



**REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD NACIONAL EXPERIMENTAL DE GUAYANA  
VICERRECTORADO ACADÉMICO  
COORDINACIÓN GENERAL DE INVESTIGACIÓN Y POSTGRADO  
MAESTRÍA EN TECNOLOGÍAS DE LA INFORMACION**

**MODELO DE DIAGNOSTICO DEL DENGUE BASADO EN INTELIGENCIA  
ARTIFICIAL. CASO REGION DE SAN MARTIN – PERÚ.**

Trabajo de Grado para optar el grado académico de Magister en Tecnologías de la  
Información.

Autor : Ing. Buenaventura Ríos Ríos  
Tutor : Mg. Carlos Guevara

Puerto Ordaz, diciembre 2021.



## TABLA DE CONTENIDO

	<u>Página</u>
RESUMEN	8
INTRODUCCIÓN	9
CAPÍTULO I: PROBLEMA.	10
Planteamiento del Problema	10
Objetivo General	15
Objetivos Específicos	16
Justificación	16
CAPÍTULO II: MARCO TEÓRICO	17
Antecedentes	17
Bases Teóricas	22
Sistemas Inteligentes	22
Inteligencia Artificial	24
Redes Neuronales Artificiales	25
Arquitectura de las Redes Neuronales Artificiales	27
Aprendizaje de la Redes Neuronales Artificiales	32
Algoritmo de entrenamiento Backpropagation	34
Características y ventajas de las RNA MLP	43
Estado del Arte	44
Diagnóstico del dengue utilizando Redes Neuronales Artificiales	44
Diagnóstico del dengue utilizando Lógica Difusa	61
Diagnóstico del dengue utilizando Algoritmos Genéticos	80
CAPÍTULO III: MARCO METODOLÓGICO	93
Diseño de la Investigación	93
Prototipo de la investigación	94
Diagrama de flujo de diseño del prototipo	95
Selección de la Muestra	96
Recolección de Datos	97
Análisis de Datos	98
Pre procesamiento de datos	98

Determinación de patrones para el entrenamiento, validación y test	98
Segmentación de datos utilizando feedforward	99
Determinación de la arquitectura de red neural	99
Entrenamiento y simulación de la red neural	100
CAPITULO IV: RESULTADOS	101
Resultados del primer objetivo específico	101
Resultados del segundo objetivo específico	103
Resultados del tercer objetivo específico	104
Resultados del cuarto objetivo específico	105
Resultados del quinto objetivo específico	106
Acondicionamiento de la red neuronal feedforward	106
Entrenamiento con 24 entradas, una capa oculta y una capa de salida	111
Entrenamiento, validación y test con dos capas ocultas	111
Entrenamiento de la red feedforward versus cascadefeedforward	112
Entrenamiento haciendo uso de MATLAB	114
Entrenamiento de la red cascadefeedforward con dos capas ocultas	116
Exportación al Simulink de MATLAB	121
Aplicación del bloque de test para efectos de validación de la red neural	122
CAPITULO V: CONCLUSIONES Y RECOMENDACIONES	125
Conclusiones	125
Recomendaciones	127
REFERENCIAS BIBLIOGRÁFICAS	128
ANEXOS	133

## INDICE DE FIGURAS

	<u>Página</u>
Figura 1. Número de casos de dengue, Perú 2014 – 2020	14
Figura 2. Evolución del dengue clásico al dengue hemorrágico.	18
Figura 3. Esquema de un sistema inteligente que percibe, razona y actúa.	23
Figura 4. Esquema de una neurona biológica.	25
Figura 5. Esquema de una neurona artificial.	26
Figura 6. Perceptrón con una capa de salida.	29
Figura 7. Limitador fuerte.	29
Figura 8. Limitador fuerte simétrico.	29
Figura 9. Red multicapa con una capa oculta.	30
Figura 10. Algoritmo que distribuye datos clínicos y datos BIA.	47
Figura 11. RNA de tres capas y atributos de entrada del dengue.	48
Figura 12. Red neuronal artificial con 10 entradas, neurona oculta y una salida.	52
Figura 13. Cuadro comparativo de precisión, sensibilidad y especificidad.	53
Figura 14. Características de operación de receptor (ROC o AUC).	55
Figura 15. Cuadro comparativo del dengue con la simulación en lógica difusa.	64
Figura 16. Resultado del modelamiento difuso para el dengue.	65
Figura 17. Resultados de FMID + MRA con rendimiento 90.06%.	66
Figura 18. Estimación de probabilidad de hábitat del Aedes Aegypti.	68
Figura 19. Simulación difusa de efectos combinados y casos de dengue.	70
Figura 20. Mapa de riesgo del dengue.	71
Figura 21. Resultados de simulación en MATLAB.	73
Figura 22. Enfermedades estudiadas por la lógica difusa.	76
Figura 23. Arquitectura de un sistema fuzzy logic.	77
Figura 24. Diagrama de flujo del sistema dengue.	77
Figura 25. Organización del genoma DENV.	82
Figura 26. Operación de cruzamiento de individuos.	88
Figura 27. Evolución de la población de mosquitos con insecticida y esterilizados.	89
Figura 28. Diagrama de flujo de diseño de prototipo.	96
Figura 29. Ventanas del MATLAB.	114

Figura 30. Arquitectura de la RNA MLP implementada.	114
Figura 31. Convergencia de datos target y datos output.	116
Figura 32. Mejor curva de validación.	117
Figura 33. Estado del entrenamiento.	118
Figura 34. Histograma del error.	119
Figura 35. Mejor factor de correlación.	120
Figura 36. Diagrama de bloques en SIMULINK.	121
Figura 37. Respuesta de prueba de la red cascadedfeedforward.	122

## INDICE DE TABLAS

	<u>Página</u>
Tabla1. Casos de dengue en la región San Martín	14
Tabla 2. AUC resultado de diferentes clasificadores	49
Tabla 3. Análisis comparativo del modelo propuesto	51
Tabla 4. Variables de las neuronas de entrada por categoría	53
Tabla 5. Mejores GAM, ANN y SARIMA empleados	56
Tabla 6. Modelos de predicción	57
Tabla 7. Parámetros utilizados como entradas	59
Tabla 8. Cuadro resumen de investigación considerando RNA	60
Tabla 9. Precisión, sensibilidad y especificidad del Fuzzy, KNN y NB.	62
Tabla 10. Parámetros espaciales y temporales considerados.	63
Tabla 11. Detalles de las imágenes de satélite utilizados.	67
Tabla 12. Evaluación de entrenamiento y pruebas del modelo difuso.	78
Tabla 13. Investigación del dengue utilizando lógica difusa.	79
Tabla 14. Comparación del host infectado total.	85
Tabla 15. Investigación sobre el dengue utilizando algoritmos genéticos.	91
Tabla 16. Investigaciones realizadas utilizando estrategias de IA	101
Tabla 17. Síntomas del dengue	104
Tabla 18. Segmentación automática para entrenamiento, validación y test.	106
Tabla 19. Comparación de funciones tansig y purelin en la capa de salida.	107
Tabla 20. Comparación de funciones de entrenamiento trainlm y trainbr.	109
Tabla 21. Cuadro comparativo de la red feedforward con dos capas ocultas.	112
Tabla 22. Comparación red feedforward versus cascadefeedforward.	113
Tabla 23. Resultados del rendimiento de la red neuronal cascadefeedforward.	122

## RESUMEN

El objetivo del estudio es el diseño e implementación de un modelo de diagnóstico del dengue, enfermedad viral producida por el mosquito *Aedes Aegypti*, de gran impacto en América Latina, basado en inteligencia artificial, en la región San Martín del Perú. El tipo de investigación realizada es aplicada ex post facto, descriptiva y transversal; la herramienta de inteligencia artificial utilizada fue una Red Neuronal Artificial – Perceptrón Multicapa (RNA-MLP) *cascadefeedforward*, de dos capas ocultas, con estructura de red 24 – 12 – 4 – 1; la muestra utilizada corresponde a 430 patrones para entrenamiento, validación y test, con 24 entradas por cada patrón, información obtenida de la Dirección Regional de Salud (DIRESA) de la región San Martín. El mejor resultado obtenido corresponde a un error de validación equivalente a 0.0457, un factor de regresión  $R=0.90715$  y una efectividad de 95.35 %.

Palabras Clave: Dengue, redes neuronales artificiales, perceptrón multicapa, retropropagación, redes de avance.

## INTRODUCCIÓN

El presente trabajo de grado, para optar el grado de Magister en Tecnologías de la Información, otorgada por la Universidad Nacional Experimental de Guayana (UNEG) – Venezuela, está orientado a la investigación y uso de las Redes Neuronales Artificiales para el diagnóstico del dengue, enfermedad viral producto de la picadura del mosquito *Aedes Aegypti* y sus diferentes serotipos que, en su forma más severa, el dengue grave, puede ser mortal.

La investigación se realizó en la región San Martín, que es el equivalente a los estados de la organización política de Venezuela, que tiene una población de 850 mil habitantes, ubicada al noreste del Perú, entre los paralelos  $5^{\circ} 15' 00''$  y  $8^{\circ} 25' 15''$  de latitud sur, y, entre los meridianos  $75^{\circ} 45' 00''$  y  $77^{\circ} 24' 00''$  de longitud oeste; enclavada en la Amazonía peruana que abarca el 57% del territorio nacional.

Este documento está estructurado en tres capítulos. El primero, contiene el planteamiento del problema, el objetivo general, los objetivos específicos, la justificación y el alcance. El Segundo Capítulo, se refiere al marco teórico el cual contiene los antecedentes del problema, bases teóricas, estado del arte y la definición de términos básicos. Finalmente, el Tercer Capítulo, que contiene el marco metodológico, en donde se describe el procedimiento seguido para determinar la red neuronal artificial utilizada, su arquitectura, funciones de activación, algoritmo de aprendizaje y entrenamiento. El documento finaliza con la presentación de los resultados, conclusiones, recomendaciones, referencias bibliográficas y los anexos.

## **CAPITULO I**

### **PROBLEMA**

#### **Planteamiento del problema**

En la actualidad, el dengue es la enfermedad viral transmitida por mosquitos con un crecimiento expansivo y dinámico en América Latina, en algunas regiones con características epidémicas, que afecta a todos por igual, sin distinción de raza, credo, sexo y edad, tanto en regiones tropicales como en aquellas que son subtropicales (Gurtler, 2009).

Situación que, en los últimos años, se ha visto complicada por la presencia de otro tipo de enfermedades virales que tienen por origen el mismo agente. Espinal (2015) advierte que en el momento en que otras arbovirosis (Chikungunya y Zika) se han introducido en la región, es cada vez más imprescindible contar con el diagnóstico oportuno y correcto de los casos de dengue, así como un seguimiento clínico adecuado.

La Organización Panamericana de la Salud (OPS) refiere que en el caso de la chikungunya el número de casos notificados hasta la semana epidemiológica (SE) 35 del 2014 a la institución asciende a 659.367, incluyendo 37 defunciones (OPS, 2014).

Según datos de la OPS el incremento del dengue en América Latina fue de 1.033.417 en la década de 1980, 2.725.405 en la década de 1990 y 4.759.007 del 2000 al 2007 (San Martín, 2010). Datos más recientes dan cuenta de que en la región se han presentado 2.439.527 casos de dengue, 12.495 de dengue hemorrágico con 1.365 muertes en el año 2015, y que, al 28 de octubre del 2016, se han registrado 2.094.443 casos de dengue, 3.787 de dengue hemorrágico y 912 muertes (OPS, 2016).

El dengue es una enfermedad transmitida por una variedad de mosquito conocido como el *Aedes Aegypti*, que se cría en agua limpia que se acumula en recipientes y objetos en desuso. En su forma grave, el dengue hemorrágico, puede llevar a la muerte si el paciente no es atendido en forma rápida y oportuna.

Esta enfermedad ha tenido y tiene un carácter endémico en la mayoría de los países de América Latina, con excepción de Chile, Cuba y Uruguay. En el Perú sus manifestaciones han sido recurrentes y han ido en aumento. Según datos del Ministerio de Salud en el año 2007 se registraron 6.315 casos, se atendieron 13.605 casos el año 2008 y más de 14.600 casos el año 2009, localizados mayormente en la región norte y la selva del país.

Sin embargo, datos más recientes confirman esta tendencia. La Organización Panamericana de la Salud señala que en el Perú se han presentado 37.837 casos de dengue, 123 casos de dengue hemorrágico y 52 muertes, presentándose los cuatro serotipos DEN 1, 2, 3 y 4 (OPS, 2016). Igualmente, según la misma referencia, hasta la 51 Semana Epidemiológica, se han presentado 28.623 casos de dengue, 115 casos de dengue hemorrágico y 38 muertes, con los serotipos DEN 1, 2 y 3.

Tanto el dengue clásico (no grave) como el dengue hemorrágico (grave), presentan los mismos síntomas durante los primeros días. Pero es recién entre el cuarto y sexto día de enfermedad en que es posible detectar los signos de alarma indicativos de un posible dengue hemorrágico, que es cuando se agravan los síntomas, se presenta una baja abrupta de fiebre, dolor abdominal intenso, sangrado en nariz, piel, encías y otras partes del cuerpo. De hecho, para prevenir esta circunstancia, la Organización Mundial de la Salud aprobó el año 2009 una nueva clasificación revisada del dengue como son: dengue sin signos de alarma, dengue con signos de alarma y dengue grave (OPS, 2009).

Es importante mencionar que no existe vacuna contra la enfermedad. La única forma conocida de prevención es impedir la presencia del mosquito transmisor *Aedes Aegypti* en las viviendas y cerca de ellas, debiendo eliminar todos los posibles criaderos de huevos. Por eso, ante la presencia de los primeros síntomas, lo más importante es acudir cuanto antes al centro de salud más cercano, a fin de que se tomen las medidas más apropiadas del caso.

Sin embargo, en los centros de salud de San Martín, los responsables de la atención médica no cuentan con tecnologías de información que permitan descartar en forma rápida y oportuna la presencia de la enfermedad del dengue en los pacientes, así como, diferenciarla de otras

enfermedades como la chikungunya y el zika de reciente aparición, así como la fiebre amarilla de la selva, que presentan síntomas similares, situación que muchas veces agudiza el cuadro clínico de los pacientes incrementando el riesgo de mortalidad y, a su vez, impidiendo el control en la propagación del virus y la enfermedad.

Así, frente a la necesidad de contar con un diagnóstico oportuno y correcto del dengue sin alarma, con alarma y grave, los centros de salud de la región San Martín no cuentan con laboratorios especializados ni con tecnologías de información que permitan diferenciar estas enfermedades de otras con síntomas similares y tomar las acciones preventivas del caso, evitando poner en riesgo la salud de la población.

Así como los centros de salud no cuentan con laboratorios ni con tecnologías de información adecuados para el tratamiento del dengue, en muchos casos no cuentan con médicos para atender las arbovirosis y no cuentan con personal técnico capacitado para atender cualquier emergencia viral.

Es preciso señalar que, en la actualidad, existen una serie de tecnologías de información en el campo de la Inteligencia Artificial (IA) como las Redes Neuronales Artificiales (RNA), Sistemas Expertos, Algoritmos Genéticos, Lógica Difusa (Fuzzy Logic), etc., que se han venido aplicando con gran éxito en diversas áreas de la medicina humana. Cada una de ellas constituye estrategias inteligentes, toda vez que imitan alguna forma de la inteligencia y comportamiento humano, para el diagnóstico y la predicción del dengue hemorrágico. Paletta (2010) define a la inteligencia artificial como “la ciencia de emular procesos propios de la inteligencia humana tales como la percepción, el razonamiento y el aprendizaje, haciendo uso de modelos no naturales” (p.15).

Se han realizado algunos estudios que utilizan algún sistema de inteligencia artificial para el diagnóstico y predicción del dengue, como el estudio realizado en Singapur por Aburas, Cetiner y Sari (2010), quienes refieren que un modelo de redes neuronales artificiales refleja que factores como la temperatura media, humedad relativa y precipitación total fue “muy eficaz para predecir el número de casos confirmados de dengue”. (p.1). Como se puede observar, en el presente caso el objeto de estudio son las características ambientales de temperatura, humedad y precipitación que predicen el número de casos de dengue.

Igualmente, otro estudio da cuenta que el uso de redes neuronales artificiales permite predecir a partir de síntomas y signos clínicos el día de la defervescencia (disminución de la fiebre) en pacientes con dengue con un 90% de precisión (Ibrahim, Nasir, Abu, Chong y Sulaiman, 2005).

Sin embargo, no permite discriminar entre pacientes con diversas enfermedades virales con hipertermia y el dengue.

También se han hecho aplicaciones de algoritmos genéticos para investigar el crecimiento poblacional del vector del dengue, el mosquito *Aedes Aegypti*, para estudiar el efecto de la inserción de machos estériles en el medio ambiente para controlar la población y prevenir la enfermedad, como los estudios de Barsante, Cardoso y Acebal (2011). En el mismo sentido se orientan los estudios de Lyra (2014), pero con la variante de comparar dos modelos de control de la población de vectores, a decir, el control biológico con el lanzamiento de mosquitos estériles en el entorno y el control de la química del insecticida. En ambos casos, el algoritmo genético se utiliza para el control de la población de mosquitos, pero no para diagnosticar la enfermedad a partir de los síntomas.

La lógica difusa también se emplea para analizar imágenes satelitales, aportados por Sistemas de Información Geográficas (SIG), a fin de localizar zonas que pueden indicar una mayor presencia de reproducción del *Aedes Aegypti*, tal como se da cuenta en los estudios de Torres (2011), en el estado de Colima - México, en la que concluye que “el algoritmo difuso puede identificar zonas consideradas como probable foco de infección para la reproducción del mosquito causante del dengue”, (p.122).

Un aporte importante es sin duda el realizado por Carlos y Jesús (2015), con el diseño de un sistema experto basado en lógica difusa para la detección de la influenza y el dengue, utilizando plataformas móviles como un Smartphone. Sin embargo, el estudio no presenta rigurosidad en el manejo de los síntomas y no acredita un nivel de confiabilidad del sistema.

Estudios más recientes, Mussumeci y Codeco (2020), dan cuenta de los esfuerzos de pronóstico del dengue utilizando modelos de aprendizaje automáticos, que incorporan la selección de características como LASSO y la regresión de bosque aleatorio con LSTM, una red neuronal recurrente profunda, para pronosticar la incidencia semanal del dengue en 790 ciudades de Brasil.

De igual manera, se han hecho esfuerzos como en Makasaar (Indonesia), utilizando modelos espacio-temporales bayesianos apropiados (Aswi, Cramb, Duncan, Hu, White y Mengersen, 2021), que permite la identificación de diferentes grupos de áreas y el impacto de las covariables climáticas que pueden ayudar a conocer patrones espaciales y climáticos del dengue, para facilitar las decisiones políticas.

De acuerdo al Centro Nacional de Epidemiología, Prevención y Control de Enfermedades (MINSA, 2020), a nivel nacional, en el Perú, entre los años 2014 al 2020, se presentaron los siguientes casos de dengue, tal como se muestra en la Figura 1.

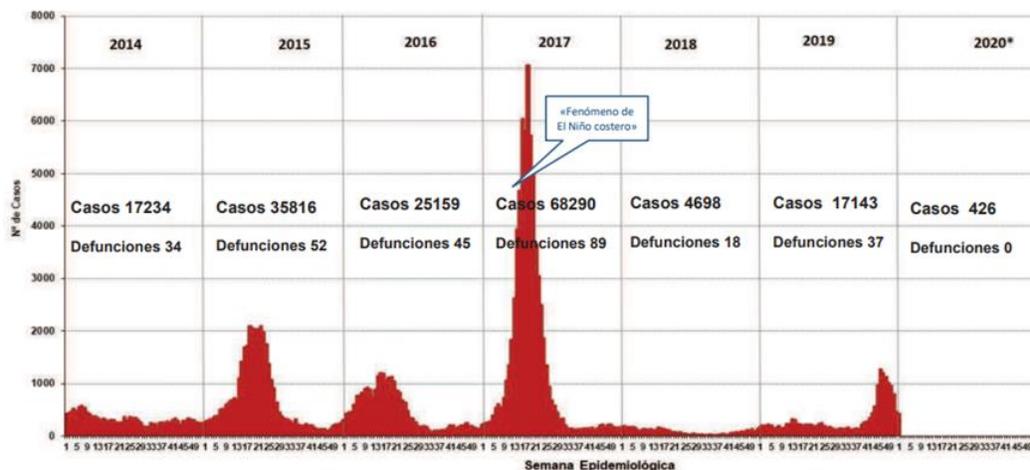


Figura 1. Número de casos de Dengue, Perú 2014 – 2020\*. Fuente: MINSA, 2020.

En el periodo observado a nivel nacional, se da cuenta que el año 2017 se reportó el mayor número de casos de dengue, ascendente a 68 290 casos y 89 defunciones. Según el diario El Comercio de Lima, durante “todo el 2020 se presentaron 38 879 casos de dengue”. Mientras tanto, en la región San Martín, se presentaron los siguientes casos de dengue (ver Tabla 1):

Tabla 1

Casos de dengue en la región San Martín periodo 2013-2017\*

Año	Confirmados	Probables	Total
2013	118	1090	1208
2014	503	1071	1574
2015	64	156	220
2016	24	422	446
2017	26	55	81
<b>Total</b>	<b>735</b>	<b>2794</b>	<b>3529</b>

Fuente propia. (\*) Los datos corresponden a la SE 05 del 2017.

Se observa que, entre los años 2013 y 2017 (SE 05) se presentaron un total de 3.529 casos de dengue en la región San Martín, de los cuales 735 fueron confirmados y 2.794 se refirieron como probables; eso quiere decir que del total de casos reportados sólo el 21% fueron confirmados como dengue y el 79% restante correspondían a otro tipo de enfermedades que presentaban síntomas similares pero que luego de las observaciones y pruebas correspondientes se descartaron como dengue.

