se activa el **CheckBox1** y se desactivan los checkboxes **CheckBox3** y **CheckBox4**.

Si el **CheckBox1** y el **CheckBox2** no están seleccionados, porque el usuario por error eligió el tipo de transporte equivocado y es necesario volver a establecer este valor desmarcando el checkbox respectivo, entonces se establecen el valor de las variables globales **Tipo_Transporte**, **Accidente** y **Muertos_o_Heridos** a un valor vacío; se deshabilitan y desmarcan los checkboxes **CheckBox3**, **CheckBox4**, **CheckBox5** y **CheckBox6**; y se deshabilita el botón **INICIAR (Button1)**.

A continuación, se muestra la función de cambio de estado del **CheckBox2**:

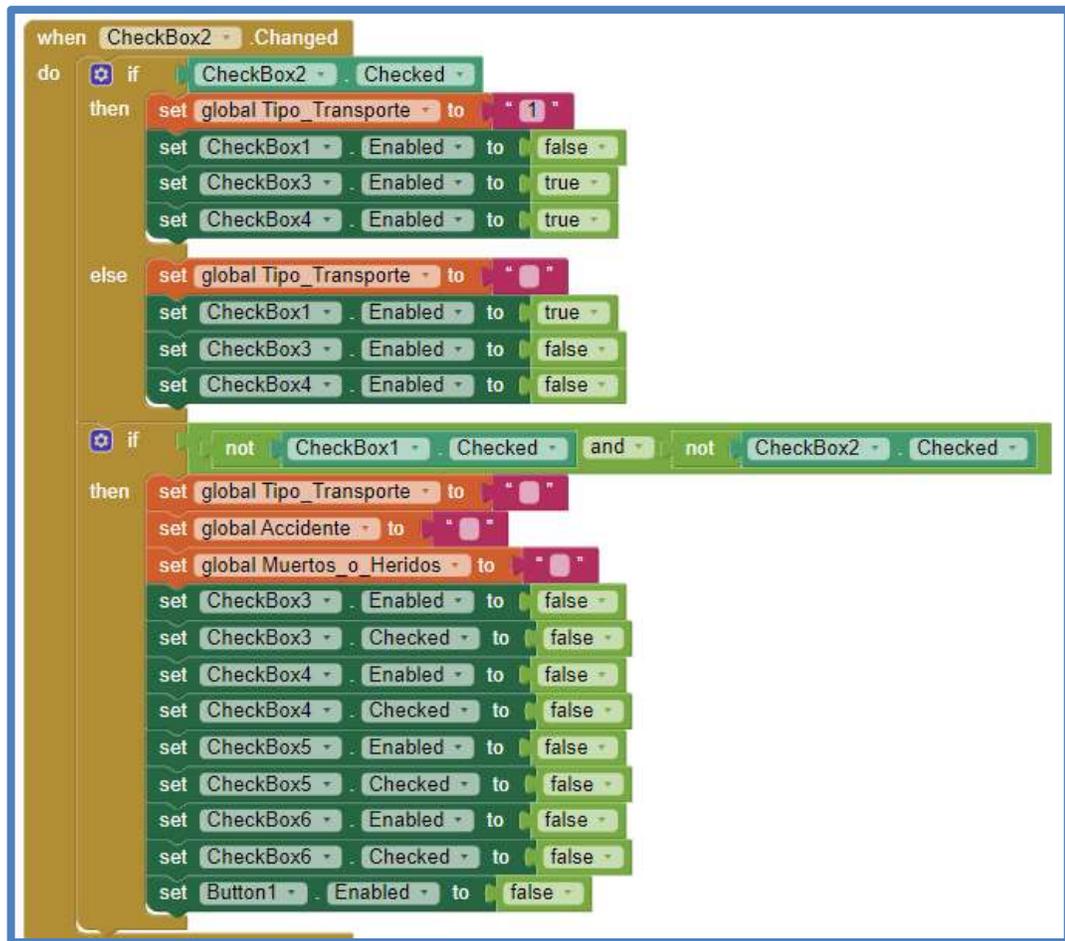


Fig. 104 Aplicación Android - Evento de Cambio de Estado del "CheckBox2" -
X1: Tipo de Transporte - Público=1

-Etapa 7:

En la función de cambio de estado del **CheckBox3**, si el **CheckBox1** o **CheckBox2** están seleccionados y el **CheckBox3** está seleccionado, esto indico que el conductor participo en un accidente de tránsito, por lo que se establecio el valor de la variable global **Accidente** a uno, adicionalmente se desactiva el **CheckBox4** el cuál es utilizado para indicar que el conductor no participó en un accidente de tránsito y se activan los checkboxes **CheckBox5** y **CheckBox6**; los cuales se utilizaron para indicar si el accidente de tránsito en el cuál participó el conductor dejo muertos o heridos; o no. En caso contrario si el

CheckBox1 o **CheckBox2** están seleccionados y el **CheckBox3** no está seleccionado, entonces se establece el valor de la variable global **Accidente** a vacío, se activa el **CheckBox4** y se desactivan los checkboxes **CheckBox5** y **CheckBox6**.

Si el **CheckBox3** y el **CheckBox4** no están seleccionados, porque el usuario por error indicó de manera errónea si el conductor participó en un accidente de tránsito o no; y es necesario volver a establecer este valor desmarcando el checkbox respectivo, entonces se establecen el valor de las variables globales **Accidente** y **Muertos_o_Heridos** a un valor vacío; se deshabilitan y desmarcan los checkboxes **CheckBox5** y **CheckBox6**; y se deshabilita el botón **INICIAR (Button1)**.

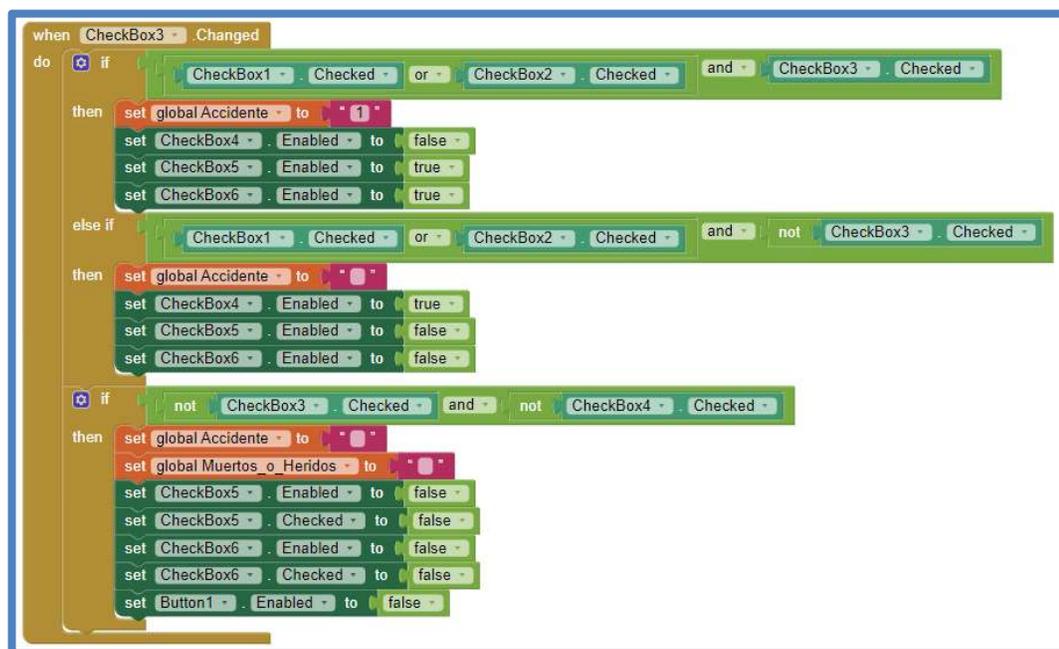


Fig. 105 Aplicación Android - Evento de Cambio de Estado del “CheckBox3” - X2: ¿El Conductor Participó en un Accidente de Tránsito? - Si=1

-Etapa 8:

De manera similar, en la función de cambio de estado del **CheckBox4**, si el **CheckBox1** o **CheckBox2** están seleccionados y el **CheckBox4** está seleccionado, esto indica que el conductor no participó en un accidente de tránsito, por lo que se establece el valor de la variable global **Accidente** a cero y el valor de la variable global **Muertos_o_Heridos** a cero, adicionalmente se desactiva el **CheckBox3** el cuál es utilizado para indicar que el conductor participó en un accidente de tránsito, se marca el **CheckBox6** el cual se utiliza para indicar que el accidente de tránsito en el cuál participó el conductor no dejó muertos o heridos y se habilita el botón **INICIAR (Button1)**. En caso contrario si el **CheckBox1** o **CheckBox2** están seleccionados y el **CheckBox4** no está seleccionado, entonces se establece el valor de las variables globales **Accidente** y **Muertos_o_Heridos** a un valor vacío; se activa el **CheckBox3**, se desmarca el **CheckBox6** y se desactiva el botón **INICIAR (Button1)**.

Si el **CheckBox3** y el **CheckBox4** no están seleccionados, porque el usuario por error indicó de manera errónea si el conductor participó en un accidente de tránsito o no; y es necesario volver a establecer este valor desmarcando el checkbox respectivo, entonces se establecen el valor de las variables globales **Accidente** y **Muertos_o_Heridos** a un valor vacío; se deshabilitan y desmarcan los checkboxes **CheckBox5** y **CheckBox6**; y se deshabilita el botón **INICIAR (Button1)**.



Fig. 106 Aplicación Android - Evento de Cambio de Estado del “CheckBox4” - X2: ¿El Conductor Participó en un Accidente de Tránsito? - No=0

-Etapa 9:

En la función de cambio de estado del **CheckBox5**, si el **CheckBox3** o **CheckBox4** están seleccionados y el **CheckBox5** está seleccionado, esto indico que el accidente de tránsito en el cuál participó el conductor dejo muertos o heridos, por lo que se establece el valor de la variable global **Muertos_o_Heridos** a uno, adicionalmente se desactiva el **CheckBox6** el cuál es utilizado para indicar que el accidente de tránsito en el cuál participó el conductor no dejo muertos o heridos y se habilita el botón **INICIAR (Button1)**. En caso contrario si el **CheckBox3** o **CheckBox4** están seleccionados y el **CheckBox5** no esta seleccionado, entonces se establece el valor de la variable global **Muertos_o_Heridos** a un valor vacío; se activa el **CheckBox6** y se desactiva el botón **INICIAR (Button1)**.

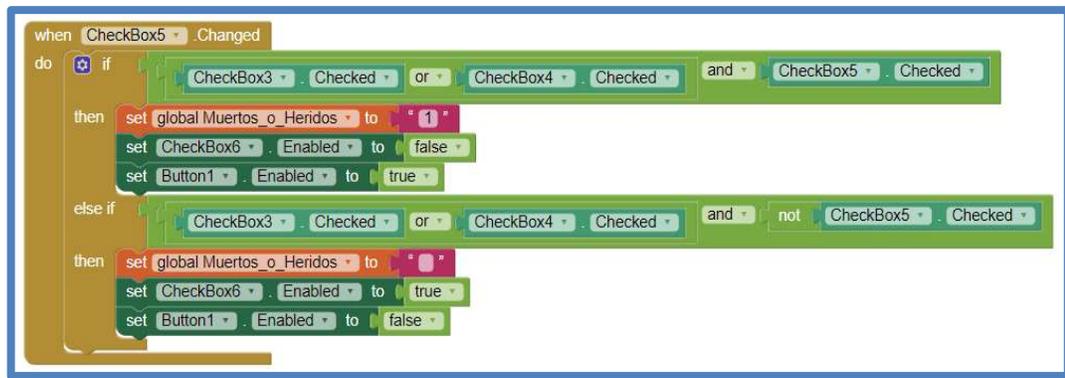


Fig. 107 Aplicación Android - Evento de Cambio de Estado del "CheckBox5" - X11: ¿El Accidente Dejó Muertos o Heridos? - Si=1

-Etapa 10:

De manera similar, en la función de cambio de estado del **CheckBox6**, si el **CheckBox3** o **CheckBox4** están seleccionados y el **CheckBox6** está seleccionado, esto indico que el accidente de tránsito en el cuál participó el conductor no dejo muertos o heridos, por lo que se establecio el valor de la variable global **Muertos_o_Heridos** a cero, adicionalmente se desactiva el **CheckBox5** el cuál es utilizado para indicar que el accidente de tránsito en el cuál participó el conductor dejo muertos o heridos y se habilita el botón **INICIAR (Button1)**. En caso contrario si el **CheckBox3** o **CheckBox4** están seleccionados y el **CheckBox6** no esta seleccionado, entonces se establece el valor de la variable global **Muertos_o_Heridos** a un valor vacío; se activa el **CheckBox5** y se desactiva el botón **INICIAR (Button1)**.

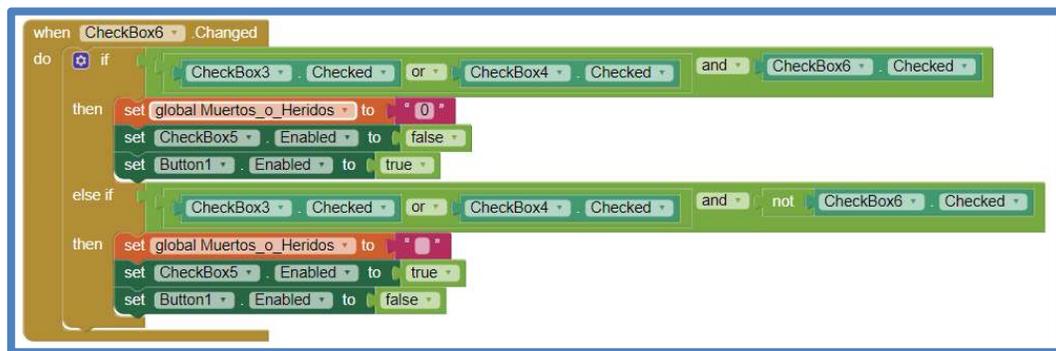


Fig. 108 Aplicación Android - Evento de Cambio de Estado del "CheckBox6" - X11: ¿El Accidente Dejó Muertos o Heridos? - No=0

-Etapa 11:

Al presionar el botón **INICIAR (Button1)** se envían vía bluetooth hacia el microcontrolador los bits de **Tipo_Transporte**, **Accidente** y **Muertos_o_Heridos** seleccionados en el menú principal de la aplicación Android, con la función **Disable_Options** se deshabilitan el estado de los checkboxes **CheckBox1**, **CheckBox2**, **CheckBox3**, **CheckBox4**, **CheckBox5** y **CheckBox6**; el estado del botón **Button1** y de la lista desplegable **ListPicker1**, se utiliza el **Notificador(Notifier1)** para mostrar el mensaje de calibración del sensor MQ3, mostrando como mensaje el valor de la variable **calibration_seconds** cuyo valor inicial es 15, finalmente, se activa el timer **Calentamiento_de_Sensor_MQ3** el cuál irá reduciendo el valor de la variable **calibration_seconds** hasta 0.

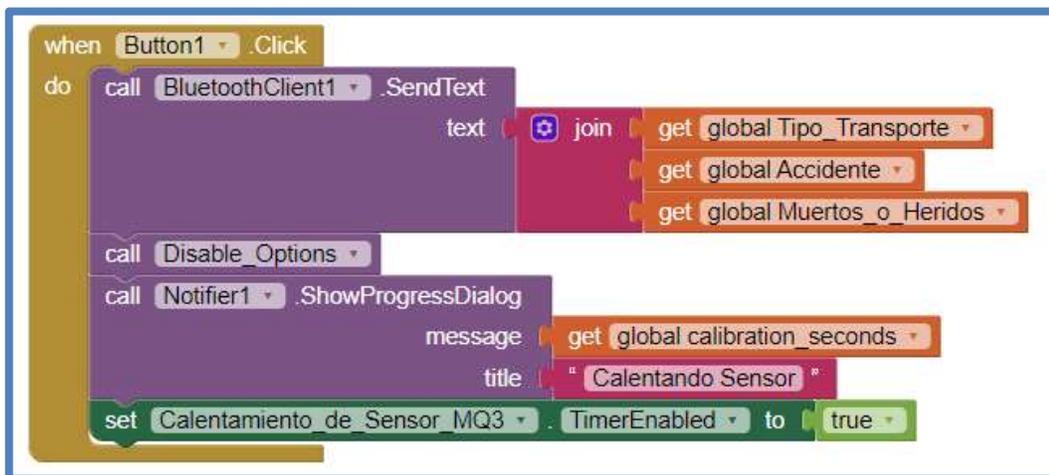


Fig. 109 Aplicación Android - Evento Pulsar Botón "INICIAR(Button1)"

A continuación, se muestra la función **Disable_Options**:

```
to Disable_Options
do
  set CheckBox1 Enabled to false
  set CheckBox2 Enabled to false
  set CheckBox3 Enabled to false
  set CheckBox4 Enabled to false
  set CheckBox5 Enabled to false
  set CheckBox6 Enabled to false
  set Button1 Enabled to false
  set ListPicker1 Enabled to false
```

Fig. 110 Aplicación Android - Función "Disable_Options"

-Etapa 12:

Cuando se dispara el timer **Calentamiento_de_Sensor_MQ3** se resta una unidad a la variable global **calibration_seconds**; si el valor de esta variable es mayor a cero, entonces se utiliza el **Notificador (Notifier1)** para mostrar el valor de la variable **calibration_seconds** lo cual representa el tiempo restante para que termine la calibración del sensor MQ3. Si en caso contrario el valor de la variable global **calibration_seconds** es menor o igual a cero, entonces se desactiva el timer **Calentamiento_de_Sensor_MQ3**, se establece el valor de la variable global **calibration_seconds** a 15, se utilizó el **Notificador(Notifier1)** se mostraron el valor de la variable global **measurement_seconds**, la cual representa el tiempo restante para que termine la medición del BAC y finalmente, se activa el timer **Medición_de_BAC**, el cual se utiliza en la medición del BAC.

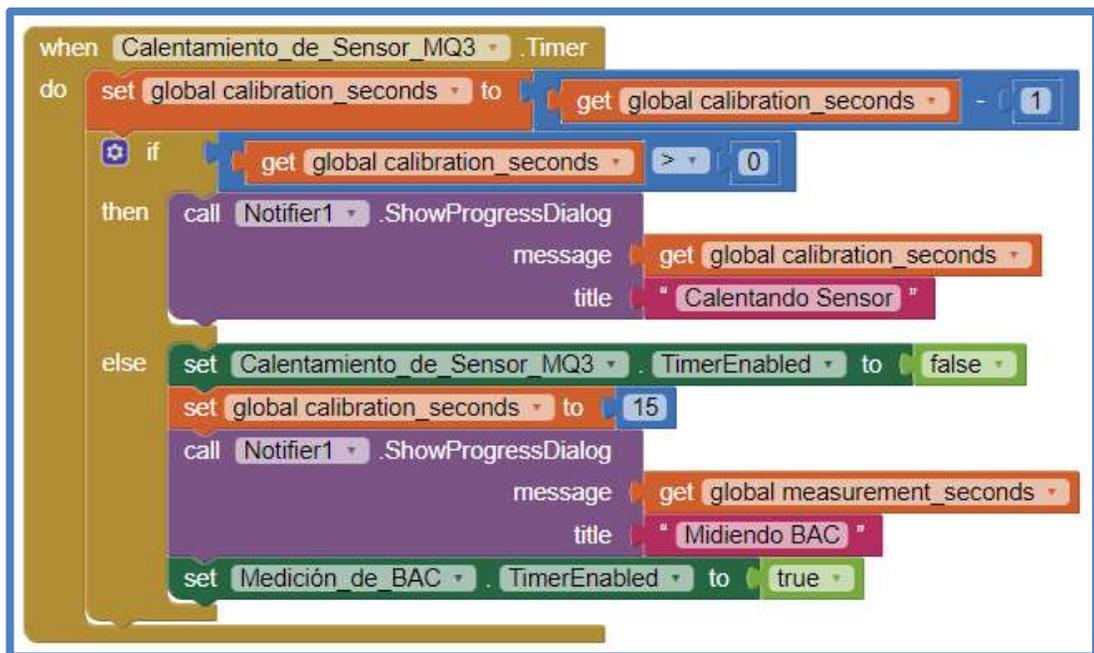


Fig. 111 Aplicación Android - Evento Disparar Timer
"Calentamiento_de_Sensor_MQ3"

-Etapa 13:

De manera similar, cuando se dispara el timer **Medición_de_BAC** se resta una unidad a la variable global **measurement_seconds**; si el valor de esta variable es mayor a cero, entonces se utilizó el **Notificador(Notifier1)** para mostraron el valor de la variable **measurement_seconds** lo cual representa el tiempo restante para que termine la medición del BAC. Si en caso contrario el valor de la variable global **calibration_seconds** es menor o igual a cero, entonces se desactiva el timer **Medición_de_BAC**, se establece el valor de la variable global **measurement_seconds** a 5 y finalmente, se cierra el cuadro de diálogo de progreso del **Notificador (Notifier1)**.

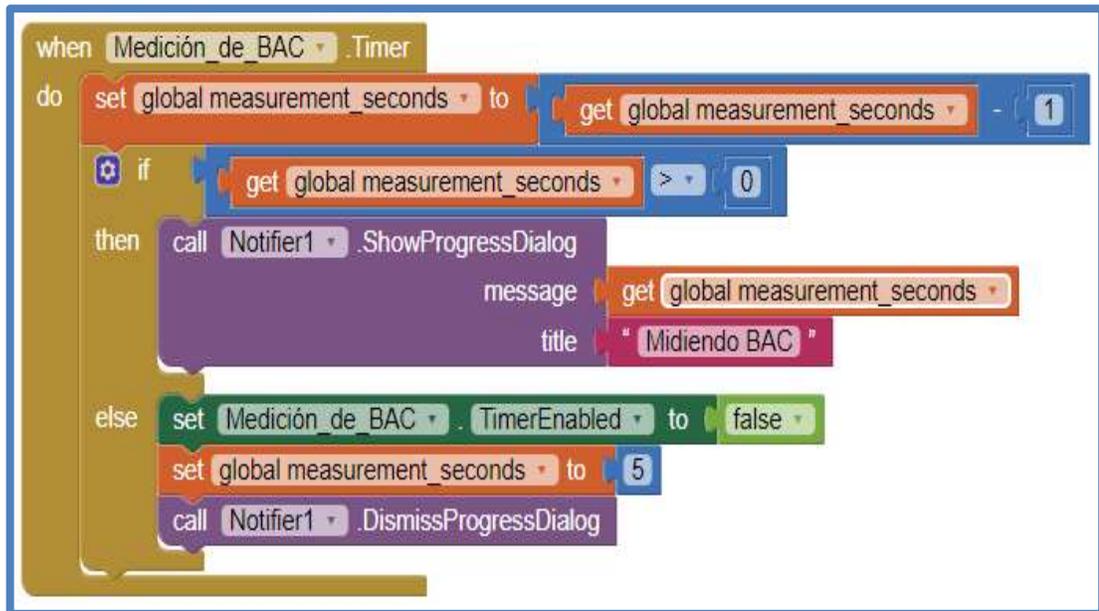


Fig. 112 Aplicación Android - Evento Disparar Timer "Medición_de_BAC"

-Etapa 14:

Cada 500 milisegundos, se dispara el timer **Bluetooth_Capture_String**, con el fin de capturar los datos de clasificación enviados por el microcontrolador. Se verifica si el cliente bluetooth está conectado y si el bus de transmisión está libre, si es así entonces se utilizó la función **trim** para eliminar los espacios al inicio y el final de la cadena recibida y la función **split at spaces** para almacenar como lista los datos de clasificación en la variable global **Recived_String**, el elemento 1 de la lista **Recived_String** representa el **BAC medido**, el elemento 2 representa la **clasificación del BAC** y el elemento 3 representaron la **clasificación de la multa**; los cuales se almaceno respectivamente en las variables globales **BAC**, **BAC_Classification** y **Multas_y_Sanciones**.

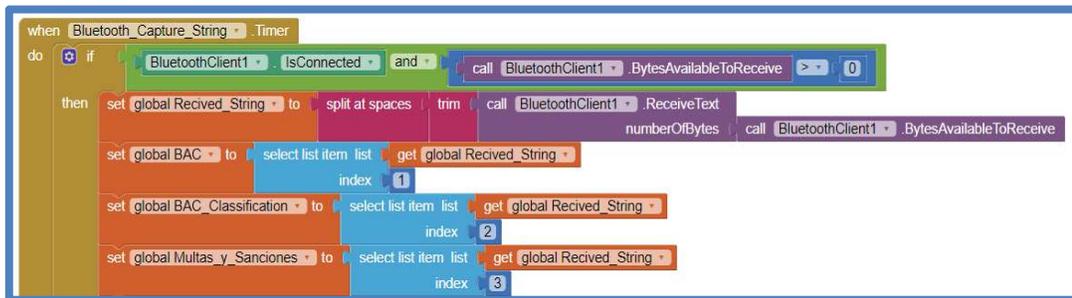


Fig. 113 Aplicación Android - Evento Disparar Timer “Bluetooth_Capture_String”
- Recepción de Datos del Módulo Bluetooth del Microcontrolador

-Etapa 15:

Finalmente, se procede a mostrar la clasificación del BAC medido y de la multa respectiva.

Si el valor de la variable global **Multas_y_Sanciones** enviada desde el módulo bluetooth del microcontrolador es igual a la cadena "EN_ORDEN", entonces a través del **Notificador (Notifier1)** se muestra el valor del BAC calculado utilizó la variable global **BAC**, se muestra la clasificación del BAC medido utilizaron la variable global **BAC_Classification**, se indica la sanción respectiva, en este caso que todo está en orden. Finalmente, se ejecuta la función **Restart_Values** para restablecer el valor de las variables globales **BAC**, **BAC_Classification**, **Multas_y_Sanciones**, **Tipo_Transporte**, **Accidente** y **Muertos_o_Heridos**; el estado de los checkboxes **CheckBox1**, **CheckBox2**, **CheckBox3**, **CheckBox4**, **CheckBox5** y **CheckBox6**; el estado del botón **Button1** y de la lista desplegable **ListPicker1**.

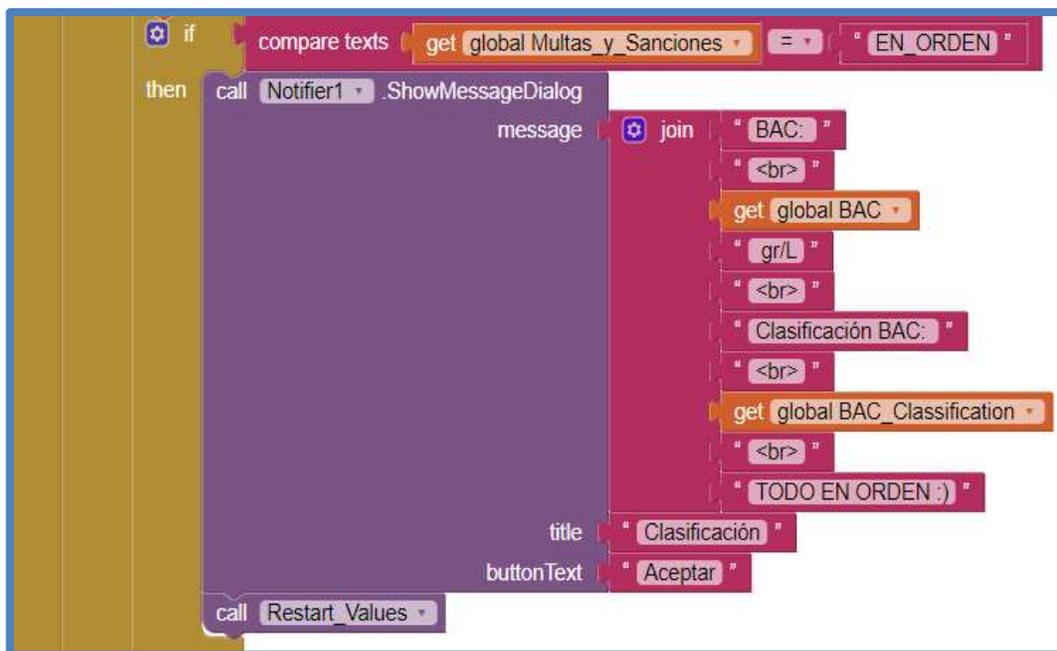
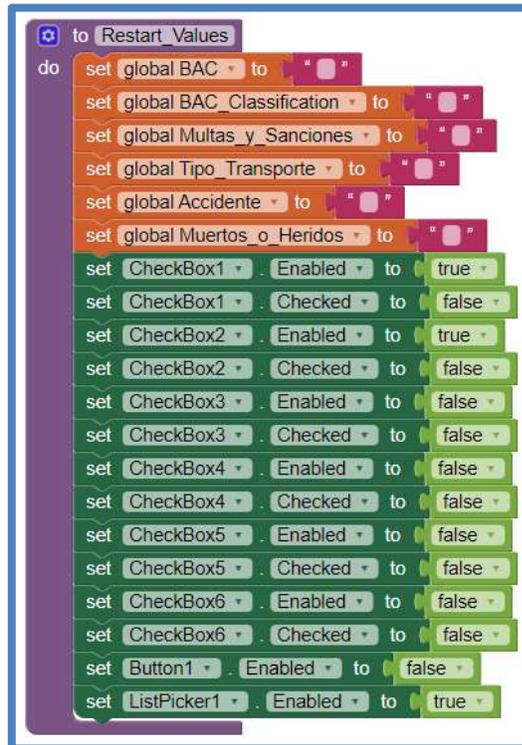


Fig. 114 Aplicación Android - Clasificación "Todo en Orden"

A continuación, se muestra la función **Restart_Values**:



```
to Restart_Values
do
    set global BAC to ""
    set global BAC_Classification to ""
    set global Multas_y Sanciones to ""
    set global Tipo_Transporte to ""
    set global Accidente to ""
    set global Muertos_o_Heridos to ""
    set CheckBox1.Enabled to true
    set CheckBox1.Checked to false
    set CheckBox2.Enabled to true
    set CheckBox2.Checked to false
    set CheckBox3.Enabled to false
    set CheckBox3.Checked to false
    set CheckBox4.Enabled to false
    set CheckBox4.Checked to false
    set CheckBox5.Enabled to false
    set CheckBox5.Checked to false
    set CheckBox6.Enabled to false
    set CheckBox6.Checked to false
    set Button1.Enabled to false
    set ListPicker1.Enabled to true
```

Fig. 115 Aplicación Android - Función “Restart_Values”

De manera similar se muestra la clasificación respectiva cuando el valor de la variable global **Multas_y Sanciones** es igual a “TIPO_1”, “TIPO_2”, “TIPO_3”, “TIPO_4” y “TIPO_5”.

A continuación, se muestran las imágenes respectivas:

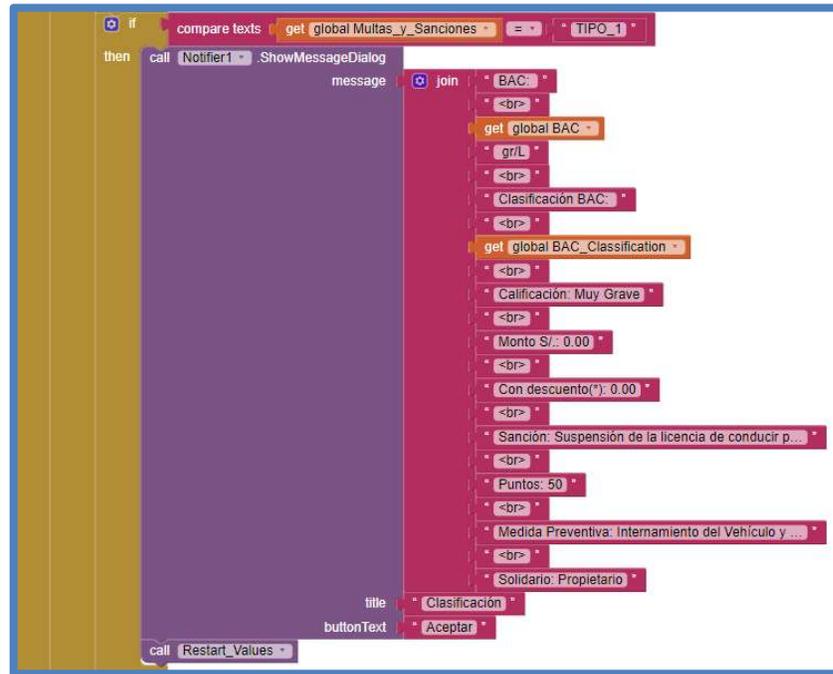


Fig. 116 Aplicación Android - Clasificación de Multas y Sanciones de Tipo 1

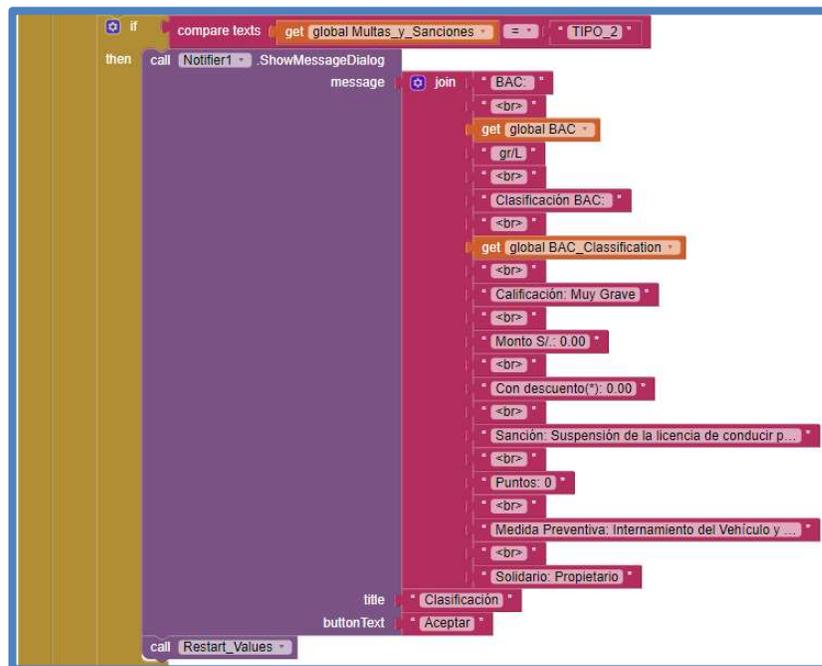


Fig. 117 Aplicación Android - Clasificación de Multas y Sanciones de Tipo 2

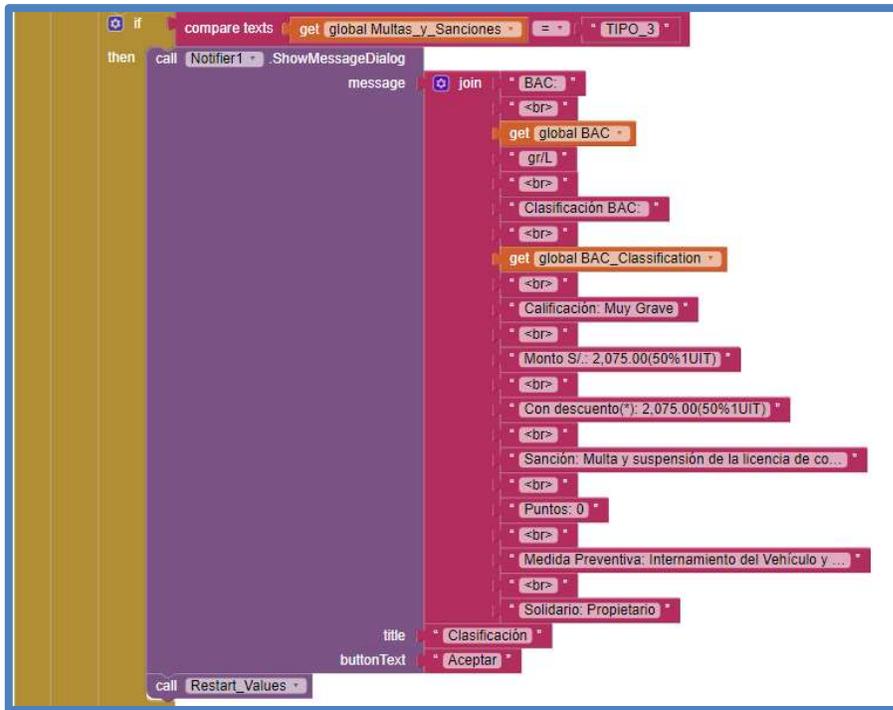


Fig. 118 Aplicación Android - Clasificación de Multas y Sanciones de Tipo 3

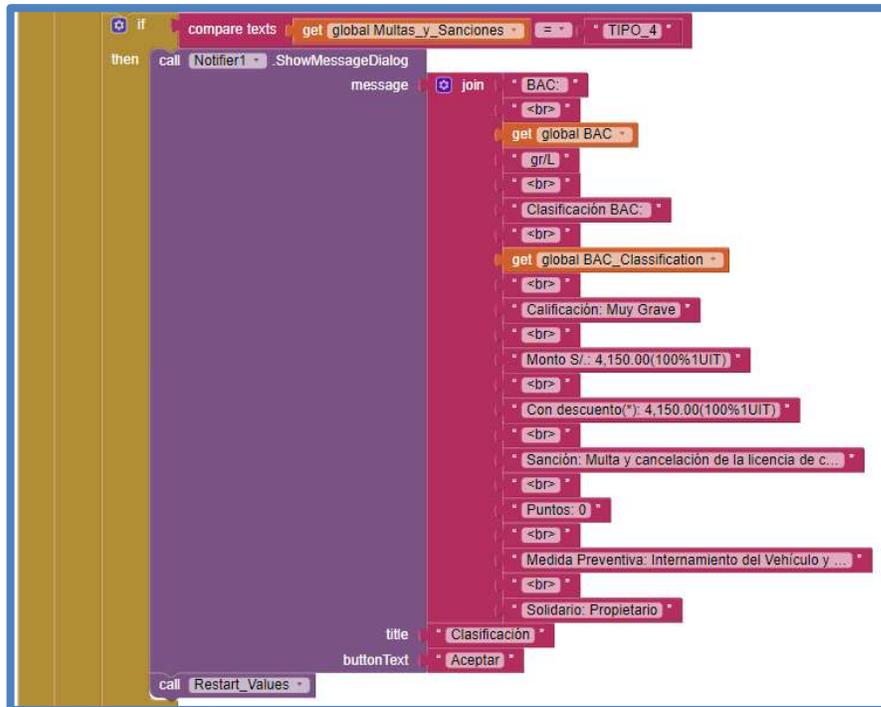


Fig. 119 Aplicación Android - Clasificación de Multas y Sanciones de Tipo 4

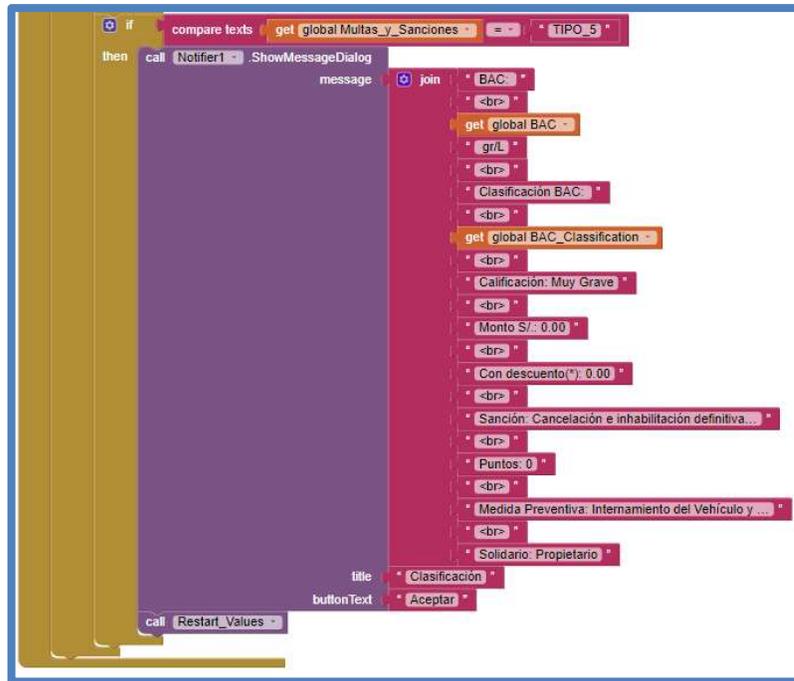


Fig. 120 Aplicación Android - Clasificación de Multas y Sanciones de Tipo 5

El diagrama de flujo general de la aplicación se encuentra en el **ANEXO8**

CONCLUSIONES

1. Se concluyó que con el sensor MQ3 se pueden realizar proyectos para medición de la concentración de alcohol en la sangre.
2. Se concluyó que el diseño de filtros pasabajos es necesario para cumplir con el teorema de muestreo de Nyquist en la conversión análoga-digital.
3. Se concluyó que mediante redes neuronales es posible implementar un sistema para la detección y clasificación de patrones de ebriedad.
4. Se concluyó que, para evitar caídas en mínimos locales de la función de error de clasificación, es importante el uso del coeficiente de momentum y de otras reglas heurísticas.

5. Se concluyó que la clasificación de los patrones de ebriedad en el presente proyecto es óptima, ya que el error de clasificación de es cercano a cero.

6. Se concluyó que un microcontrolador ATMEGA328P es posible programar un algoritmo de clasificación de perceptrón multicapa, con los valores de las matrices de pesos y umbrales respectivos.

7. Se concluyó mediante el cálculo del coeficiente de correlación de Pearson que la medición del etanol en el ambiente con el proyecto desarrollado es buena, ya que presenta dependencia positiva.

RECOMENDACIONES

1. Se recomienda hacer el test de calibración del sensor MQ3 con más personas para tener una lectura más exacta.
2. Se recomienda enlazar los resultados de clasificación que brinda el proyecto con una base de datos de la PNP que almacene el historial de multas.
3. Se recomienda elaborar un modelo más compacto que el prototipo diseñado para que se pueda utilizar con mayor facilidad.
4. Se recomienda implementar un sistema de recarga para la batería del prototipo.
5. Se recomienda la publicación de la aplicación de clasificación neuronal desarrollada para el sistema operativo Android en el Google Play Store para su uso comercial junto con el prototipo.

FUENTES DE INFORMACIÓN

Amazon.com, Inc. (2015). *Módulo de transceptor HC-05 Wireless Bluetooth Serial*. Seattle, EU.: Amazon.com, Inc. Recuperado de <https://www.amazon.com/LeaningTech-HC-05-Module-Pass-Through-Communication/dp/B00INWZRNC>

Andrade Spatz, G. L. (2009). *Regulación para los pilotos de vehículos automotores, de la prohibición de conducir bajo influencia de bebidas alcohólicas y consecuencias jurídicas y sociales de las personas que conduzcan en estado de ebriedad* (tesis de pregrado). Universidad de San Carlos de Guatemala, Ciudad de Guatemala, Guatemala.

Atmel Corporation. (2014). *ATmega48A/PA/88A/PA/168A/PA/328/P Datasheet*. Recuperado de <https://www.es.co.th/Schemetic/PDF/ATMEGAXXA-P-PA.PDF>

Automotive Research & Testing Center. (2010). *Driver Status Monitoring System, DSM*. Changhua County, TW.: ARTC. Recuperado de https://www.artc.org.tw/english/02_research/02_01detail.aspx?pdid=7

Canales Martínez, C. A. (2011). *Variación de la concentración de alcohol etílico en cadáveres en relación al tiempo* (tesis de pregrado). Universidad Nacional Mayor de San Marcos, Lima, Perú.

DADSS.org. (2015). *Touch-Based Technology*. Recuperado de <https://www.dadss.org/touch-based-technology/>

Draeger, Inc. (2015). *Dispositivo de Bloqueo de Encendido Draeger*. Lübeck, DE.: Draeger, Inc. Recuperado de <https://draegerinterlock.com/es/dispositivos-interlock/>

Electronilab. (2015). *Sensor de Etanol MQ3*. Bogota, CO.: Electronilab. Recuperado de <https://electronilab.co/tienda/sensor-de-alcohol-mq-3/>

Escuela Superior de Computo - Instituto Politécnico Nacional de México DF. (2009). *Diseño y Entrenamiento de Redes Neuronales Artificiales*. Recuperado de <https://es.slideshare.net/mentelibre/diseo-y-entrenamiento-de-redes-neuronales-artificiales>

Galván León, I. M., y Valls Ferrán, J. M. (2015). *Redes Neuronales Artificiales - Tema 3 Perceptron Multicapa*. Madrid, ES.: Universidad Carlos III de Madrid. Recuperado de <http://ocw.uc3m.es/ingenieria-informatica/redes-de-neuronas/transparencias/Tema3%20MLP.pdf>

Guillen, J. (2015). *Redes Neuronales Artificiales de Base Radial*. California, EU.: SlideShare. Recuperado de <https://es.slideshare.net/Jeffo92/redes-neuronales-50674867>

Hanwei Electronics Co.,Ltd. (2014). *Technical Data MQ3 Gas Sensor*. Recuperado de <https://www.sparkfun.com/datasheets/Sensors/MQ-3.pdf>

- Huircán Quilaqueo, J. I. (2014). *Filtros Activos, Conceptos Básicos y Diseño*. Temuco, CL.: Universidad de la Frontera - Departamento de Ingeniería Eléctrica. Recuperado de http://quidel.inele.ufro.cl/~jhuircan/PDF_CTOSII/Filtros%20Activos17.pdf
- Lavry, D. (2004). *Sampling Theory For Digital Audio*. Washington, EU.: Lavry Engineering, Inc. Recuperado de <http://lavryengineering.com/pdfs/lavry-sampling-theory.pdf>
- Lorenzo Fernández, P., Ladero, J., Leza Cerro, J. C., y Lizasoain Hernández, I. (2009). *Drogodependencias Farmacología. Patología. Psicología. Legislación*. (3ª ed.). Madrid, España: Panamericana.
- Maza Rivera, K. G. (2011). *Algoritmos de aprendizaje en visión artificial para la captura de objetos en un espacio tridimensional* (tesis de pregrado). Universidad Nacional de Ingeniería, Lima, Perú.
- MIT. (2015). *App Inventor 2*. Recuperado de <http://ai2.appinventor.mit.edu/>
- Organización Mundial de la Salud. (2015). *Alcohol*. Recuperado de <https://www.who.int/es/news-room/fact-sheets/detail/alcohol>
- Park, E. J. (27 de Abril de 2008). Sensor Report - MQ3 Gas Sensor [Mensaje en un blog]. Recuperado de <http://sensorworkshop.blogspot.com/2008/04/sensor-report-mq3-gas-sensor.html>
- Redacción EC. (20 de Abril de 2014). Muertes por accidentes de tránsito aumentaron en 25%. *El Comercio*. Recuperado de <https://elcomercio.pe/peru/la-libertad/muertes-accidentes-transito-aumentaron-25-312033>

Redacción Perú21. (28 de Diciembre de 2013). Accidentes viales dejan 2,430 muertos. *Perú21*. Recuperado de <https://peru21.pe/opinion/accidentes-viales-dejan-2-430-muertos-138152>

Redacción Perú21. (08 de Abril de 2014). Perú: Unas 550 víctimas en accidentes vehiculares en primer trimestre de 2014. *Perú21*. Recuperado de <https://peru21.pe/lima/peru-550-victimas-accidentes-vehiculares-primer-trimestre-2014-152699>

Redacción Publimetro. (29 de Diciembre de 2013). Imprudencia y ebriedad del conductor causa el 87% de accidentes de tránsito. *Publimetro*. Recuperado de <https://publimetro.pe/actualidad/noticia-imprudencia-y-ebriedad-conductor-causa-87-accidentes-transito-19238>

Sánchez Castellanos, L., y Meseas Teruel, D. (2009). *Redes Neuronales - Aplicaciones Empresariales Avanzadas*. California, EU.: SlideShare. Recuperado de https://es.slideshare.net/iquark/redes-neuronales-1426985?from_action=save

Servicio de Administración Tributaria. (2015). *Decreto Supremo N° 016-2009-MTC. Texto Único Ordenado del Reglamento Nacional de Tránsito – Código de Tránsito y modificatorias*. Lima, PE.: SAT. Recuperado de https://www.sat.gob.pe/WebSiteV8/Modulos/contenidos/mult_Papeletas_ti_rntv2.aspx

Universidad Tecnológica de Pereira. (2000). *Backpropagation*. Recuperado de <http://medicinaycomplejidad.org/pdf/redes/Backpropagation.pdf>

ÍNDICE DE ANEXOS

	Pág.
Anexo 1 Cálculo del Coeficiente de Correlación de Pearson	169
Anexo 2 Diagrama de Flujo de Generación de Entradas y de Salidas Deseadas de la RNA	175
Anexo 3 Código en Matlab del Algoritmo de Generación de Entradas y de Salidas Deseadas de la Red Neuronal	188
Anexo 4 Tabla Completa de Entrada a la Red Neuronal	192
Anexo 5 Tabla Completa de Salida Deseada de la Red Neuronal	231
Anexo 6 Diagrama de Flujo del Algoritmo de Retropropagación	270
Anexo 7 Código en Matlab del Algoritmo de Retropropagación	286
Anexo 8 Diagrama de Flujo General de la Aplicación	292
Anexo 9 Diagrama Circuitual Completo	293
Anexo 10 PCB Completo	294
Anexo 11 Glosario	295

Anexo 1 **Cálculo del Coeficiente de Correlación de Pearson**

El presente anexo ha sido desarrollado en su totalidad por los autores de la tesis.

Tabla 6. Medición del BAC detectado por un Alcoholímetro Calibrado VS BAC detectado por el sensor MQ3

Muestra	BAC Alcoholímetro Calibrado	BAC MQ3
1	0.57g/L	0.41g/L
2	0.54g/L	0.37g/L
3	0.53g/L	0.49g/L
4	0.47g/L	0.51g/L
5	0.58g/L	0.55g/L
6	0.61g/L	0.53g/L
7	0.45g/L	0.61g/L
8	0.55g/L	0.53g/L
9	0.57g/L	0.55g/L
10	0.53g/L	0.63g/L
11	0.53g/L	0.55g/L
12	0.56g/L	0.43g/L
13	0.31g/L	0.27g/L
14	0.45g/L	0.51g/L
15	0.57g/L	0.55g/L
16	0.36g/L	0.42g/L
17	0.49g/L	0.51g/L
18	0.57g/L	0.55g/L
19	0.22g/L	0.35g/L
20	0.37g/L	0.29g/L
21	0.48g/L	0.39g/L
22	0.43g/L	0.39g/L

23	0.57g/L	0.45g/L
24	0.63g/L	0.49g/L
25	0.52g/L	0.38g/L
26	0.57g/L	0.54g/L
27	0.54g/L	0.53g/L
28	0.51g/L	0.63g/L
29	0.56g/L	0.47g/L
30	0.53g/L	0.51g/L

Para calcular la correlación entre la concentración de alcohol en la sangre medida por el alcoholímetro calibrado de referencia y el sensor MQ3; se utilizó el coeficiente de correlación de Pearson, el cual está definido por la siguiente ecuación:

$$r = \frac{\sigma_{XY}}{\sigma_X \cdot \sigma_Y} \quad (139)$$

Donde σ_{XY} : covarianza entre las variable X e Y; σ_X : desviación estándar de la variable X y σ_Y : desviación estándar de la variable Y.

La Covarianza entre 2 variables X e Y, está definida por la siguiente ecuación:

$$\sigma_{XY} = \frac{\sum_{i=1}^N X_i \cdot Y_i}{N} - \bar{X} \cdot \bar{Y} \quad (140)$$

Donde N : número de datos de la muestra; \bar{X} : media aritmética de la variable X e \bar{Y} : media aritmética de la variable Y.

La desviación estándar de una variable X está definida por la siguiente ecuación:

$$\sigma_X = \sqrt{\frac{\sum_{i=1}^N X_i^2}{N} - \bar{X}^2} \quad (141)$$

Donde N : número de datos de la muestra y \bar{X} : media aritmética de la variable X.

La media aritmética de una variable X , está definida por la siguiente ecuación:

$$\bar{X} = \frac{\sum_{i=1}^N X_i}{N} \quad (142)$$

Donde N : número de datos de la muestra.

Para calcular las variables definidas en las ecuaciones desde la 142 hasta la 145, se construye la siguiente tabla, donde X : **BAC Alcoholímetro Calibrado** e Y : **BAC MQ3**

Tabla 7. Cálculo de Sumatorias Requeridas para Determinar el Coeficiente de Correlación de Pearson

X_i	Y_i	X_i^2	Y_i^2	$X.Y$
0.57	0.41	0.3249	0.1681	0.2337
0.54	0.37	0.2916	0.1369	0.1998
0.53	0.49	0.2809	0.2401	0.2597
0.47	0.51	0.2209	0.2601	0.2397
0.58	0.55	0.3364	0.3025	0.319
0.61	0.53	0.3721	0.2809	0.3233
0.45	0.61	0.2025	0.3721	0.2745
0.55	0.53	0.3025	0.2809	0.2915
0.57	0.55	0.3249	0.3025	0.3135
0.53	0.63	0.2809	0.3969	0.3339
0.53	0.55	0.2809	0.3025	0.2915
0.56	0.43	0.3136	0.1849	0.2408
0.31	0.27	0.0961	0.0729	0.0837
0.45	0.51	0.2025	0.2601	0.2295
0.57	0.55	0.3249	0.3025	0.3135
0.36	0.42	0.1296	0.1764	0.1512
0.49	0.51	0.2401	0.2601	0.2499
0.57	0.55	0.3249	0.3025	0.3135
0.22	0.35	0.0484	0.1225	0.077
0.37	0.29	0.1369	0.0841	0.1073
0.48	0.39	0.2304	0.1521	0.1872
0.43	0.39	0.1849	0.1521	0.1677
0.57	0.45	0.3249	0.2025	0.2565
0.63	0.49	0.3969	0.2401	0.3087
0.52	0.38	0.2704	0.1444	0.1976
0.57	0.54	0.3249	0.2916	0.3078
0.54	0.53	0.2916	0.2809	0.2862
0.51	0.63	0.2601	0.3969	0.3213
0.56	0.47	0.3136	0.2209	0.2632
0.53	0.51	0.2809	0.2601	0.2703
15.17	14.39	7.9141	7.1511	7.413

Con las sumatorias calculadas en la tabla anterior, se calculan las variables definidas en las ecuaciones desde la 142 hasta la 145:

Cálculo de la media aritmética de la variable X:

$$\bar{X} = \frac{\sum_{i=1}^N X_i}{N} = \frac{15.17}{30} \approx 0.50567 \quad (143)$$

Cálculo de la media aritmética de la variable Y:

$$\bar{Y} = \frac{\sum_{i=1}^N Y_i}{N} = \frac{14.39}{30} \approx 0.47967 \quad (144)$$

Cálculo de la desviación estándar de la variable X:

$$\sigma_X = \sqrt{\frac{\sum_{i=1}^N X_i^2}{N} - \bar{X}^2} = \sqrt{\frac{7.9141}{30} - 0.50567^2} \approx 0.09001 \quad (145)$$

Cálculo de la desviación estándar de la variable Y:

$$\sigma_Y = \sqrt{\frac{\sum_{i=1}^N Y_i^2}{N} - \bar{Y}^2} = \sqrt{\frac{7.1511}{30} - 0.47967^2} \approx 0.09103 \quad (146)$$

Cálculo de la covarianza entre las variables X e Y:

$$\sigma_{XY} = \frac{\sum_{i=1}^N X_i \cdot Y_i}{N} - \bar{X} \cdot \bar{Y} = \frac{7.413}{30} - (0.50567)(0.47967) \approx 0.00454 \quad (147)$$

Finalmente, se calcula el coeficiente de correlación de Pearson entre las variables X e Y:

$$r = \frac{\sigma_{XY}}{\sigma_X \cdot \sigma_Y} = \frac{0.00454}{(0.09001)(0.09103)} \approx 0.55409 \quad (148)$$

Por tanto, se concluye que la relación que existe entre las 2 variables X : **BAC Alcoholímetro Calibrado** e Y : **BAC MQ3** tienen dependencia positiva, es decir que las 2 variables están relacionadas de manera directamente proporcional y debido a que el coeficiente de correlación de Pearson calculado tiene un valor más cercano a 1 que a 0, la dependencia entre ambas variables es bastante fuerte.

Anexo 2 Diagrama de Flujo de Generación de Entradas y de Salidas Deseadas de la RNA

Este anexo ha sido desarrollado en su totalidad por los autores de la presente tesis mediante la aplicación web code2flow.