Por otro lado, se conoce que el 69% del área de la región San Martín está infestada por *Aedes Aegypti* (35.312,62 Km<sup>2</sup>), el 61% de la población reside en estas áreas de riesgo (449.241 habitantes) y que el vector está presente en al menos 70 localidades distribuidas en 53 distritos de 77 existentes y que 107.653 viviendas estaban en riesgo (MINSA, 2011).

De acuerdo a datos recientes, entregados por el Dirección Regional de Salud de San Martín (DIRESA) para el presente estudio, entre los años 2017 al 2019, se presentaron en la región San Martín 807 casos de dengue, de los cuales fueron confirmados 397 casos. Igualmente, de conformidad con la información publicada por el diario La República, del 2 de marzo del 2020, que pone como fuente al Centro Nacional de Epidemiología, Prevención y Control de Enfermedades del MINSA, en los primeros meses del año 2020, se han reportado 10 228 casos de Dengue a nivel nacional, concentrados en su gran mayoría en las regiones amazónicas de Madre de Dios, Loreto y San Martín, de los cuales 4 950 reportaron diagnóstico positivo, registrándose 15 casos de muerte, seis en Madre de Dios, ocho en Loreto y uno en San Martín.

Por lo expuesto, resulta pertinente plantear las siguientes interrogantes:

¿Qué modelo de diagnóstico del Dengue basado en Inteligencia Artificial se requiere en la región San Martín – Perú?

¿Cuál es el estado del arte del diagnóstico del dengue utilizando inteligencia artificial?

¿Cuáles son las características sintomáticas del dengue sin alarma, con alarma y grave?

¿Qué técnica informática de inteligencia artificial utilizar para el diagnóstico del dengue?.

¿Qué prototipo implementar como técnica informática de inteligencia artificial?

### **Objetivo general**

Diseñar e implementar un prototipo de diagnóstico del dengue basado en inteligencia artificial, para ser aplicada en la región San Martín – Perú.

## **Objetivos específicos**

1. Elaborar el estado del arte del diagnóstico del dengue utilizando inteligencia artificial.
2. Determinar la técnica informática de inteligencia artificial adecuada para el diagnóstico del dengue.
3. Identificar las características sintomáticas del dengue sin alarma, con alarma y grave.
4. Realizar el análisis de datos asociados al dengue sin alarma, con alarma y grave.
5. Proponer el prototipo óptimo para el diagnóstico del dengue en la región San Martín.

## **Justificación**

El presente trabajo de investigación, es el diseño de un modelo de diagnóstico del dengue basado en inteligencia artificial, en la región San Martín del Perú, en sus diversas variantes como son el dengue sin alarma, con alarma y grave, en forma rápida y oportuna, a fin de evitar los efectos nocivos de la enfermedad viral producida por el vector *Aedes Aegypti* en la población afectada.

En ese sentido, la implementación del estudio beneficiará en primer orden a la población de la región San Martín que, a junio del 2020, se proyecta a 840 520 habitantes, así como, indirectamente a otros departamentos de la Amazonía como Loreto y Madre de Dios, que recurrentemente son víctimas del dengue por un mal diagnóstico, factor limitante para un tratamiento adecuado y oportuno, cuya población, en conjunto, se proyecta para el mismo periodo a 2 065 455 habitantes (INEI, 2018).

Igualmente, el estudio está dirigido a los profesionales médicos, epidemiólogos y técnicos de los hospitales, clínicas y centros de salud de la región San Martín, quienes podrán contar con una herramienta informática basada en inteligencia artificial de apoyo, útil para diagnosticar el dengue en sus diferentes manifestaciones, tomando las medidas preventivas y de atención más adecuadas garantizando la salud y bienestar de los pacientes.

La Universidad Nacional de San Martín – Tarapoto, podrá contar con tecnología, conocimiento y experiencia en inteligencia artificial, particularmente las redes neuronales artificiales, para la formación académica de los estudiantes de la Facultad de Ingeniería de Sistemas e Informática, así como, de las facultades de Medicina Humana y Enfermería.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **Antecedentes del problema**

San Martín (2010) analiza el impacto del dengue clásico y del dengue hemorrágico en el continente americano entre los años 1980 al 2007. De acuerdo al informe, se presentaron 8.491.416 casos, de los cuales solo 2.359 casos se presentaron en América de Norte y el resto, es decir, 8.489.057 se registraron en América Latina representando el 99.9% de casos. Del total registrado 8.307.875 corresponden al dengue clásico y 183.541 al dengue hemorrágico. Lamentablemente 2.210 casos fueron con consecuencias fatales. Eso significa que se han presentado en promedio 307.699 casos por año. Sin embargo, en informes recientes de la OPS/OMS se han presentado el 2015 un total de 2.439.527 casos en América Latina, y, hasta el 28 de octubre del 2016, SE 41, 2.094.855 casos; lo que significa un incremento de 700% respecto al promedio anual.

La Organización Panamericana de la Salud informa que, el año 2016, a nivel subregional se registraron los siguientes casos de dengue: Cono sur (1.599.505), subregión Andina (195.896), México y América Central (231.684) y los países del Caribe (67.358). (OPS, 2016).

En lo que respecta al dengue hemorrágico el mayor impacto se traslada a la subregión Andina (1.219), seguido de México y América Central (1.122), países del Caribe (613) y el Cono Sur (833). Considerando que el dengue hemorrágico se produce como resultado de una segunda picadura del mosquito *Aedes Aegyptis* es previsible que su impacto en el Cono Sur varíe ostensiblemente. En total se presentaron 912 defunciones.

A nivel de países, sin duda Brasil es el que ha reportado el mayor número de casos de dengue clásico con 1.452.484 de hechos registrados, que representa el 69.35% en América Latina, seguido

de México (99.759) y Colombia (94.012). La República Bolivariana de Venezuela alcanza 32.474 y Perú como ya se indicó líneas arriba 28.623 casos reportados, que mantiene una tendencia ascendente dado los antecedentes del año 2007 (6.315) casos, el 2008 (13.605) y el 2009 (14.600); en siete años se ha duplicado el número de casos.

Históricamente, el mayor impacto del dengue hemorrágico, desde 1980 al 2007, se ha registrado en Venezuela (64.437), Colombia (50.194), México (25.288) y Honduras (11.158). Brasil, Colombia, Honduras y República Dominicana presentan los mayores indicadores.

En cuanto a la evolución de la epidemia, se puede observar que en la década de 1980 la mayoría de los países registraba indicadores inferiores a los 50/100.000 habitantes, sólo Cuba registraba un indicador superior a los 200/100.000, Paraguay de 100 a 200/100.000 y Nicaragua de 50 a 100/100.000. En la década de 1990 la situación cambia dramáticamente, pues, Venezuela, Colombia y Brasil se ubican entre los 50 a 100/100.000 habitantes, Nicaragua se incrementa a 200/100.000, así como también Surinam.

Durante los años 2000 al 2007, el cuadro se agudiza, la enfermedad se masifica en Brasil, Surinam y la Guayana Francesa, así como, Costa Rica y Honduras, que pasan a 200/100.000 habitantes afectados. Venezuela, Colombia, Surinam y Paraguay se ubican entre 100 a 200/100.000 habitantes, mientras que Ecuador y Bolivia entre 50 a 100/100.000 afectados.

En general, tal como se observa en la Figura 2, tanto el dengue clásico como el dengue hemorrágico presentan una tendencia de crecimiento exponencial en América Latina, lo cual representa un desafío para evitar la propagación epidémica de la enfermedad.

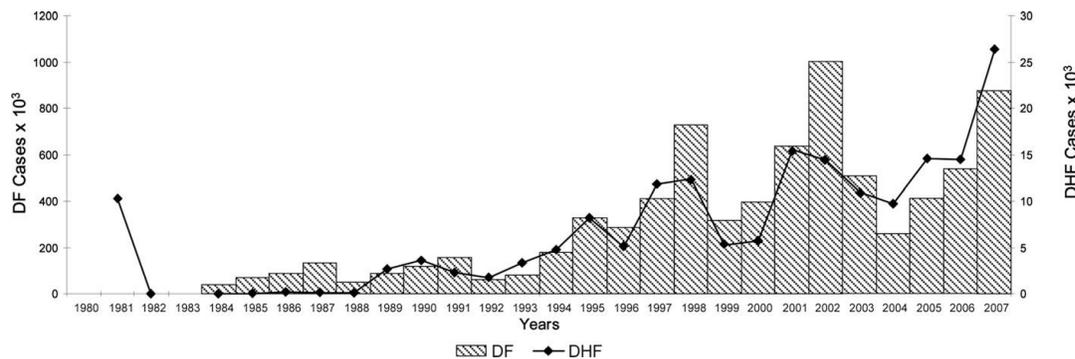


Figura 2. Evolución del dengue clásico y el dengue hemorrágico de 1980 al 2007. Fuente: OPS, 2009.

Una vez más, la Organización Panamericana de la Salud da cuenta del impacto del dengue y el dengue hemorrágico en América Latina, de enero al 01 de julio del 2011. Ahí se informa que el impacto del dengue clásico en América del Norte registra 412 casos, frente al Cono Sur (1.599.505), subregión Andina (195.896), América Central (231.684) y los países del Caribe (67.358). El Perú registra 28.623 casos y Venezuela 32.474, más que Perú. (OPS, 2011).

Igualmente, en lo que respecta al dengue hemorrágico se da la siguiente información: No se registra ningún caso en Norteamérica, Cono Sur (833), subregión Andina (1.219), América Central (1.122) y los países del Caribe (613).

En el periodo analizado se han registrado 912 fallecidos por efectos del dengue hemorrágico, de los cuales 607 se reportan en el Cono Sur y 233 en los países Andinos: 38 en Perú y 30 en Venezuela.

En lo que respecta al Perú, Cabezas (2005) refiere que el vector transmisor *Aedes Aegypti*:

Fue eliminado del Perú en 1956, pero reingresó en 1984 trayendo consigo al dengue, cuya manifestación máxima ocurrió en forma explosiva en 1990 al presentarse el dengue clásico, debido al serotipo 1 del virus dengue, en las principales ciudades de nuestra Amazonía... a partir de entonces, somos testigos de su inexorable expansión por las ciudades de la costa norte y de la Amazonía, en las cuales han circulado en los últimos años, los cuatro serotipos del virus, y han aparecido en ambas regiones, casos de dengue hemorrágico... complicando este escenario, el *Aedes Aegypti* reapareció el año 2000 en Lima, pues también estuvo desde la Colonia hasta el siglo XIX y fue el transmisor de la fiebre amarilla urbana. (p.4).

Más adelante, se establece que: “Estando el vector *Aedes Aegypti* en Lima y existiendo entre los inmigrantes, personas infectadas por el virus del dengue en etapa de viremia, era previsible la infección del vector y su transmisión a pobladores residentes en Lima, lo cual ocurrió entre marzo y abril de 2005 en Comas, un distrito densamente poblado, al norte de esta metrópoli que es Lima” (Cabezas, 2005, p.4).

Por otro lado, entre los meses de abril y mayo del 2005, se registraron 1.385 casos de pacientes en Lima, de los cuales 479 (34.6%) residían en el populoso distrito de Comas. De las 315 personas incluidas en el estudio, 142 fueron de sexo masculino (44.9%) y los síntomas más frecuentes fueron fiebre (89.6%), cefalea (86.7%), mialgias (75.0%), artralgias (69.6%), dolor abdominal (60.1%) y dolor ocular (54.1%) (Juárez, 2005).

De conformidad con las estadísticas del Ministerio de Salud del Perú, se detectaron 16.842 casos de dengue el 2010; 28.084 el 2011; 28.505 el 2012; 13.076 el 2013; 17.284 el 2014; 35.837 el 2015 y en lo que va del 2016, al 28 de octubre, 28.623 casos.

Asimismo, de acuerdo a la Dirección General de Epidemiología, en las primeras semanas del año 2015, de manera específica hasta la semana quince (SE 15), la mayor intensidad de dengue se detectó en las regiones de Piura (3,479), Tumbes (2,506) y Loreto (1,425). En tanto que en la región San Martín se observaron 442 casos, 82 confirmados y 360 probables (MINSA, 2017).

### **Predicción de enfermedades tropicales**

En el estudio de Castillo (2009), se describe un modelo de minería de datos para centros de salud que se dedican al tratamiento, cura y prevención de enfermedades transmitidas por vectores artrópodos como el Aedes 2005 en; el objeto fundamental del modelo es determinar y predecir los casos y condiciones por las cuales sucedió la enfermedad, todo ello debido al crecimiento en número de pacientes con enfermedades tropicales y de la detección de nuevas enfermedades infectocontagiosas; existiendo la necesidad de crear un sistema que permita llevar un control automatizado de los pacientes.

La investigación realizada concluye que muchos son los factores que ayudan a la reproducción del mosquito del dengue y paludismo, por lo que otras deben ser las medidas de prevención y campañas de capacitación.

### **Control de la malaria en Venezuela**

Según Delgado (2003) existe la reemergencia a nivel mundial de enfermedades transmitidas por vectores, tal como la malaria, el dengue, la encefalitis y otras, está asociada a múltiples factores de diversa índole, que abarcan desde los factores políticos y sociales hasta los ambientales. Esta naturaleza multifactorial ha llevado a los investigadores a obtener la mayor información posible sobre la dinámica espacio-temporal de estas endemias y a trabajarlas bajo un enfoque sistémico.

El referido autor sostiene el volumen de datos que estas investigaciones generan no pueden manejarse con las técnicas tradicionales, tal volumen de información sobre las variables relevantes demanda una metodología que permita la integración de la información geoespacial con los datos que cuantifiquen las variables y permita el análisis espacial como una forma de identificar patrones

que solo tiene sentido en su expresión espacial a diferentes niveles de resolución o abstracción que requiere su interpretación.

Asimismo, considera importante que el uso combinado de tecnologías digitales que incluyen modelos de terreno de tres dimensiones en formato digital (MDT), el procesamiento digital de imágenes satelitales a través de sus píxeles constituyentes (PDI), los sistemas de posicionamiento global por satélite (GPS) y la integración y análisis final de esta información con los Sistemas de Información Geográfica (SIG), constituyen poderosas herramientas de trabajo para lograr un control sobre estas enfermedades.

Una de las conclusiones del estudio antes identificado es aquella que se refiere al procesamiento de imágenes satelitales y los modelos digitales de terreno que facilitan la detección de variables a nivel “gran visión”, conjuntamente con su integración en SIG que permiten la localización y descripción de los hábitats de los vectores y se han convertido en herramientas indispensables en el soporte para la toma de decisiones en el área de salud pública.

### **Redes neuronales artificiales**

Estudios realizados por Montaña (2002), refieren sobre el enorme potencial que tienen las Redes Neuronales Artificiales (RNA), como una nueva y eficaz metodología para la descripción, prevención y predicción de la conducta adictiva de las personas, confirmando la validez de las redes hetero asociativas para diagnosticar y predecir el estatus de un sujeto como adicto o no adicto con un margen de error pequeño.

Del mismo modo, Montaña (2002) concluye que una RNA es capaz de predecir el consumo de éxtasis a partir de las respuestas dadas a un cuestionario, con un grado de eficacia de 96.66%; lo cual significa que conociendo las respuestas del sujeto a 25 preguntas se puede anticipar si ese sujeto es consumidor o no de éxtasis, con un margen de error muy pequeño. Lo que coloca a esta estrategia inteligente como una herramienta capaz de diagnosticar y predecir determinadas conductas adictivas.

Otro aspecto estudiado por Montaña (2002) es si las redes neuronales jerárquicas son más precisas que el modelo de Cox y si el modelo de redes neuronales secuenciales supone una mejora respecto al modelo de redes neuronales jerárquicas, arribando a la conclusión de que el modelo de redes secuenciales tiene una resolución significativamente mejor que el modelo de Cox y que, el

modelo de redes secuenciales no ha supuesto una mejora en rendimiento respecto al modelo de redes jerárquicas.

Finalmente, a partir de los estudios de Montaña (ob. cit.) se demuestra que una RNA “no es una caja negra”, sino un instrumento de predicción capaz de proporcionar claves acerca de las variables que están determinando la estimación realizada por el modelo.

Muñoz, Espinoza y Rivera (2012) concluyen que las redes neuronales artificiales constituyen una alternativa tecnológica computacional para el diagnóstico de la anemia ferropénica y que, el modelo de 27 factores de entrada, dos capas ocultas con 40 y 27 neuronas respectivamente y una capa de salida, es un sistema válido que tiene una certeza de 99% en sus resultados.

### **Bases teóricas**

En este apartado, abordamos los conceptos teóricos más importantes que han sido materia de nuestro estudio, tales como los sistemas inteligentes y las redes neuronales artificiales.

#### **Sistemas inteligentes**

Como refiere Paletta (2007), software inteligente, sistemas inteligentes e inteligencia artificial son algunas de las maneras de identificar todos aquellos programas de computación que muestran algún tipo de “inteligencia”, cualidad inteligente a la que asocia el conocimiento y la decisión, que permiten entender el paso de tratar de emular el concepto de inteligencia de manera artificial, es decir, mediante el uso de máquinas, principalmente computadoras. (p.77, 79).

Los requisitos generales de un sistema inteligente son los siguientes:

- a) El sistema debe ser capaz de aprender. - Esto implica que el sistema debe tener un recuerdo para el aprendizaje que se mantenga.
  - a.1 Tiene que ser capaz de aprender de sus propias experiencias:
    - 1 El aprendizaje a través de ensayo y error.
    - 2 El aprendizaje a través de la observación.
    - 3 El aprendizaje a través del activo razonamiento.
  - a.2 Debe ser capaz de aprender de la instrucción.
  
- b) El sistema debe ser autónomo. - Es decir, debe ser capaz de hacer las cosas por sí mismo.

- 1 El sistema debe ser capaz de afectar el medio ambiente sobre la base de sus propias conclusiones independientes.
  - 2 El sistema debe ser su propio maestro y, por lo tanto, no requerir de la intervención del operador para funcionar.
  - 3 El sistema debe ser motivado. Debe tener necesidades y requerimientos que pueden ser satisfechos por sus propias acciones.
- c) El sistema debe ser capaz de razonar. Es decir, debe utilizar algún tipo de razonamiento basado en hechos conocidos y capaces de producir las ideas que más tarde se hacen hechos conocidos.
- d) El sistema debe ser capaz de desarrollar su autoconciencia. Esto no sólo se relaciona con la autonomía, el razonamiento y el aprendizaje, sino también con la consideración de la necesidad de los sentidos internos y externos.

En suma, un Sistema inteligente es aquél que es capaz de percibir, razonar y actuar (ver Figura 3). Razonar se funda en el conocimiento y actuar implica un nivel de decisión.

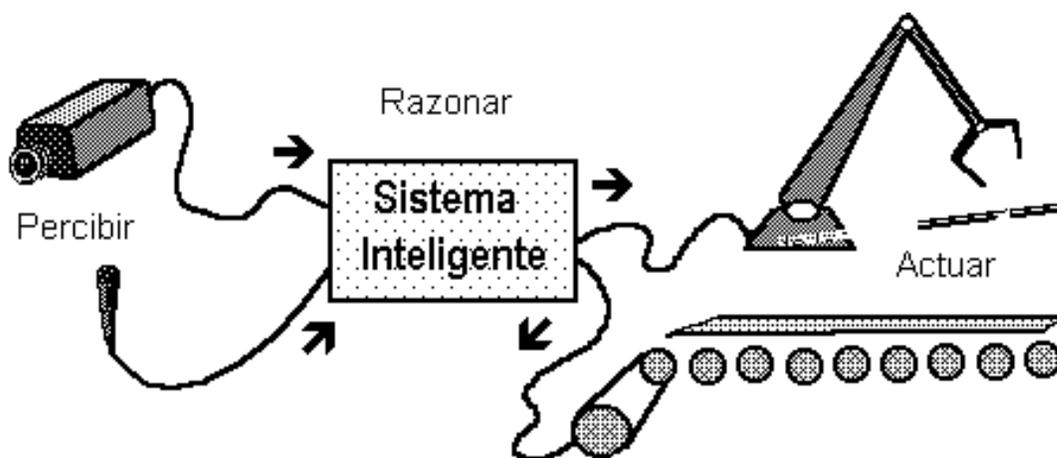


Figura 3. Esquema de un sistema inteligente que percibe, razona y actúa.

## **Inteligencia Artificial**

Definir que es inteligencia artificial (IA) no es ni ha sido tarea fácil, según Hangeland (1988), el epítome de todo el drama es la inteligencia artificial, ese nuevo y apasionante esfuerzo por hacer pensar a las computadoras. Pensar, es una facultad o habilidad propia del ser humano.

Pero, ¿Qué es la inteligencia artificial?. Carretero (1989) entiende que es el estudio de los procesos cognitivos, lo que se puede resumir en memorización, aprendizaje, solución de problemas, inferencia y deducción lógica, percepción y reconocimiento de formas, toma de decisiones y comprensión del lenguaje natural. Por su parte Hayes (1973) considera que la IA es el estudio de la inteligencia como proceso, por medio de los cuales se pueden generar comportamientos inteligentes, por lo que el objetivo de esta disciplina no son los programas o el propio ordenador sino la conducta inteligente. O sea, de un lado tenemos los procesos cognitivos y por otro los comportamientos inteligentes, dos niveles u objetos de estudio.

De otro lado, Boden (1984) manifiesta que la IA no es el estudio de las computadoras y que estas son máquinas metálicas de interés intrínseco para la ingeniería electrónica, sino el uso de programas de computadoras y de técnicas de programación para emular la inteligencia humana. Es decir, no son los ordenadores ni la programación de los ordenadores, sino ambas interactuando para lograr una conducta inteligente. Mientras Rich (1985) dice que la IA es cualquier cosa que los ordenadores aún son incapaces de hacer, se refiere a situaciones diferentes a las operaciones numéricas, el almacenamiento y el procesamiento de información.