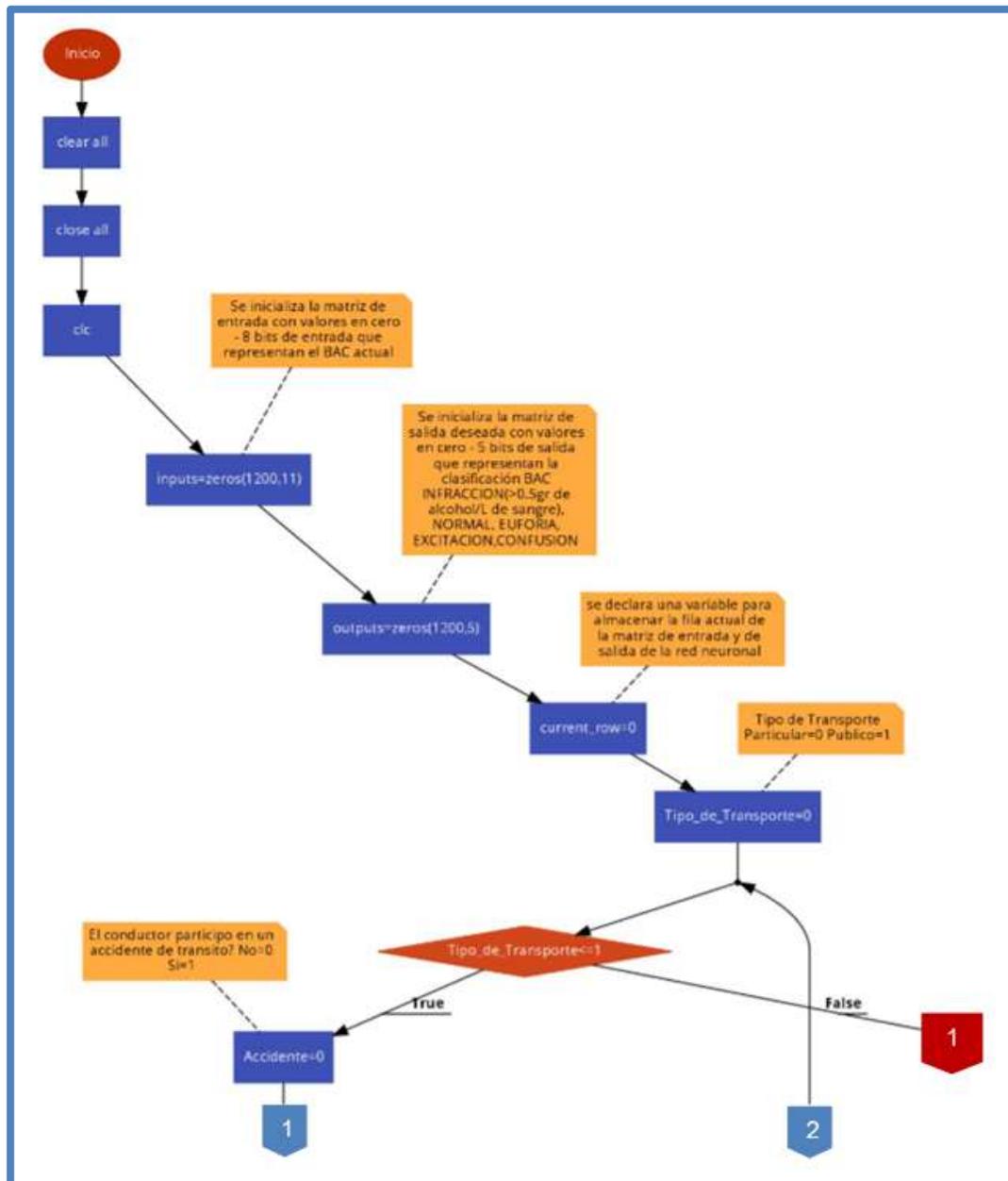


Fig. 121 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Principal

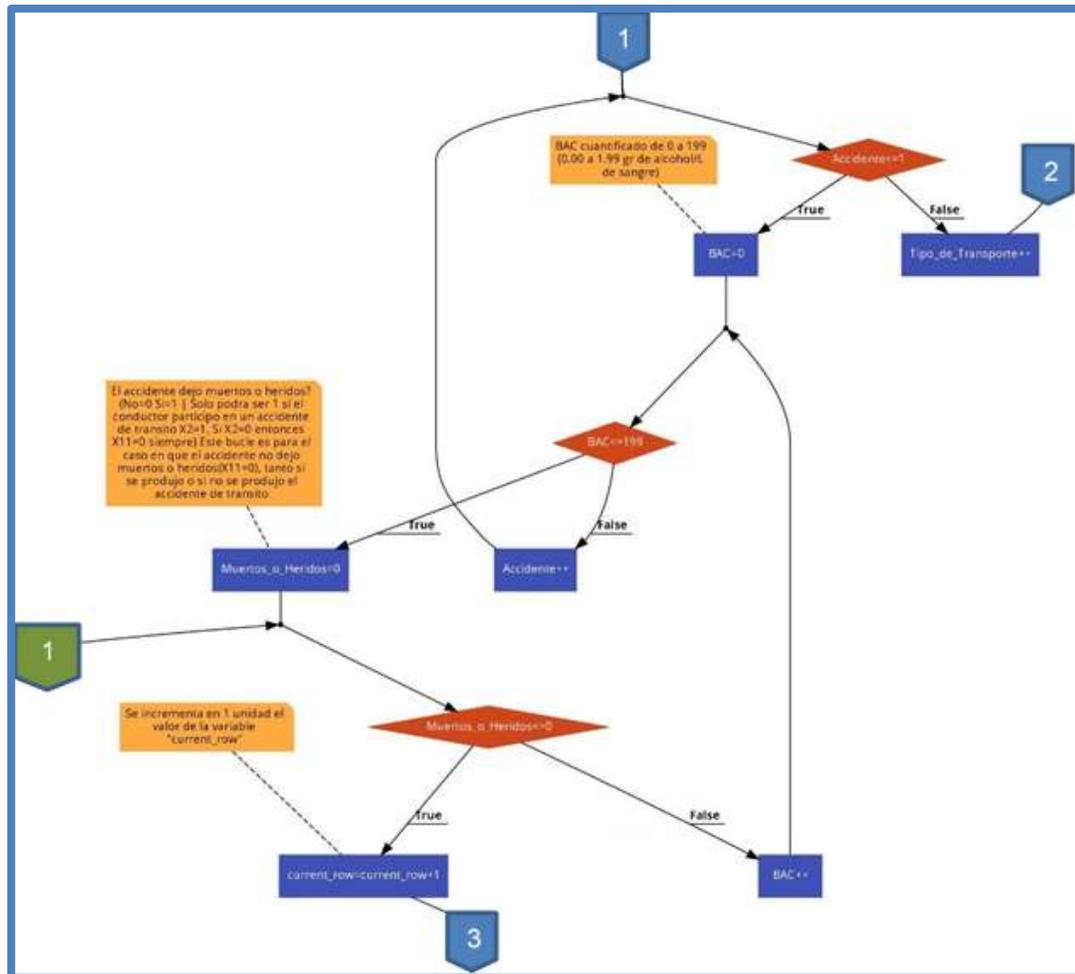


Fig. 122 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal- Rama Izquierda – Parte 1

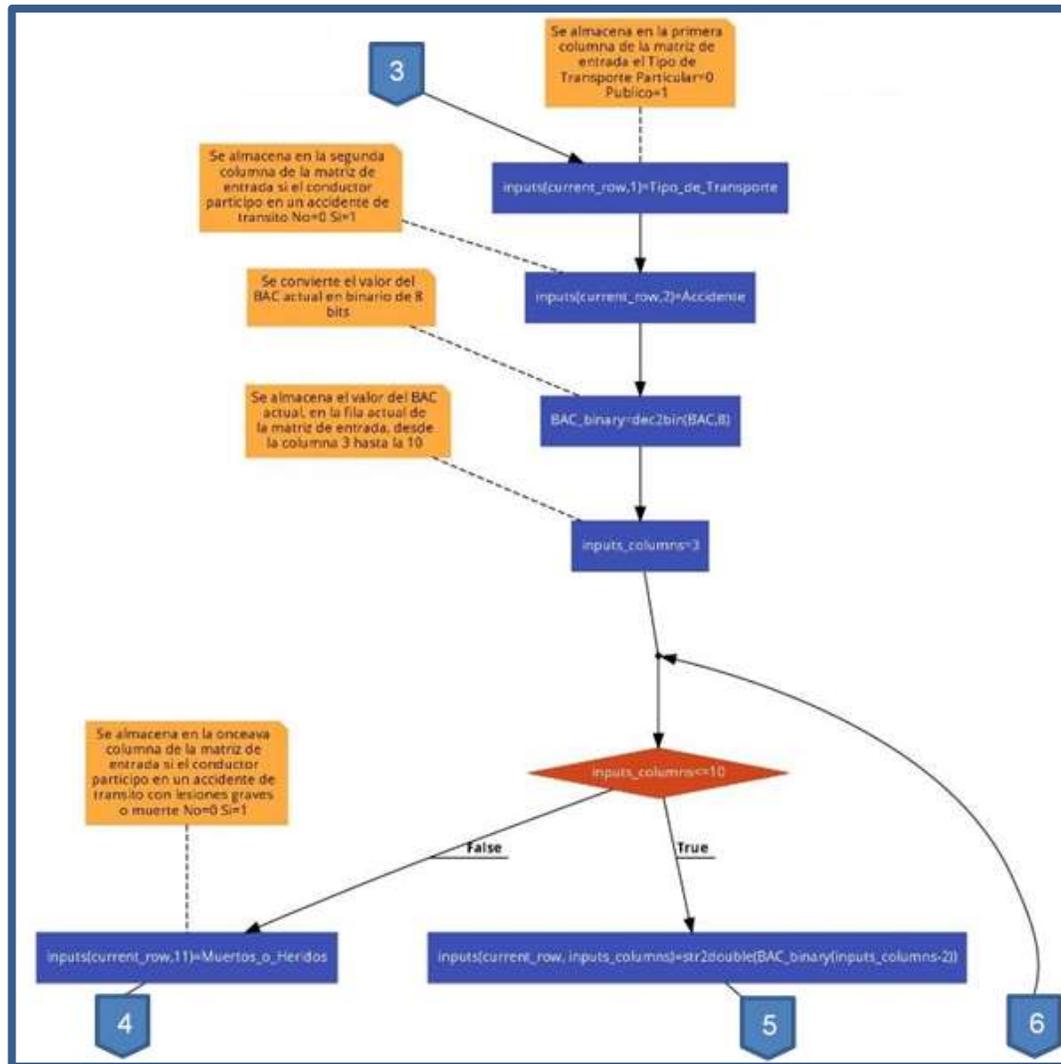


Fig. 123 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Izquierda – Parte 2

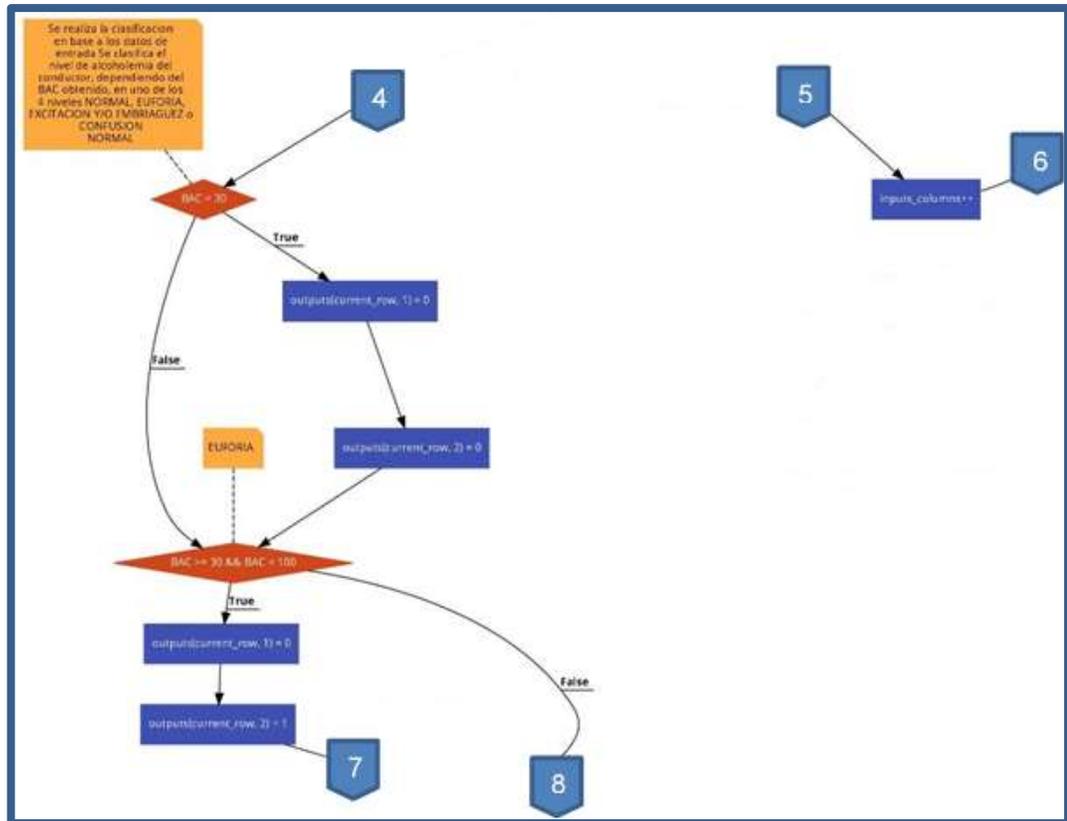


Fig. 124 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Izquierda – Parte 3

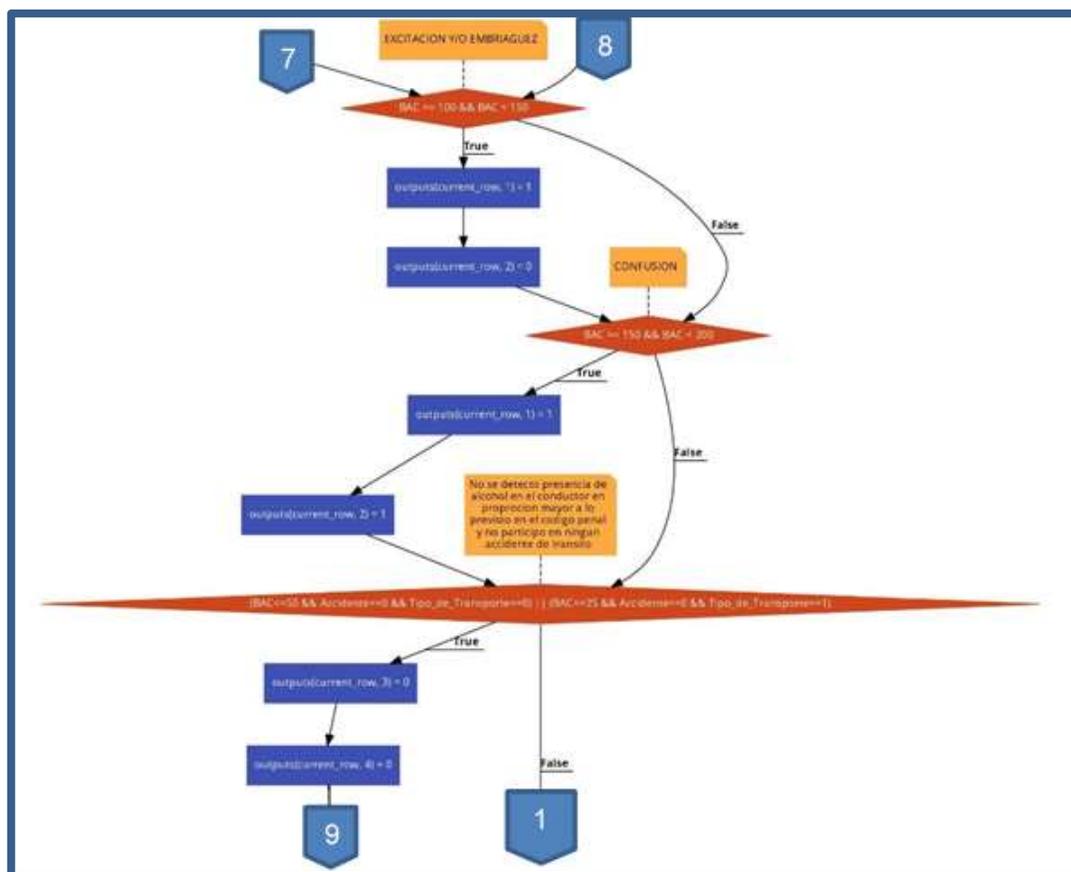


Fig. 125 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Izquierda – Parte 4

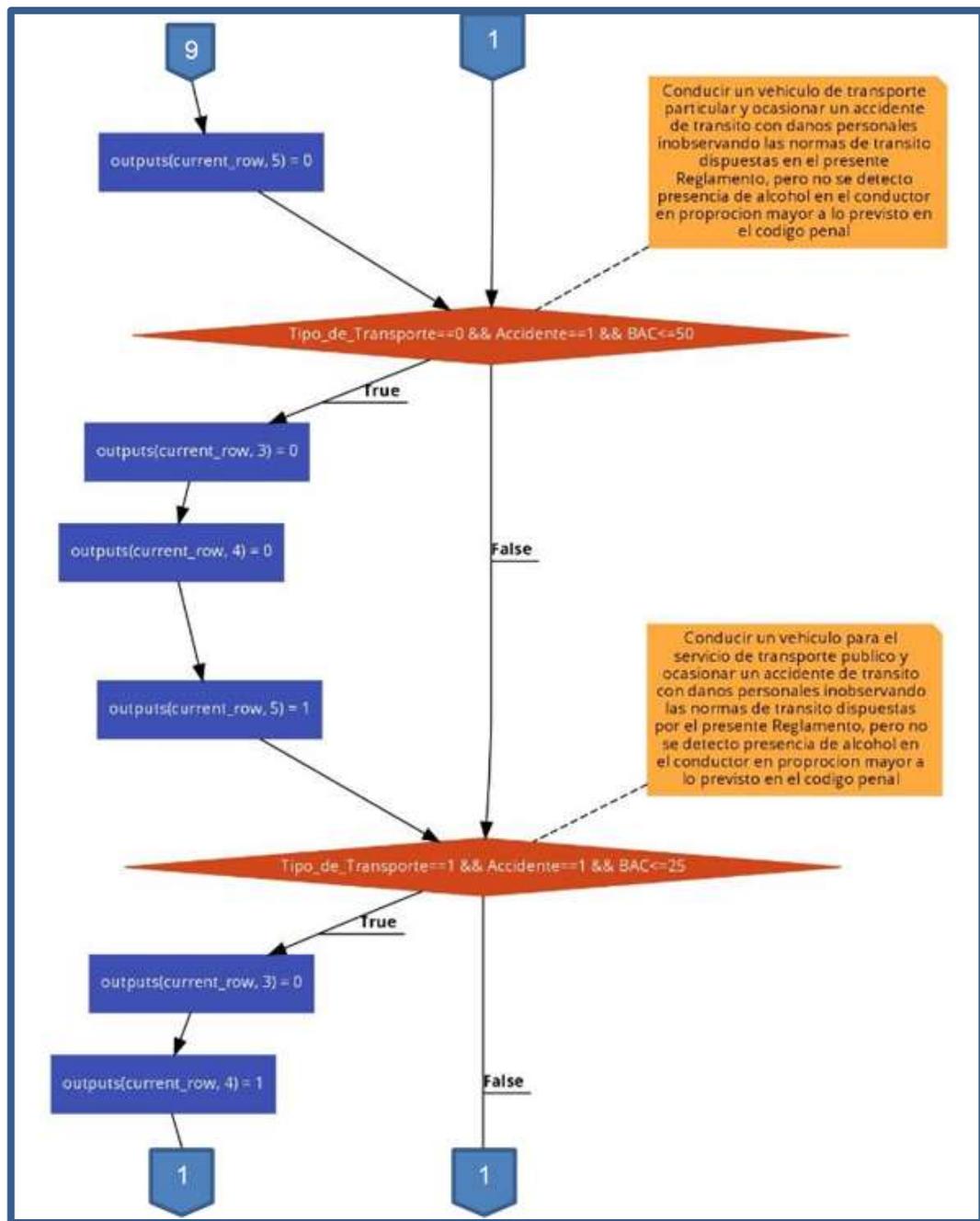


Fig. 126 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Izquierda – Parte 5

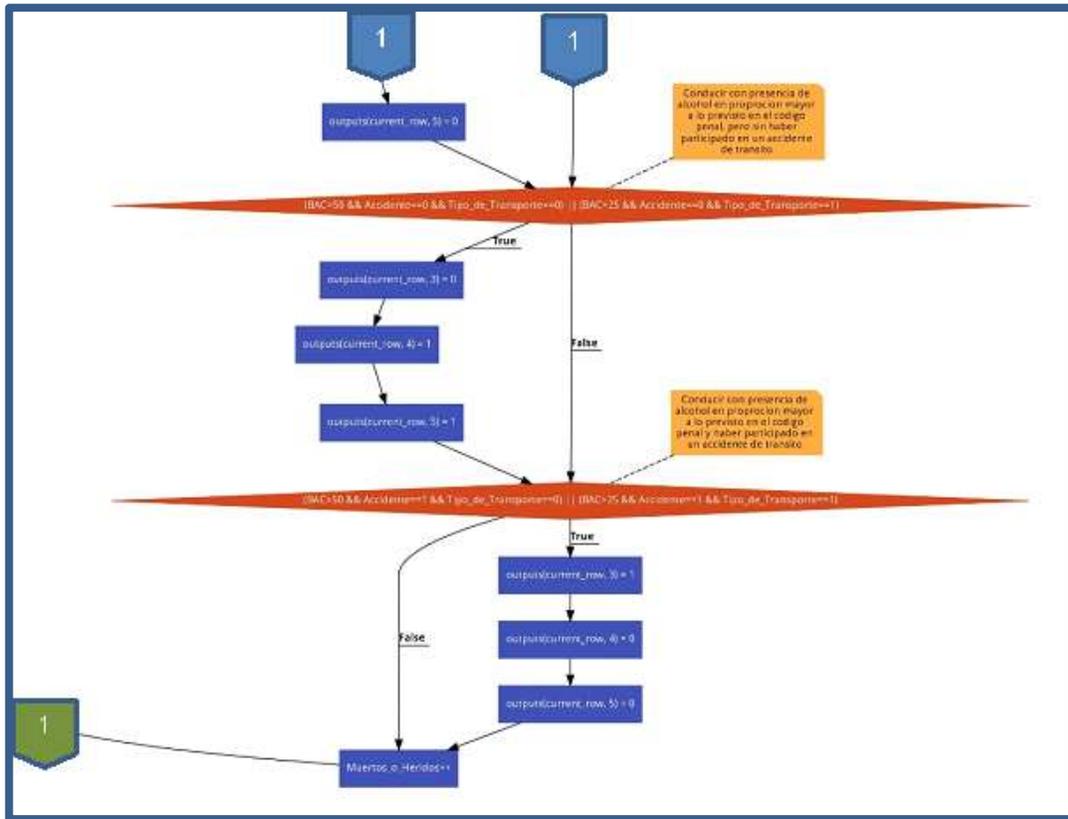


Fig. 127 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Izquierda – Parte 6