¿Pero qué cosas pueden hacer los humanos mejor que los ordenadores?. Amador (1996) responde que todas aquellas acciones que comportan inteligencia, toda vez que el hombre va más allá del simple procesamiento de la información ya que es capaz de entenderla.

Minsky (1988) sostiene que la IA se dedica a programar ordenadores de forma que realicen tareas que, si fuesen llevadas a cabo por un ser humano exigiría inteligencia por parte de la persona que la ejecuta. Una vez más la programación y los ordenadores aparecen juntos para lograr una conducta inteligente.

Finalmente, Paletta (2010) dice que la IA es “la ciencia de emular procesos propios de la inteligencia humana tales como la percepción, el razonamiento y el aprendizaje, haciendo uso de modelos no naturales” (p.15). Entendemos por emular al hecho de imitar procesos y por modelos no naturales a la acción humana utilizando herramientas creadas por el propio hombre como las

máquinas y los ordenadores. La IA entonces, imita por un lado los procesos cognoscitivos y por el otro el comportamiento humano que implica algún nivel de inteligencia.

## Redes neuronales artificiales

Las Redes Neuronales Artificiales (RNA) o sistemas conexionistas son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas (Hilera y Martínez, 1995).

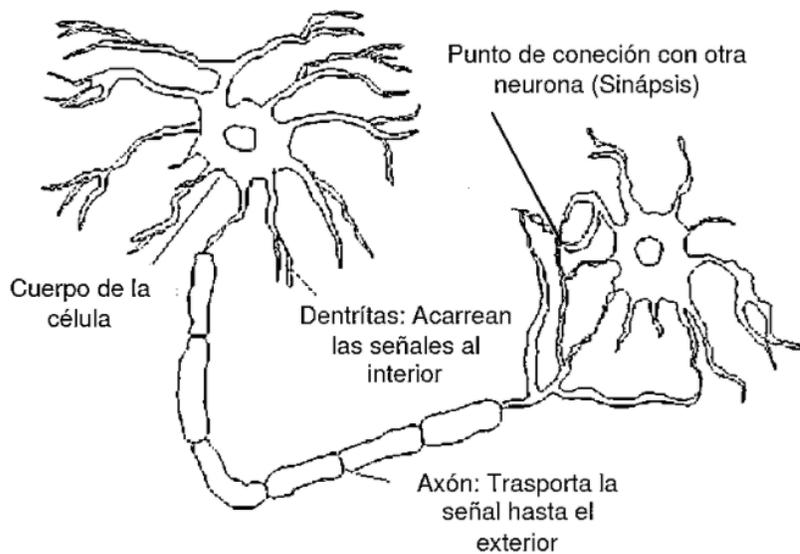


Figura 4. Esquema de una neurona biológica. Copyright © 2013: Rojas, J.

La Figura 4 muestra un modelo simple de una neurona biológica. Las dendritas y el cuerpo celular reciben señales de entradas excitatorias o inhibitorias de las neuronas vecinas; el cuerpo celular las combina e integra y emite señales de salida. El axón transporta esas señales de salida a los terminales axónicos, que se encargan de distribuir información a otras neuronas. Una neurona recibe información de miles de neuronas y, a su vez, transmite información a miles de neuronas.

Desde un punto de vista técnico, las RNAs consisten en:

Un gran número de elementos simples de procesamiento llamados nodos o neuronas que están organizados en capas. Cada neurona está conectada con otras neuronas mediante enlaces de comunicación,

cada uno de los cuales tiene asociado un peso. Los pesos representan la información que será usada por la red neuronal para resolver un problema determinado. (Montaño, 2002, p.120).

Por su parte, la neurona artificial (ver Figura 5), recibe las señales de entrada  $X_j$  de las neuronas vecinas ponderada por los pesos de las conexiones  $W_{ij}$ . La sumatoria de estas señales ponderadas proporciona la entrada total o neta de la neurona, y, mediante la aplicación de una función matemática denominada función de salida  $f()$  sobre la entrada neta, se calcula su valor de salida  $Y_i$ , el cual es enviado a otras neuronas.

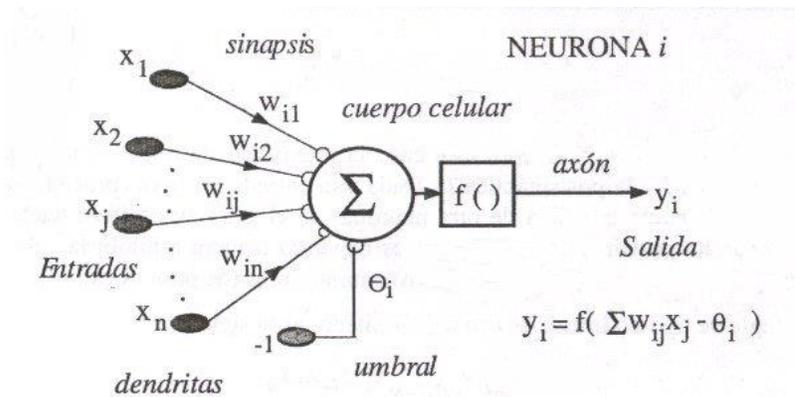


Figura 5. Esquema de neurona artificial. Copyright © 2005: Palacios, F.

Los valores de salida proporcionan un resultado en función de las variables de entrada. Desde el punto de vista de la implementación práctica, los nodos son elementos computacionales simples que emulan la respuesta de una neurona a un determinado estímulo. Estos elementos, como las neuronas en el sistema nervioso, funcionan como interruptores: Cuando la suma de señales de entrada es suficientemente alta (en el caso de una neurona se diría que se acumula suficiente neurotransmisor), la neurona manda una señal a las neuronas con las que mantiene contacto (se genera un potencial de acción).

Esta situación se modela matemáticamente como una suma de pesos de todas las señales de llegada al nodo que se compara con un umbral característico, haciendo uso de ecuaciones de activación como la sigmoidea, tangente hiperbólica y lineal, que se muestran más adelante. Si el umbral se supera, entonces el nodo se dispara, mandando una señal a otros nodos, que a su vez procesarán esa información juntamente con la que reciben de nodos adyacentes. Evidentemente,

la respuesta de cada nodo dependerá del valor de las interacciones con los nodos precedentes dentro de la estructura de la red.

Función sigmoidea:

$$f(x) = \text{log sig} = \frac{1}{1 + e^{-x}}$$

Función tangente hiperbólica:

$$f(x) = \text{tan sig} \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Función lineal:

$$f(x) = \text{purelin} = x$$

Como en el caso del sistema nervioso, el poder computacional de una RNA deriva no de la complejidad de cada unidad de proceso sino de la densidad y complejidad de sus interconexiones. Estas complejidades, según Trujillano, March y Sorribas (2004) se han ido resolviendo progresivamente desde:

La primera implementación práctica de estas ideas se describe en los trabajos de McCulloch y Pitts en 1946. A partir de este punto, algunos de los hitos principales en la investigación de este tipo de técnicas fueron: El diseño por Widrow y Hoff (1961) de la red conocida como *Adaline* (capaz de resolver problemas de regresión lineal), el desarrollo de la red con estructura de *perceptrón simple* en 1959 (con equivalencia al análisis discriminante y regresión logística) y las redes *multicapa* por Rosenblatt en 1986 (que permiten la resolución de situaciones no lineales). Por otra parte, los trabajos teóricos de Bishop y la aportación sobre redes autoorganizadas de Kohonen dotaron de fundamentos formales a este tipo de técnica. (p.60-61).

## Arquitectura de las redes neuronales artificiales

Una neurona artificial tiene baja capacidad de procesamiento, de hecho, por si sola es capaz de resolver algunos problemas en la que las variables se relacionan linealmente, pero se muestran torpes e incapaces cuando las variables no son lineales o la información es incompleta. Por eso, su

verdadero potencial radica en la interconexión de las mismas formando capas sucesivas, lo que ha motivado el desarrollo de diversas arquitecturas.

Veamos algunos conceptos claves. Dependiendo del número de capas, se pueden clasificar en monocapa y multicapa; de otro lado, si se toma en cuenta el modo en que fluye la información las podemos separar en feedforward y recurrentes.

### **Redes monocapa.**

Esta arquitectura muy simple, solo tiene una capa de procesamiento de información, de allí su nombre Monocapa, hasta allí fluye la información desde las entradas siendo transferidas a otro nivel de salida, modificados por los respectivos pesos sinápticos. Es preciso señalar que todas las entradas están conectadas con todas las neuronas de salida, es decir, existe una conectividad total entre entradas y salidas. Ergo, si tenemos  $n$  entradas y  $m$  salidas, se producirá una matriz de conexión  $m \times n$ . La información fluye desde la entrada hacia la salida.

La primera RNA conocida fue el Perceptrón, presentado en 1958 por el psicólogo Frank Rosenblatt, despertando gran interés por su capacidad de reconocer patrones sencillos con una superficie de separación lineal entre variables. Según Caicedo y López (2017) este perceptrón es:

Una red monocapa pues la capa de entrada, al no realizar procesamiento alguno sobre la señal de estímulo, no se tiene en la cuenta y, por tanto, esta red posee solo una capa de procesamiento que es la misma de salida. Posee conectividad total, ya que la neurona de la capa de salida está conectada a todas las unidades o neuronas de entrada... La función de activación que utiliza la neurona de salida es el tipo escalón binario  $[0, +1]$  o bipolar  $[-1, +1]$ . (p.38).

El perceptrón de Rosenblatt tiene una sola neurona de salida, a la que convergen todas las neuronas de entrada, que realiza la suma ponderada de las entradas multiplicada por sus respectivos pesos de conexión ( $x_i w_i$ ) a las que se suma un valor de umbral ( $\theta$ ), el resultado se pasa a una función de activación binaria o bipolar. Si es binaria la salida será 1 ó 0 (ver Figura 6).

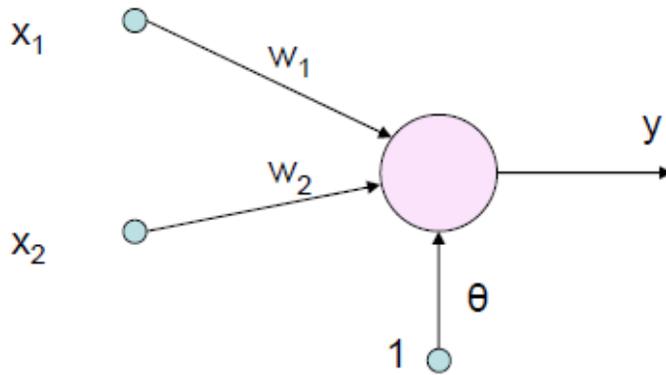


Figura 6. Perceptrón con una capa de salida. Copyright © 2012: Carcini, M.

La notación matemática se representa de la siguiente manera:

$$Y = x_1 * w_1 + x_2 * w_2 + \theta$$

Las funciones de transferencia o de activación con los que opera el perceptrón son el limitador fuerte (hardlim) y el limitador fuerte simétrico (hardlims), que se muestran seguidamente en las figuras 7 y 8:

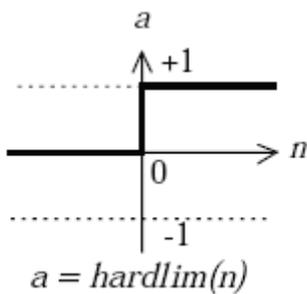


Figura 7. Limitador fuerte.  
Copyright © 2012: López-Juarez, I.

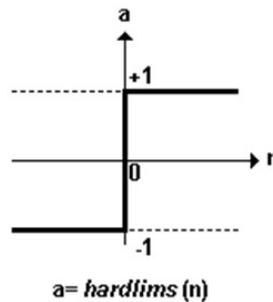


Figura 8. Limitador fuerte simétrico.  
Copyright © 2012: Hugo.

El cálculo del error del p-ésimo patrón de entrada es importante, lo que se calcula de acuerdo a la siguiente ecuación:

$$e_p = \frac{1}{2} \sum_{j=1}^M (d_{pj} - y_{pj})^2$$

Donde:

$d_{pj}$  : Valor de salida deseada de la j-ésima neurona.

$Y_{pj}$  : Valor de salida de la j-ésima neurona.

Sin embargo, el perceptrón tiene una gran limitación en su incapacidad para separar regiones que no son linealmente separables. Es idóneo para resolver funciones AND u OR, pero tiene limitaciones insuperables cuando se trata de funciones tipo XOR o no lineales, situación que si puede ser resuelta por redes multicapa.

### **Redes multicapa.**

A diferencia de las redes monocapa, las redes multicapa tienen capas de neuronas intermedias llamadas “ocultas” entre los niveles de entrada y la capa de salida, las que pueden ser más de una e incluso muchas capas como es el caso de las redes Deep Learning (aprendizaje profundo), de amplia aplicación en nuestros días. En este caso, la información también fluye en un solo sentido, de la entrada a la salida. Se llaman capas ocultas por el hecho de que están en la parte intermedia de la red y no entran en contacto con los datos entregados por el mundo real. Cada capa oculta puede tener una o más neuronas como parte de ella, aunque es aconsejable no incrementar demasiadas neuronas para evitar saturar el procesamiento de la información al momento del aprendizaje (ver Figura 9).

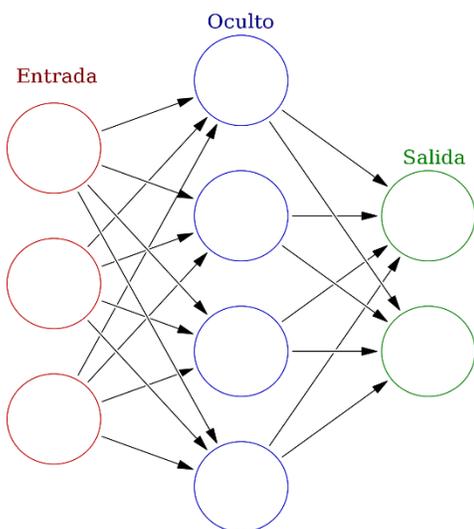


Figura 9. Red multicapa con una capa oculta. Copyright © 2019: Cburnet.

Las redes multicapa MLP (Multilayer Perceptron) logran superar las limitaciones del perceptrón simple, para solucionar problemas de clasificación no lineales, gracias al desarrollo del Algoritmo Backpropagation (retropropagación o propagación hacia atrás) realizado en 1974 por el profesor Paul Werbos, para el entrenamiento del perceptrón multicapa. Así, mientras los datos se propagan hacia adelante “feedforward”, el error se propaga de la capa de salida hacia las capas ocultas, haciendo uso de la llamada Regla Delta, más conocido como Error Cuadrático Medio (MSE) o por sus siglas en ingles Least Mean Square (LMS).

La Regla Delta, cuya ecuación la hemos explicitado como el error “ $e_p$ ”, nos permite calcular el error generado por una sola neurona, mientras que el Algoritmo Backpropagation hace posible calcular el error global de toda la red neuronal. Es obvio que lo que se busca es el menor error posible, cuanto menor es el error mejor es el resultado.

### **Redes feedforward.**

Lo que caracteriza a estas redes neuronales es la forma como fluye la información, de la entrada hacia la salida pasando por las capas intermedias. Así, la información evoluciona del nivel de entrada a la capa oculta uno, de la capa oculta uno a la capa oculta dos y así, sucesivamente, hasta llegar a la capa de salida.

Estas redes también son conocidas como redes de avance y se pueden utilizar para relacionar entradas con salidas conocidas, haciendo una suerte de mapeo y resolver problemas de aprendizaje, como en el caso del diagnóstico del dengue en el que se relaciona los síntomas de entrada con las salidas con presencia o ausencia de dengue, sobre cuya base la red se entrena y aprende. La experiencia indica que una RNA con una capa oculta y suficientes neuronas en la capa oculta puede adaptarse a cualquier problema de relación entrada – salida finita.

Por defecto, las RNA MLP feedforward trabajan con funciones de transferencia u activación tansig (sigmoidea tangencial) en las capas ocultas y con funciones purelin

(lineal) en la capa de salida. Del mismo modo, la función de entrenamiento generalmente utilizado es el trainlm (Levenberg – Marquardt) al que nos referiremos más adelante.

Existen diferentes versiones de redes neuronales feedforward como las redes de adaptación fitnet, de reconocimiento de patrones patternnet y la red en cascada hacia adelante cascadeforwardnet. Esta última es una variación de la red de avance hacia adelante que tiene conexiones adicionales desde la entrada directamente hacia cada una de las capas ocultas y de salida, así como, de cada capa anterior a la capa posterior; estas redes tienen la capacidad de aprender relaciones de entrada – salida finita de manera óptima.

### **Redes recurrentes.**

Son aquellas redes neuronales en las que la información no fluye en un solo sentido, digamos de la entrada hacia la salida, como es el caso de las redes de avance feedforward, sino que puede realimentarse desde capas posteriores hacia las anteriores a través de nuevas conexiones sinápticas. Estas pueden ser monocapa o multicapa.

## **Aprendizaje de las redes neuronales artificiales**

Según el diccionario de la Real Academia Española, aprendizaje es “la adquisición por la práctica de una conducta duradera” (2020). Eso es más o menos lo que sucede con las RNA cuando son entrenadas, por la práctica repetitiva (iteraciones) va actualizando los pesos de sus conexiones internas, hasta encontrar un resultado óptimo mejorando su performance. Luego del entrenamiento y agotado el proceso de aprendizaje, la red puede ser utilizada en aplicaciones duraderas dando respuesta a los problemas del entorno.

Según Caicedo y López (2017) “el aprendizaje para la red neuronal artificial, es la habilidad para aprender del entorno y mejorar su desempeño, es un proceso interactivo que permite ajustar los pesos sinápticos” (p.28). Este proceso de aprendizaje se puede generalizar de la siguiente forma:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \Delta\mathbf{w}(t)$$

Donde:

$w(t+1)$  : Valor actualizado del peso sináptico.

$w(t)$  : Valor actual del peso sináptico.

$\Delta w(t)$  : Variación del peso sináptico.

El peso actual es igual al peso anterior más una variación.

### **Aprendizaje supervisado.**

El aprendizaje supervisado, en redes neuronales artificiales, es cuando el proceso de entrenamiento se realiza bajo el control de un supervisor o maestro externo, quien monitorea el comportamiento del error cuadrático medio resultado de la comparación de la salida objetivo o deseada (target) respecto a la salida real del sistema (output), cuando se le ingresa un conjunto de entradas.

Sin embargo, en una RNA, este debe aprender todo el conjunto de patrones de entrenamiento, por lo que no puede aprender con un error local sino con el conjunto de errores generados o error global, que se calcula en base a la ecuación que se muestra a continuación, que se define como el Error Cuadrático Medio (MSE por sus siglas en inglés), donde “n” es el número de puntos a evaluar de la regresión y el término entre paréntesis elevado al cuadrado es para evitar valores negativos del error:

$$e_p = \frac{1}{2} \sum_{j=1}^M (d_{pj} - y_{pj})^2$$

### **Aprendizaje no supervisado.**

El aprendizaje no supervisado es aquel que ejecuta el entrenamiento de la red neuronal sin la presencia del supervisor o maestro, en este caso, solo se ingresan los datos de entrada

organizados en patrones sin tener conocimiento de la salida deseada o target, los pesos se calculan en base a las características de las entradas y el objetivo específico que se espera alcanzar.

### Algoritmo de entrenamiento backpropagation

Sabido es que, un algoritmo de entrenamiento de una RNA, modifica el valor de los pesos de las conexiones entre las neuronas de entrada, ocultas y de salida. Cada vez que se entrena la red se modifican los pesos. La clave está en entrenar la red neuronal cuantas veces sea necesaria, hasta encontrar el menor error de entrenamiento y sobre todo de validación, de donde se obtienen los pesos definitivos.

El algoritmo backpropagation o de retropropagación se basa en el cálculo del gradiente negativo o gradiente descendente del error cuadrático medio ( $\nabla E_p$ ) en la salida de la RNA MLP, considerando  $E_p$  como el error global de la red neuronal. El error se calcula en la capa de salida, luego se calcula en la capa oculta anterior, continuando por las demás capas ocultas hasta llegar a la capa de entrada, actualizando los pesos sinápticos de cada capa de acuerdo al error obtenido en cada una de ellas. Como se verá en este caso el error se propaga de manera inversa al funcionamiento de avance de la red o feedforward.