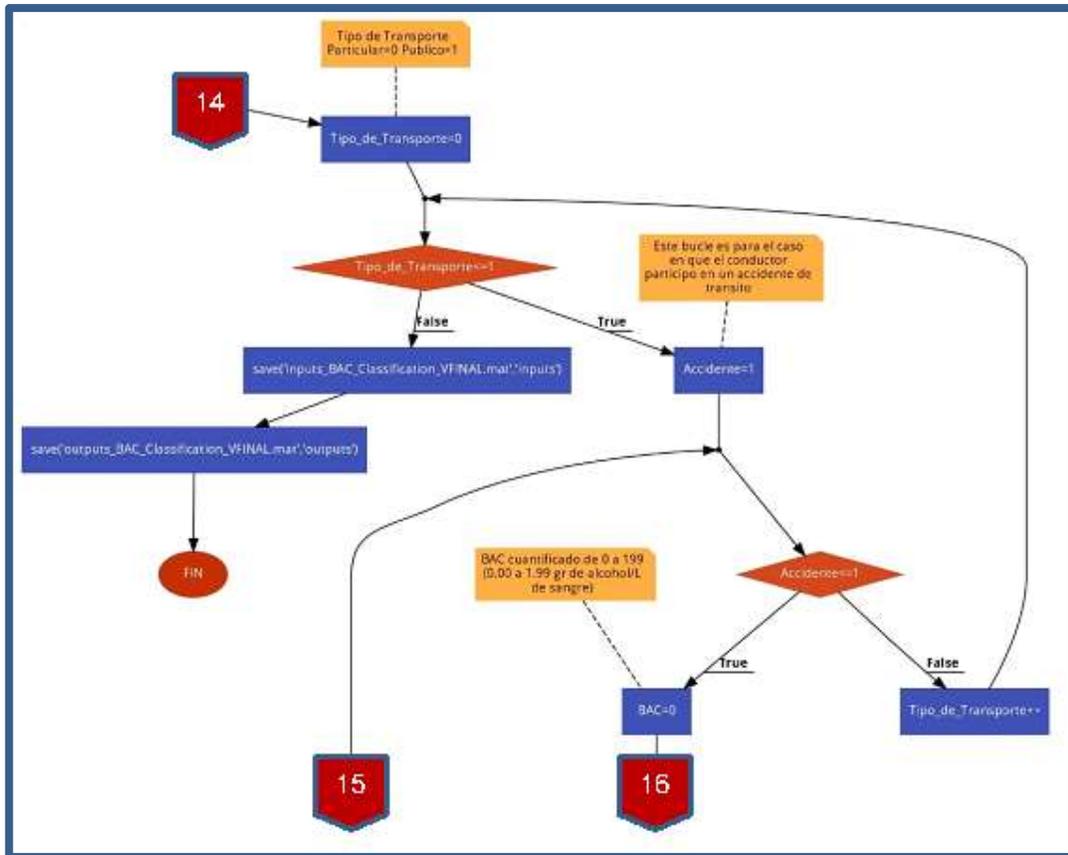


Fig. 128 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Derecha – Parte 1

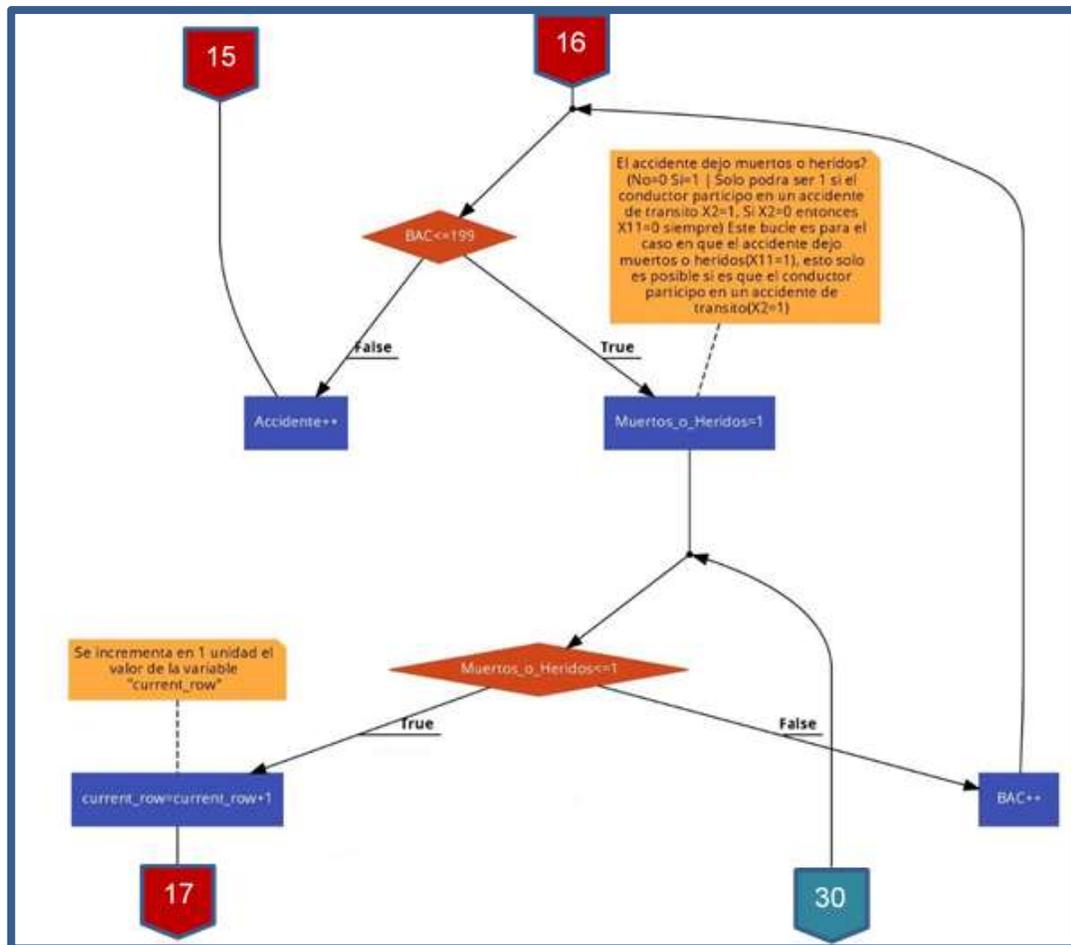


Fig. 129 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Derecha – Parte 2

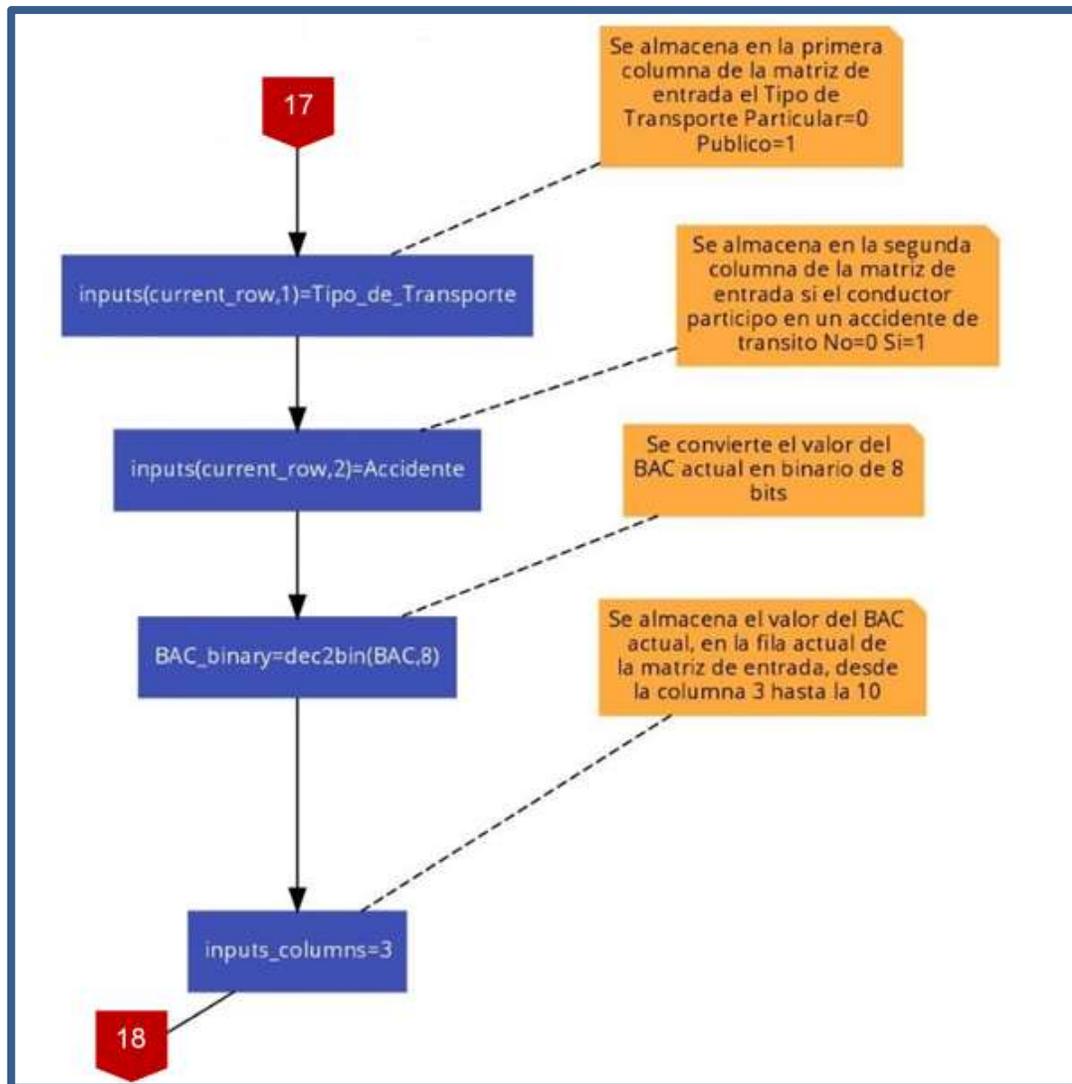


Fig. 130 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Derecha – Parte 3

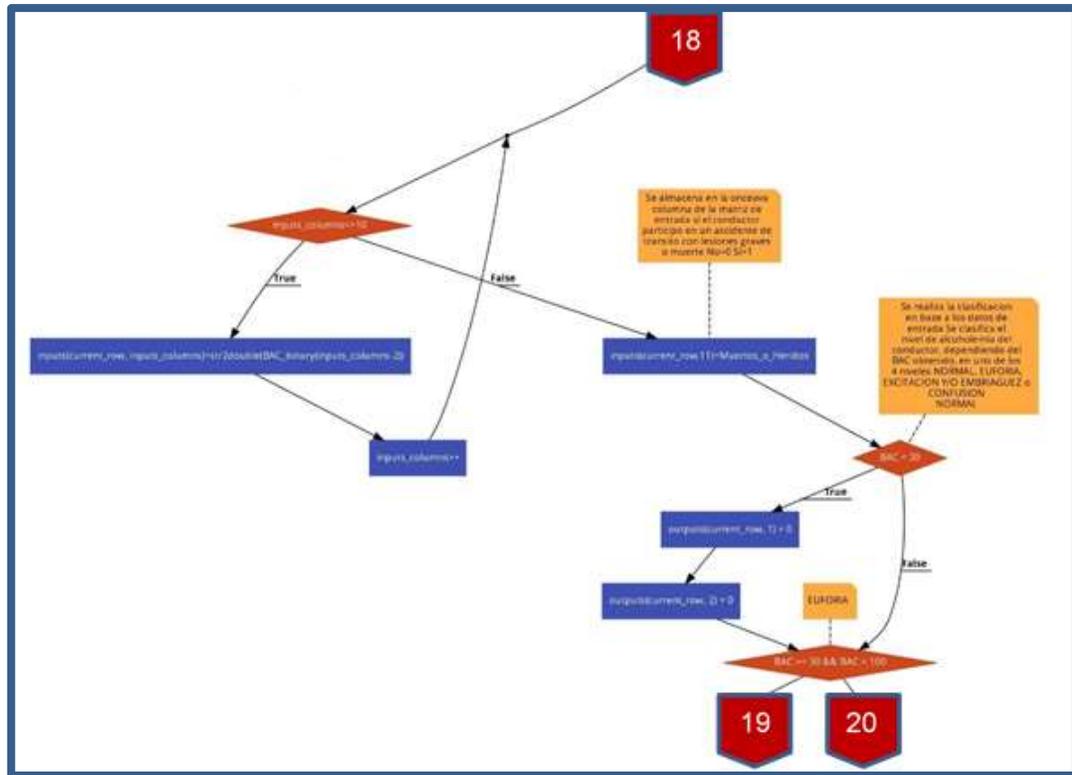


Fig. 131 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Derecha – Parte 4

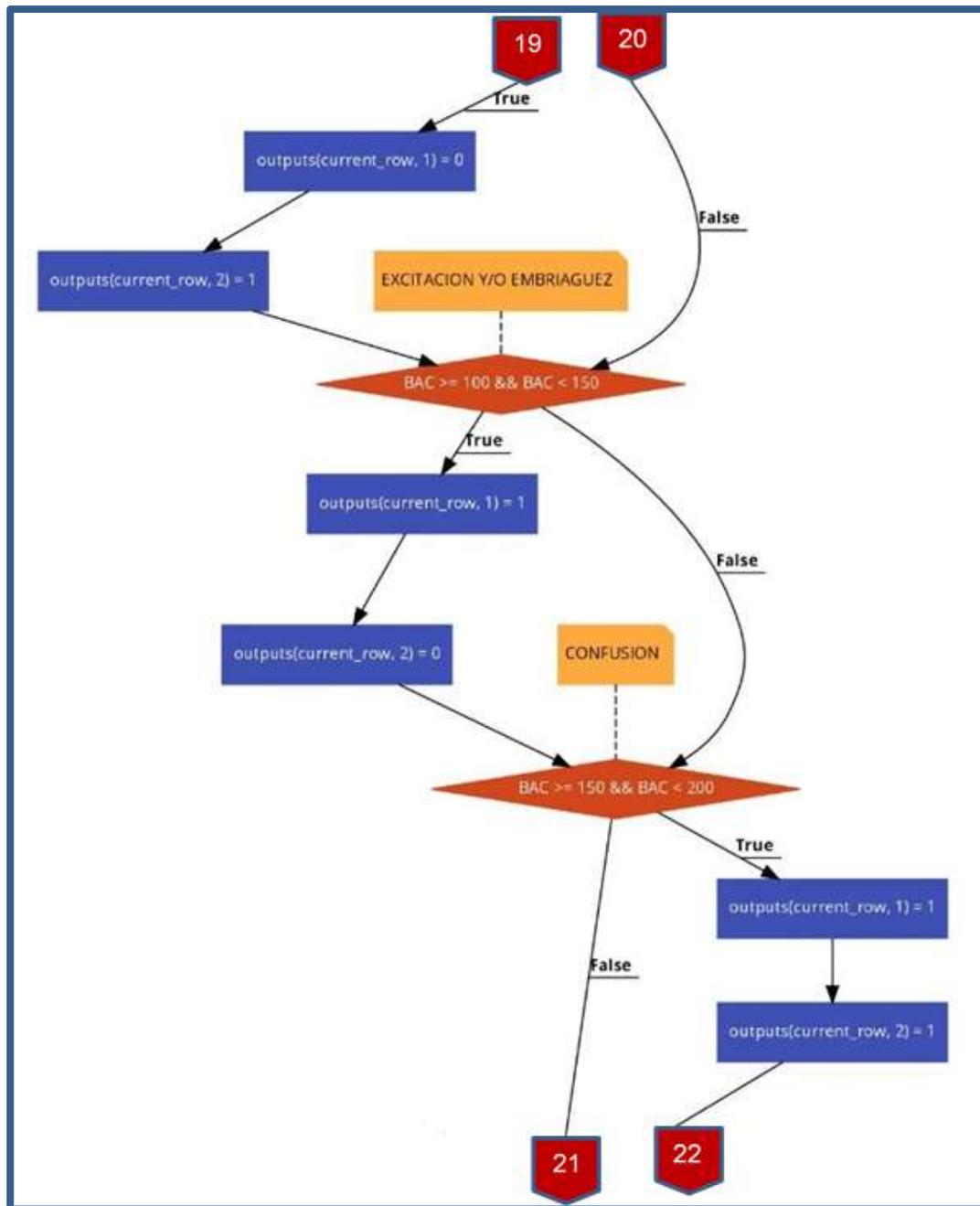


Fig. 132 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Derecha – Parte 5

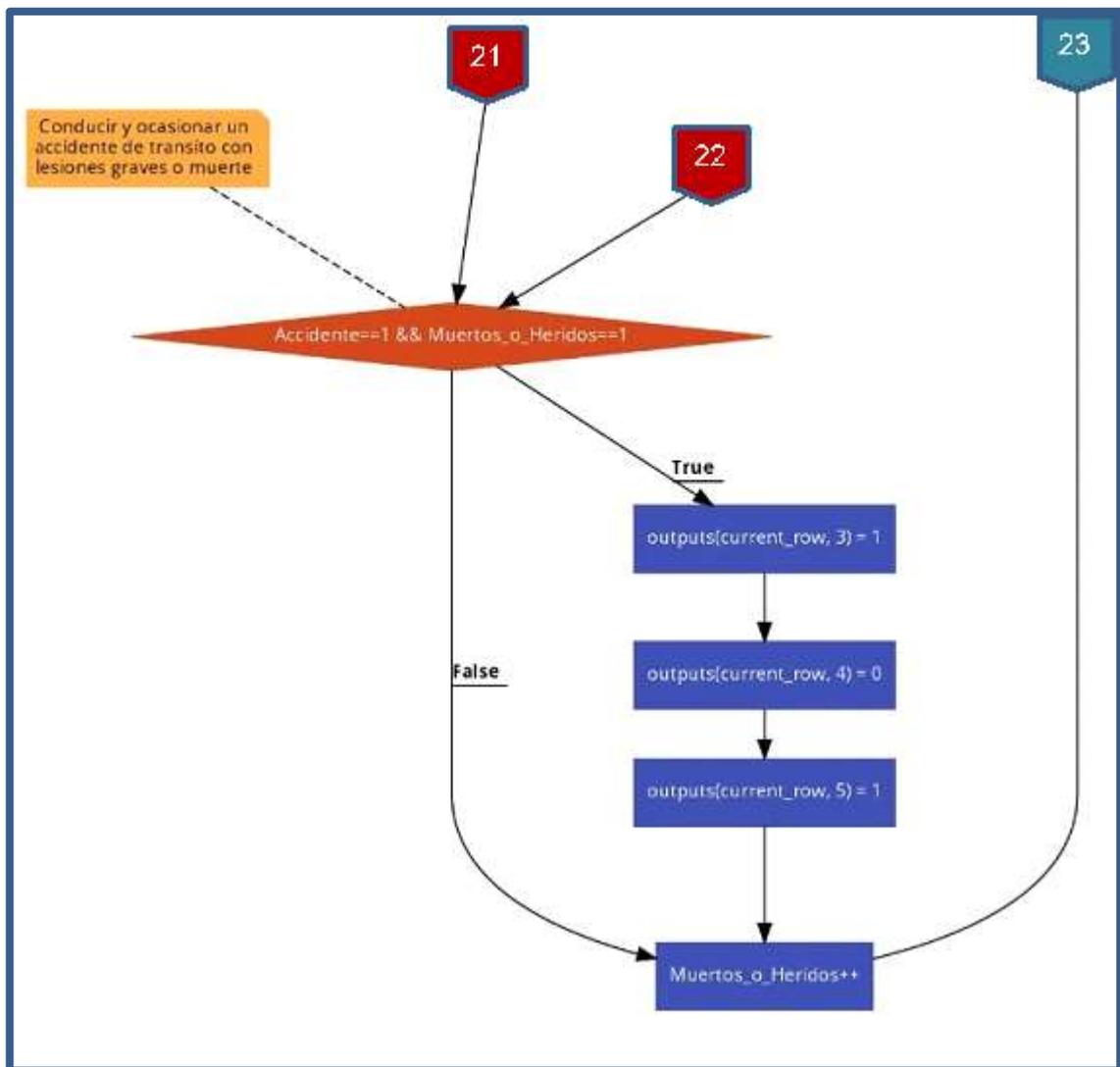


Fig. 133 Diagrama de Flujo del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal– Rama Derecha – Parte 6

Anexo 3 Código en Matlab del Algoritmo de Generación de Entradas y de Salidas Deseadas de la Red Neuronal

Este anexo ha sido desarrollado en su totalidad por los autores de la presente tesis mediante el software Matlab.

```
1 clear all;
2
3 close all;
4
5 clc;
6
7 %Se inicializa la matriz de entrada con valores en cero - 8 bits de
8 %entrada que representan el BAC actual
9 inputs=zeros(1200,11);
10
11 %Se inicializa la matriz de salida deseada con valores en cero - 5 bits de
12 %salida que representan la clasificaci?n BAC INFRACCION(>0.5gr de alcohol/L de
sangre), NORMAL, EUPORIA, EXCITACION,
13 %CONFUSION
14 outputs=zeros(1200,5);
15
16 %se declara una variable para almacenar la fila actual de la matriz de entrada y
17 %de salida de la red neuronal
18 current_row=0;
19
20 %Tipo de Transporte Particular=0 Publico=1
21 for Tipo_de_Transporte=0:1
22     %El conductor participo en un accidente de transito? No=0 Si=1
23     for Accidente=0:1
24         %BAC cuantificado de 0 a 199 (0.00 a 1.99 gr de alcohol/L de sangre)
25         for BAC=0:199
26             %El accidente dejo muertos o heridos? (No=0 Si=1 | Solo podra ser 1 si
el conductor participo en un accidente de transito X2=1, Si X2=0 entonces X1=
siempre)
27             %Este bucle es para el caso en que el accidente no dejo
28             %muertos o heridos(X1=0), tanto si se produjo o si no se produjo
29             %el accidente de transito
30             for Muertos_o_Heridos=0
31
32                 %Se incrementa en 1 unidad el valor de la variable "current_row"
33                 current_row=current_row+1;
34
35                 %Se almacena en la primera columna de la matriz de entrada el
36                 %Tipo de Transporte Particular=0 Publico=1
37                 inputs(current_row,1)=Tipo_de_Transporte;
38
39                 %Se almacena en la segunda columna de la matriz de entrada si
40                 %el conductor participo en un accidente de transito No=0 Si=1
41                 inputs(current_row,2)=Accidente;
42
43                 %Se convierte el valor del BAC actual en binario de 8 bits
44                 BAC_binary=dec2bin(BAC,8);
45
46                 %Se almacena el valor del BAC actual, en la fila actual de la
matriz de
47                 %entrada, desde la columna 3 hasta la 10
48                 for inputs_columns=3:10
49                     inputs(current_row, inputs_columns)=str2double(BAC_binary
(inputs_columns-2));
50                 end
51
52                 %Se almacena en la onceava columna de la matriz de entrada si
53                 %el conductor participo en un accidente de transito con lesiones
54                 % graves o muerte No=0 Si=1
55                 inputs(current_row,11)=Muertos_o_Heridos;
```

Fig. 134 Código en Matlab del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal – Parte 1

```

56
57      %Se realiza la clasificacion en base a los datos de entrada
58      %Se clasifica el nivel de alcoholemia del conductor, dependiendo
del BAC obtenido, en uno de los 4 niveles NORMAL, EUFORIA, EXCITACION Y/O EMBRIAGUEZ
CONFUSION
59
60      %NORMAL
61      if(BAC < 30)
62          outputs(current_row, 1) = 0;
63          outputs(current_row, 2) = 0;
64      end
65
66      %EUFORIA
67      if(BAC >= 30 && BAC < 100)
68          outputs(current_row, 1) = 0;
69          outputs(current_row, 2) = 1;
70      end
71
72      %EXCITACION Y/O EMBRIAGUEZ
73      if(BAC >= 100 && BAC < 150)
74          outputs(current_row, 1) = 1;
75          outputs(current_row, 2) = 0;
76      end
77
78      %CONFUSION
79      if(BAC >= 150 && BAC < 200)
80          outputs(current_row, 1) = 1;
81          outputs(current_row, 2) = 1;
82      end
83
84      %No se detecto presencia de alcohol en el conductor en
85      %proporcion mayor a lo previsto en el codigo penal y no
86      %participo en ningun accidente de transito
87      if((BAC<=50 && Accidente==0 && Tipo_de_Transporte==0) || (BAC<=25
&& Accidente==0 && Tipo_de_Transporte==1))
88          outputs(current_row, 3) = 0;
89          outputs(current_row, 4) = 0;
90          outputs(current_row, 5) = 0;
91      end
92
93      %Conducir un vehiculo de transporte particular y ocasionar
94      %un accidente de transito con danos personales inobservando
95      %las normas de transito dispuestas en el presente Reglamento,
96      %pero no se detecto presencia de alcohol en el conductor
97      %en proporción mayor a lo previsto en el código penal
98      if(Tipo_de_Transporte==0 && Accidente==1 && BAC<=50)
99          outputs(current_row, 3) = 0;
100         outputs(current_row, 4) = 0;
101         outputs(current_row, 5) = 1;
102     end
103
104     %Conducir un vehiculo para el servicio de transporte publico y
105     %ocasionar un accidente de transito con danos personales
106     %inobservando las normas de transito dispuestas por el presente
107     %Reglamento, pero no se detecto presencia de alcohol en el
108     %conductor en proporción mayor a lo previsto en el código penal
109     if(Tipo_de_Transporte==1 && Accidente==1 && BAC<=25)
110         outputs(current_row, 3) = 0;
111         outputs(current_row, 4) = 1;
112         outputs(current_row, 5) = 0;

```

Fig. 135 Código en Matlab del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal – Parte 2

```

113         end
114
115         %Conducir con presencia de alcohol en proporción mayor
116         %a lo previsto en el código penal, pero sin haber participado
117         %en un accidente de tránsito
118         if((BAC>50 && Accidente==0 && Tipo_de_Transporte==0) || (BAC>25 &&
Accidente==0 && Tipo_de_Transporte==1))
119             outputs(current_row, 3) = 0;
120             outputs(current_row, 4) = 1;
121             outputs(current_row, 5) = 1;
122         end
123
124         %Conducir con presencia de alcohol en proporción mayor
125         %a lo previsto en el código penal y haber participado
126         %en un accidente de tránsito
127         if((BAC>50 && Accidente==1 && Tipo_de_Transporte==0) || (BAC>25 &&
Accidente==1 && Tipo_de_Transporte==1))
128             outputs(current_row, 3) = 1;
129             outputs(current_row, 4) = 0;
130             outputs(current_row, 5) = 0;
131         end
132     end
133 end
134 end
135 end
136 end
137
138
139
140 %Tipo de Transporte Particular=0 Publico=1
141 for Tipo_de_Transporte=0:1
142     %Este bucle es para el caso en que el conductor participo en un
143     %accidente de tránsito
144     for Accidente=1
145         %BAC cuantificado de 0 a 199 (0.00 a 1.99 gr de alcohol/L de sangre)
146         for BAC=0:199
147             %El accidente dejo muertos o heridos? (No=0 Si=1 | Solo podra ser 1 si
el conductor participo en un accidente de tránsito X2=1, Si X2=0 entonces X11=
siempre)
148             %Este bucle es para el caso en que el accidente dejo
149             %muertos o heridos(X11=1), esto solo es posible si es que el
150             %conductor participo en un accidente de tránsito(X2=1)
151             for Muertos_o_Heridos=1
152
153                 %Se incrementa en 1 unidad el valor de la variable "current_row"
154                 current_row=current_row+1;
155
156                 %Se almacena en la primera columna de la matriz de entrada el
157                 %Tipo de Transporte Particular=0 Publico=1
158                 inputs(current_row,1)=Tipo_de_Transporte;
159
160                 %Se almacena en la segunda columna de la matriz de entrada si
161                 %el conductor participo en un accidente de tránsito No=0 Si=1
162                 inputs(current_row,2)=Accidente;
163
164                 %Se convierte el valor del BAC actual en binario de 8 bits
165                 BAC_binary=dec2bin(BAC,8);
166
167                 %Se almacena el valor del BAC actual, en la fila actual de la
matriz de

```

Fig. 136 Código en Matlab del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal – Parte 3

```

168         %entrada, desde la columna 3 hasta la 10
169         for inputs_columns=3:10
170             inputs(current_row, inputs_columns)=str2double(BAC_bina
(inputs_columns-2));
171         end
172
173         %Se almacena en la onceava columna de la matriz de entrada si
174         %el conductor participo en un accidente de transito con lesiones
175         % graves o muerte No=0 Si=1
176         inputs(current_row,11)=Muertos_o_Heridos;
177
178         %Se realiza la clasificacion en base a los datos de entrada
179         %Se clasifica el nivel de alcoholemia del conductor, dependiendo
del BAC obtenido, en uno de los 4 niveles NORMAL, EUFORIA, EXCITACION Y/O EMBRIAGUEZ
CONFUSION
180
181         %NORMAL
182         if(BAC < 30)
183             outputs(current_row, 1) = 0;
184             outputs(current_row, 2) = 0;
185         end
186
187         %EUFORIA
188         if(BAC >= 30 && BAC < 100)
189             outputs(current_row, 1) = 0;
190             outputs(current_row, 2) = 1;
191         end
192
193         %EXCITACION Y/O EMBRIAGUEZ
194         if(BAC >= 100 && BAC < 150)
195             outputs(current_row, 1) = 1;
196             outputs(current_row, 2) = 0;
197         end
198
199         %CONFUSION
200         if(BAC >= 150 && BAC < 200)
201             outputs(current_row, 1) = 1;
202             outputs(current_row, 2) = 1;
203         end
204
205         %Conducir y ocasionar un accidente de transito con lesiones grave
o muerte
206         if(Accidente==1 && Muertos_o_Heridos==1)
207             outputs(current_row, 3) = 1;
208             outputs(current_row, 4) = 0;
209             outputs(current_row, 5) = 1;
210         end
211     end
212 end
213 end
214 end
215 end
216
217 save('inputs_BAC_Classification_VFINAL.mat','inputs');
218
219 save('outputs_BAC_Classification_VFINAL.mat','outputs');
220
221
222
223

```

Fig. 137 Código en Matlab del Algoritmo de Generación de entradas y de salidas deseadas de la Red Neuronal – Parte 4

Anexo 4 **Tabla Completa de Entrada a la Red Neuronal**

Este anexo ha sido desarrollado en su totalidad por los autores de la presente tesis.

Inputs =

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	1	0
0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	1	1	0	1	0
0	0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	0	1	1	1	1	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	1	0	0	1	0	0
0	0	0	0	0	1	0	0	1	1	0
0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	1	0	1	0	1	0
0	0	0	0	0	1	0	1	1	0	0
0	0	0	0	0	1	0	1	1	1	0

0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	1	1	0	0	1	0
0	0	0	0	0	1	1	0	1	0	0
0	0	0	0	0	1	1	0	1	1	0
0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	1	1	1	0	1	0
0	0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	1	1	1	1	1	0
0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	1	0	0	0
0	0	0	0	1	0	0	1	1	0	0
0	0	0	0	1	0	0	1	1	1	0
0	0	0	0	1	0	1	0	0	0	0
0	0	0	0	1	0	1	0	0	1	0
0	0	0	0	1	0	1	0	1	0	0
0	0	0	0	1	0	1	0	1	1	0
0	0	0	0	1	0	1	1	0	0	0
0	0	0	0	1	0	1	1	0	1	0
0	0	0	0	1	0	1	1	1	0	0
0	0	0	0	1	0	1	1	1	1	0
0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	1	1	0	0	0	1	0
0	0	0	0	1	1	0	0	1	0	0
0	0	0	0	1	1	0	0	1	1	0
0	0	0	0	1	1	0	1	0	0	0
0	0	0	0	1	1	0	1	0	1	0
0	0	0	0	1	1	0	1	1	0	0

0	0	0	0	1	1	0	1	1	1	0
0	0	0	0	1	1	1	0	0	0	0
0	0	0	0	1	1	1	0	0	1	0
0	0	0	0	1	1	1	0	1	0	0
0	0	0	0	1	1	1	0	1	1	0
0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	0	1	0
0	0	0	0	1	1	1	1	1	0	0
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	1	1	0
0	0	0	1	0	0	0	1	0	0	0
0	0	0	1	0	0	0	1	1	0	0
0	0	0	1	0	0	0	1	1	1	0
0	0	0	1	0	0	1	0	0	0	0
0	0	0	1	0	0	1	0	0	1	0
0	0	0	1	0	0	1	0	1	0	0
0	0	0	1	0	0	1	0	1	1	0
0	0	0	1	0	0	1	1	0	0	0
0	0	0	1	0	0	1	1	1	0	0
0	0	0	1	0	0	1	1	1	1	0
0	0	0	1	0	1	0	0	0	0	0
0	0	0	1	0	1	0	0	0	1	0
0	0	0	1	0	1	0	0	1	0	0
0	0	0	1	0	1	0	0	1	0	0
0	0	0	1	0	1	0	0	1	1	0
0	0	0	1	0	1	0	1	0	0	0
0	0	0	1	0	1	0	1	0	1	0

0	0	0	1	0	1	0	1	1	0	0
0	0	0	1	0	1	0	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	0	1	0	1	1	0	0	1	0
0	0	0	1	0	1	1	0	1	0	0
0	0	0	1	0	1	1	0	1	1	0
0	0	0	1	0	1	1	1	0	0	0
0	0	0	1	0	1	1	1	0	1	0
0	0	0	1	0	1	1	1	1	0	0
0	0	0	1	0	1	1	1	1	1	0
0	0	0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	1	0
0	0	0	1	1	0	0	0	1	0	0
0	0	0	1	1	0	0	0	1	1	0
0	0	0	1	1	0	0	1	0	0	0
0	0	0	1	1	0	0	1	1	0	0
0	0	0	1	1	0	0	1	1	1	0
0	0	0	1	1	0	1	0	0	0	0
0	0	0	1	1	0	1	0	0	1	0
0	0	0	1	1	0	1	0	1	0	0
0	0	0	1	1	0	1	0	1	1	0
0	0	0	1	1	0	1	1	0	0	0
0	0	0	1	1	0	1	1	0	1	0
0	0	0	1	1	0	1	1	1	0	0
0	0	0	1	1	0	1	1	1	1	0
0	0	0	1	1	1	0	0	0	0	0
0	0	0	1	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0	0
0	0	0	1	1	1	0	0	1	1	0
0	0	0	1	1	1	0	1	0	0	0

0	0	0	1	1	1	0	1	0	1	0	
0	0	0	1	1	1	0	1	1	0	0	
0	0	0	1	1	1	0	1	1	1	0	
0	0	0	1	1	1	1	0	0	0	0	
0	0	0	1	1	1	1	0	0	1	0	
0	0	0	1	1	1	1	0	1	0	0	
0	0	0	1	1	1	1	0	1	1	0	
0	0	0	1	1	1	1	1	0	0	0	
0	0	0	1	1	1	1	1	0	1	0	
0	0	0	1	1	1	1	1	1	0	0	
0	0	0	1	1	1	1	1	1	1	0	
0	0	1	0	0	0	0	0	0	0	0	
0	0	1	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	1	0	0
0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1	0	1	0	0
0	0	1	0	0	0	0	1	1	1	0	0
0	0	1	0	0	0	1	1	0	0	0	0
0	0	1	0	0	0	1	1	0	1	0	0
0	0	1	0	0	0	1	1	1	0	0	0
0	0	1	0	0	0	1	1	1	1	0	0
0	0	1	0	0	1	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	1	0	0
0	0	1	0	0	1	0	0	1	0	0	0
0	0	1	0	0	1	0	0	1	1	0	0

0	0	1	0	0	1	0	1	0	0	0
0	0	1	0	0	1	0	1	0	1	0
0	0	1	0	0	1	0	1	1	0	0
0	0	1	0	0	1	0	1	1	1	0
0	0	1	0	0	1	1	0	0	0	0
0	0	1	0	0	1	1	0	0	1	0
0	0	1	0	0	1	1	0	1	0	0
0	0	1	0	0	1	1	0	1	1	0
0	0	1	0	0	1	1	1	0	0	0
0	0	1	0	0	1	1	1	0	1	0
0	0	1	0	0	1	1	1	1	1	0
0	0	1	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	1	0
0	0	1	0	1	0	0	0	1	0	0
0	0	1	0	1	0	0	0	1	1	0
0	0	1	0	1	0	0	1	0	0	0
0	0	1	0	1	0	0	1	0	1	0
0	0	1	0	1	0	0	1	1	0	0
0	0	1	0	1	0	0	1	1	1	0
0	0	1	0	1	0	1	0	0	0	0
0	0	1	0	1	0	1	0	0	1	0
0	0	1	0	1	0	1	0	1	0	0
0	0	1	0	1	0	1	0	1	1	0
0	0	1	0	1	0	1	1	0	0	0
0	0	1	0	1	0	1	1	0	1	0
0	0	1	0	1	0	1	1	1	0	0
0	0	1	0	1	0	1	1	1	1	0
0	0	1	0	1	1	0	0	0	0	0
0	0	1	0	1	1	0	0	0	1	0
0	0	1	0	1	1	0	0	1	0	0

0	0	1	0	1	1	0	0	1	1	0
0	0	1	0	1	1	0	1	0	0	0
0	0	1	0	1	1	0	1	0	1	0
0	0	1	0	1	1	0	1	1	0	0
0	0	1	0	1	1	0	1	1	1	0
0	0	1	0	1	1	1	0	0	0	0
0	0	1	0	1	1	1	0	0	1	0
0	0	1	0	1	1	1	0	1	0	0
0	0	1	0	1	1	1	0	1	1	0
0	0	1	0	1	1	1	1	0	0	0
0	0	1	0	1	1	1	1	0	1	0
0	0	1	0	1	1	1	1	1	0	0
0	0	1	0	1	1	1	1	1	1	0
0	0	1	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	0	1	1	0	0	0	1	0	0	0
0	0	1	1	0	0	0	1	0	1	0
0	0	1	1	0	0	0	1	1	1	0
0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	1	1	0	0
0	1	0	0	0	0	0	1	1	1	0
0	1	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	1	0	0	1	0

0	1	0	0	0	0	1	0	1	0	0
0	1	0	0	0	0	1	0	1	1	0
0	1	0	0	0	0	1	1	0	0	0
0	1	0	0	0	0	1	1	0	1	0
0	1	0	0	0	0	1	1	1	0	0
0	1	0	0	0	0	1	1	1	1	0
0	1	0	0	0	1	0	0	0	0	0
0	1	0	0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0	1	0	0
0	1	0	0	0	1	0	0	1	1	0
0	1	0	0	0	1	0	1	0	0	0
0	1	0	0	0	1	0	1	1	0	0
0	1	0	0	0	1	0	1	1	1	0
0	1	0	0	0	1	1	0	0	0	0
0	1	0	0	0	1	1	0	1	0	0
0	1	0	0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1	0	0	0
0	1	0	0	0	1	1	1	1	0	0
0	1	0	0	0	1	1	1	1	1	0
0	1	0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	1	0
0	1	0	0	1	0	0	0	1	0	0
0	1	0	0	1	0	0	0	1	1	0
0	1	0	0	1	0	0	1	0	0	0
0	1	0	0	1	0	0	1	0	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	0	1	0	0	1	1	1	0
0	1	0	0	1	0	1	0	0	0	0

0	1	0	0	1	0	1	0	0	1	0
0	1	0	0	1	0	1	0	1	0	0
0	1	0	0	1	0	1	0	1	1	0
0	1	0	0	1	0	1	1	0	0	0
0	1	0	0	1	0	1	1	0	1	0
0	1	0	0	1	0	1	1	1	0	0
0	1	0	0	1	0	1	1	1	1	0
0	1	0	0	1	1	0	0	0	0	0
0	1	0	0	1	1	0	0	0	1	0
0	1	0	0	1	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0	0
0	1	0	0	1	1	0	1	0	1	0
0	1	0	0	1	1	0	1	1	0	0
0	1	0	0	1	1	0	1	1	1	0
0	1	0	0	1	1	1	0	0	0	0
0	1	0	0	1	1	1	0	0	1	0
0	1	0	0	1	1	1	0	1	0	0
0	1	0	0	1	1	1	1	0	0	0
0	1	0	0	1	1	1	1	0	1	0
0	1	0	0	1	1	1	1	1	1	0
0	1	0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0
0	1	0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	0	0	1	1	0
0	1	0	1	0	0	0	1	0	0	0
0	1	0	1	0	0	0	1	0	1	0
0	1	0	1	0	0	0	1	1	0	0
0	1	0	1	0	0	0	1	1	1	0

0	1	0	1	0	0	1	0	0	0	0
0	1	0	1	0	0	1	0	0	1	0
0	1	0	1	0	0	1	0	1	0	0
0	1	0	1	0	0	1	0	1	1	0
0	1	0	1	0	0	1	1	0	0	0
0	1	0	1	0	0	1	1	0	1	0
0	1	0	1	0	0	1	1	1	0	0
0	1	0	1	0	0	1	1	1	1	0
0	1	0	1	0	1	0	0	0	0	0
0	1	0	1	0	1	0	0	0	1	0
0	1	0	1	0	1	0	0	1	0	0
0	1	0	1	0	1	0	0	1	1	0
0	1	0	1	0	1	0	1	0	0	0
0	1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	1	0	0
0	1	0	1	0	1	0	1	1	1	0
0	1	0	1	0	1	1	0	0	0	0
0	1	0	1	0	1	1	0	0	1	0
0	1	0	1	0	1	1	0	1	0	0
0	1	0	1	0	1	1	0	1	1	0
0	1	0	1	0	1	1	1	0	0	0
0	1	0	1	0	1	1	1	1	1	0
0	1	0	1	1	0	0	0	0	0	0
0	1	0	1	1	0	0	0	0	1	0
0	1	0	1	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0	1	1	0
0	1	0	1	1	0	0	1	0	0	0
0	1	0	1	1	0	0	1	0	1	0
0	1	0	1	1	0	0	1	1	0	0

0	1	0	1	1	0	0	1	1	1	0
0	1	0	1	1	0	1	0	0	0	0
0	1	0	1	1	0	1	0	0	1	0
0	1	0	1	1	0	1	0	1	0	0
0	1	0	1	1	0	1	0	1	1	0
0	1	0	1	1	0	1	1	0	0	0
0	1	0	1	1	0	1	1	0	1	0
0	1	0	1	1	0	1	1	1	0	0
0	1	0	1	1	0	1	1	1	1	0
0	1	0	1	1	1	0	0	0	0	0
0	1	0	1	1	1	0	0	0	1	0
0	1	0	1	1	1	0	0	1	0	0
0	1	0	1	1	1	0	0	1	1	0
0	1	0	1	1	1	0	1	0	0	0
0	1	0	1	1	1	0	1	1	0	0
0	1	0	1	1	1	0	1	1	1	0
0	1	0	1	1	1	1	0	0	0	0
0	1	0	1	1	1	1	0	0	1	0
0	1	0	1	1	1	1	0	1	0	0
0	1	0	1	1	1	1	0	1	1	0
0	1	0	1	1	1	1	1	0	0	0
0	1	0	1	1	1	1	1	1	0	0
0	1	0	1	1	1	1	1	1	1	0
0	1	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	1	1	0
0	1	1	0	0	0	0	1	0	0	0
0	1	1	0	0	0	0	1	0	1	0

0	1	1	0	0	0	0	1	1	0	0
0	1	1	0	0	0	0	1	1	1	0
0	1	1	0	0	0	1	0	0	0	0
0	1	1	0	0	0	1	0	0	1	0
0	1	1	0	0	0	1	0	1	0	0
0	1	1	0	0	0	1	0	1	1	0
0	1	1	0	0	0	1	1	0	0	0
0	1	1	0	0	0	1	1	0	1	0
0	1	1	0	0	0	1	1	1	0	0
0	1	1	0	0	0	1	1	1	1	0
0	1	1	0	0	1	0	0	0	0	0
0	1	1	0	0	1	0	0	0	1	0
0	1	1	0	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0	1	1	0
0	1	1	0	0	1	0	1	0	0	0
0	1	1	0	0	1	0	1	1	0	0
0	1	1	0	0	1	0	1	1	1	0
0	1	1	0	0	1	1	0	0	0	0
0	1	1	0	0	1	1	0	0	1	0
0	1	1	0	0	1	1	0	1	0	0
0	1	1	0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	1	1	1	0
0	1	1	0	1	0	0	0	0	0	0
0	1	1	0	1	0	0	0	0	1	0
0	1	1	0	1	0	0	0	1	0	0
0	1	1	0	1	0	0	0	1	1	0
0	1	1	0	1	0	0	1	0	0	0

0	1	1	0	1	0	0	1	0	1	0
0	1	1	0	1	0	0	1	1	0	0
0	1	1	0	1	0	0	1	1	1	0
0	1	1	0	1	0	1	0	0	0	0
0	1	1	0	1	0	1	0	0	1	0
0	1	1	0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0	1	1	0
0	1	1	0	1	0	1	1	0	0	0
0	1	1	0	1	0	1	1	0	1	0
0	1	1	0	1	0	1	1	1	0	0
0	1	1	0	1	0	1	1	1	1	0
0	1	1	0	1	1	0	0	0	0	0
0	1	1	0	1	1	0	0	0	1	0
0	1	1	0	1	1	0	0	1	0	0
0	1	1	0	1	1	0	0	1	1	0
0	1	1	0	1	1	0	1	1	0	0
0	1	1	0	1	1	0	1	1	1	0
0	1	1	0	1	1	1	0	0	0	0
0	1	1	0	1	1	1	0	0	1	0
0	1	1	0	1	1	1	0	1	1	0
0	1	1	0	1	1	1	1	0	0	0
0	1	1	0	1	1	1	1	0	1	0
0	1	1	0	1	1	1	1	1	0	0
0	1	1	0	1	1	1	1	1	1	0
0	1	1	1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	1	0
0	1	1	1	0	0	0	0	1	0	0
0	1	1	1	0	0	0	0	1	1	0

0	1	1	1	0	0	0	1	0	0	0
0	1	1	1	0	0	0	1	0	1	0
0	1	1	1	0	0	0	1	1	0	0
0	1	1	1	0	0	0	1	1	1	0
1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	1	0
1	0	0	0	0	0	0	1	1	1	0
1	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	1	0	0	1	0
1	0	0	0	0	0	1	0	1	0	0
1	0	0	0	0	0	1	0	1	1	0
1	0	0	0	0	0	1	1	0	0	0
1	0	0	0	0	0	1	1	0	1	0
1	0	0	0	0	0	1	1	1	0	0
1	0	0	0	0	0	1	1	1	1	0
1	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	1	0	0	0	1	0
1	0	0	0	0	1	0	0	1	0	0
1	0	0	0	0	1	0	0	1	1	0
1	0	0	0	0	1	0	1	0	0	0
1	0	0	0	0	1	0	1	1	0	0
1	0	0	0	0	1	0	1	1	1	0
1	0	0	0	0	1	1	0	0	0	0
1	0	0	0	0	1	1	0	0	1	0
1	0	0	0	0	1	1	0	1	0	0

1	0	0	0	0	1	1	0	1	1	0
1	0	0	0	0	1	1	1	0	0	0
1	0	0	0	0	1	1	1	0	1	0
1	0	0	0	0	1	1	1	1	0	0
1	0	0	0	0	1	1	1	1	1	0
1	0	0	0	1	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	1	0
1	0	0	0	1	0	0	0	1	0	0
1	0	0	0	1	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	1	0
1	0	0	0	1	0	0	1	0	0	0
1	0	0	0	1	0	0	1	1	0	0
1	0	0	0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0	0	0	0
1	0	0	0	1	0	1	0	0	1	0
1	0	0	0	1	0	1	0	1	1	0
1	0	0	0	1	0	1	1	0	1	0
1	0	0	0	1	0	1	1	1	0	0
1	0	0	0	1	0	1	1	1	1	0
1	0	0	0	1	1	0	0	0	0	0
1	0	0	0	1	1	0	0	1	0	0
1	0	0	0	1	1	0	0	1	1	0
1	0	0	0	1	1	0	1	0	0	0
1	0	0	0	1	1	0	1	1	0	0
1	0	0	0	1	1	0	1	1	0	0
1	0	0	0	1	1	0	1	1	1	0
1	0	0	0	1	1	1	0	0	0	0
1	0	0	0	1	1	1	0	0	1	0

1	0	0	0	1	1	1	0	1	0	0
1	0	0	0	1	1	1	0	1	1	0
1	0	0	0	1	1	1	1	0	0	0
1	0	0	0	1	1	1	1	0	1	0
1	0	0	0	1	1	1	1	1	0	0
1	0	0	0	1	1	1	1	1	1	0
1	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	1	0	0
1	0	0	1	0	0	0	0	1	1	0
1	0	0	1	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1	1	0	0
1	0	0	1	0	0	0	1	1	1	0
1	0	0	1	0	0	1	0	0	0	0
1	0	0	1	0	0	1	0	0	1	0
1	0	0	1	0	0	1	0	1	0	0
1	0	0	1	0	0	1	1	0	0	0
1	0	0	1	0	0	1	1	0	1	0
1	0	0	1	0	0	1	1	1	0	0
1	0	0	1	0	0	1	1	1	1	0
1	0	0	1	0	1	0	0	0	0	0
1	0	0	1	0	1	0	0	0	1	0
1	0	0	1	0	1	0	0	1	0	0
1	0	0	1	0	1	0	0	1	1	0
1	0	0	1	0	1	0	1	0	0	0
1	0	0	1	0	1	0	1	0	1	0
1	0	0	1	0	1	0	1	1	0	0
1	0	0	1	0	1	0	1	1	1	0
1	0	0	1	0	1	1	0	0	0	0

1	0	0	1	0	1	1	0	0	1	0
1	0	0	1	0	1	1	0	1	0	0
1	0	0	1	0	1	1	0	1	1	0
1	0	0	1	0	1	1	1	0	0	0
1	0	0	1	0	1	1	1	0	1	0
1	0	0	1	0	1	1	1	1	0	0
1	0	0	1	0	1	1	1	1	1	0
1	0	0	1	1	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	1	0
1	0	0	1	1	0	0	0	1	0	0
1	0	0	1	1	0	0	0	1	1	0
1	0	0	1	1	0	0	1	0	0	0
1	0	0	1	1	0	0	1	0	1	0
1	0	0	1	1	0	0	1	1	0	0
1	0	0	1	1	0	1	0	0	0	0
1	0	0	1	1	0	1	0	0	1	0
1	0	0	1	1	0	1	0	1	1	0
1	0	0	1	1	0	1	1	0	0	0
1	0	0	1	1	0	1	1	0	1	0
1	0	0	1	1	0	1	1	1	1	0
1	0	0	1	1	1	0	0	0	0	0
1	0	0	1	1	1	0	0	0	1	0
1	0	0	1	1	1	0	0	1	0	0
1	0	0	1	1	1	0	0	1	1	0
1	0	0	1	1	1	0	1	0	0	0
1	0	0	1	1	1	0	1	0	1	0
1	0	0	1	1	1	0	1	1	0	0
1	0	0	1	1	1	0	1	1	1	0

1	0	0	1	1	1	1	0	0	0	0
1	0	0	1	1	1	1	0	0	1	0
1	0	0	1	1	1	1	0	1	0	0
1	0	0	1	1	1	1	0	1	1	0
1	0	0	1	1	1	1	1	0	0	0
1	0	0	1	1	1	1	1	0	1	0
1	0	0	1	1	1	1	1	1	0	0
1	0	0	1	1	1	1	1	1	1	0
1	0	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0	1	0	0
1	0	1	0	0	0	0	0	1	1	0
1	0	1	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	1	0	1	0
1	0	1	0	0	0	0	1	1	0	0
1	0	1	0	0	0	1	0	0	0	0
1	0	1	0	0	0	1	0	0	1	0
1	0	1	0	0	0	1	0	1	0	0
1	0	1	0	0	0	1	1	0	0	0
1	0	1	0	0	0	1	1	1	1	0
1	0	1	0	0	1	0	0	0	0	0
1	0	1	0	0	1	0	0	0	1	0
1	0	1	0	0	1	0	0	1	0	0
1	0	1	0	0	1	0	0	1	1	0
1	0	1	0	0	1	0	1	0	0	0
1	0	1	0	0	1	0	1	0	1	0
1	0	1	0	0	1	0	1	1	0	0

1	0	1	0	0	1	0	1	1	1	0
1	0	1	0	0	1	1	0	0	0	0
1	0	1	0	0	1	1	0	0	1	0
1	0	1	0	0	1	1	0	1	0	0
1	0	1	0	0	1	1	0	1	1	0
1	0	1	0	0	1	1	1	0	0	0
1	0	1	0	0	1	1	1	0	1	0
1	0	1	0	0	1	1	1	1	0	0
1	0	1	0	0	1	1	1	1	1	0
1	0	1	0	1	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	1	0
1	0	1	0	1	0	0	0	1	0	0
1	0	1	0	1	0	0	0	1	1	0
1	0	1	0	1	0	0	1	0	0	0
1	0	1	0	1	0	0	1	0	1	0
1	0	1	0	1	0	0	1	1	0	0
1	0	1	0	1	0	0	1	1	1	0
1	0	1	0	1	0	1	0	0	0	0
1	0	1	0	1	0	1	0	0	1	0
1	0	1	0	1	0	1	0	1	0	0
1	0	1	0	1	0	1	0	1	1	0
1	0	1	0	1	0	1	1	1	0	0
1	0	1	0	1	0	1	1	1	1	0
1	0	1	0	1	1	0	0	0	0	0
1	0	1	0	1	1	0	0	0	1	0
1	0	1	0	1	1	0	0	1	0	0
1	0	1	0	1	1	0	0	1	1	0
1	0	1	0	1	1	0	1	0	0	0
1	0	1	0	1	1	0	1	0	1	0

1	0	1	0	1	1	0	1	1	0	0
1	0	1	0	1	1	0	1	1	1	0
1	0	1	0	1	1	1	0	0	0	0
1	0	1	0	1	1	1	0	0	1	0
1	0	1	0	1	1	1	0	1	0	0
1	0	1	0	1	1	1	0	1	1	0
1	0	1	0	1	1	1	1	0	0	0
1	0	1	0	1	1	1	1	0	1	0
1	0	1	0	1	1	1	1	1	0	0
1	0	1	0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	1	0
1	0	1	1	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	1	0
1	0	1	1	0	0	0	1	0	0	0
1	0	1	1	0	0	0	1	1	0	0
1	0	1	1	0	0	0	1	1	1	0
1	1	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	0	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	1	1	0
1	1	0	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0
1	1	0	0	0	0	1	0	0	1	0
1	1	0	0	0	0	1	0	1	0	0
1	1	0	0	0	0	1	0	1	1	0
1	1	0	0	0	0	1	1	0	0	0

1	1	0	0	0	0	1	1	0	1	0
1	1	0	0	0	0	1	1	1	0	0
1	1	0	0	0	0	1	1	1	1	0
1	1	0	0	0	1	0	0	0	0	0
1	1	0	0	0	1	0	0	0	1	0
1	1	0	0	0	1	0	0	1	0	0
1	1	0	0	0	1	0	0	1	1	0
1	1	0	0	0	1	0	1	0	0	0
1	1	0	0	0	1	0	1	0	1	0
1	1	0	0	0	1	0	1	1	0	0
1	1	0	0	0	1	1	0	0	0	0
1	1	0	0	0	1	1	0	0	1	0
1	1	0	0	0	1	1	0	1	0	0
1	1	0	0	0	1	1	0	1	1	0
1	1	0	0	0	1	1	1	1	1	0
1	1	0	0	0	1	1	1	1	1	0
1	1	0	0	0	1	1	1	1	1	0
1	1	0	0	1	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	1	0
1	1	0	0	1	0	0	0	1	0	0
1	1	0	0	1	0	0	0	1	1	0
1	1	0	0	1	0	0	1	0	0	0
1	1	0	0	1	0	0	1	0	1	0
1	1	0	0	1	0	0	1	1	0	0
1	1	0	0	1	0	0	1	1	1	0
1	1	0	0	1	0	1	0	0	0	0
1	1	0	0	1	0	1	0	0	1	0
1	1	0	0	1	0	1	0	1	0	0
1	1	0	0	1	0	1	0	1	1	0

1	1	0	0	1	0	1	1	0	0	0
1	1	0	0	1	0	1	1	0	1	0
1	1	0	0	1	0	1	1	1	0	0
1	1	0	0	1	0	1	1	1	1	0
1	1	0	0	1	1	0	0	0	0	0
1	1	0	0	1	1	0	0	0	1	0
1	1	0	0	1	1	0	0	1	0	0
1	1	0	0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	1	0	0	0
1	1	0	0	1	1	0	1	0	1	0
1	1	0	0	1	1	0	1	1	1	0
1	1	0	0	1	1	1	0	0	0	0
1	1	0	0	1	1	1	0	0	1	0
1	1	0	0	1	1	1	0	1	0	0
1	1	0	0	1	1	1	1	0	0	0
1	1	0	0	1	1	1	1	0	1	0
1	1	0	0	1	1	1	1	1	0	0
1	1	0	0	1	1	1	1	1	1	0
1	1	0	0	1	1	1	1	1	1	0
1	1	0	1	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	0
1	1	0	1	0	0	0	0	1	0	0
1	1	0	1	0	0	0	0	1	1	0
1	1	0	1	0	0	0	1	0	0	0
1	1	0	1	0	0	0	1	1	0	0
1	1	0	1	0	0	0	1	1	1	0
1	1	0	1	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	1	0
1	1	0	1	0	0	1	0	1	0	0

1	1	0	1	1	0	1	0	1	0	0
1	1	0	1	1	0	1	0	1	1	0
1	1	0	1	1	0	1	1	0	0	0
1	1	0	1	1	0	1	1	0	1	0
1	1	0	1	1	0	1	1	1	0	0
1	1	0	1	1	0	1	1	1	1	0
1	1	0	1	1	1	0	0	0	0	0
1	1	0	1	1	1	0	0	0	1	0
1	1	0	1	1	1	0	0	1	0	0
1	1	0	1	1	1	0	0	1	1	0
1	1	0	1	1	1	0	1	0	0	0
1	1	0	1	1	1	0	1	0	1	0
1	1	0	1	1	1	0	1	1	0	0
1	1	0	1	1	1	0	1	1	1	0
1	1	0	1	1	1	1	0	0	0	0
1	1	0	1	1	1	1	0	0	1	0
1	1	0	1	1	1	1	0	1	0	0
1	1	0	1	1	1	1	1	0	1	0
1	1	0	1	1	1	1	1	0	0	0
1	1	0	1	1	1	1	1	0	1	0
1	1	0	1	1	1	1	1	1	0	0
1	1	0	1	1	1	1	1	1	1	0
1	1	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	1	1	0
1	1	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	1	1	0
1	1	1	0	0	0	0	1	0	0	0
1	1	1	0	0	0	0	1	0	1	0
1	1	1	0	0	0	0	1	1	0	0
1	1	1	0	0	0	0	1	1	1	0
1	1	1	0	0	0	1	0	0	0	0

1	1	1	0	0	0	1	0	0	1	0
1	1	1	0	0	0	1	0	1	0	0
1	1	1	0	0	0	1	0	1	1	0
1	1	1	0	0	0	1	1	0	0	0
1	1	1	0	0	0	1	1	0	1	0
1	1	1	0	0	0	1	1	1	0	0
1	1	1	0	0	0	1	1	1	1	0
1	1	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	1	0
1	1	1	0	0	1	0	0	1	0	0
1	1	1	0	0	1	0	0	1	1	0
1	1	1	0	0	1	0	1	0	0	0
1	1	1	0	0	1	0	1	0	1	0
1	1	1	0	0	1	0	1	1	0	0
1	1	1	0	0	1	1	0	0	0	0
1	1	1	0	0	1	1	0	0	1	0
1	1	1	0	0	1	1	0	1	0	0
1	1	1	0	0	1	1	1	0	1	0
1	1	1	0	0	1	1	1	1	0	0
1	1	1	0	0	1	1	1	1	1	0
1	1	1	0	1	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	1	0
1	1	1	0	1	0	0	0	1	0	0
1	1	1	0	1	0	0	0	1	1	0
1	1	1	0	1	0	0	1	0	0	0
1	1	1	0	1	0	0	1	0	1	0
1	1	1	0	1	0	0	1	1	0	0
1	1	1	0	1	0	0	1	1	1	0

1	1	1	0	1	0	1	0	0	0	0
1	1	1	0	1	0	1	0	0	1	0
1	1	1	0	1	0	1	0	1	0	0
1	1	1	0	1	0	1	0	1	1	0
1	1	1	0	1	0	1	1	0	0	0
1	1	1	0	1	0	1	1	0	1	0
1	1	1	0	1	0	1	1	1	0	0
1	1	1	0	1	0	1	1	1	1	0
1	1	1	0	1	1	0	0	0	0	0
1	1	1	0	1	1	0	0	0	1	0
1	1	1	0	1	1	0	0	1	0	0
1	1	1	0	1	1	0	0	1	1	0
1	1	1	0	1	1	0	1	0	0	0
1	1	1	0	1	1	0	1	0	1	0
1	1	1	0	1	1	0	1	1	0	0
1	1	1	0	1	1	1	0	0	0	0
1	1	1	0	1	1	1	0	0	1	0
1	1	1	0	1	1	1	1	0	0	0
1	1	1	0	1	1	1	1	0	0	0
1	1	1	0	1	1	1	1	1	0	0
1	1	1	0	1	1	1	1	1	1	0
1	1	1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	1	0	0
1	1	1	1	0	0	0	0	1	1	0
1	1	1	1	0	0	0	1	0	0	0
1	1	1	1	0	0	0	1	0	1	0
1	1	1	1	0	0	0	1	1	0	0

1	1	1	1	0	0	0	1	1	1	0
0	1	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	1	1
0	1	0	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	1	0	0	1
0	1	0	0	0	0	0	1	0	1	1
0	1	0	0	0	0	0	1	1	0	1
0	1	0	0	0	0	0	1	1	1	1
0	1	0	0	0	0	1	0	0	0	1
0	1	0	0	0	0	1	0	0	1	1
0	1	0	0	0	0	1	0	1	1	1
0	1	0	0	0	0	1	1	0	0	1
0	1	0	0	0	0	1	1	1	1	1
0	1	0	0	0	1	0	0	0	0	1
0	1	0	0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	0	1	0	1
0	1	0	0	0	1	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	1
0	1	0	0	0	1	0	1	1	1	1
0	1	0	0	0	1	1	0	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	0	0	1	1	0	1	0	1
0	1	0	0	0	1	1	0	1	1	1
0	1	0	0	0	1	1	1	0	0	1
0	1	0	0	0	1	1	1	0	1	1

0	1	0	0	0	1	1	1	1	0	1
0	1	0	0	0	1	1	1	1	1	1
0	1	0	0	1	0	0	0	0	0	1
0	1	0	0	1	0	0	0	0	1	1
0	1	0	0	1	0	0	0	1	0	1
0	1	0	0	1	0	0	0	1	1	1
0	1	0	0	1	0	0	1	0	0	1
0	1	0	0	1	0	0	1	0	1	1
0	1	0	0	1	0	0	1	1	0	1
0	1	0	0	1	0	0	1	1	1	1
0	1	0	0	1	0	1	0	0	0	1
0	1	0	0	1	0	1	0	1	0	1
0	1	0	0	1	0	1	0	1	1	1
0	1	0	0	1	0	1	1	0	0	1
0	1	0	0	1	0	1	1	1	0	1
0	1	0	0	1	0	1	1	1	1	1
0	1	0	0	1	1	0	0	0	0	1
0	1	0	0	1	1	0	0	1	0	1
0	1	0	0	1	1	0	0	1	1	1
0	1	0	0	1	1	0	1	0	0	1
0	1	0	0	1	1	0	1	0	1	1
0	1	0	0	1	1	0	1	1	0	1
0	1	0	0	1	1	0	1	1	1	1
0	1	0	0	1	1	1	0	0	0	1
0	1	0	0	1	1	1	0	0	1	1
0	1	0	0	1	1	1	0	1	0	1
0	1	0	0	1	1	1	0	1	1	1
0	1	0	0	1	1	1	1	0	0	1