De acuerdo a Caicedo y López (2017) el algoritmo backpropagation corre siguiendo la secuencia que se muestra a continuación:

1. Inicializamos los pesos del MLP.
2. Mientras la condición de parada sea falsa ejecutamos los pasos 3 a 12.
3. Aplicamos un vector de entrada  $\mathbf{x}_p = [x_{p1}, x_{p2}, \dots, x_{pi}, \dots, x_{pn}]^T$ .
4. Calculamos los valores de las entradas netas para la capa oculta.

$$Neta_{pj}^h = \sum_i^N w_{ji}^h x_{pi} + \theta_j^k$$

5. Calculamos la salida de la capa oculta.

$$i_{pj}^h = f_j^h (Neta_{pj}^h)$$

6. Calculamos los valores netos de entrada para cada capa de salida.

$$Neta_{pk}^o = \sum_{j=1}^L w_{kj}^o i_{pj}^o + \theta_k^o$$

7. Calculamos las salidas de la red.

$$y_{pk} = f_k^o(Neta_{pk}^o)$$

8. Calculamos los términos de error para las unidades de salida.

$$\delta_{pk}^o = (d_{pk} - y_{pk}^o) f_k^{o'}(Neta_{pk}^o)$$

9. Estimamos los términos de error para las unidades ocultas.

$$\delta_{pj}^h = f_j^{h'}(Neta_{pj}^h) \sum_{k=1}^M \delta_{pk}^o w_{kj}^o$$

10. Actualizamos los pesos en la capa de salida.

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \alpha \delta_{pk}^o i_{pj}^h$$

11. Actualizamos pesos en la capa oculta.

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \alpha \delta_{pj}^h x_{pi}$$

12. Verificamos si el error global cumple con la condición de finalizar.

$$E_p = \frac{1}{2} \sum_{p=1}^P \sum_{k=1}^M (d_{pk} - y_{pk})^2$$

Asumiendo que el error global  $e_{pk}$  se define como el error de la  $k$ -ésima neurona, con  $d_{pk}$  representando la salida deseada, así como,  $y_{pk}$  la salida real de la red neuronal, entonces el error global se escribe como:

$$e_{pk} = (d_{pk} - y_{pk})$$

Considerando además que, para efectos de análisis, hacemos uso del  $p$ -ésimo patrón de aprendizaje, no siendo necesaria considerar toda la sumatoria de la ecuación del error global que va de  $p=1$  hasta  $P$ , en consecuencia, el gradiente de  $E_p$  queda como sigue:

$$\nabla E_p = \frac{\partial E_p}{\partial w_{kj}^o}$$

Sin embargo, como nuestro problema trata del concepto de gradiente descendente, con el fin de encontrar progresivamente un punto de error mínimo, se requiere un valor negativo de la derivada parcial del error global, lo que nos lleva a la siguiente ecuación:

$$-\nabla E_p = - \frac{\partial E_p}{\partial w_{kj}^o}$$

El proceso de entrenamiento de la RNA MLP nos lleva a modificar el peso  $w_{kj}^o$ , lo que se realiza de acuerdo a la siguiente ecuación:

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \Delta w_{kj}^o(t) = w_{kj}^o(t) + \alpha (-\nabla E_p)$$

Lo que en la práctica significa que el peso en tiempo futuro (t+1) es igual al peso actual en tiempo (t) más una variación del peso, que se expresa por un factor  $\alpha$  multiplicado por el gradiente descendente. Donde  $\alpha$  representa la tasa de aprendizaje de la RNA. El desarrollo de esta ecuación nos lleva finalmente a la ecuación de modificación de pesos.

### **Algoritmos de entrenamiento.**

A lo largo del tiempo se han desarrollado una serie de algoritmos de entrenamiento, algunos de los cuales enumeramos a continuación.

#### ***Traingd.***

Es el algoritmo backpropagation por excelencia, que hace uso de la regla delta o método del gradiente descendente para el cálculo del error cuadrático medio de la RNA.

#### ***Traingda.***

Un inconveniente del algoritmo backpropagation es que el parámetro de aprendizaje  $\alpha$  es fijo, parámetro que se puede modificar para variar los pesos y mejorar la convergencia del algoritmo de aprendizaje. Este método se denomina algoritmo de Gradiente Descendente con Velocidad de Aprendizaje Variable, traingda.

### ***Algoritmos de gradiente conjugado.***

Muchas veces los algoritmos `traingd` y `traingda` resultan siendo métodos de entrenamiento demasiado lentos, por lo que se ha desarrollado otros métodos de aprendizaje como los denominados de gradiente conjugado. Intuitivamente, el concepto es el siguiente, se trata de minimizar el error siguiendo a la conjugada del gradiente descendente que, como se sabe, es tangencial a la curva del error, lo que nos permite llegar al error mínimo global de forma más rápida. Aquí se utiliza el concepto de Matriz Hessiana (H) que implica una alta carga computacional; Caicedo y López (2017) refieren que “en problemas de optimización complejos como el entrenamiento de una red neuronal para ser aplicada en reconocimiento de voz o de caracteres manuscritos, es difícil o imposible calcularlas”. (p.90). Una matriz hessiana relaciona a la variable de salida de la red neuronal con los pesos de las neuronas a través de derivadas parciales de segundo orden.

Existen varios algoritmos que se basan en este método: El algoritmo de actualización de Fletcher – Reeves (`traincgf`), de actualización de Polak – Ribière (`traincgp`), de restablecimiento de Powell – Beale (`traincgb`), y, Gradiente Conjugado Escalado (`traincsg`).

### ***Algoritmos cuasi - newton.***

Este método es una interesante alternativa a los métodos de entrenamiento de gradiente conjugado, toda vez que converge a menudo más rápidamente, aunque es complejo y requiere alta computabilidad (Castro, 2006). Existen un par de algoritmos de esta naturaleza: El algoritmo de Broyden, Fletcher, Goldfarb y Shanno (BFGS) comúnmente conocido como `trainbfgf`, y, el Algoritmo Secante de Un Paso (`trainoss`).

### ***Algoritmo levenberg – marquardt.***

Tal como se define en Mathworks, es una función de entrenamiento en red que actualiza los valores de peso y sesgo (bías) según la optimización de Levenberg – Marquardt y lleva

por denominación **trainlm**. En consonancia con Santillán (2015) “es a menudo el algoritmo más rápido en el toolbox, y es altamente recomendado como primera opción para entrenamientos supervisados a pesar de requerir más espacio de memoria que otros algoritmos” (p.73). De hecho, en redes feedforward es el algoritmo utilizado por defecto.

Lo interesante del algoritmo trainlm es que combina en método de Gradiente Descendente y el método de Newton. Tal como señalan Caicedo y López (2017) “el método Levenberg Marquardt mezcla, suavemente, el método de Newton y el método Gradiente Descendente en una única ecuación para estimar  $\mathbf{W}(t+1)$ ” (p.96). Ecuación que se escribe de la siguiente forma:

$$\mathbf{W}(t+1) = \mathbf{W}(t) - (\mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{G}$$

$$\mathbf{W}(t+1) = \mathbf{W}(t) - \Delta$$

$$(\mathbf{H} + \lambda \mathbf{I}) \Delta = \mathbf{G}$$

Donde:

- H : Matriz Hessiana.
- I : Matriz Identidad.
- G : Gradiente para el conjunto de patrones.
- $\lambda$  : Parámetro.

Algunos autores recomiendan elegir un valor inicial fijo de  $\lambda=0.001$ .

La RNA MLP que utiliza el algoritmo Levenberg – Marquardt (trainlm) actualiza su vector de pesos en un proceso de aprendizaje por lotes, según Caicedo y López (2017) con los siguientes pasos:

1. Inicializamos los parámetros de la red:
  - Definimos los valores iniciales de los pesos y bias de la red.
  - Definimos el valor inicial del parámetro de aprendizaje  $\alpha$ .
  - Definimos el error mínimo deseado  $E_{\min}$ .
  - Definimos la condición parar = Falsa y RESET = Falso.

2. Si RESET=False ejecutamos los pasos 3 a 13, sino vamos al paso 14.

3. Presentamos todos los patrones de entrenamiento de la red.

$$X_p = \{x_1, x_2, \dots, x_i, \dots, x_p\}$$

4. Calculamos los valores de las entradas netas para la capa oculta.

$$Neta_{pj}^h = \sum_{i=1}^N w_{ji}^h x_{pi} + \theta_j^h$$

5. Calculamos la salida de la capa oculta.

$$i_{pj}^h = f(Neta_{pj}^h)$$

6. Calculamos los valores netos de entrada para la capa de salida.

$$Neta_{pk}^o = \sum_{j=1}^l w_{kj}^o i_{pj}^o + \theta_k^o$$

7. Calculamos la salida de la red neuronal.

$$y_{pk} = f_k^o(Neta_{pk}^o)$$

8. Calculamos el error global de la red y lo almacenamos en la variable.

$E_{prev}$  (Error previo)

$$E_{prev} = E = \frac{1}{2} \sum_{p=1}^P \sum_{k=1}^M (d_{pk} - y_{pk})^2$$

9. Calculamos el gradiente para el conjunto de patrones.

$$G_{save} = -\nabla E(0)$$

10. Calculamos la matriz H.

$$H_{save} = H$$

11. Si  $E_{prev} > E_{min}$  (el error previo es mayor al error mínimo deseado), ejecutamos los pasos 12 a 18, sino hacemos condición parar = Verdadera y salimos.

12. Inicializamos el parámetro  $\lambda$ .

$$\lambda = H_{ijmax}$$

$$i = j$$

13. Inicializamos indicador RESET a verdadero y volver a paso 2.

14. Si RESET = verdadero, guardamos el vector gradiente y la matriz  $\mathbf{H}$ .

$$\mathbf{H} = \mathbf{H}_{\text{save}}$$

$$\mathbf{G} = \mathbf{G}_{\text{save}}$$

15. Incrementamos cada elemento de la diagonal de  $\mathbf{H}$  en  $\lambda$ .

$$\mathbf{H} = \mathbf{H} + \lambda \mathbf{I}$$

16. Resolvemos el sistema de ecuaciones para  $\Delta$ .

$$(\mathbf{H} + \lambda \mathbf{I})\Delta = \mathbf{G}$$

17. Adicionamos  $\Delta$  al actual vector de pesos para actualización.

$$\mathbf{W}(t+1) = \mathbf{W}(t) - \Delta$$

18. Calculamos el error  $\mathbf{E}$ , el gradiente  $\mathbf{G}$  y la matriz  $\mathbf{H}$  con el nuevo vector de pesos.

$$E = \frac{1}{2P} \sum_{p=1}^P \sum_{k=1}^M (d_{pk} - y_{pk})^2$$

19. Si  $\mathbf{E}_{\text{prev}} \leq \mathbf{E}_{\text{min}}$ , hacer condición\_parar = verdadero, en caso contrario condición\_parar = Falsa.

20. Si  $\mathbf{E} < \mathbf{E}_{\text{prev}}$  hacemos  $\mathbf{E}_{\text{prev}} = \mathbf{E}$ , en caso contrario hacemos los siguientes ajustes y volvemos al paso 14:

$$\text{RESET} = \text{verdadero}$$

$$\lambda = \lambda * \rho \text{ donde } \rho < 1$$

21. Si  $\mathbf{E} > \mathbf{E}_{\text{prev}}$  hacemos los siguientes ajustes y volvemos al paso 14.

$$\text{RESET} = \text{falso}$$

$$\lambda = \lambda * \sigma$$

$$\mathbf{H}_{\text{save}} = \mathbf{H}, \text{ donde } \sigma > 1$$

$$\mathbf{G}_{\text{save}} = \mathbf{G}. \text{ (p.97).}$$

### *Algoritmo bayesiano (trainbr).*

Una alternativa para resolver el problema del sobre entrenamiento es entrenar la RNA con un enfoque probabilístico, lo que se conoce como entrenamiento bayesiano. La idea es

suponer una función de densidad de probabilidad a priori para los pesos de la red y a medida que se obtengan datos calcular la densidad de probabilidad posterior de estos pesos.

El sobre entrenamiento de la red neuronal es algo indeseable porque hace que las salidas de la red sean erróneas cuando se trabaja con datos de entrada que no formaron parte de los patrones de entrenamiento; afectando su capacidad de generalización. Esto se debe a que por un lado se tiene demasiadas neuronas en o las capas ocultas, a que los pesos son demasiados grandes. Situación que se puede resolver a través de la poda de neuronas, la parada temprana cuando el error comienza a crecer o entrenando la red con pesos más pequeños. Ahí es donde interviene el teorema de Bayes.

Para lograr que el entrenamiento de la RNA MLP se realice con los pesos más pequeños, se requiere hacer una modificación en la forma de calcular el error total  $M(w)$  como se observa en fórmula siguiente:

$$M(w) = \beta E_D + \alpha E_w$$

En la que el error de salida  $E_D$  y el error de los pesos  $E_w$  están dados:

$$E_D = \frac{1}{2}P \sum_{p=1}^P (d_p - Y_p)^2$$

$$E_w = \sum_{j=1}^N w_j^2$$

Donde  $P$  es el número de patrones de entrenamiento y  $N$  el número de pesos. De modo que, el error deseado es controlado por el parámetro  $\beta$  y el error de los pesos por el parámetro  $\alpha$ .

En términos de Caicedo y López (2005) “la idea del enfoque bayesiano para entrenar redes neuronales es suponer una función de densidad de probabilidad a priori para los pesos de la red y a medida que se obtengan datos calcular la probabilidad de densidad posterior de estos pesos” (p.4). La regla de Bayes adaptada al caso de una RNA MLP se muestra a continuación:

$$P(w|\mathcal{D}, H) = \frac{P(\mathcal{D}|w, H) P(w, H)}{P(\mathcal{D}, H)}$$

Donde:

$W$  : vector de pesos de la RNA.

$\mathcal{D}$  : Conjunto de datos para el entrenamiento.

$H$  : modelo neuronal de entrenamiento

$P(w|\mathcal{D}, H)$ : Densidad de probabilidad posterior de los pesos.

$P(w, H)$  : Densidad de probabilidad a priori, pesos conocidos.

$P(\mathcal{D}|w, H)$ : Densidad de probabilidad condicional que datos ocurran dado  $w$ .

$P(\mathcal{D}, H)$  : Factor de normalización que garantiza probabilidad total 1.

El análisis de Caicedo y López nos lleva a la siguiente ecuación:

$$\gamma = N - 2 \alpha^{MP} \text{tr}(\mathbf{H}^{MP})^{-1}$$

En la que  $\gamma$  representa la medida de cuantos pesos de la red neuronal están siendo usados de manera efectiva para reducir el error y  $\mathbf{H}$  es la Matriz Hessiana dada por:

$$\mathbf{H} = \beta \nabla^2 E_D + \alpha \nabla^2 E_w$$

Conocida como la retropropagación de regularización bayesiana, en suma, `trainbr` según Mathworks “es una función de entrenamiento en red que actualiza los valores de peso y sesgo según la optimización de Levenberg-Marquardt. Minimiza una combinación de errores al cuadrado y pesos, y luego determina la combinación correcta para producir una red que generalice bien”. Eso supone que es una mejora del algoritmo `trainlm`.

La sintaxis utilizada es la siguiente:

```
red.trainFcn = 'trainbr'
```

```
[red,tr] = train(red,trainbr).
```

Una vez más, Caicedo y López (2005) proponen los siguientes pasos del algoritmo trainbr:

1. Inicializar  $\alpha$  y  $\beta$  y los pesos de la red neuronal ( $w$ ). Con  $\alpha=0$  y  $\beta=1$ , a fin de minimizar el  $E_D$  sin penalizar la magnitud de los pesos.
2. Minimice la función  $M(w)=\beta E_D+\alpha E_w$  con algún método convencional, una alternativa es el Algoritmo Levenberg – Marquardt que contiene a la Matriz Hessiana  $\mathbf{H}$  útil para el algoritmo.
3. Calcular la Matriz Hessiana. Si se trabaja con Levenberg – Marquardt la matriz ya está calculada.
4. Con el valor de la Matriz Hessiana calcular el número efectivo del parámetro  $\gamma$ .
5. Computar los valores para  $\alpha$  y  $\beta$ :

$$\gamma = \frac{\gamma}{2E_w(\mathbf{w})}$$
$$\beta = \frac{P - \gamma}{2E_D(\mathbf{w})}$$

6. Repetir del paso 2 al 5 hasta que el entrenamiento converja.

### **Características y ventajas de las RNA MLP.**

Las redes neuronales artificiales MLP han demostrado tener enormes capacidades para extraer conocimiento a partir de datos complejos e imprecisos como el ruido en señales de comunicaciones, procesamiento de series de tiempo o la clasificación de patrones. Así, tienen capacidad de aprendizaje, es decir, aprender de un conjunto de ejemplos; capacidad de generalización, resolver problemas trabajando con datos que no han formado parte del proceso de entrenamiento; capacidad para identificar características esenciales de los datos discriminándolas de las que no son relevantes; capacidad de asociación haciendo clustering; aprendizaje adaptativo, aquel que le permite modificar sus propios pesos sinápticos para adecuarse a nuevas circunstancias; Auto-Organización; tolerancia a fallos, porque soportan datos imperfectos, incompletos, con ruido y distorsiones, así como, una capacidad muy importante, la de procesar información en paralelo y por ende a altas velocidades.

## Estado del Arte

### Diagnóstico del dengue utilizando redes neuronales artificiales

Para la realización de este trabajo se evaluaron diez (10) artículos científicos correspondientes al diagnóstico del dengue utilizando redes neuronales artificiales, entre los años 2010 al 2019, con los siguientes resultados.

Aburas, Cetiner y Sari (2010) realizaron una aplicación de redes neuronales artificiales para la predicción del dengue en Singapur, para 14 209 casos confirmados de dengue “en función de los parámetros físicos de temperatura media, humedad relativa media y precipitación total” (p.4257).

Según los autores, los modelos de redes neuronales artificiales (ANN) empleados:

Tenían tres capas: capa de entrada, capa oculta y capa de salida. El modelo de red inicial consistía de tres parámetros solo en su capa de entrada: temperatura media, humedad relativa media y precipitación total. La salida se seleccionó como el número de casos confirmados de dengue. Teniendo en cuenta los tres parámetros como entradas, no se obtuvieron buenos resultados y el mejor coeficiente de correlación alcanzado en los datos de prueba fue de 0,46, lo que significa un rendimiento muy bajo. (Aburas et al., 2010, p.4258).

Por lo tanto, se necesitaba una nueva estructura de modelo de red, que contenga un parámetro adicional a las neuronas de la capa de entrada. Este parámetro adicional, a saber, fue el número de casos de dengue reportados la semana anterior, el mismo que ya estaba disponible en el conjunto de datos. Aburas et al. (2010) afirma que:

El nuevo modelo ANN consistió en cuatro parámetros de entrada en la capa de entrada que fueron la temperatura media, la humedad relativa media, la precipitación total y la medición realizada la semana pasada o la última vez. El resultado fue el número de casos confirmados de dengue... Las ANN fueron entrenadas y probadas para ver la correlación de redes con diferentes estructuras. Se logró una alta correlación de 0.96 como se esperaba cuando se usa el mismo conjunto de datos (330 patrones) tanto para el entrenamiento como para las pruebas. Se decidió una capa oculta para ser óptima. (p.4259).

El propósito de este trabajo fue predecir los casos de dengue confirmados a partir de las bases de las redes neuronales artificiales en datos registrados durante seis años. Según Aburas et al. (2010) los resultados mostraron que “las cuatro características importantes, a saber, la temperatura media, la humedad relativa media, la precipitación total y el número total de casos confirmados de dengue fueron muy eficaces para predecir el número de casos confirmados de dengue”. (p.4260). Por lo que el modelo de predicción propuesto se puede utilizar en todo el mundo y en cualquier período de tiempo, ya que el enfoque no utiliza la información de tiempo para construirlo; y ha producido resultados muy alentadores.

El estudio realizado, constituye una aplicación de las redes neuronales artificiales al estudio de la predicción del dengue, considerando factores externos como la temperatura media, humedad relativa y precipitación pluvial, además, de los datos estadísticos referentes al impacto de la enfermedad en Singapur. Pero, no toma en cuenta los síntomas de la enfermedad a partir de los cuales evaluar la función predictora de las RNA.

Con el propósito de ayudar a los médicos a diagnosticar correctamente a los pacientes con dengue, en el siguiente estudio se planteó como objetivo construir un sistema de diagnóstico no invasivo, utilizando los conceptos de red neuronal artificial perceptrón multicapa entrenados a través de modelos Levenberg - Marquardt (MLPLM) y algoritmos de gradiente de conjugado a escala (MLPSCG), comparando ambos resultados. Ibrahim, Faisal y Taib (2010) manifiestan que solo “se utilizaron medidas de fisiología, manifestaciones clínicas y mediciones de BIA para construir el modelo de pronóstico. Las manifestaciones clínicas se componen de 18 síntomas y signos resentidos con pacientes con dengue, mientras que las mediciones de BIA se componen de 15 parámetros”, indicando a continuación que “nuestro modelo de diagnóstico contiene los 35 parámetros como variables independientes (entradas) y una salida (variable dependiente) que es la clasificación de riesgo”. (p.666).

Ibrahim et al. (2010) refieren que para el caso de la red MLPLM (perceptrón multicapa Levenberg - Marquardt) se “utilizaron tamaño de capas ocultas de 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28 y 30 neuronas... La tasa de precisión promedio más alta es de 69%, lo que se logra con un tamaño de capa oculta de 12 neuronas”, igualmente para el caso de la red MLPSCG (algoritmos de gradiente conjugado de escala), en la que también “se utilizaron tamaños de capas ocultas de 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28 y 30

neuronas... Se puede ver que el tamaño de la capa oculta de 26 neuronas tiene la precisión de predicción promedio más alta del 74.2% con 0.02 MSE promedio". (p.669-670).

Tal como refieren los autores, en el presente estudio:

Se emplearon dos modelos de redes neuronales de perceptrón multicapa para construir el sistema de diagnóstico. Los modelos se entrenaron a través de los algoritmos de gradiente de conjugado escalado y Levenberg - Marquardt. Se realizó un ajuste preciso de los parámetros de los modelos para lograr el mayor rendimiento de los modelos. La precisión de predicción del 75% de precisión se logró utilizando el algoritmo de gradiente de conjugado escalado, mientras que el 70.7% se logró mediante el uso del algoritmo de Levenberg - Marquardt... Los resultados del área bajo la Característica Operativa del Receptor (ROC) verificaron que el modelo entrenado a través del algoritmo de gradiente de conjugado escalado tiene un mejor desempeño en comparación con el entrenamiento del modelo a través del algoritmo de Levenberg - Marquardt. (Ibrahim et al., 2010, 671).

En el presente caso, se comparan el rendimiento de dos métodos de RNA para diagnosticar el dengue, en Malasia, el primero, utilizando el algoritmo de gradiente de conjugado escalado y, el otro, el algoritmo Levenberg - Marquardt. En ambos casos se utilizaron las mismas entradas, en total 35, de las cuales 18 corresponden a manifestaciones clínicas, es decir, síntomas del paciente que son factores determinantes del dengue, las demás corresponden a datos de BIA (Bioelectrical Impedance Analysis) que no siempre están disponibles, toda vez que no siempre se cuenta con este tipo de equipos BIA en los centros de salud, sobre todo en zonas rurales. Aun así, ambos modelos han demostrado ser robustos para el diagnóstico del dengue.