0	1	0	0	1	1	1	1	0	1	1
0	1	0	0	1	1	1	1	1	0	1
0	1	0	0	1	1	1	1	1	1	1
0	1	0	1	0	0	0	0	0	0	1
0	1	0	1	0	0	0	0	0	1	1
0	1	0	1	0	0	0	0	1	0	1
0	1	0	1	0	0	0	0	1	1	1
0	1	0	1	0	0	0	1	0	0	1
0	1	0	1	0	0	0	1	0	1	1
0	1	0	1	0	0	0	1	1	0	1
0	1	0	1	0	0	1	0	0	0	1
0	1	0	1	0	0	1	0	0	1	1
0	1	0	1	0	0	1	0	1	0	1
0	1	0	1	0	0	1	0	1	1	1
0	1	0	1	0	0	1	1	0	0	1
0	1	0	1	0	0	1	1	1	0	1
0	1	0	1	0	0	1	1	1	1	1
0	1	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	0	0	0	1	1
0	1	0	1	0	1	0	0	1	0	1
0	1	0	1	0	1	0	0	1	1	1
0	1	0	1	0	1	0	1	0	0	1
0	1	0	1	0	1	0	1	0	1	1
0	1	0	1	0	1	0	1	1	0	1
0	1	0	1	0	1	0	1	1	1	1
0	1	0	1	0	1	1	0	0	0	1
0	1	0	1	0	1	1	0	0	1	1
0	1	0	1	0	1	1	0	1	0	1
0	1	0	1	0	1	1	0	1	1	1

0	1	0	1	0	1	1	1	0	0	1
0	1	0	1	0	1	1	1	0	1	1
0	1	0	1	0	1	1	1	1	0	1
0	1	0	1	0	1	1	1	1	1	1
0	1	0	1	1	0	0	0	0	0	1
0	1	0	1	1	0	0	0	0	1	1
0	1	0	1	1	0	0	0	1	0	1
0	1	0	1	1	0	0	0	1	1	1
0	1	0	1	1	0	0	1	0	0	1
0	1	0	1	1	0	0	1	0	1	1
0	1	0	1	1	0	0	1	1	1	1
0	1	0	1	1	0	1	0	0	0	1
0	1	0	1	1	0	1	0	0	1	1
0	1	0	1	1	0	1	0	1	0	1
0	1	0	1	1	0	1	0	1	1	1
0	1	0	1	1	0	1	1	0	0	1
0	1	0	1	1	0	1	1	0	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	0	1	1	1	0	0	0	0	1
0	1	0	1	1	1	0	0	0	1	1
0	1	0	1	1	1	0	0	1	0	1
0	1	0	1	1	1	0	0	1	1	1
0	1	0	1	1	1	0	1	0	0	1
0	1	0	1	1	1	0	1	0	1	1
0	1	0	1	1	1	0	1	1	0	1
0	1	0	1	1	1	0	1	1	1	1
0	1	0	1	1	1	1	0	0	0	1
0	1	0	1	1	1	1	0	0	1	1
0	1	0	1	1	1	1	0	1	0	1

0	1	0	1	1	1	1	0	1	1	1
0	1	0	1	1	1	1	1	0	0	1
0	1	0	1	1	1	1	1	0	1	1
0	1	0	1	1	1	1	1	1	0	1
0	1	0	1	1	1	1	1	1	1	1
0	1	1	0	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0	0	1	1
0	1	1	0	0	0	0	0	1	0	1
0	1	1	0	0	0	0	0	1	1	1
0	1	1	0	0	0	0	1	0	0	1
0	1	1	0	0	0	0	1	0	1	1
0	1	1	0	0	0	0	1	1	1	1
0	1	1	0	0	0	1	0	0	0	1
0	1	1	0	0	0	1	0	0	1	1
0	1	1	0	0	0	1	1	0	1	1
0	1	1	0	0	0	1	1	1	1	1
0	1	1	0	0	1	0	0	0	0	1
0	1	1	0	0	1	0	0	1	0	1
0	1	1	0	0	1	0	0	1	1	1
0	1	1	0	0	1	0	1	0	0	1
0	1	1	0	0	1	0	1	0	1	1
0	1	1	0	0	1	0	1	1	0	1
0	1	1	0	0	1	0	1	1	1	1
0	1	1	0	0	1	1	0	0	0	1
0	1	1	0	0	1	1	0	0	1	1

0	1	1	0	0	1	1	0	1	0	1
0	1	1	0	0	1	1	0	1	1	1
0	1	1	0	0	1	1	1	0	0	1
0	1	1	0	0	1	1	1	0	1	1
0	1	1	0	0	1	1	1	1	0	1
0	1	1	0	0	1	1	1	1	1	1
0	1	1	0	1	0	0	0	0	0	1
0	1	1	0	1	0	0	0	0	1	1
0	1	1	0	1	0	0	0	1	0	1
0	1	1	0	1	0	0	0	1	1	1
0	1	1	0	1	0	0	1	0	1	1
0	1	1	0	1	0	0	1	1	0	1
0	1	1	0	1	0	0	1	1	1	1
0	1	1	0	1	0	1	0	0	0	1
0	1	1	0	1	0	1	0	1	0	1
0	1	1	0	1	0	1	0	1	1	1
0	1	1	0	1	0	1	1	0	0	1
0	1	1	0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	0	1	1	0	0	0	0	1
0	1	1	0	1	1	0	0	0	1	1
0	1	1	0	1	1	0	0	1	0	1
0	1	1	0	1	1	0	0	1	1	1
0	1	1	0	1	1	0	1	0	0	1
0	1	1	0	1	1	0	1	0	1	1
0	1	1	0	1	1	0	1	1	0	1
0	1	1	0	1	1	0	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1

0	1	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	1	0	1
0	1	1	0	1	1	1	0	1	1	1
0	1	1	0	1	1	1	1	0	0	1
0	1	1	0	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	1	0	1
0	1	1	0	1	1	1	1	1	1	1
0	1	1	1	0	0	0	0	0	0	1
0	1	1	1	0	0	0	0	0	1	1
0	1	1	1	0	0	0	0	1	0	1
0	1	1	1	0	0	0	0	1	1	1
0	1	1	1	0	0	0	1	0	0	1
0	1	1	1	0	0	0	1	0	1	1
0	1	1	1	0	0	0	1	1	0	1
0	1	1	1	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	1	0	1
1	1	0	0	0	0	0	0	1	1	1
1	1	0	0	0	0	0	1	0	0	1
1	1	0	0	0	0	0	1	0	1	1
1	1	0	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	1	1	1	1
1	1	0	0	0	0	1	0	0	0	1
1	1	0	0	0	0	1	0	0	1	1
1	1	0	0	0	0	1	0	1	0	1
1	1	0	0	0	0	1	0	1	1	1
1	1	0	0	0	0	1	1	0	0	1
1	1	0	0	0	0	1	1	0	1	1
1	1	0	0	0	0	1	1	1	0	1
1	1	0	0	0	0	1	1	1	1	1

1	1	0	0	0	1	0	0	0	0	1
1	1	0	0	0	1	0	0	0	1	1
1	1	0	0	0	1	0	0	1	0	1
1	1	0	0	0	1	0	0	1	1	1
1	1	0	0	0	1	0	1	0	0	1
1	1	0	0	0	1	0	1	0	1	1
1	1	0	0	0	1	0	1	1	0	1
1	1	0	0	0	1	0	1	1	1	1
1	1	0	0	0	1	1	0	0	0	1
1	1	0	0	0	1	1	0	0	1	1
1	1	0	0	0	1	1	0	1	1	1
1	1	0	0	0	1	1	1	0	0	1
1	1	0	0	0	1	1	1	0	1	1
1	1	0	0	0	1	1	1	1	0	1
1	1	0	0	0	1	1	1	1	1	1
1	1	0	0	1	0	0	0	0	0	1
1	1	0	0	1	0	0	0	0	1	1
1	1	0	0	1	0	0	0	1	0	1
1	1	0	0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	1	0	0	1
1	1	0	0	1	0	0	1	0	1	1
1	1	0	0	1	0	0	1	1	0	1
1	1	0	0	1	0	0	1	1	1	1
1	1	0	0	1	0	1	0	0	0	1
1	1	0	0	1	0	1	0	0	1	1
1	1	0	0	1	0	1	0	1	0	1
1	1	0	0	1	0	1	0	1	1	1
1	1	0	0	1	0	1	1	0	0	1
1	1	0	0	1	0	1	1	0	1	1
1	1	0	0	1	0	1	1	1	0	1

1	1	0	0	1	0	1	1	1	1	1
1	1	0	0	1	1	0	0	0	0	1
1	1	0	0	1	1	0	0	0	1	1
1	1	0	0	1	1	0	0	1	0	1
1	1	0	0	1	1	0	0	1	1	1
1	1	0	0	1	1	0	1	0	0	1
1	1	0	0	1	1	0	1	0	1	1
1	1	0	0	1	1	0	1	1	0	1
1	1	0	0	1	1	0	1	1	1	1
1	1	0	0	1	1	1	0	0	0	1
1	1	0	0	1	1	1	0	0	1	1
1	1	0	0	1	1	1	0	1	0	1
1	1	0	0	1	1	1	0	1	1	1
1	1	0	0	1	1	1	1	0	0	1
1	1	0	0	1	1	1	1	0	1	1
1	1	0	0	1	1	1	1	1	0	1
1	1	0	0	1	1	1	1	1	1	1
1	1	0	1	0	0	0	0	0	0	1
1	1	0	1	0	0	0	0	0	1	1
1	1	0	1	0	0	0	0	1	0	1
1	1	0	1	0	0	0	0	1	1	1
1	1	0	1	0	0	0	0	1	1	1
1	1	0	1	0	0	0	1	0	0	1
1	1	0	1	0	0	0	1	1	0	1
1	1	0	1	0	0	0	1	1	1	1
1	1	0	1	0	0	0	1	1	0	1
1	1	0	1	0	0	0	1	0	0	1
1	1	0	1	0	0	1	0	0	1	1
1	1	0	1	0	0	1	0	1	0	1
1	1	0	1	0	0	1	0	1	1	1
1	1	0	1	0	0	1	1	0	0	1
1	1	0	1	0	0	1	1	0	1	1

1	1	0	1	0	0	1	1	1	0	1
1	1	0	1	0	0	1	1	1	1	1
1	1	0	1	0	1	0	0	0	0	1
1	1	0	1	0	1	0	0	0	1	1
1	1	0	1	0	1	0	0	1	0	1
1	1	0	1	0	1	0	0	1	1	1
1	1	0	1	0	1	0	1	0	0	1
1	1	0	1	0	1	0	1	0	1	1
1	1	0	1	0	1	0	1	1	0	1
1	1	0	1	0	1	0	1	1	1	1
1	1	0	1	0	1	1	0	0	0	1
1	1	0	1	0	1	1	0	1	0	1
1	1	0	1	0	1	1	0	1	1	1
1	1	0	1	0	1	1	1	0	0	1
1	1	0	1	0	1	1	0	1	1	1
1	1	0	1	0	1	1	1	0	0	1
1	1	0	1	0	1	1	1	1	1	1
1	1	0	1	1	0	0	0	0	0	1
1	1	0	1	1	0	0	0	0	1	1
1	1	0	1	1	0	0	0	1	0	1
1	1	0	1	1	0	0	0	1	1	1
1	1	0	1	1	0	0	1	0	0	1
1	1	0	1	1	0	0	1	0	1	1
1	1	0	1	1	0	0	1	1	0	1
1	1	0	1	1	0	0	1	1	1	1
1	1	0	1	1	0	1	0	0	0	1
1	1	0	1	1	0	1	0	0	1	1
1	1	0	1	1	0	1	0	1	0	1
1	1	0	1	1	0	1	0	1	1	1
1	1	0	1	1	0	1	1	0	0	1

1	1	0	1	1	0	1	1	0	1	1
1	1	0	1	1	0	1	1	1	0	1
1	1	0	1	1	0	1	1	1	1	1
1	1	0	1	1	1	0	0	0	0	1
1	1	0	1	1	1	0	0	0	1	1
1	1	0	1	1	1	0	0	1	0	1
1	1	0	1	1	1	0	0	1	1	1
1	1	0	1	1	1	0	1	0	0	1
1	1	0	1	1	1	0	1	0	1	1
1	1	0	1	1	1	0	1	1	0	1
1	1	0	1	1	1	0	1	1	1	1
1	1	0	1	1	1	1	0	0	0	1
1	1	0	1	1	1	1	0	0	1	1
1	1	0	1	1	1	1	0	1	0	1
1	1	0	1	1	1	1	0	1	1	1
1	1	0	1	1	1	1	1	0	0	1
1	1	0	1	1	1	1	1	1	0	1
1	1	0	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	1	1
1	1	1	0	0	0	0	0	1	0	1
1	1	1	0	0	0	0	0	1	1	1
1	1	1	0	0	0	0	1	0	0	1
1	1	1	0	0	0	1	0	0	0	1
1	1	1	0	0	0	1	0	0	1	1
1	1	1	0	0	0	1	0	1	0	1
1	1	1	0	0	0	1	0	1	1	1

1	1	1	0	0	0	1	1	0	0	1
1	1	1	0	0	0	1	1	0	1	1
1	1	1	0	0	0	1	1	1	0	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	0	0	1	0	0	0	0	1
1	1	1	0	0	1	0	0	0	1	1
1	1	1	0	0	1	0	0	1	0	1
1	1	1	0	0	1	0	0	1	1	1
1	1	1	0	0	1	0	1	0	0	1
1	1	1	0	0	1	0	1	0	1	1
1	1	1	0	0	1	0	1	1	1	1
1	1	1	0	0	1	1	0	0	0	1
1	1	1	0	0	1	1	0	0	1	1
1	1	1	0	0	1	1	0	1	0	1
1	1	1	0	0	1	1	0	1	1	1
1	1	1	0	0	1	1	1	0	0	1
1	1	1	0	0	1	1	1	1	0	1
1	1	1	0	0	1	1	1	1	1	1
1	1	1	0	1	0	0	0	0	0	1
1	1	1	0	1	0	0	0	0	1	1
1	1	1	0	1	0	0	0	1	0	1
1	1	1	0	1	0	0	0	1	1	1
1	1	1	0	1	0	0	1	0	0	1
1	1	1	0	1	0	0	1	1	0	1
1	1	1	0	1	0	0	1	1	1	1
1	1	1	0	1	0	1	0	0	0	1
1	1	1	0	1	0	1	0	0	1	1
1	1	1	0	1	0	1	0	1	0	1

1	1	1	0	1	0	1	0	1	1	1
1	1	1	0	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1	0	1	1
1	1	1	0	1	0	1	1	1	0	1
1	1	1	0	1	0	1	1	1	1	1
1	1	1	0	1	1	0	0	0	0	1
1	1	1	0	1	1	0	0	0	1	1
1	1	1	0	1	1	0	0	1	0	1
1	1	1	0	1	1	0	0	1	1	1
1	1	1	0	1	1	0	1	0	0	1
1	1	1	0	1	1	0	1	1	0	1
1	1	1	0	1	1	0	1	1	1	1
1	1	1	0	1	1	1	0	0	0	1
1	1	1	0	1	1	1	0	0	1	1
1	1	1	0	1	1	1	0	1	1	1
1	1	1	0	1	1	1	1	0	0	1
1	1	1	0	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	0	1
1	1	1	0	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0	0	0	1
1	1	1	1	0	0	0	0	0	1	1
1	1	1	1	0	0	0	0	1	1	1
1	1	1	1	0	0	0	1	0	0	1
1	1	1	1	0	0	0	1	0	1	1
1	1	1	1	0	0	0	1	1	0	1
1	1	1	1	0	0	0	1	1	1	1

1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1

1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0

1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	1	1
0	0	0	1	1

1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	1	0	1	1
1	1	0	1	1

1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
1	1	0	1	1
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0

1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	1	1	0	1
1	1	1	0	1
1	1	1	0	1
1	1	1	0	1
1	1	1	0	1

Anexo 6 Diagrama de Flujo del Algoritmo de Retropropagación

Este anexo ha sido desarrollado en su totalidad por los autores de esta tesis mediante la aplicación web code2flow.

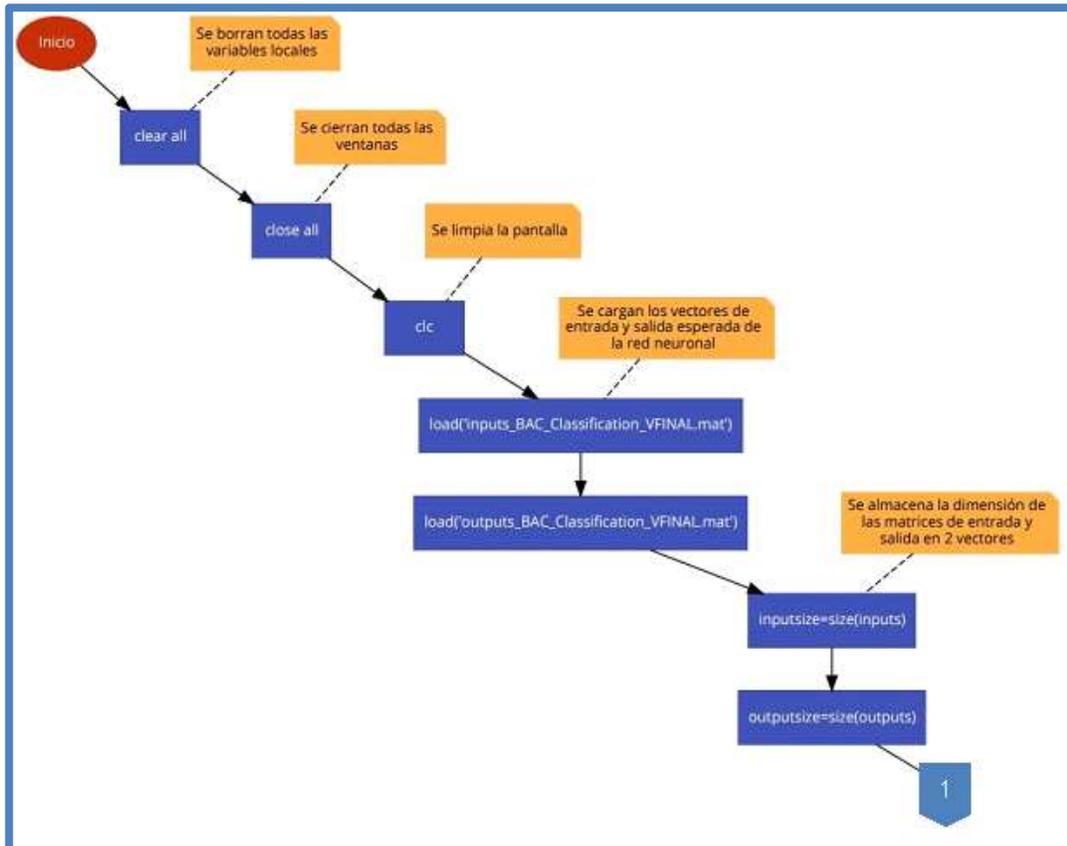


Fig. 138 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Principal – Parte 1

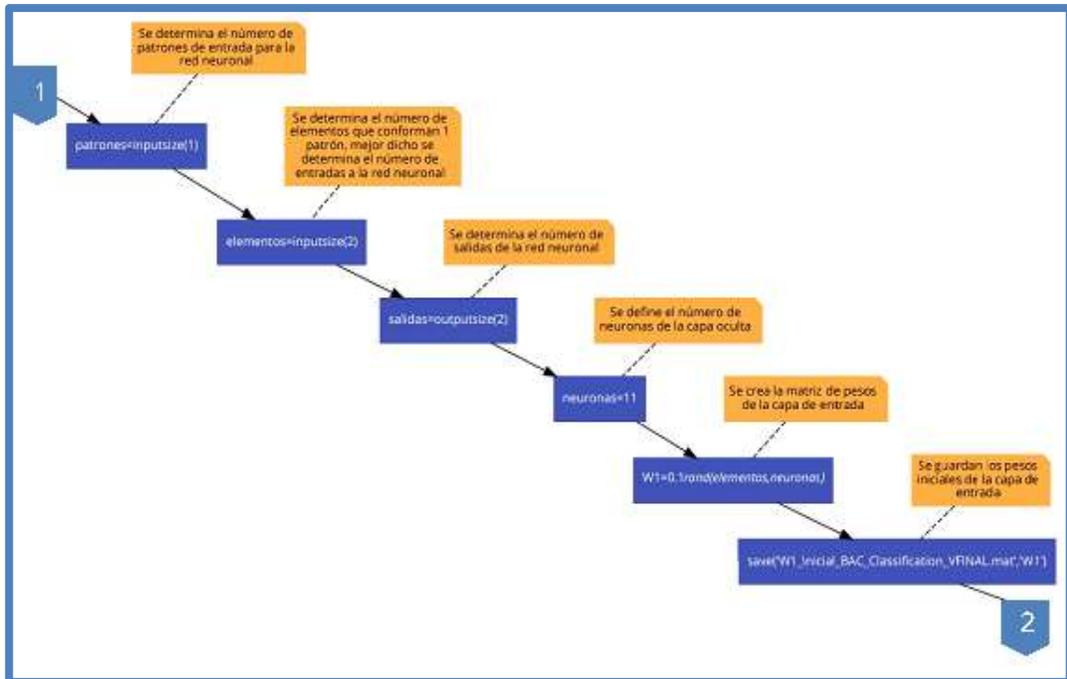


Fig. 139 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Principal – Parte 2

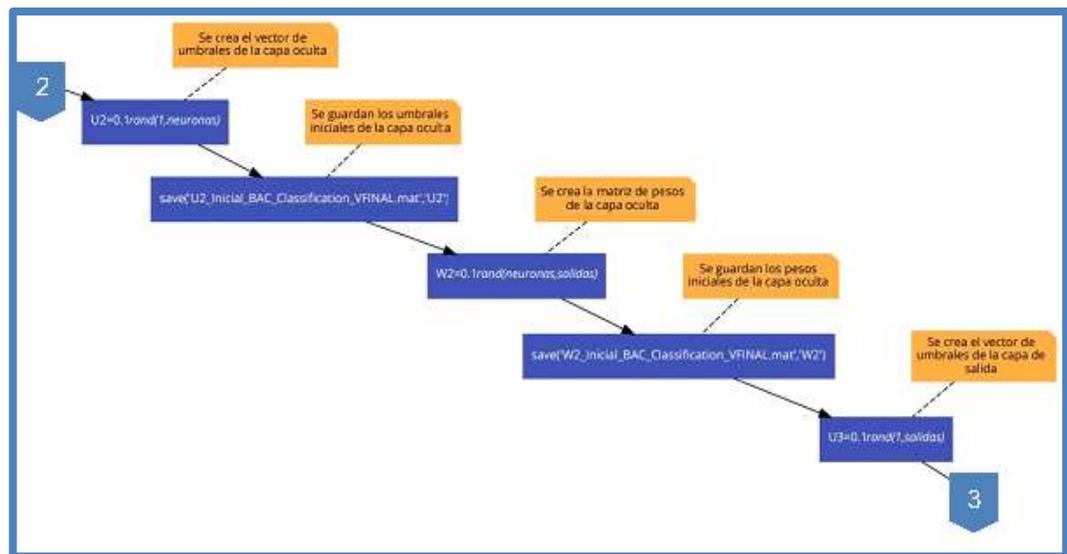


Fig. 140 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Principal – Parte 3

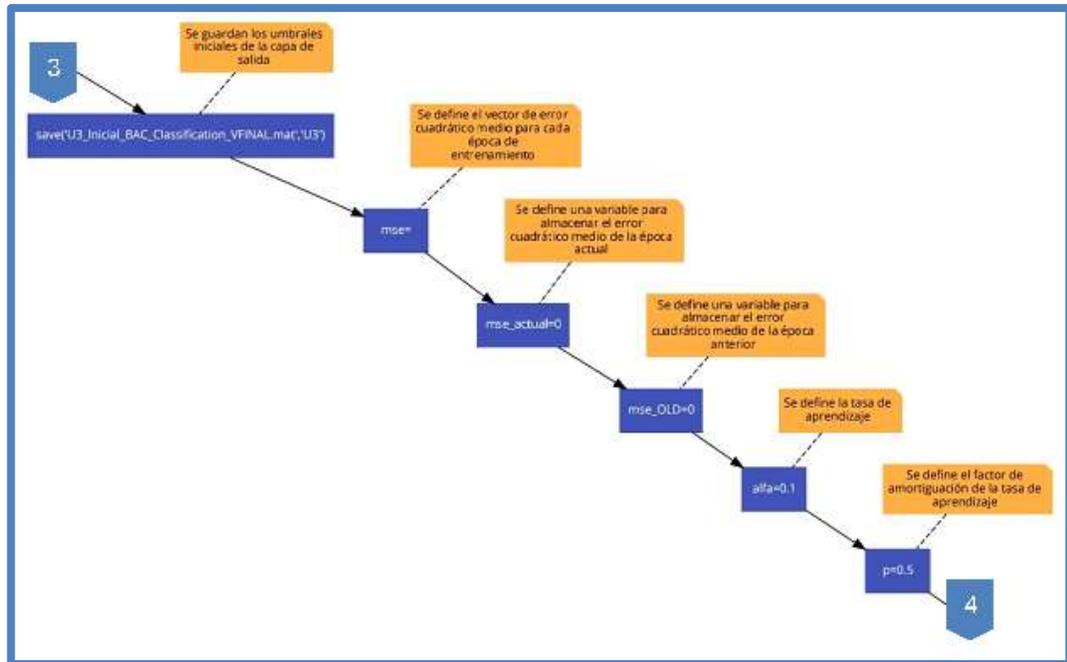


Fig. 141 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Principal – Parte 4

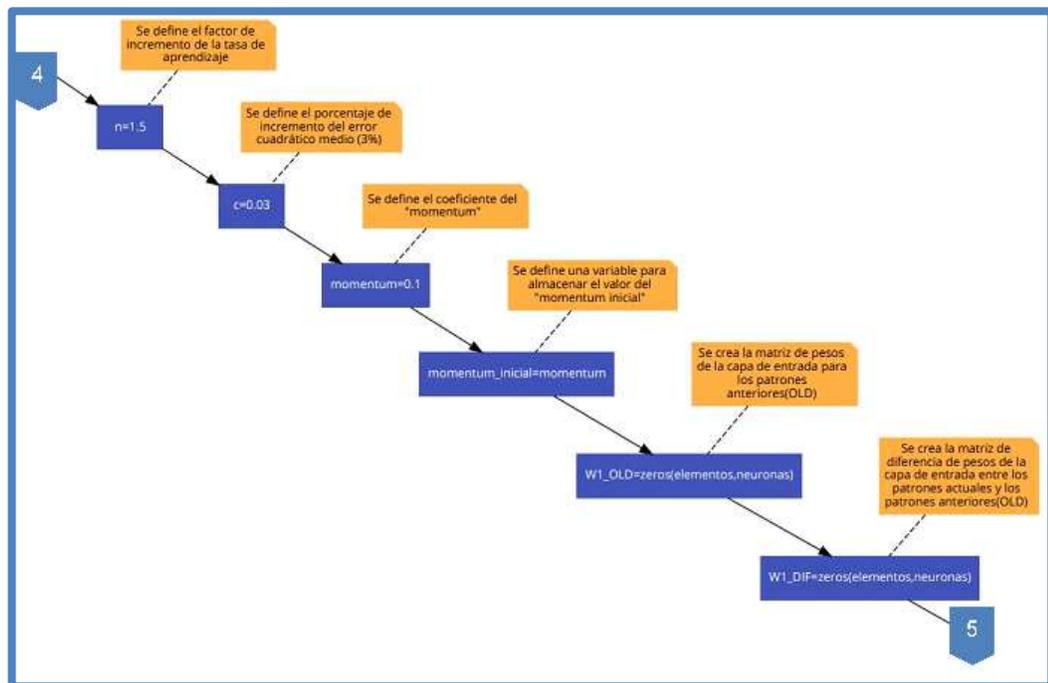


Fig. 142 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Principal – Parte 5