Un estudio similar al expuesto anteriormente para el diagnóstico del dengue, es el desarrollado en Malasia utilizando la técnica BIA y las redes neuronales artificiales Feed-Forward (MFNN). Para el caso, los autores Ibrahim, Faisal, Salim y Taib (2010) estudiaron prospectivamente un total de 223 sujetos sanos y 207 pacientes hospitalizados con dengue, los criterios de gravedad del riesgo de dengue se determinaron y agruparon en base a tres investigaciones de sangre, a saber, el recuento de plaquetas (PLT) (menor o igual a 30,000 células por mm<sup>3</sup>), hematocrito (HCT) (aumento en más de o igual al 20%) y nivel de

aspartato aminotransferasa (AST) (elevado en cinco veces el límite superior normal) o nivel de alanina aminotransferasa (ALT) (elevado en cinco veces el límite superior normal).

Ibrahim et al. (2010), refiere que la RNA decidida se entrenó a través de la más pronunciada propagación de la pendiente descendente con el algoritmo de impulso utilizando la función de transferencia log-sigmoidea, mientras que el error de suma al cuadrado se utilizó como indicador de rendimiento de la red, la mejor arquitectura lograda de RNA fue de 3-6-1 (3 entradas, 6 neuronas en la capa oculta y 1 salida). El algoritmo utilizado se muestra en la Figura 10:

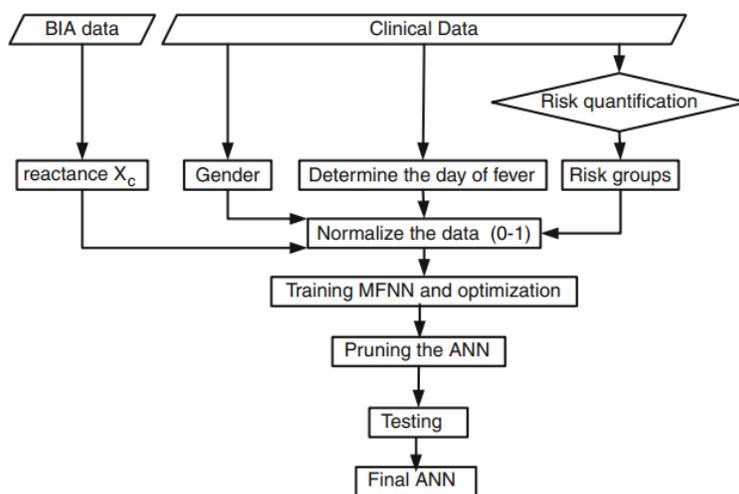


Figura 10. Algoritmo que distribuye datos clínicos y datos BIA.  
Fuente: Ibrahim, 2010.

En este estudio, según los autores, se logró el 96.86% de precisión de la predicción general para el diagnóstico de riesgo en pacientes con dengue que utilizan BIA y ANN. La precisión diagnóstica para el diagnóstico de los grupos control, alto riesgo y bajo riesgo fue de 100, 92.78 y 95.93%, respectivamente. Sin embargo, cuando se implementó la técnica de poda (eliminación de la ANN), la precisión general de la predicción fue de 96.27% con 95.88, 96.83 y 95.81% para los grupos de alto riesgo, bajo riesgo y control, respectivamente. Por lo tanto, la eliminación de la ANN redujo las precisiones generales y las precisiones de los grupos de control en 0.59 y 4.19%, respectivamente, sin embargo, es favorable para podar la ANN ya que la precisión del diagnóstico de los grupos de bajo riesgo y alto riesgo se ha incrementado. (Ibrahim et al., 2010).

Como resultado, el sistema es capaz de clasificar y diagnosticar el riesgo en los pacientes con dengue con una precisión de predicción global del 96.27%, es decir, considerando tanto la técnica BIA y la RNA Feed – Forward. Una vez más, nos encontramos con las limitaciones en cuanto a contar con equipos BIA en las zonas rurales.

Comparando los métodos MLPLM (70.7%) y MLPSCG (75%) de Faisal y la combinación BIA con MFNN (96,27%) de Ibrahim, es evidente, que este último tiene una mejor precisión para el diagnóstico del dengue.

Gambhir, Malika y Kumarb (2017), plantean que “el diagnóstico temprano de la enfermedad del dengue puede ayudar a proteger las vidas humanas haciendo un movimiento preventivo antes de que se convierta en una enfermedad infecciosa” (p.1), por lo que proponen desarrollar un modelo de diagnóstico basado en PSO-ANN para el diagnóstico temprano de la fiebre del dengue. En el modelo propuesto, la técnica de PSO (Particle Swarm Optimization) se aplica para optimizar los parámetros de peso y sesgo del método ANN (Network Neural Artificial).

Los atributos a tomar en cuenta son 16, entre los cuales podemos mencionar a datos clínicos como fiebre, vómito, dolor abdominal, escalofríos, dolor del cuerpo, cefalea y debilidad, así como, datos de entrada como la edad y el sexo, además, de inmunoglobulinas como IgM, IgG, NS1 y Elisa (Gambhir, 2017). Las inmunoglobulinas son obtenidas con el auxilio de equipos especiales. La RNA utilizada por los autores se muestra en la Figura 11 a continuación:

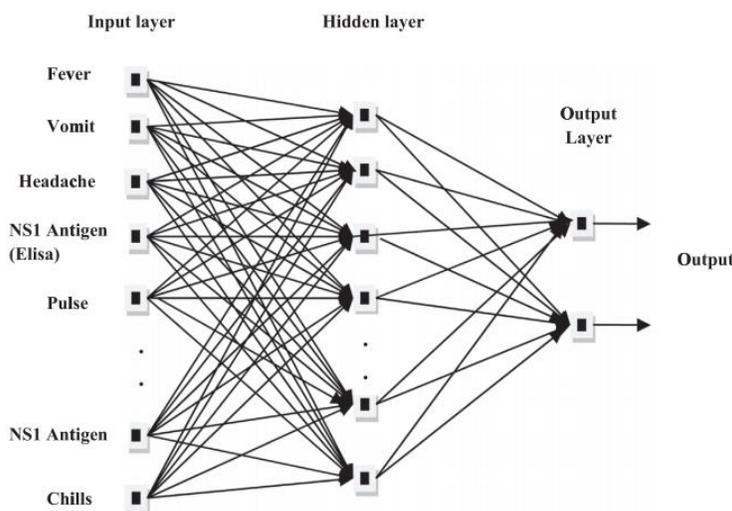


Figura 11. RNA de tres capas y atributos de entrada del dengue. Fuente: Ghambir, 2017.

Los autores, Gambhir et al. (2017) explican que el método PSO realiza búsquedas utilizando una población (llamada enjambre) de individuos (llamados partículas) que se actualizan con cada iteración. Para descubrir la solución óptima, cada partícula se mueve en la dirección de su mejor posición anterior (pbest) y su mejor posición global (gbest). La velocidad y posición de las partículas se puede actualizar utilizando las ecuaciones:

$$\mathbf{V}_{i,j}(t+1) = \mathbf{W} * \mathbf{V}_{i,j}(t) + c_1 * r_1 * (\mathbf{X}_{pbest}(t) - \mathbf{X}_{i,j}(t)) + c_2 * r_2 * (\mathbf{X}_{gbest}(t) - \mathbf{X}_{i,j}(t))$$

$$\mathbf{X}_{i,j}(t+1) = \mathbf{X}_{i,j}(t) + \mathbf{V}_{i,j}(t+1)$$

En la Tabla 2 podemos observar los resultados del trabajo de Gambhir et al., que compara el modelo PSO-ANN con otros modelos demostrando tener mejor performance.

Tabla 2

*AUC resultado de diferentes clasificadores*

	NB	DT	ANN	PSO	PSO-ANN
<b>Dengue negativo</b>	0.752	0.723	0.779	0.736	0.826
<b>Dengue positivo</b>	0.742	0.729	0.776	0.747	0.819
<b>Total</b>	0.747	0.726	0.778	0.742	0.823

El AUC del PSO-ANN es superior a los demás casos. Fuente: Gambhir, 2017.

En términos de los autores, se observa que el modelo propuesto:

Proporciona mejores resultados en comparación con otros enfoques como ANN, DT, NB y PSO. Además, la optimización de parámetros de ANN tiene un impacto significativo en el logro de un rendimiento mejorado. El enfoque de PSO-ANN tiene un tiempo de computación grande que otras metodologías. La investigación presentada en este documento muestra que el modelo de diagnóstico basado en PSO-ANN es capaz de diagnosticar el dengue en su etapa inicial con alta precisión. (Gambhir et al., 2017, p.8).

El modelo analizado, ofrece importantes ventajas para diagnosticar el dengue, con alta precisión, en sus etapas tempranas, pero requiere que en el centro de salud se cuente con

equipos modernos para la detección de los niveles de inmunoglobulina e incluso de VIH que no siempre están disponibles en zonas rurales.

Una tecnología Microarray (micromatrices) que permite extraer los diferentes niveles de expresión de los genes contenidos en muestras biológicas, fue utilizada como entradas para predecir diferentes tipos de enfermedades como el dengue. Chatterjee, Dey, Shi, Ashour, Fong y Sen (2017) combinaron esta tendencia en la investigación bioinformática con algoritmos de optimización de búsqueda Cockoo para redes neuronales artificiales (ANN-MCS); los autores manifiestan que:

La detección y clasificación de la fiebre del dengue tiene un papel vital debido a los brotes recientes de diferentes tipos de fiebre del dengue. Recientemente, el avance en la tecnología de microarrays puede emplearse para dicho proceso de clasificación. Varios estudios han establecido que la fase de selección de genes desempeña un papel importante en el rendimiento del clasificador. Posteriormente, el estudio actual se centró en la detección de dos variaciones diferentes, a saber, la fiebre del dengue (FD) y la fiebre hemorrágica del dengue (FHD). Se ha propuesto un método modificado de bag-of-features (BoF) para seleccionar la mayoría genes prometedores en el proceso de clasificación. (p.1).

Ellos manifiestan que el método propuesto (ANN-MCS) “se ha comparado con otros tres clasificadores conocidos, a saber, la red de avance de perceptrón multicapa (MLP-FFN), la red neuronal artificial (ANN) entrenada con búsqueda de cuckoo (ANN-CS), y ANN entrenada con PSO (ANN-PSO)”. (Chatterjee et al., 2017).

En general, según los autores, una enfermedad es causada por la expresión de varios genes; por lo tanto, seleccionar el número óptimo de genes, que se refiere al número de grupos, es una tarea difícil, alegando que, si el número de grupos es demasiado pequeño, los genes importantes podrían quedar fuera de los datos; por otro lado, demasiados clústeres podrían incorporar características redundantes a los datos, lo que tampoco se espera (Chatterjee et al., 2017).

En la Tabla 3 a continuación, se observa el comportamiento de cada uno de los modelos sometidos a evaluación, así, en cuanto a exactitud los modelos MLP-FNN, ANN-PSO, ANN-CS y ANN-MCS tienen los siguientes resultados 87.5, 90.91, 92.05 y 95.65 respectivamente, y, en cuanto a la precisión se obtiene 80, 91,67, 92.18 y 99.98 también respectivamente. Con lo

que se puede concluir que el modelo esbozado ANN-MCS es determinantemente superior a los otros modelos.

Tabla 3

*Análisis comparativo del modelo propuesto con otros clasificadores*

	<b>MLP-FFN</b>	<b>ANN-PSO</b>	<b>ANN-CS</b>	<b>ANN-MCS</b>
<b>Accuracy</b>	87.5	90.91	92.05	95.65
<b>Precision</b>	80	91.67	92.18	99.98
<b>Recall</b>	100	91.67	92.37	91.67
<b>F-Measure</b>	88.89	91.67	92.27	95.65

El modelo ANN-MCS muestra mejor performance que las demás. Fuente: Chatterjee, 2017.

Los resultados experimentales sugirieron que el modelo ANN-MCS propuesto, junto con el esquema modificado de Bolsas de Aire (BoF), es extremadamente exitoso en la detección del dengue, asimismo, la prueba de significación estadística reveló que los resultados experimentales obtenidos por ANN-MCS son altamente significativos (Chatterjee et al., 2017).

El modelo en referencia, hace uso de las técnicas de clasificación especiales que requieren contar previamente con información genética, que muchas veces no están al alcance del personal de salud de las zonas rurales de la región San Martín, en Perú, por lo que su uso se ve dificultado, siendo necesario contar con una técnica informática más asequible.

Una variante en las investigaciones sobre el dengue la tenemos en el siguiente estudio, que incorpora en el análisis de predicción los síntomas de la fiebre del dengue hemorrágico (DHF por sus siglas en inglés). En el estudio realizado en Indonesia por Sasongko, Wibawa, Maulana y Bahtiar (2017) se buscó encontrar el mejor algoritmo Backpropagation para resolver el problema mediante el análisis de detección temprana de DHF con la adición de optimización en el perceptrón de múltiples capas (MLP) “a través de cinco tipos de algoritmos de entrenamiento de propagación hacia atrás, Gradient Descent (GD), BFGS Quasi Newton (BQN), pendiente de gradiente conjugada - Powel (CGD), Backpropagation resistente (RB) y Levenberg Marquardt (LM)”. (p.183).

Las variables utilizadas en este estudio fueron los síntomas iniciales de los pacientes con DHF de hasta 10 variables. Todos los datos de investigación se tomaron en base a los datos de registros médicos en RSUP Dr. Karyadi Semarang un total de 76 datos. En el 2014 se presentaron 71 668 casos de dengue en 34 provincias, con 641 casos mortales. Siendo Indonesia el país con mayor afección en el sudeste asiático.

La Red Neuronal Artificial Backpropagation utilizada se observa de la Figura 5, que cuenta con 10 neuronas de entrada, neuronas ocultas y una neurona de salida, que se muestra a continuación:

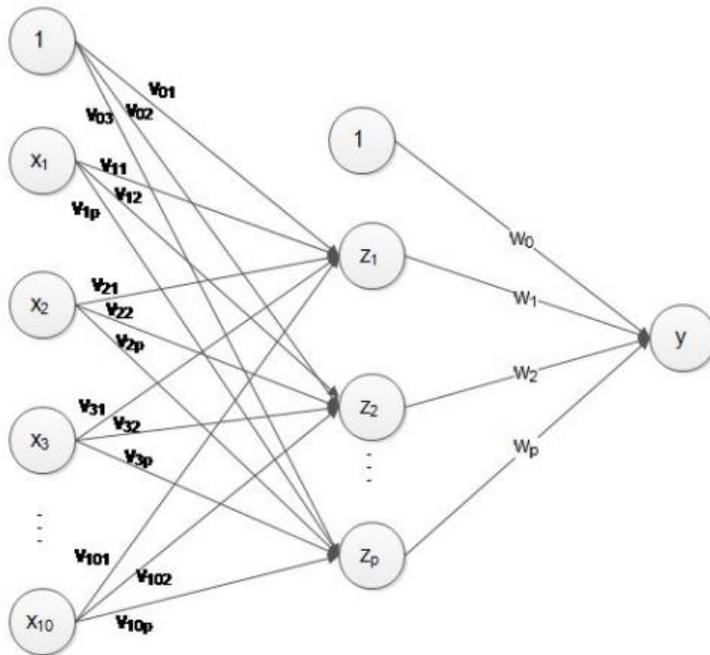


Figura 12. Red neuronal artificial con 10 entradas, neuronas ocultas y una de salida.  
Fuente: Sasongko, 2017.

Las variables de entrada son: Fiebre ( $x_1$ ), náuseas ( $x_2$ ), vómitos ( $x_3$ ), diarrea ( $x_4$ ), heces negras ( $x_5$ ), dolor de cabeza ( $x_6$ ), dolor de abdomen ( $x_7$ ), dolor muscular ( $x_8$ ), manchas rojas ( $x_9$ ) y el sangrado espontáneo ( $x_{10}$ ). Variables que fueron evaluadas en función de su presencia o ausencia como síntoma. Ver Tabla 4.

Tabla 4

*Variables de las neuronas de entrada por categoría.*

TABLE I. RESULT OF ONE HOT ENCODING DATA

Variable	Category	Result
Fever ( $x_1$ )	Yes	1
	No	0
Nausea ( $x_2$ )	Yes	1
	No	0
Vomiting ( $x_3$ )	Yes	1
	No	0
Diarrhea ( $x_4$ )	Yes	1
	No	0
Black Feses ( $x_5$ )	Yes	1
	No	0
Headache ( $x_6$ )	Yes	1
	No	0
Pain Abdomen ( $x_7$ )	Yes	1
	No	0
Muscular Pain ( $x_8$ )	Yes	1
	No	0
Red Spots ( $x_9$ )	Yes	1
	No	0
Spontaneous Bleeding ( $x_{10}$ )	Yes	1
	No	0
Target ( $t$ )	Yes	1
	No	0

Fuente: Sasongko, 2017.

Sasongko et al. (2017), concluyen que el rendimiento de los cinco algoritmos para la detección temprana de la enfermedad del dengue visto desde el valor de MSE, la precisión, la sensibilidad y la especificidad “muestran que el algoritmo de Levenberg Marquardt produce el valor de MSE más pequeño, lo que significa que el algoritmo de Levenberg Marquardt es el mejor algoritmo”. Ver Figura 13.

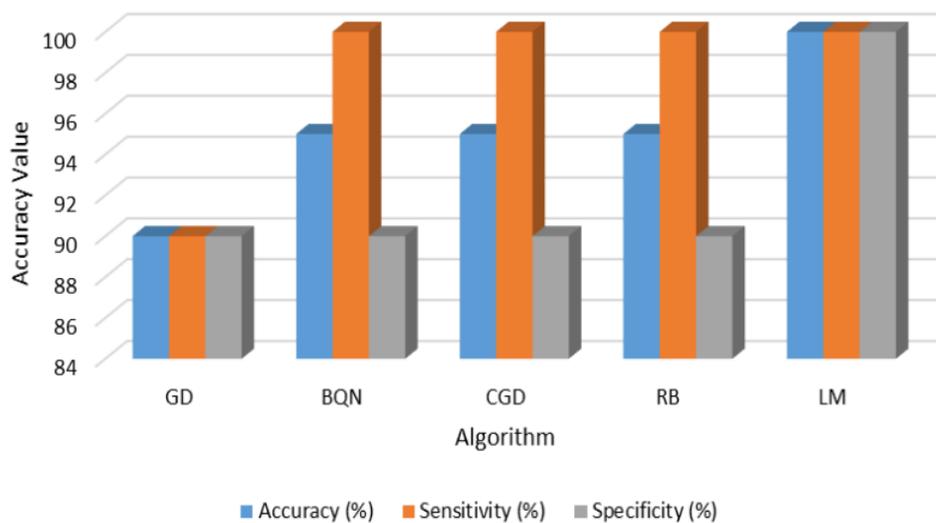


Figura 13. Cuadro comparativo de precisión, sensibilidad y especificidad. Fuente: Sasongko, 2017.

En el siguiente estudio, realizado por Laureano - Rosario, Duncan, Méndez - Lázaro, García - Rejón, Gómez - Carro, Farfán - Ale, Savic y Muller - Karger (2018), se aplicó redes neuronales artificiales (ANN) para predecir los brotes de fiebre del dengue en San Juan, Puerto Rico (EE. UU.), y, en varios municipios costeros del estado de Yucatán, México. Los modelos fueron entrenados con 19 años de datos de fiebre del dengue para Puerto Rico y seis años para México.

Los datos ambientales y demográficos incluidos en los modelos predictivos fueron la Temperatura de la Superficie del Mar (TSM), la precipitación, la temperatura del aire (es decir, la mínima, la máxima y la media), la humedad, los casos anteriores de dengue y el tamaño de la población. Para tal efecto, se aplicaron dos modelos para cada área de estudio. Uno predijo las tasas de incidencia de dengue en función de la población en riesgo (es decir, el número de personas menores de 24 años), y el otro en el tamaño de la población vulnerable (es decir, el número de personas menores de cinco años y mayores de 65 años). (Laureano et al., 2018). De acuerdo con los autores, el presente estudio:

Implementó la RNA utilizada para la identificación de inundaciones pluviales de radar para el sistema de drenaje (RAPIDS), que se desarrolló para predecir las inundaciones en los sistemas de alcantarillado de la Universidad de Exeter y se modificó para predecir los casos de brote de fiebre del dengue. El enfoque normaliza los datos (por ejemplo, la transformación natural del registro) para hacer que los valores de peso sean más sensibles a los cambios. Los modelos de ANN utilizaron un algoritmo genético de clasificación no dominado II (NSGA-II) para fines de optimización y entrenamiento; los resultados se validaron utilizando el método de validación cruzada de dejar uno fuera (LOOCV). (p.4).

Las variables ambientales utilizadas para predecir la fiebre del dengue en México fueron: humedad, precipitación acumulada de una semana, TSM, tamaño de la población, incidencia previa del dengue, temperatura mínima del aire y fecha. Para Puerto Rico, los datos incluyeron: precipitación acumulada de cuatro semanas, TSM, tamaño de la población, incidencia previa de dengue, temperatura máxima y mínima del aire y fecha.