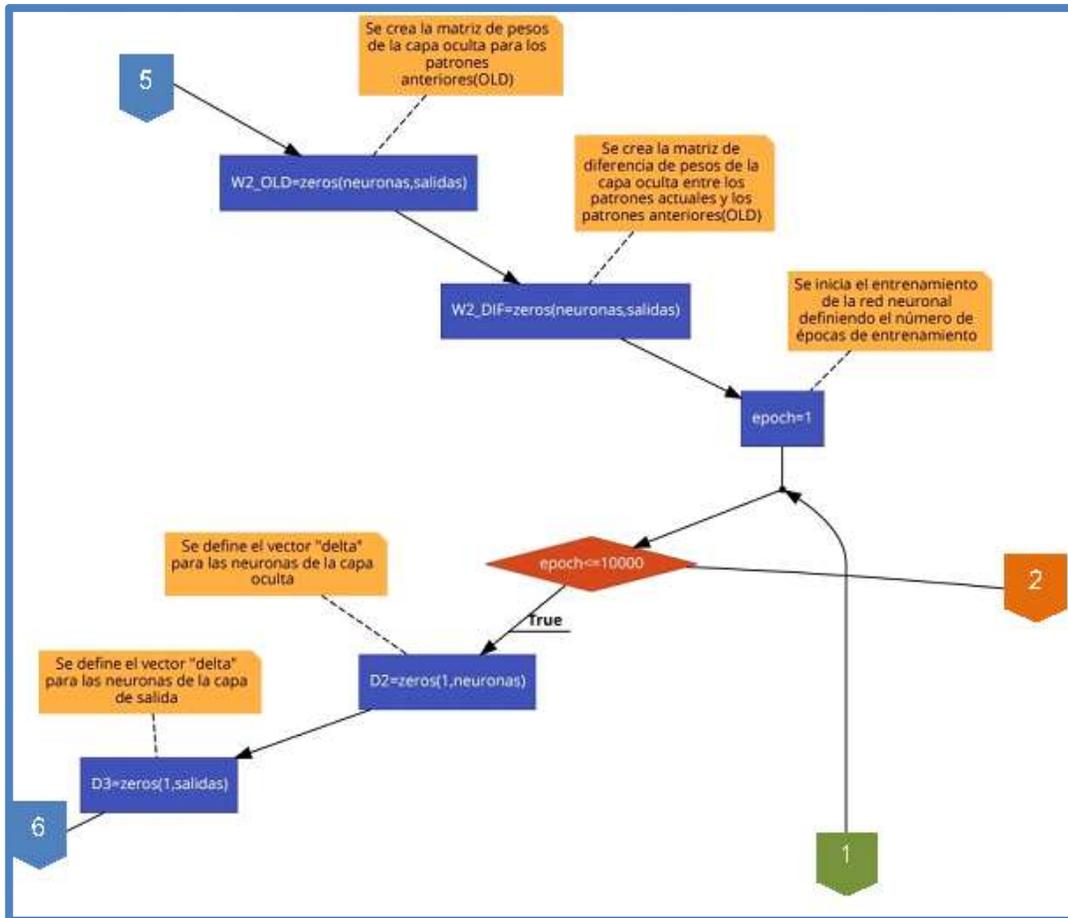


Fig. 143 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Principal – Parte 6

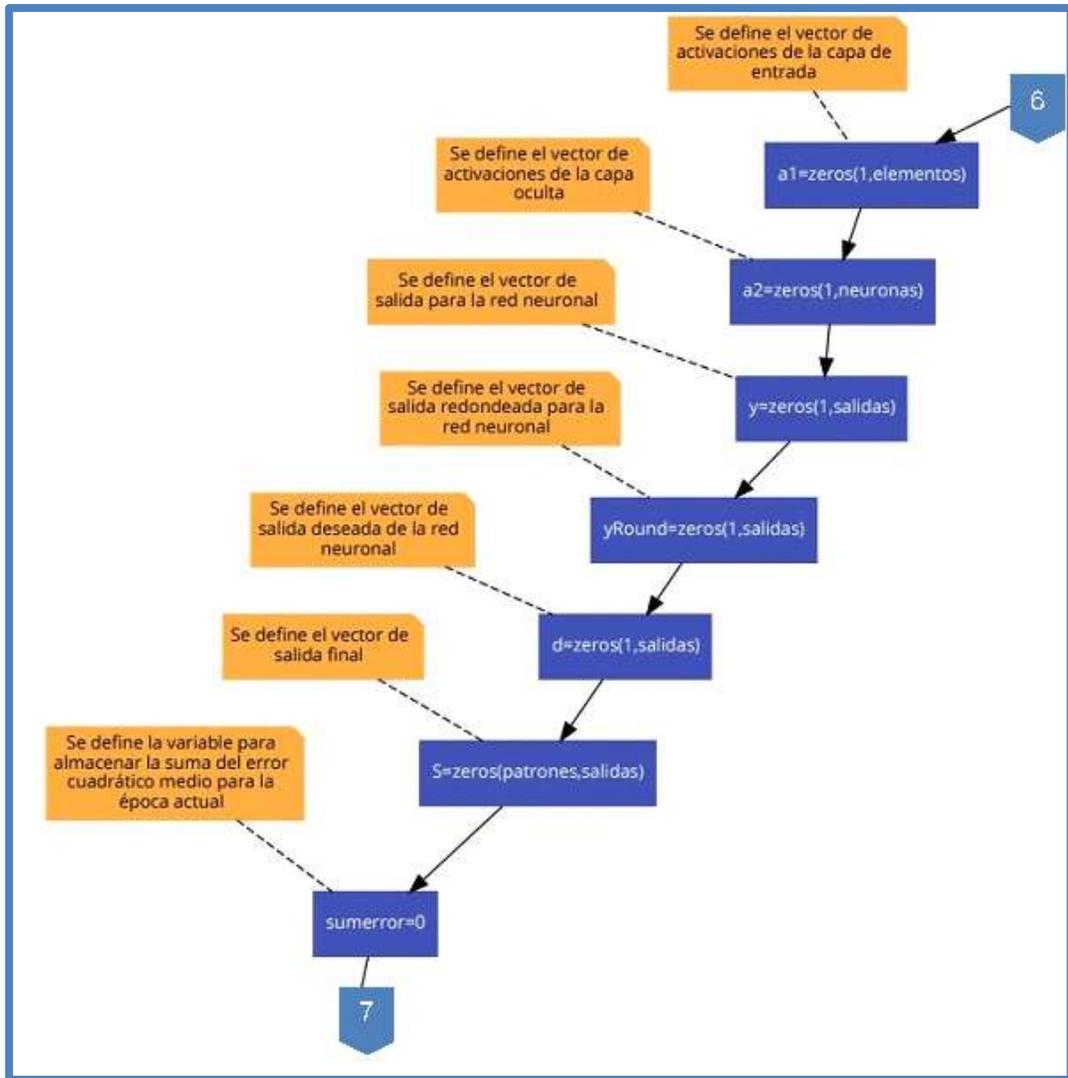


Fig. 144 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 1

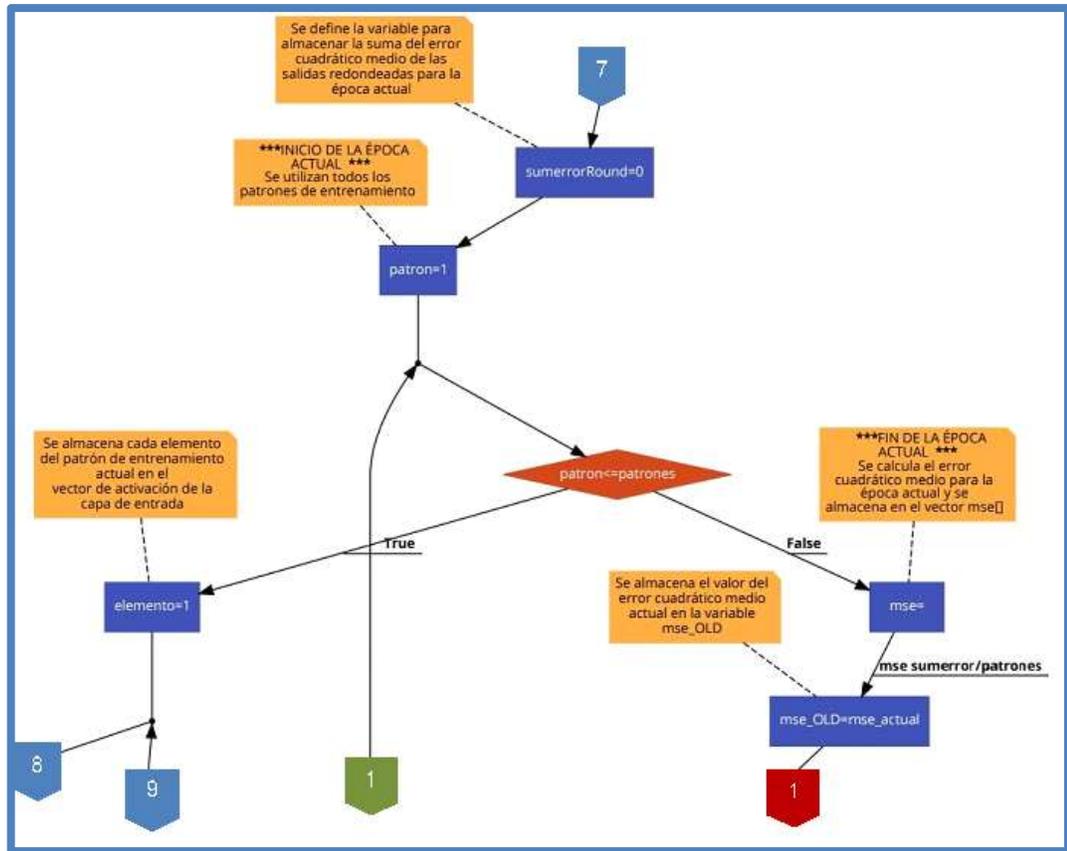


Fig. 145 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 2

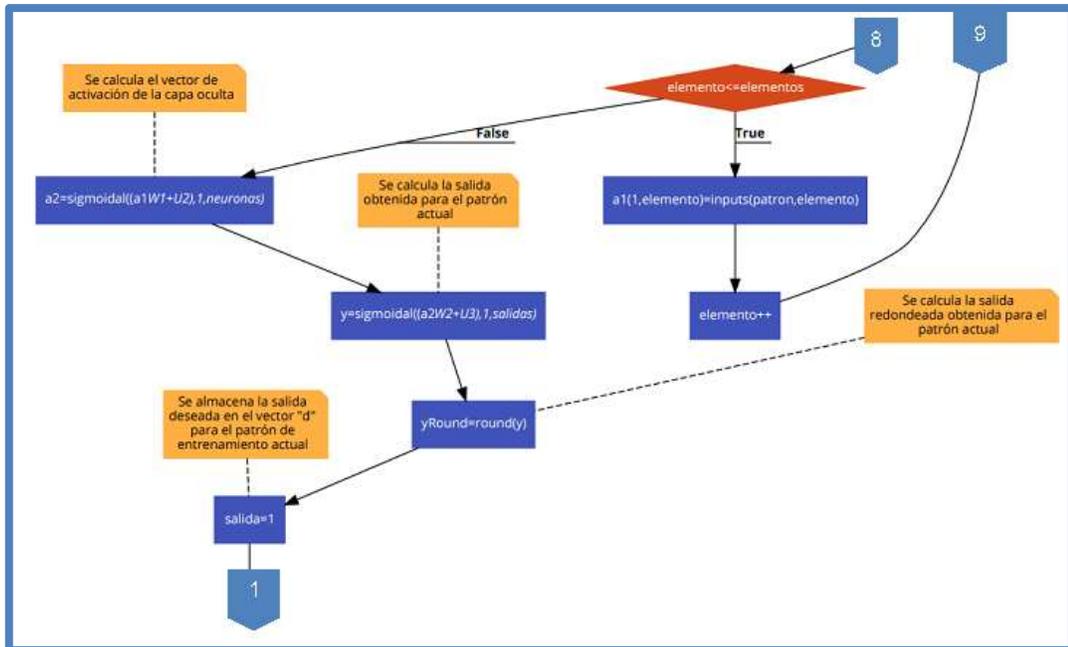


Fig. 146 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 3

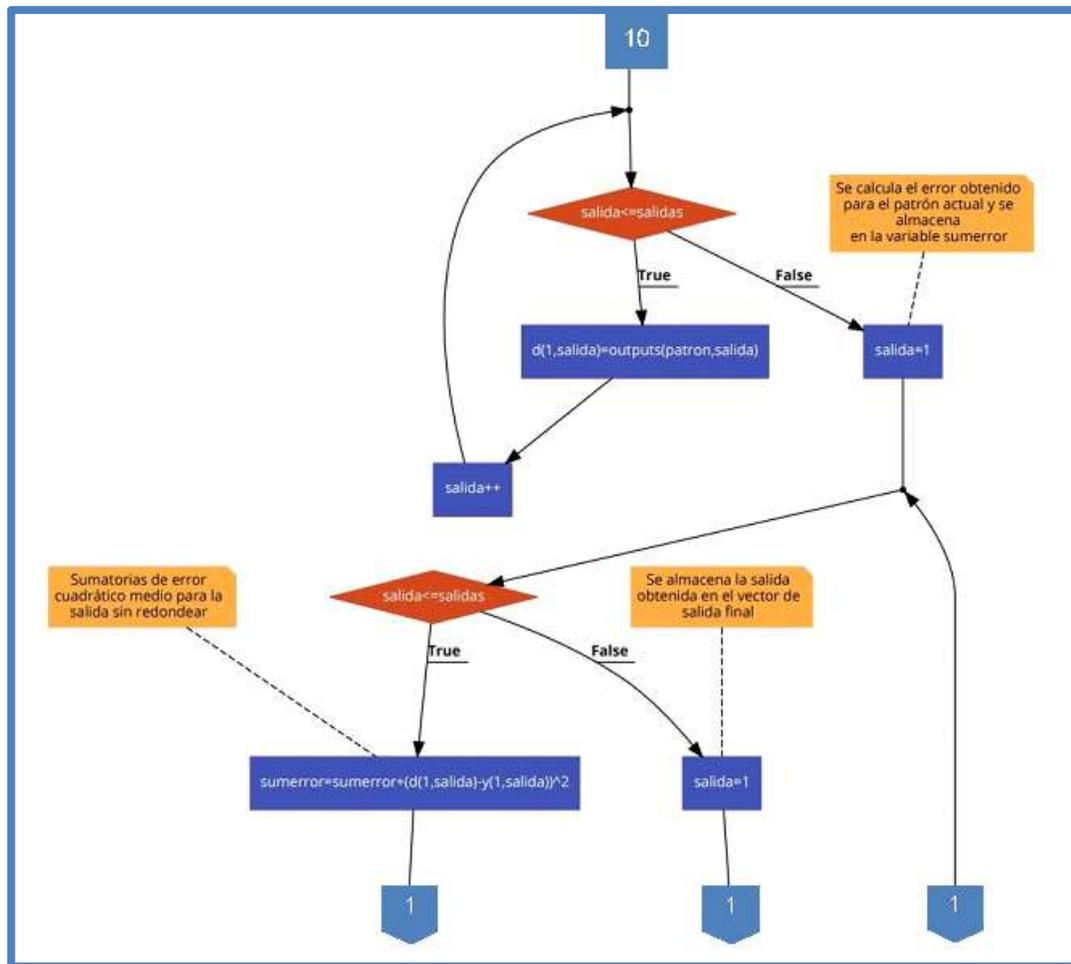


Fig. 147 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 4

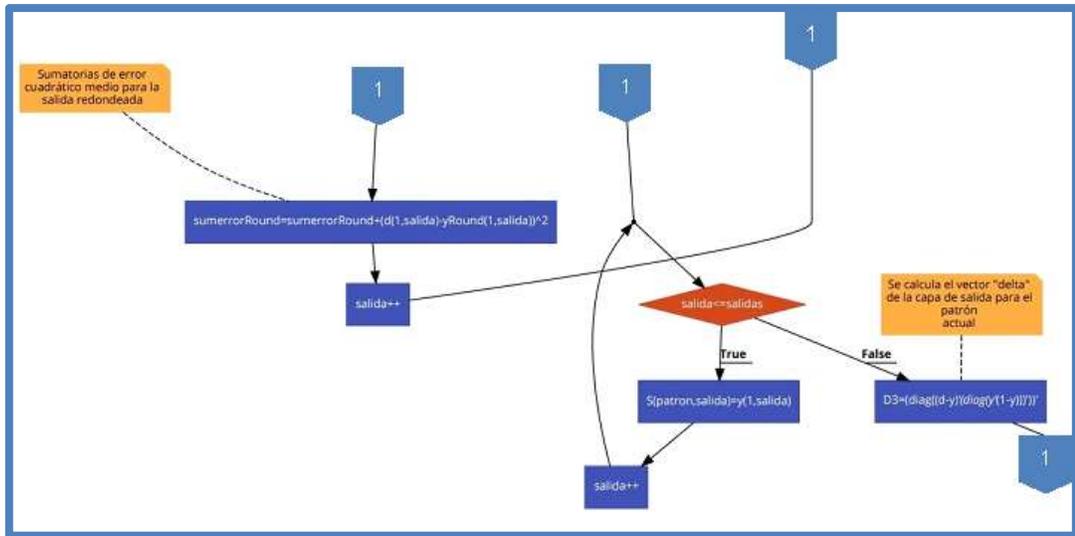


Fig. 148 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 4

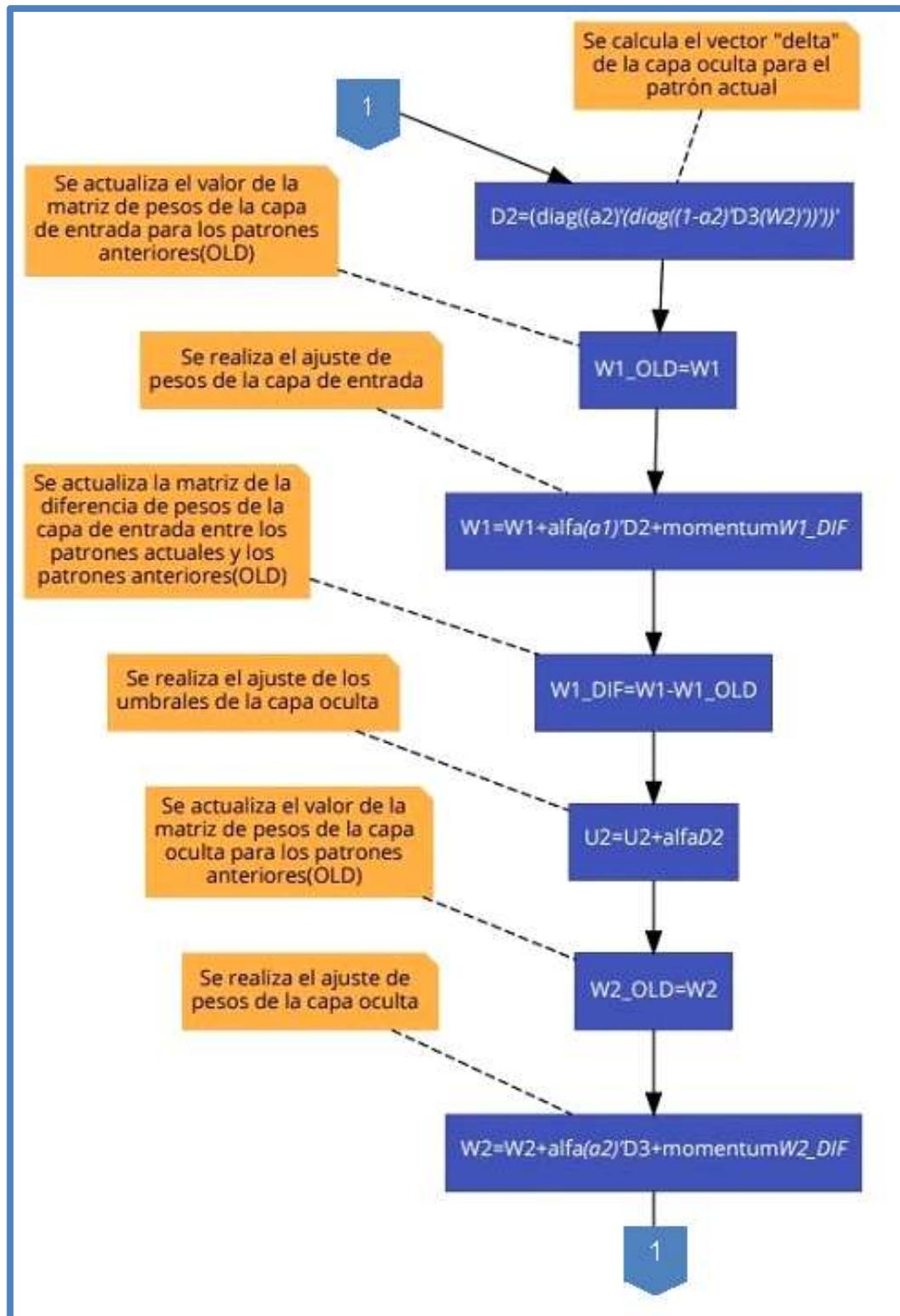


Fig. 149 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 5

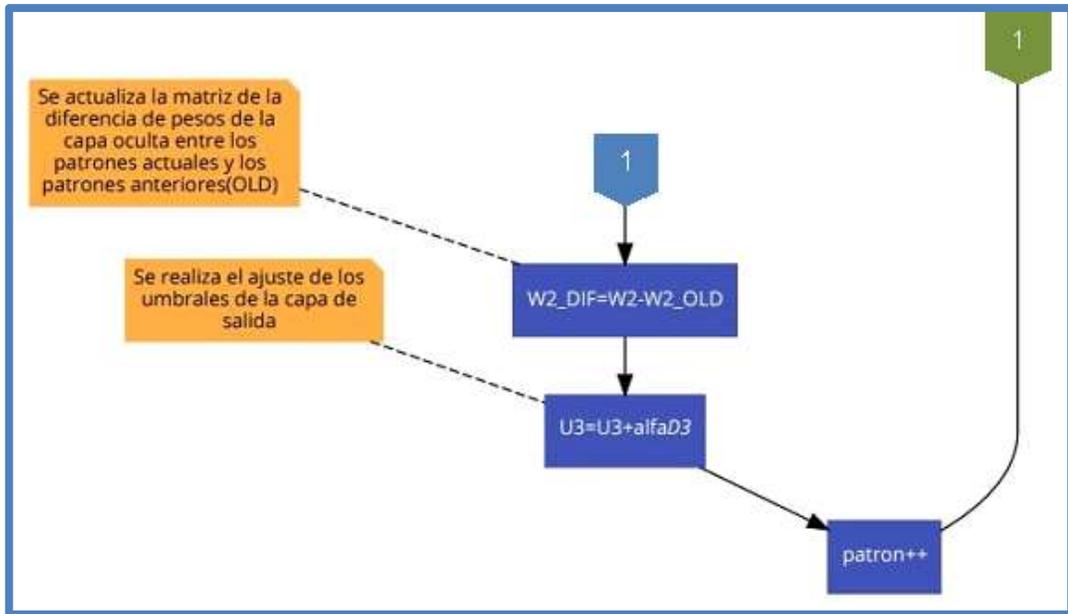


Fig. 150 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Izquierda – Parte 6