Los resultados obtenidos en las pruebas realizadas, a partir de las curvas ROC se muestran en la Figura 14, en la que destaca con mejor curva ROC el modelo predictivo C, con una curva ROC (AuC) equivalente a 0.91, que corresponde al implementado para evaluar el dengue en

Puerto Rico, para jóvenes menores de 24 años; todas las curvas ROC se establecen por encima de 0.71:

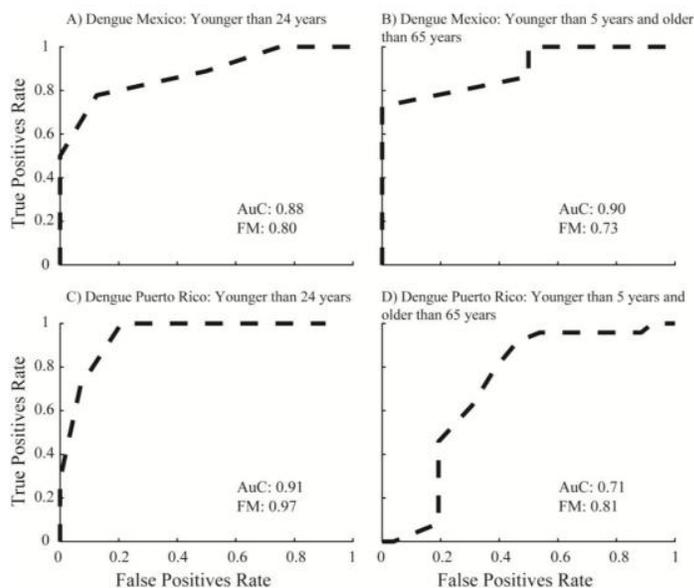


Figura 14. Características de Operación de Receptor (ROC o AUC). Fuente: Laureano, 2018.

Laureano et al. (2018) concluyen que “las RNA fueron capaces de modelar con éxito los brotes de fiebre del dengue en ambas áreas de estudio”, sin embargo, advierten en sus conclusiones que:

Se necesitan más estudios para incorporar la dinámica del virus de la fiebre del dengue y los vectores en los modelos, ya que pueden ayudar a mejorar la habilidad de las simulaciones y comprender enfermedades similares que dependen del clima y cambios ambientales. (p.12).

Como los propios autores lo sindicaron, se requieren más estudios que incorporen la dinámica del virus del dengue en los modelos predictores que, como en este caso, agregan a los análisis variables ambientales y poblacionales. Hay variables como la Temperatura en la Superficie del Mar (TSM) que no corresponden a las condiciones ambientales de la Amazonía peruana, donde se ubica la región San Martín, que es más bien mediterráneo respecto al Mar.

Una serie de tecnologías predictivas son utilizadas en el Estado de Sao Paulo para predecir el incremento de la población de mosquitos *Aedes Aegypti*, generadora del dengue en Brasil. Santos, Reis y Chiaravalloti - Neto (2018), refieren que realizaron “un estudio comparativo de

las predicciones de dengue en la ciudad de Sao Paulo para probar el desempeño de modelos de media móvil integrada autorregresiva de temporada, modelos de aditivos generalizados y redes neuronales artificiales” (p.1).

Respecto a las variables utilizadas en el estudio, los autores refieren fueron tomadas en cuenta factores como la temperatura, la precipitación y la humedad relativa, así como los valores rezagados de los casos de dengue, los que se han reportado como predictores clave; en cuanto a los modelos predictivos, las opciones adoptadas en estudios previos incluyen modelos de Media Móvil Integrada Autorregresiva Estacional (SARIMA), Modelos Aditivos Generalizados (GAM), Redes Neuronales Artificiales (RNA) y un modelo bayesiano espacio-temporal (Santos et al., 2018). Más adelante y de manera concreta se hace mención a cinco modelos, tales como: Modelo Naive, GAM, ANN, SARIMA y modelo Ensemble.

Los mejores modelos GAM, ANN y SARIMA, así como sus parámetros, empleados en el proyecto se muestran en la Tabla 5:

Tabla 5

*Mejores GAM, ANN y SARIMA empleados.*

Model	Description	
GAM	Likelihood:	Poisson
	Spline:	Shrinkage cubic
	Knots:	3
	Predictors:	Clag1, Tmaxlag2, Plag1, RHlag1
ANN	Epochs:	300
	Weights initialization:	Uniform distribution
	Activation	Rectifier
	Optimizer:	Adaptive moment estimation
	Type:	Multilayer perceptron
	Units in hidden layer 1:	20
	Units in hidden layer 2:	10
	Batch size:	50
	Dropout regularization:	0
	Predictors:	Clag1-3, Tlag1-3, Plag1-3, RHlag1-3
SARIMA	Transformation:	Natural logarithm
	Non-seasonal autoregressive order:	0
	Non-seasonal difference order:	1
	Non-seasonal moving average order:	3
	Seasonal autoregressive order:	0
	Seasonal difference order:	0
Seasonal moving average order:	1	

Fuente: Santos, 2018.

Los resultados de los modelos implementados se pueden observar en la Tabla 6, medidos a través de la Raíz de Error Cuadrático Medio (RMSE):

Tabla 6

*Modelos de predicción medidos por la Raíz del Error Cuadrático Medio.*

<b>Model</b>	<b>RMSE*</b>	<b>RMSE / RMSE<sub>Naïve</sub></b>
GAM	2152	0.316
Ensemble	3164	0.465
MLP	4422	0.650
SARIMA	5984	0.879
Naïve	6806	1.000

\* Rounded values.

Fuente: Santos, 2018.

Finalmente, Santos et al. (2018), a partir de la RMSE más baja igual a 2 152, concluyen que:

El modelo GAM tuvo el mejor desempeño e incluyó retrasos en el número de casos y variables meteorológicas, predijo epidemias de una magnitud sin precedentes y se puede actualizar y entrenar en segundos o pocos minutos, las covariables meteorológicas de la ciudad de São Paulo se actualizan diariamente y se puede acceder a ellas sin cargo. Esto significa que el modelo es adecuado para aplicaciones reales para respaldar decisiones oportunas. El modelo puede entrenarse con datos de otras ciudades, pero como las estaciones meteorológicas no están disponibles en todas las ciudades del Estado de São Paulo, se deben encontrar otras fuentes de datos (p.10).

En esencia este fue un estudio para predecir la población del *Aedes Aegypti*, generadora del dengue, pero no implica el diagnóstico de la enfermedad a partir de sus síntomas. Por otro lado, los propios investigadores han explicado que se requieren contar con estaciones meteorológicas como fuentes de datos que muchas veces no están disponibles en las zonas rurales de la Amazonía peruana.

Koh, Spindler, Sandgren y Jiang (2018) realizan un estudio en Singapur, para predecir la fiebre del dengue (DF) a partir de la precipitación semanal, respecto al cual destacan:

La importancia y la utilidad de los datos de la precipitación total semanal, debidamente rezagados, en la predicción de la incidencia semanal del DF en Singapur. También explicamos el uso y las ventajas del paradigma bayesiano en el modelado de series de tiempo. También se introduce el modelo de Red Neuronal Autorregresivo AR (NN (AR)) para el modelado en series temporales de la incidencia de la enfermedad. Finalmente, proponemos un método para la selección de modelos con respecto a la precisión del pronóstico que se puede implementar en cualquier modelo de series de tiempo competidoras y demostramos las consecuencias del sobreajuste en series de tiempo más largas sobre la precisión de la predicción. (p.1).

Se utilizaron dos modelos de predicción, el primero, el Modelo 1 es el modelo AR (2) especificado en la Ecuación (1) y el Modelo 2 que es el modelo NN (AR (2)) especificado en la Ecuación (2), mostrados a continuación, donde la variable de respuesta tiene la siguiente forma:

$$\gamma_t = \mu + \varphi_1\gamma_{t-1} + \varphi_2\gamma_{t-2} + \epsilon_t$$

$$\gamma_t = \beta + \frac{\beta_1}{1 + e^{-(\alpha_0 + \alpha_1\gamma_{t-1} + \alpha_2\gamma_{t-2} + \alpha_3x_{t-3})}} + \epsilon_t$$

De acuerdo a lo expresado por los autores, el modelo NN (AR (2)) ofreció una mayor precisión de pronóstico para series de tiempo con menos puntos de datos que el modelo AR (2). Koh et al. (2018). Sin embargo, estas tecnologías son importantes para la predicción de las poblaciones de dengue, sin tomar en cuenta los síntomas que nos permitan diagnosticar la enfermedad.

El dengue, la chikungunya y el zika son arbovirus transmitidos por mosquitos del género Aedes y han causado varios brotes en el mundo durante los últimos diez años, por eso, la identificación morfológica de los mosquitos está actualmente restringida debido a la pequeña cantidad de profesionales adecuadamente capacitados para enfrentar este embate. Al respecto, los científicos Motta, Bandeira, Winkler, Souza, Dias, Morais, Lins, Kirchner y Badaro (2019) decidieron impulsar una investigación utilizando redes neuronales convolucionales (CNN), para la clasificación de mosquitos adultos en el campo, utilizando imágenes o fotos, de las

especies *Aedes Aegypti*, *Aedes Albopictus* y *Culex Quinquefasciatus*. Según los autores para entrenar a la CNN y realizar la clasificación morfológica automática de los mosquitos, utilizaron “un conjunto de datos que incluía 4,056 imágenes de mosquitos” y “se utilizaron tres redes neuronales, incluyendo LeNet, AlexNet y GoogleNet” (p.1).

Las redes neuronales convolucionales, que no son más que una variación de las redes neuronales perceptrón multicapa, tienen la capacidad de procesar información en forma de imágenes, de manera similar a las neuronas en la corteza visual primaria de un cerebro biológico. Al respecto, los autores mencionados manifiestan que:

Durante un largo período, LeNet fue considerado el estado del arte de las redes neuronales artificiales (ANN). LeNet fue uno de los primeros ANN utilizados para mejorar el algoritmo original de propagación hacia atrás; el notable desarrollo de esta conocida red fue el primer paso hacia la aplicación del método Deep Learning. En 2012, Alex Krizhevsky lanzó AlexNet, que era una versión más profunda y mucho más amplia de LeNet que ganó, por un amplio margen, la difícil competencia de ImageNet. AlexNet tiene una arquitectura muy similar a la de LeNet, pero puede describir mejor las imágenes. Esta red se basa en ocho capas, incluidas las convolucionales, las respuestas locales, la agrupación máxima y las capas totalmente conectadas. En 2014, Christian Szegedy, de Google, comenzó a investigar métodos para reducir la carga computacional de las redes neuronales profundas. Al hacerlo, se centró en la eficiencia de la arquitectura de la red neuronal profunda (nombre en código Inception, más tarde llamado GoogLeNet). Esta red neuronal fue la ganadora de ILSVRC 2014, superando a AlexNet, y se proclamó el nuevo paradigma para las redes neuronales convolucionales. (Motta et al., 2019, p.6).

Los datos y su distribución utilizados en esta investigación se observan en la Tabla 7:

Tabla 7

*Parámetros utilizados como entradas.*

Class name	Total number of images.	Non-application images (used in training)	Application images	Application images used for training and validation	Application images use for testing
<i>Ae- aegypti female</i>	947	723	224	210	14
<i>Ae. aegypti male</i>	282	201	81	67	14
<i>Ae. albopictus female</i>	1050	821	229	215	14
<i>Ae. Albopictus male</i>	436	327	109	95	14
<i>C. quinquefasciatus female</i>	965	895	70	56	14
<i>C. quinquefasciatus male</i>	376	266	110	96	14

Fuente: Motta, 2019.

LeNet y AlexNet tuvieron un rendimiento inferior al de GoogLeNet, lo que según los autores indica que el uso de redes más complejas con más capas es necesario para mejorar la precisión. La clasificación taxonómica de los insectos se puede realizar automáticamente mediante la adquisición de características de la imagen que diferencian a un insecto de otro. Solo utilizando tanto el género como el sexo conseguimos una alta precisión en la identificación de hembras Aedes (Motta et al., 2019).

Este modelo se puede utilizar para mejorar las operaciones de control de vectores que están vinculadas con la identificación rápida y confiable de las especies objetivo y para proporcionar conocimiento de su biología y ecología. En ese sentido el GoogLeNet aporta muchas ventajas, pero resulta inapropiado para diagnosticar el dengue a partir de los síntomas de la enfermedad.

A manera de resumen podemos observar que, en los estudios del dengue utilizando redes neuronales artificiales, en cuatro se tomaron como datos de entrada a factores ambientales; uno los síntomas del dengue exclusivamente; tres los síntomas del dengue mezclados con factores bioeléctricos, de laboratorio o demográficos; uno genético y uno de procesamiento de imágenes fotográficas. Ver Tabla 8.

Tabla 8

*Cuadro resumen de investigaciones considerando RNA.*

<b>Datos de entrada</b>	<b>Cantidad</b>
Factores ambientales	4
Síntomas del dengue	1
Síntomas del dengue, de laboratorio, demográficos y/o bioeléctricos	3
Genético	1
Imágenes fotográficas	1
<b>TOTAL</b>	<b>10</b>

Fuente: Propia.

Como se desprende de la información mostrada, sólo una de las investigaciones tratadas, se realizaron tomando en cuenta datos provenientes exclusivamente de los síntomas físicos

o biológicos de la enfermedad del dengue, como la fiebre, náuseas, vómito, diarrea, heces negras, dolor de cabeza, dolor de abdomen, dolor muscular, manchas rojas y sangrado. Tres consideraron síntomas clínicos del paciente con datos de laboratorio, lo que implica contar con equipos, otros con datos demográficos o en su defecto bioeléctricos que también requieren contar con equipos especializados.

### **Diagnóstico del dengue utilizando lógica difusa**

También se realizó la búsqueda de información científica sobre la aplicación de Lógica Difusa para el diagnóstico del dengue, habiéndose encontrado diez (10) estudios, sobre los cuales damos cuenta a continuación.

En el 2011 se realizó en Kuala Lumpur, capital de Malasia, una importante investigación para diagnosticar tres enfermedades, la varicela, dengue y gripe, utilizando lógica difusa, comparando sus resultados con dos métodos de clasificación estadística utilizados en el diagnóstico de enfermedades, a saber, los clasificadores K-Nearest Neighbor y Naïve Bayes.

Según los autores del estudio, Mahdi, Razali y AlWakil (2011) el procedimiento seguido fue el que se indica:

El primer paso fue utilizar una relación difusa, como la relación de ocurrencia y la de confirmabilidad en una muestra de 149 pacientes que padecen varicela, dengue y gripe de diferentes hospitales y clínicas generales y privadas en Kuala Lumpur para diagnosticar las tres enfermedades. Se utilizaron catorce síntomas en los diagnósticos como fiebre alta, dolor de cabeza, náuseas, vómitos, erupción cutánea, dolor articular, dolor muscular, sangrado, pérdida de apetito, diarrea, tos, dolor de garganta, dolor abdominal y secreción nasal. El segundo paso fue utilizar el método de clasificación de KNearest Neighbor y el método de clasificación de Naïve Bayes en la misma muestra para diagnosticar las tres enfermedades. El paso final fue la comparación entre los tres métodos que utilizan las pruebas de rendimiento, las pruebas de McNemar y Kappa. (p.58).

El método de clasificación K Nearest Neighbor o del vecino más cercano, es un método no paramétrico que a veces se denomina algoritmos de aprendizaje basados en instancias o basados en memoria, ya que lo que hacen es almacenar los datos de entrenamiento en una tabla de búsqueda

e interpolarlos, mientras que el método de clasificación Naive Bayes es un método práctico de aprendizaje bayesiano; se basa en el llamado teorema bayesiano, a pesar de su simplicidad, a menudo puede superar métodos de clasificación más sofisticados (Mahdi et al., 2011).

Realizadas las pruebas, se arribaron a los siguientes resultados: el diagnóstico difuso fue correcto en 136 casos de 149, lo que significa 91%, KNNC fue correcto en 131 casos, lo que significa 88%, mientras que NB fue correcto en 108 casos, lo que significa una precisión del 72.4% (Madhi et al, 2011).

En cuanto a los resultados de precisión, sensibilidad y especificidad los podemos observar en la Tabla 9.

Tabla 9

*Precisión, sensibilidad y especificidad del Fuzzy, KNN y NB.*

<b>Classification Method</b>	<b>Accuracy</b>	<b>Sensitivity</b>	<b>Specificity</b>
<b>FUZZY</b>	<b>0.91</b>	<b>0.94</b>	<b>0.90</b>
<b>KNN</b>	0.88	0.66	0.95
<b>NB</b>	0.72	0.88	0.67

Fuente: Madhi, 2011.

El diagnóstico Fuzzy demuestra tener mejor performance que los otros dos métodos.

Mahdi et al. (2011) manifiestan que “los resultados de la comparación mostraron que el K-Nearest Neighbor y Naïve Bayes Classifier tienen una utilidad limitada y el diagnóstico de enfermedades difusas es mejor y más útil en el diagnóstico de estas enfermedades que los otros dos métodos” (p. 65).

No obstante, debemos tomar en cuenta que los métodos utilizados resultaron bastante útiles para diagnosticar varias enfermedades como la varicela, la gripe y el dengue, enfermedades que dicho sea de paso son bastantes diferentes en sus orígenes, aunque pueden tener síntomas similares; por ejemplo, el dengue se origina por el vector *Aedes Aegypti* y la varicela zoster y la gripe por el virus de la influenza. Sin embargo, el diagnóstico difuso tuvo mejor performance.

Silveira y de Barros (2013) realizaron un estudio espacial y temporal del dengue en el estado de Campiñas, Brasil; según ellos mismos expresan:

Propusimos un modelo matemático integrado (tipo SIR: susceptible, infectado, recuperado) para estudiar la evolución tanto en el espacio como en el tiempo de la enfermedad del dengue. El modelo está dado por ecuaciones diferenciales parciales (PDE) cuyas soluciones numéricas se obtienen mediante esquemas híbridos, lógica difusa y métodos estocásticos... Para la evolución del tiempo, hemos elegido el tercer orden Runge– Kutta TVD (Total Variation Diminishing). Los parámetros inciertos relacionados con el comportamiento de *Aedes Aegypti* son extremadamente importantes para el desarrollo y/o control de la enfermedad. De esta manera, para incorporar esta información en el modelo, los parámetros se estimaron utilizando sistemas basados en reglas difusas e información proporcionada por especialistas. (p.39).

Se trabajaron con poblaciones humanas (susceptible, infectada y recuperada) y poblaciones de mosquitos *Aedes Aegypti* acuáticas (huevo, larva, pupa) y aladas (susceptibles e infectadas). Las implementaciones se desarrollaron en el entorno MATLAB 2010.

Los datos con los que se ha trabajado se pueden observar en la Tabla 10, que da cuenta de las cuatro regiones consideradas del estado de Campiñas, el número de habitantes por región, las áreas consideradas y los contenedores disponibles.

Tabla 10

*Parámetros espaciales y temporales considerados.*

<b>Región</b>	<b>Inhabitants</b>	<b>Area</b>	<b>Containers available</b>
<b>A</b>	184,478	53.2	1081
<b>B</b>	34,220	16.4	311
<b>C</b>	8069	30.7	56
<b>D</b>	21,809	27.4	311

Número de habitantes, área en Km<sup>2</sup> y contenedores por regiones.  
Fuente: Silveira y da Barros, 2013.

El modelo pudo monitorear el número de personas infectadas registradas en la epidemia real, y ha demostrado que, en estos períodos, también se necesitan grandes medidas para reducir los criaderos de mosquitos. A modo de conclusión Silveira y Barros (2013) enfatizan que “las simulaciones que ilustran la evolución espacial/temporal del dengue, según nuestro enfoque, han sido bien aceptadas por los especialistas de salud que trabajan con esta enfermedad”.

La Figura 15 muestra un cuadro de barras comparativo sobre la validez de las simulaciones respecto al proceso epidémico del dengue real.

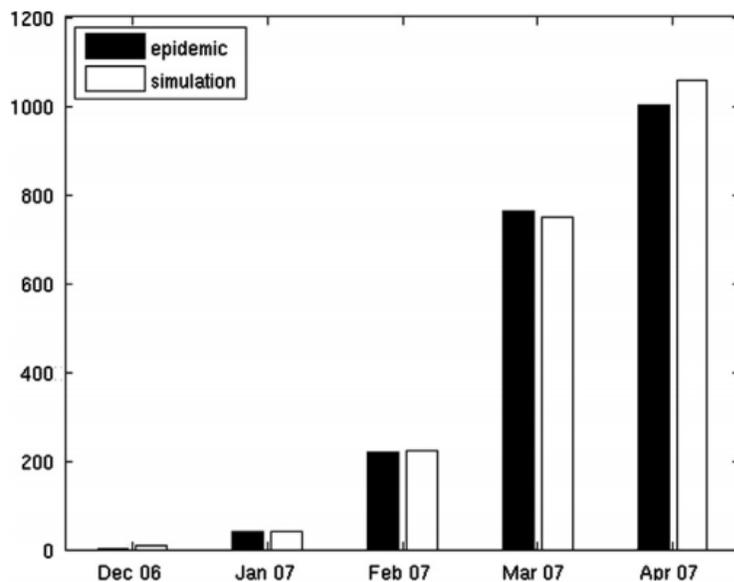


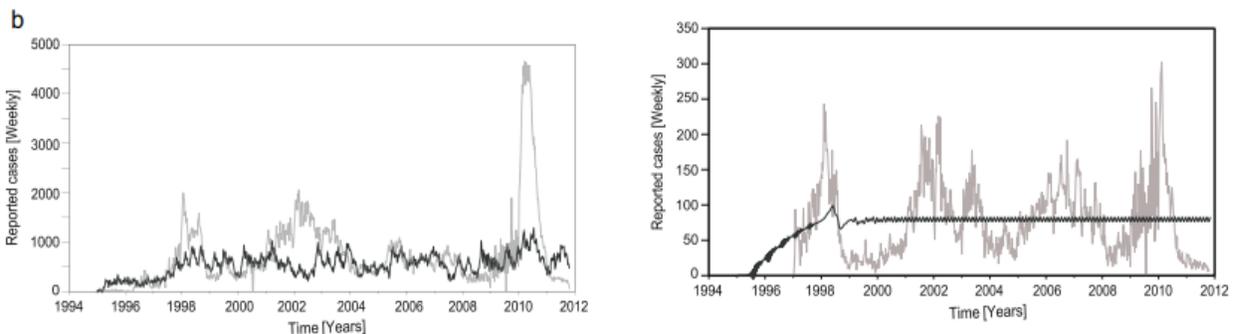
Figura 15. Cuadro comparativo de la epidemia del dengue y la simulación con Lógica Difusa. Fuente: Silveira y da Barros, 2013.

En todo caso, esta aplicación nos permite conocer la impronta del mosquito *Aedes Aegypti* en una zona determinada, a partir de condiciones espaciales (medioambientales) y temporales, lo que nos permite inferir la presencia o no de la enfermedad del dengue en esa zona. En ese propósito, el uso de sistemas de lógica difusa, resulta ser apropiada para determinar la presencia del dengue.

A continuación, se presenta el estudio realizado en Colombia, sobre el uso de un enfoque metodológico que combina el análisis multiresolución y los sistemas difusos para representar los casos de dengue y de dengue grave. En efecto, Torres, Barguil, Melgarejo y Olarte (2014) refieren que:

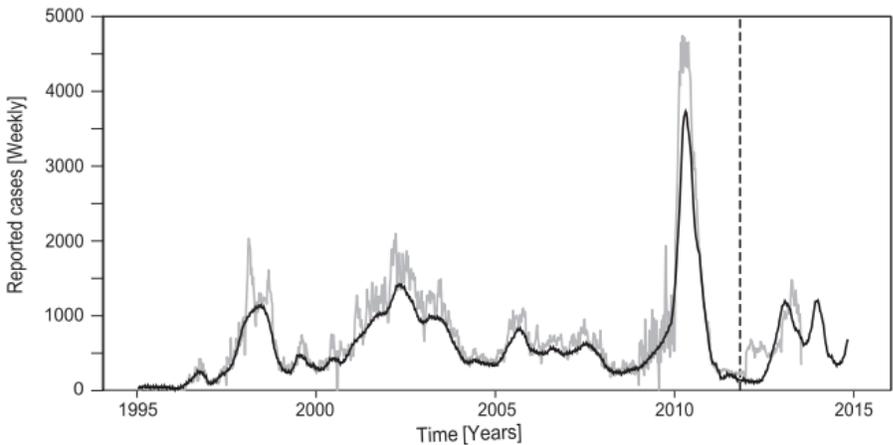
Es bien sabido que los sistemas difusos son apropiados para modelar sistemas con incertidumbre y no linealidad, lo que motivó a los autores a construir un modelo dinámico de dengue y epidemia de dengue grave en Colombia basado en datos reales y utilizando un enfoque difuso. Para lograr esto, se presentó una propuesta metodológica, donde la técnica de identificación de modelado difuso (FMID) y el análisis de resolución múltiple (MRA) se combinaron por primera vez en el análisis epidemiológico. (p.42).

Se realizaron pruebas, para el dengue clásico y el dengue severo. En el primer caso, la mejor performance del modelo FMID se encontró para 32 grupos, falta de claridad 2 y 8 retrasos, con un rendimiento de 21% y un RMSE de 689.38, mientras que, en el segundo caso, el dengue severo, el mejor comportamiento se encontró para 4 grupos, fuzziness 4.1 y 8 retrasos, alcanzando un rendimiento de apenas 18.90 % y un RMSE de 53.49. Como sostienen los autores, “no se encontraron resultados satisfactorios después de una extensa experimentación con la técnica de FMID, ya que los mejores modelos obtenidos dieron un bajo rendimiento con VAF por debajo del 22%” (Torres et al., 2014, p.46). Ver resultados en la Figura 16.



*Figura 16.* Resultados de modelamiento difuso para el dengue (izquierda) y dengue severo.  
Fuente: Torres, 2014.

Mejores resultados se obtuvieron cuando se empleó la combinación de la técnica de identificación de modelado difuso (FMID) y el análisis de resolución múltiple (MRA), los que se pueden observar en la Figura 17. En este caso se obtuvo un rendimiento de 90.06 % para el caso del dengue clásico y de 76.83 % para el dengue hemorrágico; muy superior al modelo FMID solamente.



*Figura 17.* Resultados FMID + MRA, con rendimiento de 90.06 %.  
Fuente: Torres, 2014.

Torres et al. (2014), concluyen que la técnica propuesta permitió la creación de un modelo que representa adecuadamente la dinámica de la epidemia del dengue clásico y el dengue severo en Colombia, dicen que esta técnica logra “un rendimiento significativamente superior al obtenido con las técnicas tradicionales de modelado difuso: La similitud entre los datos originales y la señal aproximada aumenta de 21.13% a 90.06% y de 18.90% a 76.83% en el caso del dengue y el dengue severo, respectivamente” (p.41).

Un rendimiento de 90.06 % para el dengue clásico y de 76.83 % para el dengue hemorrágico o grave es importante sin duda, pero, habrá que ver si existen otras técnicas o modelos de lógica difusa que presenten mejores resultados.

Con el propósito de analizar si la propagación del dengue depende de la disponibilidad de sitios de reproducción favorables para el mosquito *Aedes Aegypti*, se realizó una importante investigación en Phitsanulok (Tailandia), haciendo uso una vez más de la lógica difusa.

Así Shahzad, Tripathi, Faruque, Bajwa, Kitamoto y Souris (2014) plantean que para investigar los diversos factores que influyen en los hábitats de reproducción del mosquito vector, estos pueden cartografiarse desde el espacio, determinando como factores más influyentes “la temperatura, la humedad, las precipitaciones, la densidad de población, la elevación y la cobertura del suelo” (p.1).

El hecho es que se plantearon modelos utilizando parámetros derivados principalmente de imágenes de satélite de libre acceso y software de lógica difusa con sincronización de

parámetros y un algoritmo de predicción basado en la extracción de datos y el método del Arbol de Decisión. Se encontró que los modelos desarrollados eran lo suficientemente flexibles para acomodar parámetros adicionales y datos de muestreo que podrían mejorar la predicción de lugares de reproducción favorables.

Shahzad et al. (2014) advierten que “los objetivos de este estudio fueron crear un modelo de simulación, capaz de predecir posibles hábitats de reproducción de los vectores Aedes al considerar información entomológica (índices de dengue) junto con datos climáticos y ambientales”. Se utilizaron datos de imágenes satelitales o de teledetección que se resumen en la Tabla 11.

Tabla 11

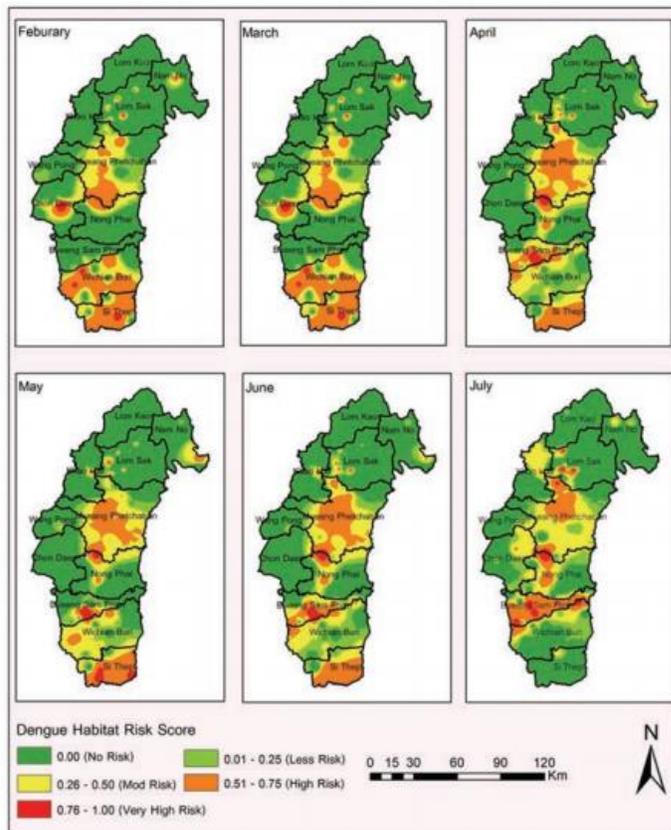
*Detalles de las imágenes de satélite utilizados.*

Source/information	Temperature	Rainfall	Humidity	Elevation
Satellite platform	AQUA	TRMM	AQUA	SRTM
Instrument	MODIS	Precipitation radar	AIRS	Synthetic aperture radar
Image name	MYD11C3	3B43	RetQuant	SRTM_57_09
Spatial resolution	0.05 degree	0.25 degree	13.5 km	90 m
Availability	Day/Night	3-hourly	5 days	11-day mission
Version	Version 5	Version 6	Version 5	Version 1

Fuente: Shahzad, 2014.

Entre los diversos modelos de hábitat y los sistemas automáticos de predicción de brotes desarrollados, el modelo de predicción de enfermedades transmitidas por vectores es considerablemente complejo, ya que requiere evaluación y predicción bajo incertidumbre basada en factores sociales, climáticos, ambientales y de comportamiento. Sarfraz (como se citó en Shahzad et al., 2014). De otro lado se sabe que el proceso de jerarquía analítica (AHP) es una forma sólida de abordar diversos problemas, ya que este método es cualitativo y fácil de implementar desde el punto de vista de la validación y el requisito de datos. Sin embargo, el AHP no es útil para resolver problemas de predicción basados en la incertidumbre, como los brotes de dengue. Afirman los autores que “con el fin de excluir esta restricción, desarrollamos el enfoque de aptitud de hábitat basado en difusos, una técnica que elimina el requisito de agregación adicional, así como los procedimientos de clasificación” (Shahzad et al., 2014, p.8).

La Figura 18 muestra los resultados de la probabilidad de hábitat del dengue debido a los factores considerados en el modelo. Se encontró que el período de abril - julio es un período de gran idoneidad, después de lo cual comienzan las fuertes lluvias y el número de hábitats de reproducción exitosos disminuye repentinamente; el modelo propuesto puede adoptarse para monitorear la reproducción de vectores de dengue durante todo el año, lo que permite a los administradores de salud monitorear las áreas favorables para la reproducción de mosquitos diariamente e instituir medidas inmediatas de precaución. (Shahzad et al., 2014).



*Figura 18.* Estimación de probabilidad de hábitat del *Aedes Aegypti*. El rojo implica mayor riesgo. Fuente: Shahzad, 2014.

Como disciernen los autores, el estudio enfatiza la posibilidad de utilizar datos climáticos gratuitos basados en satélites e integrarlos con datos geoespaciales recopilados en el terreno. El enfoque reduce la dependencia de la recopilación de datos de campo de los parámetros climáticos de las estaciones terrestres que requieren mucho tiempo y de forma extenuante y la integración con los datos de salud, lo que reduce el retraso en la predicción.

La utilización de datos geoespaciales y de detección remota con métodos estadísticos adecuados, como el análisis de lógica difusa, podría predecir con eficacia la tendencia de las enfermedades transmitidas por vectores. (Shahzad et al., 2014).

Como podemos colegir de este apretado resumen sobre el modelo implementado, la lógica difusa es una herramienta informática muy importante para predecir la probabilidad de ocurrencia del dengue, a partir de variables de entrada medio ambientales, obtenidas a través de imágenes vía satélite. Pero, así como es posible precisar esta probabilidad, las mismas también pueden ser obtenidas para otras enfermedades que tienen como causa la presencia del *Aedes Aegypti* en algunas zonas del planeta. Pero, habría que evaluar su eficacia para diagnosticar a partir de los síntomas de la enfermedad cuando se da la emergencia.

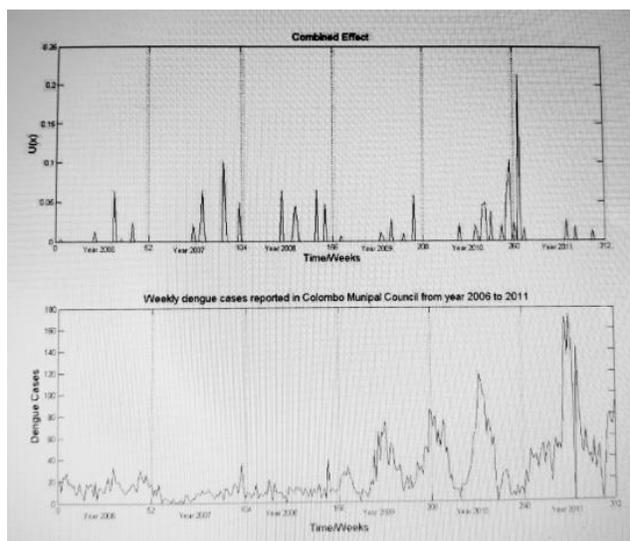
En un estudio realizado en Colombo (Sri Lanka), los autores Wickramaarachchi y Jayasinghe (2015) manifiestan que:

La dinámica de la transmisión de la enfermedad del dengue depende de varios factores externos, como el clima, el comportamiento humano, la geografía y la demografía. Modelar la dinámica del dengue es difícil con los enfoques clásicos debido a esta complejidad. La lógica difusa y la teoría de conjuntos difusos son una herramienta útil en matemáticas para modelar sistemas bajo incertidumbre. (p.20).

En el presente caso, para el análisis, se toma en cuenta el efecto de la lluvia (RF, rainfall) y la temperatura (TEMP), como factores determinantes de la enfermedad del dengue. Asignando al operador difuso las siguientes inferencias:

- a. Si RF es desfavorable y TEMP es desfavorable, entonces puede ser altamente desfavorable para la transmisión del dengue.
- b. Si RF es desfavorable y TEMP es favorable o la RF es favorable y la TEMP es desfavorable, entonces puede ser favorable para la transmisión del dengue.
- c. Si RF es favorable y TEMP es favorable, entonces puede ser altamente favorable para la transmisión del dengue.

Los efectos de la simulación de la lluvia y la temperatura, para la presencia del dengue en la zona de Colombo, se observa en la Figura 19 que se muestra a continuación.



*Figura 19.* Simulación difusa de efectos combinados y casos de dengue.  
Fuente: Wickramaarachchi, 2015.

Las observaciones indicaron que los valores de membresía simulados por el operador de Hamacher varían entre 0 y 0.2 a lo largo del período comprendido entre el año 2006 y 2011 considerado para este estudio. Esto implica que la medida desfavorable es muy pequeña y la precipitación y la temperatura apoyan la transmisión del dengue durante todo el período de tiempo que se tomó en el estudio.

La teoría de conjuntos difusos puede ser sumamente útil cuando se modelan y analizan sistemas inciertos y complejos como la propagación de enfermedades. La teoría de los operadores difusos se puede usar para investigar el potencial general de la transmisión del dengue de varios parámetros sensibles como la lluvia y la temperatura. Consideramos el efecto general de la lluvia y la temperatura para producir condiciones ambientales desfavorables para la transmisión del dengue. (Wickramaarachchi y Jayasinghe, 2015).

Se debe considerar que, en este documento, solo se ha tomado en cuenta los efectos de la lluvia y la temperatura en el del modelo informático. No se han tomado en cuenta factores

como la movilidad humana, la demografía y la geografía que son sensibles a la dinámica de la transmisión de la enfermedad del dengue, mucho menos los síntomas de la enfermedad.

En otro artículo se propone un análisis de la evolución del riesgo de dengue, a partir de un modelo del tipo Takagi-Sugeno, en el que la consecuencia de cada regla difusa es una ecuación diferencial parcial (PDE). Silveira y de Barros (2015) establecen: “los parámetros de incertidumbre  $k_i$ , relacionados con el riesgo de propagación del dengue, están directamente relacionados con el comportamiento de la población de mosquitos y se evaluaron utilizando un sistema basado en reglas difusas e información proporcionada por expertos” (p.1).

En el estudio, realizado en la ciudad de Campiñas, estado de Sao Paulo (Brasil), los autores refieren que los parámetros dependen de la población humana, que proporciona sangre para la maduración de los huevos y de los posibles contenedores de reproducción de mosquitos y aún depende de la lluvia, explican así que la cantidad de lluvia presenta dependencia estocástica en los valores muestreados y por esta razón es que se elige el método de la Cadena de Markov para modelarla.

Silveira y de Barros (2015) consideraron varios aspectos de la población humana involucrada que se tuvieron en cuenta, como el sexo, el nivel de educación, o si no ha tenido dengue, suministro de agua, presencia de potenciales criaderos de mosquitos, entre otros. Con ellos fue posible generar un mapa de riesgo de dengue en la región, los mismos que fueron transcritos al lenguaje MATLAB obteniendo el mapa que se muestra en la Figura 20, donde los riesgos más altos están asociados con los colores cálidos, los tonos rojizos.

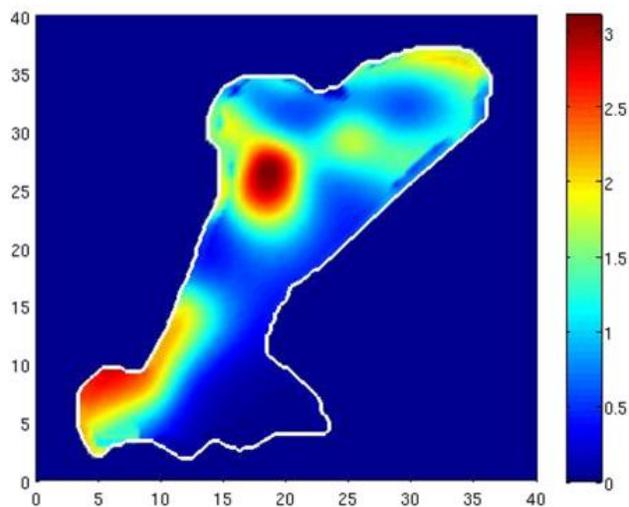


Figura 20. Mapa de riesgo del dengue. Fuente: Silveira y de Barros, 2015.

Sobre la base del modelado matemático y su simulación con MATLAB, Silveira y de Barros (2015) concluyen que:

Una buena estrategia para combatir la propagación del dengue es invertir en campañas para reducir o eliminar los receptores de la cría de mosquitos *Aedes Aegypti*. Entonces, aunque en el presente trabajo existen herramientas matemáticas sofisticadas, los resultados tienen interpretaciones intuitivas y las autoridades de salud pública pueden indicar posibles políticas de salud pública. Como trabajo futuro, podemos sugerir el desarrollo de otro modelo de Takagi-Sugeno, teniendo como uno de los insumos el riesgo ya analizado y cuyos resultados son PDE, que modelan poblaciones de mosquitos humanos y/o infectados. En este caso, es importante considerar el término advectivo y estimar el parámetro, dadas las características de vuelo de los mosquitos”. (p.12).

Como en el caso de investigación anterior el modelo Takagi-Sugeno de la lógica difusa es utilizado para diagnosticar el dengue, a partir de variables ambientales como la lluvia y, algunos aspectos culturales como el hecho de mantener recipientes con agua durante mucho tiempo, que hacen proclives a los diferentes hábitats de sostener criaderos de *Aedes Aegypti* propagadores de la enfermedad.

Un sistema experto difuso, para predecir de manera temprana el dengue, fue implementado en Taiwan a partir de síntomas físicos y de pruebas de laboratorio. Saikia y Dutta (2016), los autores, señalan que:

El sistema de inferencia difusa, se utiliza un conjunto de reglas para representar el conocimiento o los datos de un problema en particular... este trabajo tuvo como objetivo diseñar un sistema experto para el diagnóstico temprano de la enfermedad del dengue utilizando el Sistema de Inferencia Fuzzy (FIS), una herramienta poderosa para tratar la imprecisión y la incertidumbre. El FIS diseñado se puede usar para el diagnóstico temprano de la enfermedad del dengue de un paciente utilizando sus síntomas físicos e informes de pruebas médicas como variables de entrada y convirtiendo estas variables de entrada en funciones de membresía difusa. (p.1).

Se seleccionaron once signos, de los cuales seis corresponden a síntomas físicos del dengue y cinco pruebas de laboratorio sobre la misma enfermedad, los que se enumeran a continuación:

1. Síntomas clínicos:
  - a) Fiebre.
  - b) Síntomas gastrointestinales (náuseas, diarrea, vómito).
  - c) Dolor de cabeza.
  - d) Dolores (mialgias, artralgias, dolor óseo).
  - e) Erupción cutánea.
  - f) Dolor de bola ocular.
2. Hallazgos de Laboratorio:
  - a) Leucopenia (GB<4000mm<sup>3</sup>).
  - b) Trombocitopenia (PLT<100,000/mm<sup>3</sup>).
  - c) Disfunción hepática (AST/ALT>30U/L).
  - d) Proteína C reactiva (CRP<20mg/L).
  - e) Tiempo de tromboplastina parcial (PTT<30s).

Basándose en los efectos de los parámetros de entrada en el parámetro de salida, se generaron 860 reglas mediante el proceso de IF-THEN (Si - Entonces), utilizando una conexión (AND) para las variables de entrada. Para efectos de simulación se utilizó la opción de visor de reglas de lógica difusa de MATLAB. El rango para la variable de salida del sistema de inferencia de dengue se considera como (0-30) y en base a esto, se puede tomar una decisión sobre tres clasificaciones de diagnóstico (no dengue, dengue probable, confirmar dengue) para un conjunto particular de síntomas de entrada. Ver Figura 21.

Val	Fever (35-40)	GI (0-11)	Headache (0-11)	Pains (0-11)	Skin Rash (0-11)	Eye Ball pain (0-11)	WBC (0-11)	PLT (0-11)	AST/ALT (0-11)	CRP (0-11)	PTT (0-11)	Output (0-30)	Remarks
Val.1	35.5	4	4.50	7	3.3	8	2.7	1.4	1.5	2.4	1.7	2.09 (6.9%)	No Dengue
Val.2	36.9	5.6	1.5	5.5	10	8.5	2.8	10.5	4.3	2.4	10.5	7.62 (25.4%)	Probable Dengue
Val.3	38.7	7.6	7.3	7	3.3	7.5	2.8	1.7	1.4	6	6	10 (33.33%)	Probable Dengue
Val.4	36.9	5.3	5.6	1.5	5.5	10	8.5	10.5	4.3	10	10.5	18.2 (60.67%)	Dengue Confirm
Val.5	38.7	7.6	7.3	7	3.3	7	2.5	1.7	1.5	2.4	1.7	2.1 (7%)	No Dengue
Val.6	37.6	6	5.2	8.6	5.6	6	8	3.3	10	9.5	10	20.2 (67.33%)	Dengue Confirm
Val.7	36.2	3	5.1	4.3	4.4	4.5	4.5	3.5	3.3	4.6	9	2.96 (9.8%)	No Dengue
Val.8	36.8	5.2	5.6	5.9	6	1.5	8.7	1.7	2.5	2.4	1.7	6.87 (22.9%)	Probable Dengue
Val.9	37.5	6	6	6	6	6	6	6	6	6	6	23 (76.67%)	Dengue Confirm

Figura 21. Resultados de simulación en MATLAB. Fuente: Saikia y Dutta, 2016.