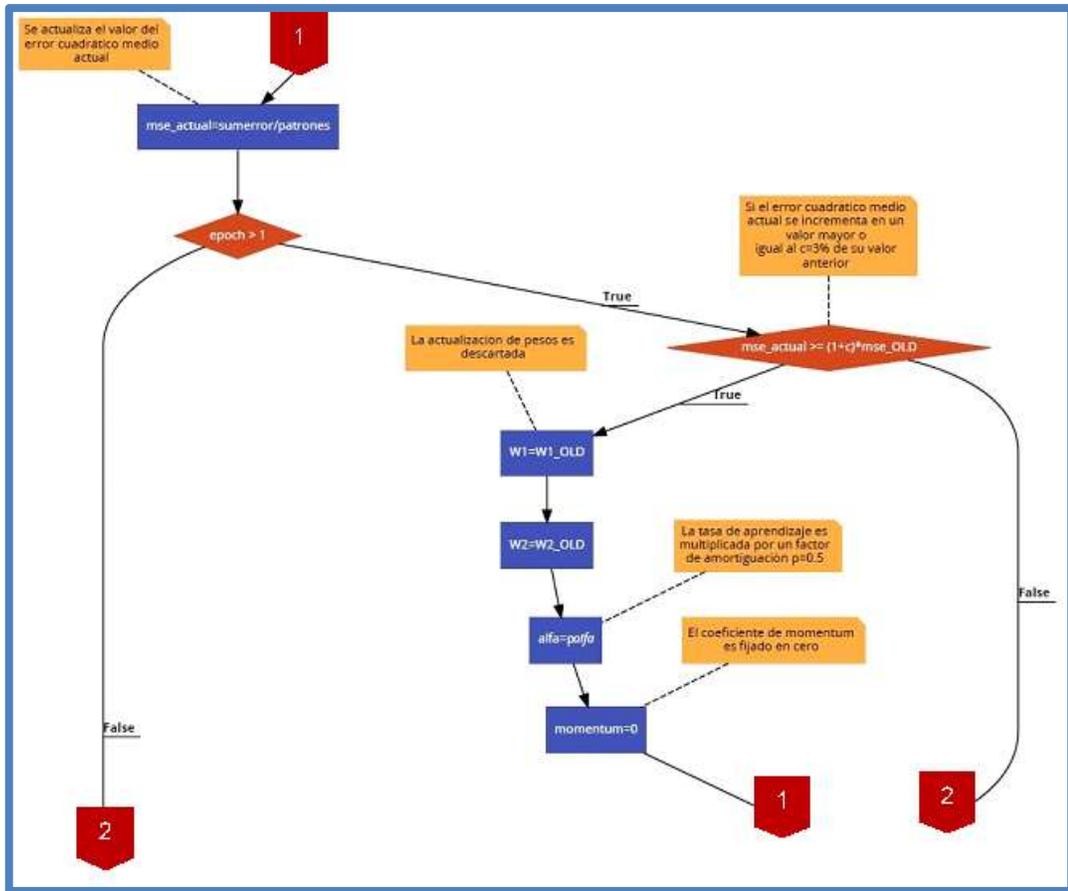


Fig. 151 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Central
– Parte 1

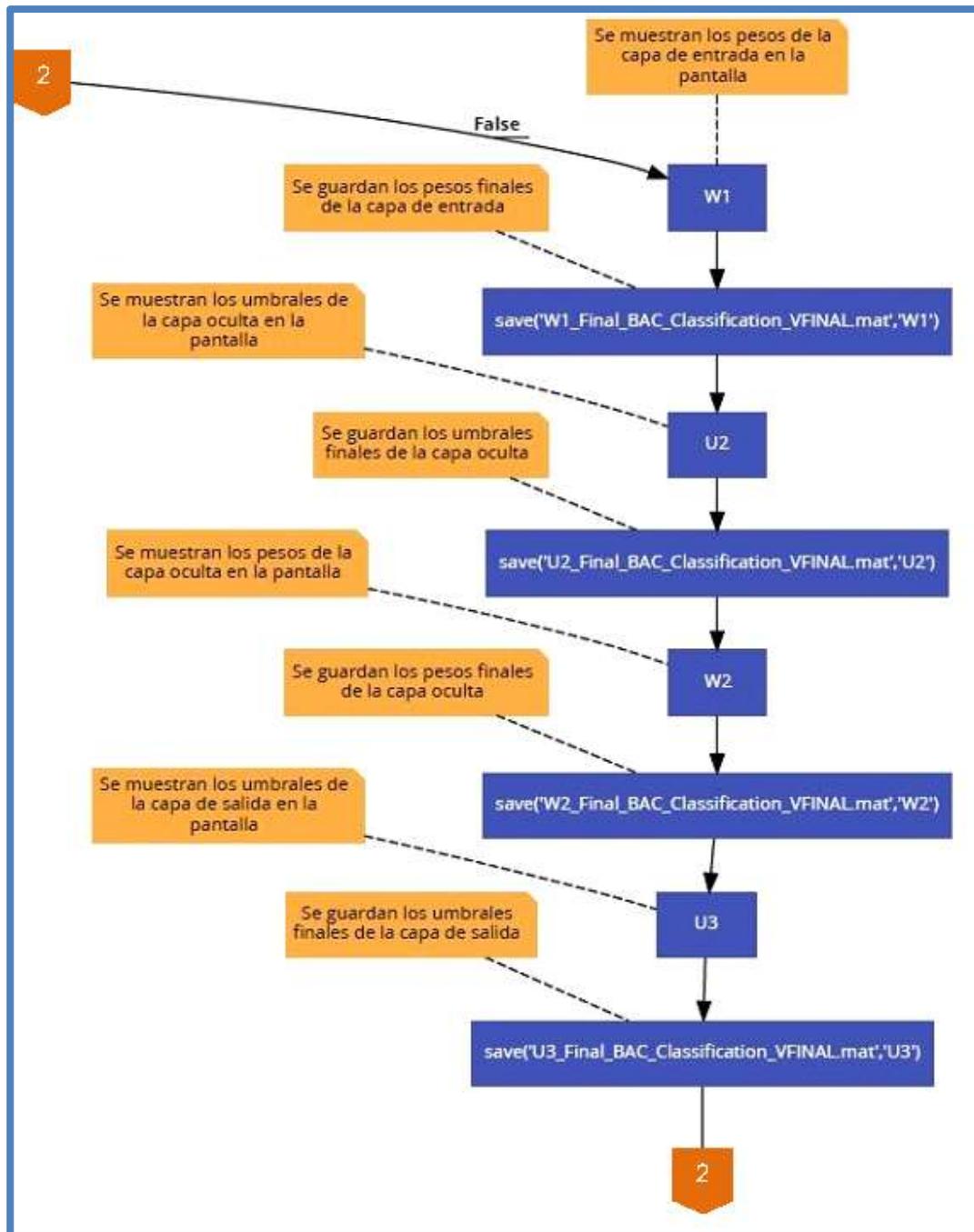


Fig. 153 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Derecha – Parte 1

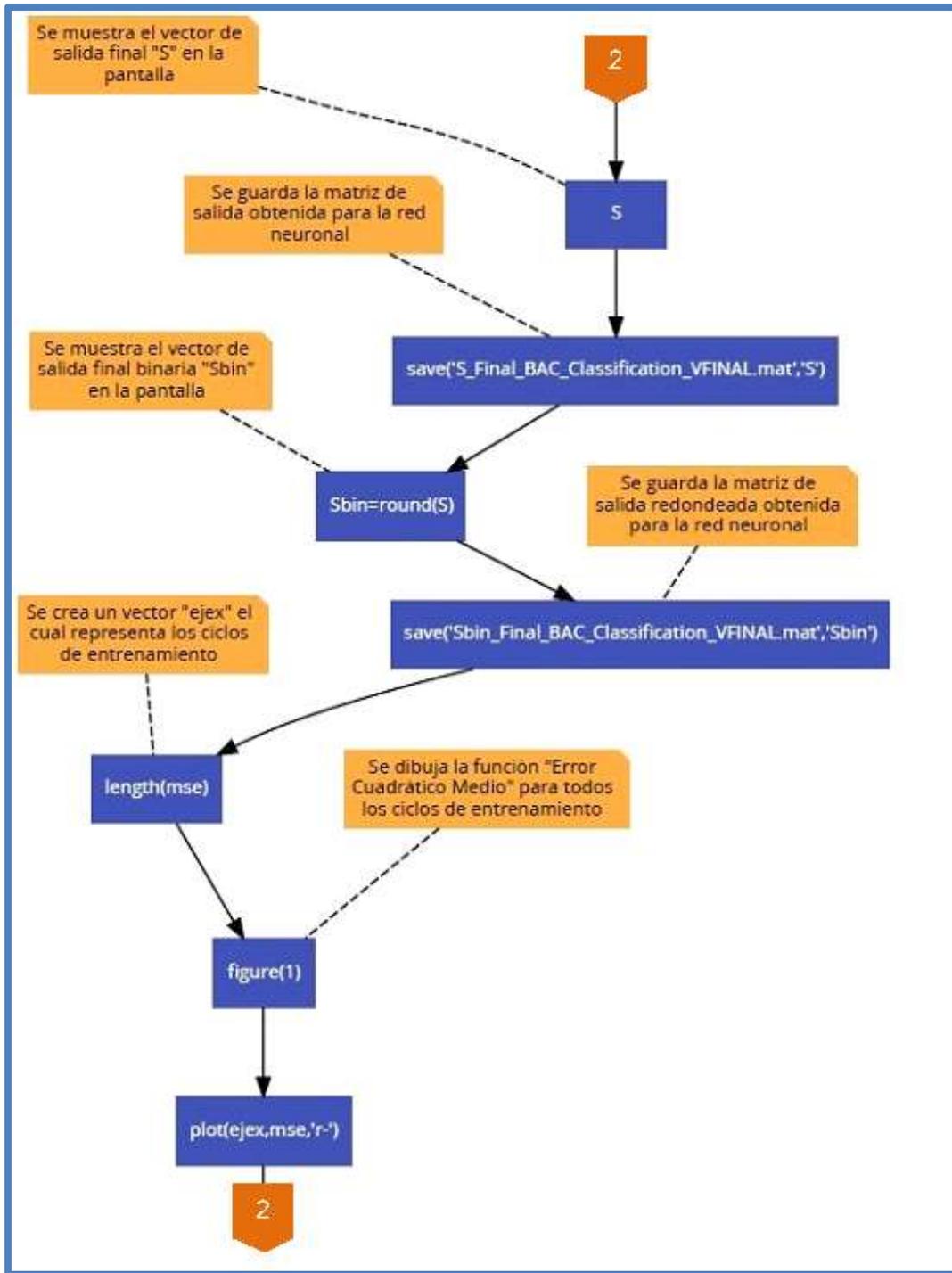


Fig. 154 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Derecha
– Parte 2

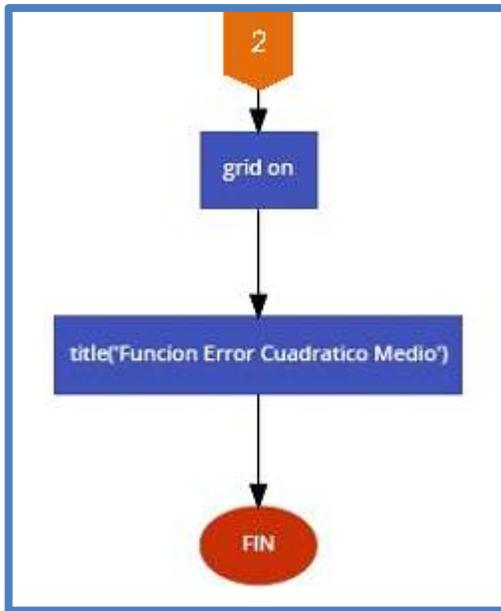


Fig. 155 Diagrama de Flujo del Algoritmo de Retropropagación – Rama Derecha
– Parte 3

Anexo 7 Código en Matlab del Algoritmo de Retropropagación

Este anexo ha sido desarrollado en su totalidad por los autores de la presente tesis mediante el software Matlab.

```
1 %Se borran todas las variables locales
2 clear all;
3 %Se cierran todas las ventanas
4 close all;
5 %Se limpia la pantalla
6 clc;
7
8 %Se cargan los vectores de entrada y salida esperada de la red neuronal
9 load('inputs_BAC_Classification_VFINAL.mat');
10 load('outputs_BAC_Classification_VFINAL.mat');
11
12 %Se almacena la dimensión de las matrices de entrada y salida en 2 vectores
13 inputsize=size(inputs);
14 outputsize=size(outputs);
15
16 %Se determina el número de patrones de entrada para la red neuronal
17 patrones=inputsize(1);
18
19 %Se determina el número de elementos que conforman 1 patrón, mejor dicho se
20 %determina el número de entradas a la red neuronal
21 elementos=inputsize(2);
22
23 %Se determina el número de salidas de la red neuronal
24 salidas=outputsize(2);
25
26 %Se define el número de neuronas de la capa oculta
27 neuronas=11;
28
29 %Se crea la matriz de pesos de la capa de entrada
30 W1=0.1*rand(elementos,neuronas)
31
32 %Se guardan los pesos iniciales de la capa de entrada
33 save('W1_Inicial_BAC_Classification_VFINAL.mat','W1');
34
35 %Se crea el vector de umbrales de la capa oculta
36 U2=0.1*rand(1,neuronas)
37
38 %Se guardan los umbrales iniciales de la capa oculta
39 save('U2_Inicial_BAC_Classification_VFINAL.mat','U2');
40
41 %Se crea la matriz de pesos de la capa oculta
42 W2=0.1*rand(neuronas,salidas)
43
44 %Se guardan los pesos iniciales de la capa oculta
45 save('W2_Inicial_BAC_Classification_VFINAL.mat','W2');
46
47 %Se crea el vector de umbrales de la capa de salida
48 U3=0.1*rand(1,salidas)
49
50 %Se guardan los umbrales iniciales de la capa de salida
51 save('U3_Inicial_BAC_Classification_VFINAL.mat','U3');
52
53 %Se define el vector de error cuadrático medio para cada época de
54 %entrenamiento
55 mse=[];
56
57 %Se define una variable para almacenar el error cuadrático medio de la época
58 mse_actual=0;
59
```

Fig. 156 Código en Matlab del Algoritmo de Retropropagación – Parte 1

```

60 %Se define una variable para almacenar el error cuadrático medio de la época
anterior
61 mse_OLD=0;
62
63 %Se define la tasa de aprendizaje
64 alfa=0.1;
65
66 %Se define el factor de amortiguación de la tasa de aprendizaje
67 p=0.5;
68
69 %Se define el factor de incremento de la tasa de aprendizaje
70 n=1.5;
71
72 %Se define el porcentaje de incremento del error cuadrático medio (3%)
73 c=0.03;
74
75 %Se define el coeficiente del "momentum"
76 momentum=0.1;
77
78 %Se define una variable para almacenar el valor del "momentum inicial"
79 momentum_inicial=momentum;
80
81 %Se crea la matriz de pesos de la capa de entrada para los patrones
anteriores (OLD)
82 W1_OLD=zeros(elementos,neuronas);
83
84
85 %Se crea la matriz de diferencia de pesos de la capa de entrada entre los patrones
actuales y los patrones anteriores (OLD)
86 W1_DIF=zeros(elementos,neuronas);
87
88 %Se crea la matriz de pesos de la capa oculta para los patrones
anteriores (OLD)
89 W2_OLD=zeros(neuronas,salidas);
90
91
92 %Se crea la matriz de diferencia de pesos de la capa oculta entre los patrones
actuales y los patrones anteriores (OLD)
93 W2_DIF=zeros(neuronas,salidas);
94
95 %Se inicia el entrenamiento de la red neuronal definiendo el número de
épocas de entrenamiento
96 for epoch=1:10000
97
98
99 %Se define el vector "delta" para las neuronas de la capa oculta
100 D2=zeros(1,neuronas);
101
102 %Se define el vector "delta" para las neuronas de la capa de salida
103 D3=zeros(1,salidas);
104
105 %Se define el vector de activaciones de la capa de entrada
106 a1=zeros(1,elementos);
107
108 %Se define el vector de activaciones de la capa oculta
109 a2=zeros(1,neuronas);
110
111 %Se define el vector de salida para la red neuronal
112 y=zeros(1,salidas);
113
114 %Se define el vector de salida redondeada para la red neuronal
115 yRound=zeros(1,salidas);
116

```

Fig. 157 Código en Matlab del Algoritmo de Retropropagación – Parte 2

```

117 %Se define el vector de salida deseada de la red neuronal
118 d=zeros(1,salidas);
119
120 %Se define el vector de salida final
121 S=zeros(patrones,salidas);
122
123 %Se define la variable para almacenar la suma del error cuadrático medio para la
época actual
124 sumerror=0;
125
126 %Se define la variable para almacenar la suma del error cuadrático medio de las
salidas redondeadas para la época actual
127 sumerrorRound=0;
128
129 %*****INICIO DE LA ÉPOCA ACTUAL*****%
130
131 %Se utilizan todos los patrones de entrenamiento
132 for patron=1:patrones
133
134     %Se almacena cada elemento del patrón de entrenamiento actual en el
135     %vector de activación de la capa de entrada
136     for elemento=1:elementos
137         a1(1,elemento)=inputs(patron,elemento);
138     end
139
140     %Se calcula el vector de activación de la capa oculta
141     a2=sigmoidal((a1*W1+U2),1,neuronas);
142
143     %Se calcula la salida obtenida para el patrón actual
144     y=sigmoidal((a2*W2+U3),1,salidas);
145
146     %Se calcula la salida redondeada obtenida para el patrón actual
147     yRound=round(y);
148
149     %Se almacena la salida deseada en el vector "d" para el patrón de
entrenamiento actual
150     for salida=1:salidas
151         d(1,salida)=outputs(patron,salida);
152     end
153
154     %Se calcula el error obtenido para el patrón actual y se almacena
155     %en la variable sumerror
156     for salida=1:salidas
157
158         %Sumatorias de error cuadrático medio para la salida sin redondear
159         sumerror=sumerror+(d(1,salida)-y(1,salida))^2;
160
161         %Sumatorias de error cuadrático medio para la salida redondeada
162         sumerrorRound=sumerrorRound+(d(1,salida)-yRound(1,salida))^2;
163
164     end
165
166     %Se almacena la salida obtenida en el vector de salida final
167     for salida=1:salidas
168         S(patron,salida)=y(1,salida);
169     end
170
171     %Se calcula el vector "delta" de la capa de salida para el patrón
172     %actual
173     D3=(diag((d-y)'*(diag(y'*(1-y))'))');

```

Fig. 158 Código en Matlab del Algoritmo de Retropropagación – Parte 3

```

174
175 %Se calcula el vector "delta" de la capa oculta para el patrón actual
176 D2=(diag((a2)'+(diag((1-a2)'+D3*(W2)'))'))';
177
178 %Se actualiza el valor de la matriz de pesos de la capa de entrada para los
patrones anteriores(OLD)
179 W1_OLD=W1;
180
181 %Se realiza el ajuste de pesos de la capa de entrada
182 W1=W1+alfa*(a1)'+D2+momentum*W1_DIF;
183
184 %Se actualiza la matriz de la diferencia de pesos de la capa de entrada entre
los patrones actuales y los patrones anteriores(OLD)
185 W1_DIF=W1-W1_OLD;
186
187 %Se realiza el ajuste de los umbrales de la capa oculta
188 U2=U2+alfa*D2;
189
190 %Se actualiza el valor de la matriz de pesos de la capa oculta para los
patrones anteriores(OLD)
191 W2_OLD=W2;
192
193 %Se realiza el ajuste de pesos de la capa oculta
194 W2=W2+alfa*(a2)'+D3+momentum*W2_DIF;
195
196 %Se actualiza la matriz de la diferencia de pesos de la capa oculta entre los
patrones actuales y los patrones anteriores(OLD)
197 W2_DIF=W2-W2_OLD;
198
199 %Se realiza el ajuste de los umbrales de la capa de salida
200 U3=U3+alfa*D3;
201
202 end
203
204 %*****FIN DE LA ÉPOCA ACTUAL*****%
205
206 %Se calcula el error cuadrático medio para la época actual y se
207 %almacena en el vector mse[]
208 mse=[mse sumerror/patrones];
209
210 %Se almacena el valor del error cuadrático medio actual en la variable mse_OLD
211 mse_OLD=mse_actual;
212
213 %Se actualiza el valor del error cuadrático medio actual
214 mse_actual=sumerror/patrones;
215
216 if(epoch > 1)
217
218 %Si el error cuadrático medio actual se incrementa en un valor mayor o
219 %igual al c=3% de su valor anterior
220 if(mse_actual >= (1+c)*mse_OLD)
221
222 %La actualización de pesos es descartada
223 W1=W1_OLD;
224
225 W2=W2_OLD;
226
227 %La tasa de aprendizaje es multiplicada por un factor de amortiguación
p=0.5
228 alfa=p*alfa;

```

Fig. 159 Código en Matlab del Algoritmo de Retropropagación – Parte 4

```

229
230     %El coeficiente de momentum es fijado en cero
231     momentum=0;
232
233     end
234
235     %Si el error cuadrático medio actual se decrementa
236     if(mse_actual < mse_OLD)
237
238         %La actualización de pesos es aceptada
239
240         %La tasa de aprendizaje es multiplicada por un factor de incremento
241         n=1.5
242         alfa=n*alfa;
243
244         %Si el momentum había sido previamente puesto en cero, entonces se
245         retorna a su valor original
246         if(momentum==0)
247             momentum=momentum_inicial;
248         end
249     end
250
251     %Si el error cuadrático medio se incrementa en un valor menor al c=3% de
252     %su valor anterior, los pesos actualizados son aceptados, pero la tasa de
253     aprendizaje y el coeficiente de momentum no son cambiados
254
255     end
256
257     %Se muestra el error cuadrático medio redondeado actual
258     mse_Round_actual=sumerrorRound/patrones
259
260     %Se muestra la época actual
261     epoch
262
263 end
264
265 %Se muestran los pesos de la capa de entrada en la pantalla
266 W1
267
268 %Se guardan los pesos finales de la capa de entrada
269 save('W1_Final_BAC_Classification_VFINAL.mat','W1');
270
271 %Se muestran los umbrales de la capa oculta en la pantalla
272 U2
273
274 %Se guardan los umbrales finales de la capa oculta
275 save('U2_Final_BAC_Classification_VFINAL.mat','U2');
276
277 %Se muestran los pesos de la capa oculta en la pantalla
278 W2
279
280 %Se guardan los pesos finales de la capa oculta
281 save('W2_Final_BAC_Classification_VFINAL.mat','W2');
282
283 %Se muestran los umbrales de la capa de salida en la pantalla
284 U3
285
286 %Se guardan los umbrales finales de la capa de salida
287 save('U3_Final_BAC_Classification_VFINAL.mat','U3');

```

Fig. 160 Código en Matlab del Algoritmo de Retropropagación – Parte 5

```

286
287 %Se muestra el vector de salida final "S" en la pantalla
288 S
289
290 %Se guarda la matriz de salida obtenida para la red neuronal
291 save('S_Final_BAC_Classification_VFINAL.mat','S');
292
293 %Se muestra el vector de salida final binaria "Sbin" en la pantalla
294 Sbin=round(S)
295
296 %Se guarda la matriz de salida redondeada obtenida para la red neuronal
297 save('Sbin_Final_BAC_Classification_VFINAL.mat','Sbin');
298
299 %Se crea un vector "ejex" el cual representa los ciclos de entrenamiento
300 ejex=1:length(mse);
301
302 %Se dibuja la función "Error Cuadrático Medio" para todos los ciclos de
entrenamiento
303 figure(1);
304 plot(ejex,mse,'r-');
305 grid on;
306 title('Funcion Error Cuadrático Medio');
307

```

Fig. 161 Código en Matlab del Algoritmo de Retropropagación – Parte 6

Anexo 8 Diagrama de Flujo General de la Aplicación

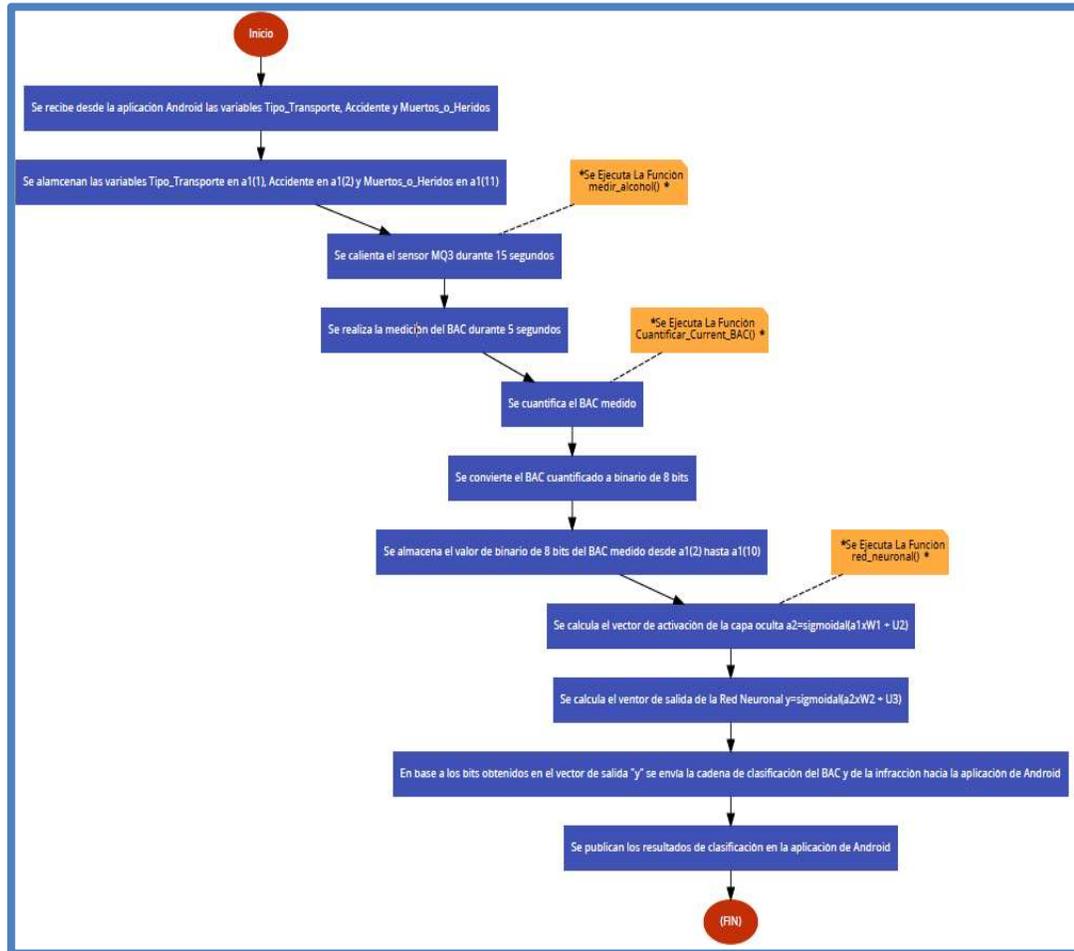


Fig. 162 Diagrama de Flujo General de la Aplicación

Elaboración: los autores

Fuente: code2flow

Anexo 9 Diagrama Circuital Completo

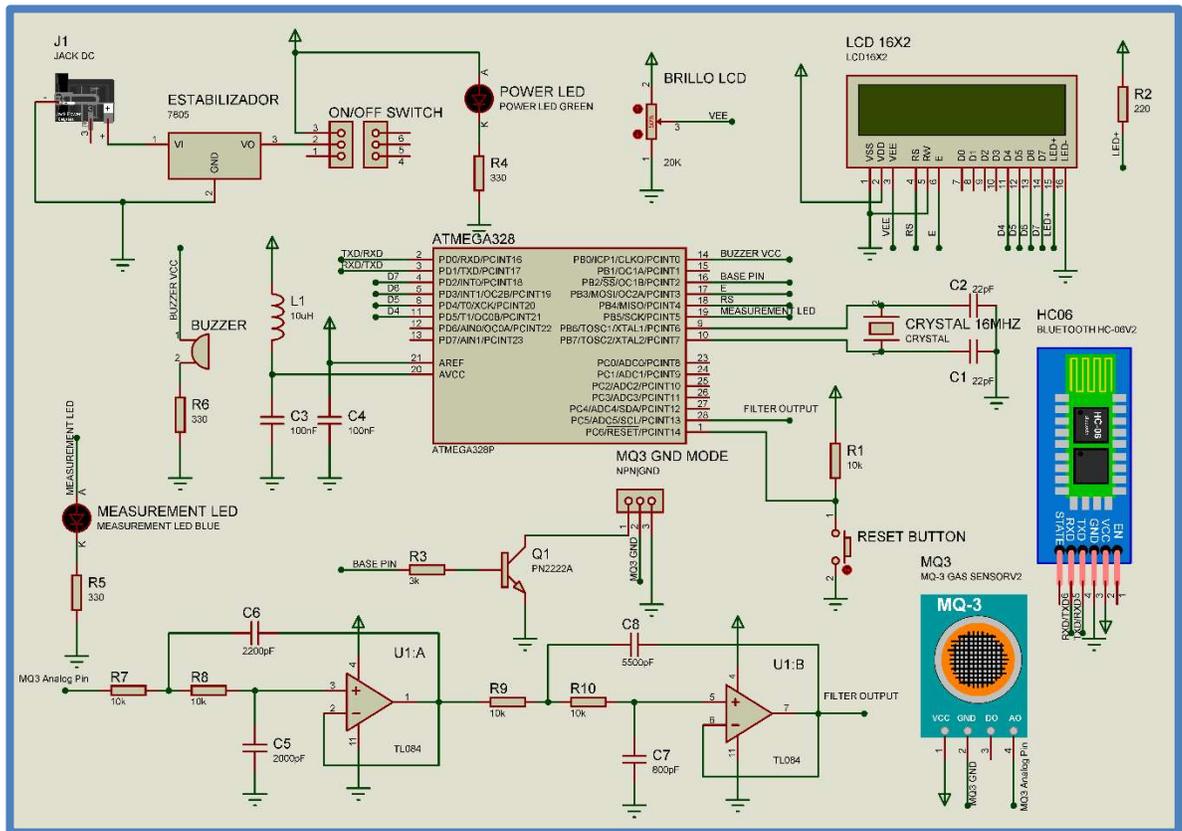


Fig. 163 Diagrama Circuital Completo

Elaboración: los autores

Fuente: Proteus 8 Professional

Anexo 10 PCB Completo

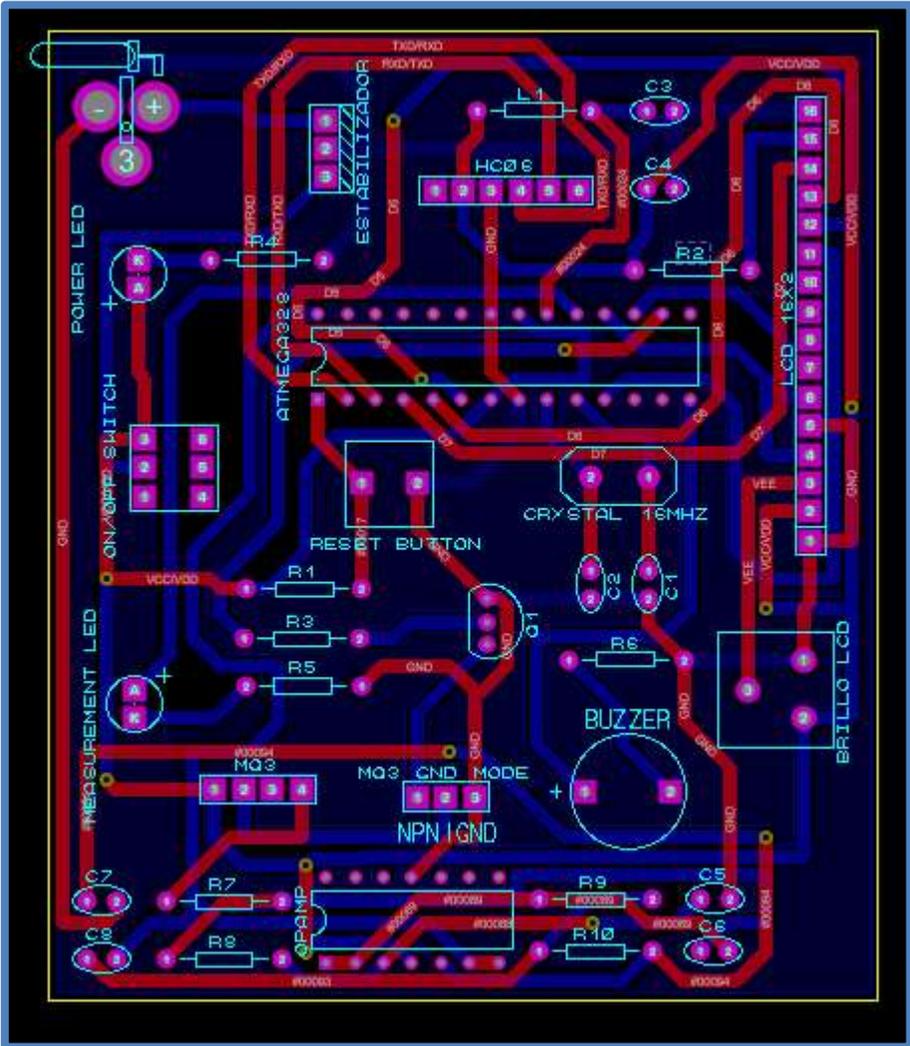


Fig. 164 PCB Completo

Elaboración: los autores

Fuente: Proteus 8 Professional

Anexo 11 Glosario

Este anexo ha sido desarrollado en su totalidad por los autores de la presente tesis.

Tabla 8. Glosario

Término	Significado
BAC	Del Inglés Blood alcohol concentration (BAC) , que quiere decir concentración de alcohol en la sangre, es comúnmente usado como métrica de la intoxicación por alcohol para propósitos legales o médicos.
BIT	Bit es el acrónimo de Binary digit (o sea de 'dígito binario', en español denominado como bit, y en menor medida como bitio). Un bit es un dígito del sistema de numeración binario. La capacidad de almacenamiento de una memoria digital también se mide en bits, pues esta palabra tiene varias acepciones.
LCD	Sigla del inglés Liquid Cristal Display, 'representación visual por cristal líquido', sistema que utilizan determinadas pantallas electrónicas para mostrar información visual.
LED	Sigla de la expresión inglesa light-emitting diode, 'diodo emisor de luz',

	que es un tipo de diodo empleado en computadoras, paneles numéricos (en relojes digitales, calculadoras de bolsillo...), etc.
MLP	Sigla de la expresión MultiLayer Perceptron, el cual se utiliza para hacer referencia a la arquitectura de Perceptrón Multicapa en una red neuronal.
MQ3	Sensor de alcohol el cuál puede detectar etanol en el aire.
PCB	Del inglés: Printed Circuit Board, PCB), es la superficie constituida por caminos, pistas o buses de material conductor laminadas sobre una base no conductora.
PIN	Pequeña clavija terminal de un circuito eléctrico.
RNA	Siglas de Red Neuronal Artificial.
RXD	Siglas que hacen referencia a la data recibida (Receive Data en inglés).
TXD	Siglas que hacen referencia a la data transmitida (Transceive Data en inglés).