En conclusión, los síntomas físicos solos pueden realizar el diagnóstico precoz de pacientes con probabilidad de dengue, pero, en áreas remotas, “uno puede realizar el diagnóstico considerando los síntomas físicos utilizando este sistema de inferencia difuso diseñado en ausencia de un médico. Pero para el resultado de la confirmación, el proceso de diagnóstico requiere síntomas físicos y resultados de exámenes médicos” (Saikia y Dutta, 2016, p. 6).

Refieren los autores que, el diagnóstico del dengue basados en lógica difusa, considerando solamente los síntomas de la enfermedad, sin atender a las pruebas de laboratorio, inexistente muchas veces en las zonas rurales de la región San Martín en Perú, solo es apreciable en las etapas tempranas de la enfermedad, para que los médicos y los propios pacientes puedan tomar medidas preventivas para afrontar los aspectos más graves de la enfermedad. Se advierte que no se puede realizar la clasificación a nivel de riesgo, es decir, fiebre del dengue (DF), fiebre hemorrágica del dengue (DHF) y síndrome de shock del dengue (DSS). Esto sucede porque los parámetros de entrada del sistema diseñado no son suficientes para ese propósito.

Interesante es el estudio sobre la presencia del dengue hemorrágico realizado por Jayasundara, Perera, Malavige y Jayasinghe (2017), quienes manifiestan que un mecanismo clave de la gravedad “es la fuga de líquido de los vasos sanguíneos a los tejidos circundantes y la disminución resultante en los volúmenes dentro del compartimento vascular y la hipotensión, agregando que esto ocurre durante aproximadamente 48 horas y se conoce como fase crítica” (p.2). Tomando en cuenta de que actualmente no hay medicamentos específicos contra la enfermedad, el diagnóstico clínico temprano y el manejo cuidadoso de los fluidos corporales son fundamentales para el cuidado de los enfermos grave.

En la línea de Jayasundara et al. (2017):

El aumento de la permeabilidad vascular es la causa principal de la DHF y se cree que las citoquinas, los mediadores de lípidos inflamatorios y el antígeno NS1 del dengue contribuyen significativamente a este aumento de la permeabilidad vascular. Por lo tanto, varios estudios han intentado identificar las relaciones entre las citoquinas y el dengue. En este estudio, hemos intentado usar varias citoquinas y otros mediadores inflamatorios para desarrollar un modelo matemático para predecir la probabilidad de desarrollar DHF.

Para este modelo, hemos elegido tres citoquinas y dos mediadores de lípidos inflamatorios debido a su asociación con la fuga vascular en el dengue y también con la enfermedad clínica grave". (p.2).

Las tres citoquinas o citosinas más los dos mediadores son la Sphingosine 1-phosphate (S1P), Interleukin-  $1\beta$  (IL- $1\beta$ ), Tumor Necrosis Factor (TNF- $\alpha$ ), Platelet Activating Factor (PAF) and Interleukin -10 (IL-10); utilizados como parámetros de medición. Al respecto, los autores expresan que en este estudio, el objetivo es desarrollar un modelo matemático que pueda detectar a los pacientes que pasan al nivel de DHF en una etapa temprana mediante el análisis del efecto combinado de los parámetros de Esfingosina 1-fosfato (S1P), Interleukin- $1\beta$  (IL- $1\beta$ ), Factor de Necrosis Tumoral (TNF- $\alpha$ ), Factor de Activación de Plaquetas (PAF) e Interleukin -10 (IL-10); recientemente se demostró que concentraciones más altas de antígeno NS1 y niveles séricos de IL-10 se asocian con una enfermedad clínica grave en la infección aguda por dengue. La agrupación jerárquica se utiliza para detectar factores que se correlacionan entre sí. Sus interacciones se mapean utilizando mecanismos de lógica difusa con la combinación de operadores de Hamacher y OWA modificado. Los resultados de la aplicación muestran que la precisión de este modelo para predecir el nivel de severidad del dengue es de 71.43% a las 96 horas desde el inicio de la enfermedad, 85.00% a las 108 horas y 76.92% a las 120 horas. Se detecta una región de ambigüedad en el modelo para el rango de valores de 0,36 a 0,51. Jayasundara et al. (2017).

Así, los resultados muestran un modelo matemático robusto que explica la evolución del dengue a sus formas serias en pacientes individuales con alta precisión. Sin embargo, este modelo debería mejorarse aún más al incluir parámetros adicionales y debe validarse en otros conjuntos de datos, toda vez que tiende a sobrestimar el riesgo de los pacientes con DF, lo cual implica una gran limitación.

En abril del 2018, se realizó una revisión sistemática de bibliografía científica, para determinar la contribución de la utilización de métodos de lógica difusa en el diagnóstico de enfermedades en diferentes prácticas médicas. Se seleccionaron ocho bases de datos científicas como PubMed, IEEE, Science Direct, Web of Science, Taylor & Francis, Willey Online Library, Emerald y Google Scholar.

El análisis de métodos difusos aplicados en el diagnóstico de enfermedades por disciplinas médicas se puede ver en la Figura 22. Las disciplinas médicas más estudiadas con

lógica difusa fueron cardiología, neurología, reumatología y enfermedades infecciosas; todo esto entre los años 2004 al 2018. Se analizaron 46 artículos de los cuales el único tema estudiado a través de la lógica difusa referido al dengue fue la de los autores Saikia y Dutta (2016) titulado “Early Diagnosis of Dengue Disease Using Fuzzy Inference System”, que ya ha sido tratado in extenso en líneas anteriores.

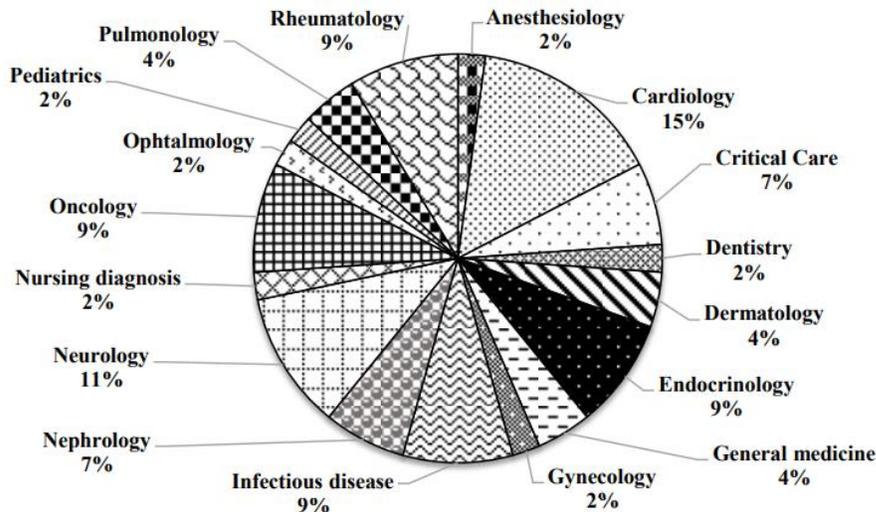


Figura 22. Enfermedades estudiadas por la lógica difusa. Fuente: Ahmadi, 2018.

Ahmadi, Gholamzadeh, Shahmoradi, Nilashi y Rashvand (2018), al respecto de lo expuesto, dicen que “el hallazgo más obvio que se desprende de este estudio es que la aplicación del método difuso, además de mejorar el diagnóstico de enfermedades, puede proporcionar una detección temprana de la enfermedad para prevenir la progresión de enfermedades complejas” (p.33). Otro punto importante a recordar es que se identificaron 13 métodos difusos que son los más utilizados en estudios recientes y se representaron dichos resultados en base a esta clasificación.

En Malasia, se aplicaron dos técnicas para obtener un diagnóstico temprano a los pacientes, ya sea que se sospeche que tienen fiebre del dengue o no, a saber, el sistema experto difuso y la técnica de extracción de dato. Según Husin, Alharogi, Mustapha, Hamdan y Amalina Husin (2018), se trata de “desarrollar un modelo preciso que pudiera detectar mejor los signos y síntomas tempranos de la fiebre del dengue y un sistema práctico para la auto notificación de la enfermedad” (p.2); en dicho modelo las reglas del diagnóstico de dengue

basadas en una entrevista con un médico (experto) se aplicarán en un sistema experto utilizando una lógica difusa, pero, antes de aplicar las reglas extraídas, la precisión de las mismas serán probadas a través de la herramienta de minería de datos.

La arquitectura del sistema fuzzy logic en esta investigación se muestra en la Figura 23, y, el diagrama de flujo correspondiente, que funciona como una guía para el sistema de diagnóstico de los pacientes con dengue y notifica si necesitan buscar atención médica, en la Figura 24.

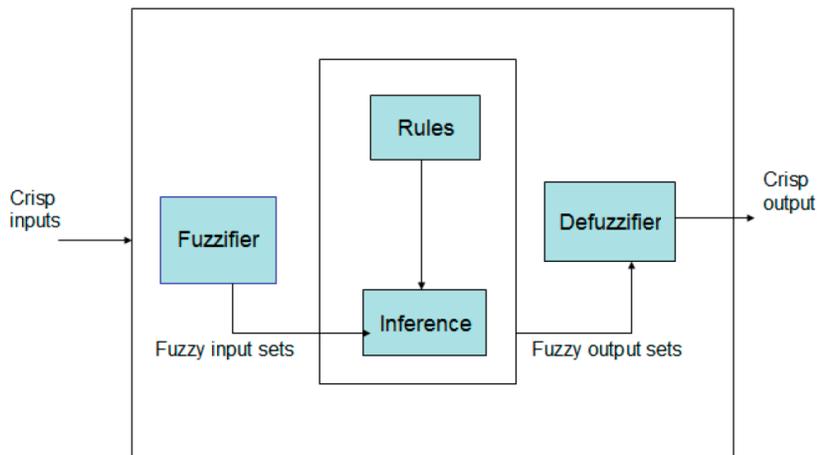


Figura 23. Arquitectura de un sistema fuzzy logic. Fuente: Husin, 2018.

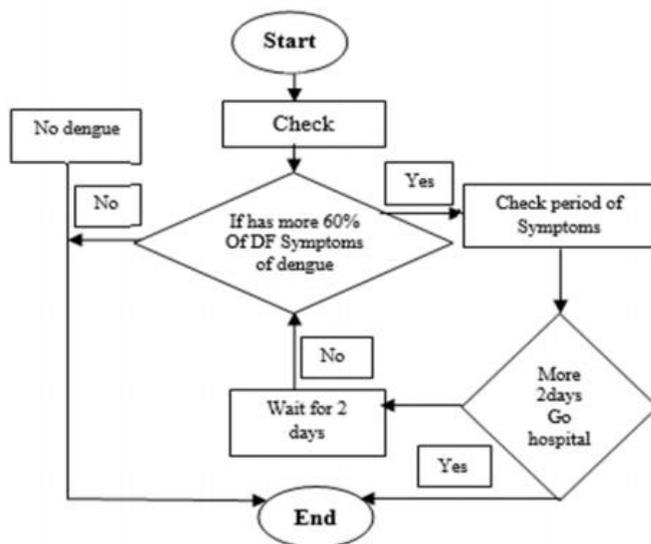


Figura 24. Diagrama de flujo del sistema que determina la presencia o ausencia de dengue. Fuente: Husin, 2018.

Los autores anuncian que se utiliza una técnica de minería de datos para extraer reglas precisas, a través de la conversión de las reglas a un conjunto de datos para examinar la herramienta de minería de datos que ha sido entrenada por un dataset de dengue, para tal efecto, se ha entrevistado a dos expertos de la Universidad de Malaya y obtener la información correcta sobre la enfermedad del dengue, así como, proporcionar las sugerencias correctas para la enfermedad del dengue. (Husin et al., 2018).

Los síntomas del dengue, considerados por los autores, luego de entrevistar a los médicos expertos se sintetizan en: Fiebre súbita y alta, dolor severo en articulaciones y músculos, vómitos, erupción cutánea, dolor detrás de los ojos, fatiga, dolores de cabeza severos, náuseas y pérdida de apetito. También se consideran los síntomas de la fiebre del dengue hemorrágico: Dolor abdominal, vómitos persistentes, acumulación de fluidos, sangrado bucal interno y agotamiento. Después de realizar la sesión de entrevista “hemos extraído 20 reglas sobre DF y DHF” (Husin et al., 2018, p.6).

Los resultados obtenidos por la investigación se observan en la Tabla 12.

Tabla 12

*Evaluación de entrenamiento y pruebas del modelo difuso.*

<b>Evaluation training dataset</b>	<b>Training rate</b>	<b>Testing rate</b>
Correctly Classified Instances	1055 (100%)	29 (100%)
Incorrectly Classified Instances	0 (0%)	0 (0%)
Kappa statistic	1	1
Mean absolute error	0.0009	0.0009
Root mean squared error	0.0011	0.0011
Coverage of cases (0.95 level)	100%	100%
Mean rel. region size (0.95 level)	50%	50%
Total number of Instances	1055	29

Se observa 100% de efectividad en entrenamiento y 100% en las pruebas.  
Fuente: Husin, 2018.

Husin et al. (2018), explican la evaluación de la capacitación y los resultados de las pruebas, deduciendo que:

Estos resultados muestran la clasificación del conjunto de datos cerca del 100% para la capacitación, lo que significa que el conjunto de datos está limpio y la máquina ha detectado cada atributo del

conjunto de datos. De este experimento también se muestra la clasificación del conjunto de datos para probar las reglas del médico igual al 100%, lo que significa que el resultado asegura que nuestro conjunto de datos es muy correcto y se ajusta al conjunto de datos de entrenamiento. Además, estos resultados también nos dicen que las reglas que se han extraído de las entrevistas con los médicos tienen una gran precisión. (p.7).

Este sistema ayuda a los médicos a facilitar su trabajo de diagnóstico del dengue y proporciona una plataforma práctica para que los pacientes infectados obtengan la información de si necesitan buscar atención médica inmediatamente antes de que los síntomas empeoren. El hecho de que el propio paciente pueda auto diagnosticarse es una gran ventaja, sin embargo, eso implica que debe tener algún conocimiento de computación y contar con las herramientas de cómputo necesarios, no siempre disponible en la selva rural del Perú.

A modo de resumen, en la Tabla 13 observamos el conjunto de diez investigaciones consideradas para el estudio del dengue, desde la perspectiva de la lógica difusa. Cuatro de ellas corresponden a estudios considerando como entradas a factores ambientales; dos que toman en cuenta exclusivamente los síntomas físicos o biológicos de la enfermedad; uno que considera los síntomas más datos de laboratorio (que requiere contar con equipos especiales); uno de genética y dos que consideran datos estadísticos (antecedentes de la enfermedad) y otros.

Tabla 13

*Investigaciones del dengue utilizando lógica difusa.*

<b>Datos de entrada</b>	<b>Cantidad</b>
<b>Factores ambientales</b>	<b>4</b>
<b>Síntomas del dengue</b>	<b>2</b>
<b>Síntomas del dengue y laboratorio</b>	<b>1</b>
<b>Genética</b>	<b>1</b>
<b>Datos estadísticos (antecedentes) y otros</b>	<b>2</b>
<b>TOTAL</b>	<b>10</b>

Fuente: Propia.

De ellos, interesa resaltar los procesos que consideran solo a los síntomas del dengue, como por ejemplo el estudio de Mahdi et al. (2011), aplicado para discriminar a pacientes afectados con varicela, gripe y dengue, que consideran como variables a evaluar la fiebre alta, dolor de cabeza, náuseas, vómitos, erupción cutánea, dolor articular, dolor muscular, sangrado, pérdida de apetito, diarrea, tos, dolor de garganta, dolor abdominal y secreción nasal. También la investigación realizada por Husin et al. (2018), que consideran como datos de entrada materia de análisis, luego de entrevistar a expertos, la fiebre súbita y alta, dolor severo en articulaciones y músculos, vómitos, erupción cutánea, dolor detrás de los ojos, fatiga, dolor de cabeza severos, náuseas y pérdida de apetito, así como, síntomas que corresponden al dengue grave o hemorrágico como dolor abdominal, vómitos persistentes, acumulación de fluidos, sangrado bucal interno y agotamiento.

### **Diagnóstico del dengue utilizando algoritmos genéticos**

La computación evolutiva también viene siendo utilizada como una herramienta informática para investigar, modelar y diagnosticar enfermedades, como por ejemplo los algoritmos genéticos (AG). A continuación, presentamos algunos casos aplicables al diagnóstico del dengue.

Gerardi y Monteiro (2011), sostienen que la propagación de enfermedades infecciosas es uno de los principales problemas de salud pública en el mundo y que el Autómata Celular (CA) “se ha empleado para modelar la propagación de tales enfermedades, para predecir epidemias y evaluar la eficacia de las estrategias de control”. Expresan dichos autores que:

En un modelo epidemiológico de CA, cada célula que compone la red puede corresponder a un individuo único o a una fracción mayor de la población. En cada paso de tiempo, cada celda se encuentra en uno de tres estados: S, I o R. El estado S representa una celda que es susceptible y, por lo tanto, está sujeta a ser infectada; el estado I está relacionado con una célula que es infecciosa y, por lo tanto, puede transmitir la enfermedad a las susceptibles; el estado R está asociado con la eliminación del proceso de propagación de la enfermedad, lo que significa que la célula está curada e inmunizada o está muerta. Tal modelo se conoce generalmente por la abreviatura SIR. Si la recuperación de la infección no confiere inmunidad, el modelo se llama SIS. (p.1).

En el estudio que analizamos, la evolución temporal del número de individuos con fiebre de dengue registrados semanalmente en la ciudad de Río de Janeiro, Brasil, durante 2007, se utiliza para identificar los modelos de SIS (susceptibles-infecciosos-susceptibles) y SIR (susceptibles-infectados-eliminados) formulado en términos de autómatas celulares CA.

En el proceso de identificación, se utiliza un algoritmo genético (AG) para encontrar las probabilidades del estado de transición  $S \rightarrow I$  capaz de reproducir en la red de CA las series históricas de 2007. Luego, estos modelos epidemiológicos creados mediante la combinación de CA y AG se emplean para predecir los casos de personas enfermas en 2008. (Gerardi y Monteiro, 2011).

Los resultados revelan que la elección de la estructura del modelo es fundamental. Los modelos SIS parecen ser más convenientes para la identificación, mientras que los modelos SIR para la predicción. En este marco, basado en CA y AG, refieren los autores que sería pertinente aplicar a otras enfermedades y ciudades, para verificar la validez de las conclusiones sobre el rendimiento de los diferentes tipos de modelos empleados en esta investigación.

El virus del dengue (DENV) es, según Velandia y Castellanos (2011):

Un virus icosaedro de 50 nm, aproximadamente, conformado por una membrana lipídica (obtenida de las células del huésped), sobre la cual se insertan las proteínas de membrana y de envoltura. El interior del virus contiene el complejo ribo proteico conformado por la proteína de la cápside y el genoma viral que consiste en una única hebra de ARN de sentido positivo que codifica para un polipéptido único, que contiene tanto las proteínas estructurales, que harán parte de la partícula viral, como las proteínas no estructurales, que intervienen durante los procesos de ensamblaje y replicación del ARN genómico". (p.34).

Esquemáticamente, podemos observar el DENV en la Figura 25.

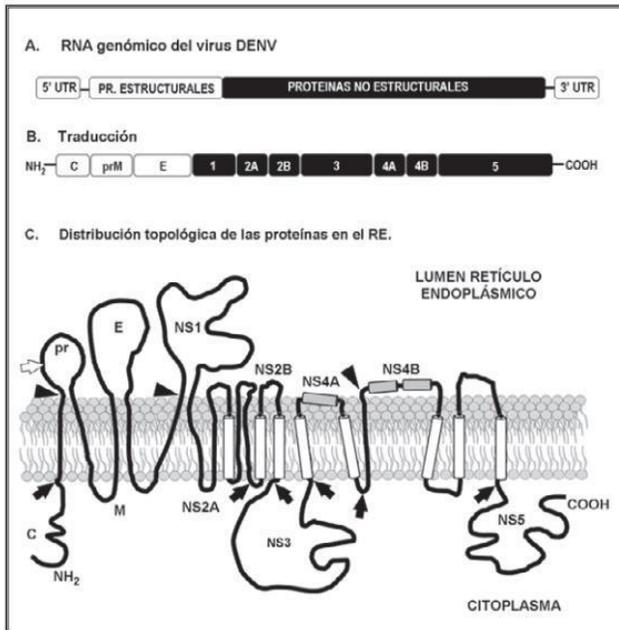


Figura 25. Organización del genoma DENV.

Se observa, el esquema de la organización del genoma del virus del dengue. Se puede notar que es un ARN de cadena sencilla con un único marco abierto de lectura que comienza hacia el extremo 5' no traducido y que contiene los genes estructurales y no estructurales. También se ve el esquema de la poliproteína sintetizada que muestra la organización de las proteínas estructurales (C, prM y E) y las no estructurales (NS1, NS2A, NS2B, NS3, NS4A, NS4B, NS5), así como, la topología propuesta de las proteínas estructurales y no estructurales del DENV en el retículo endoplásmico.

Al respecto, Velandia y Castellanos (2011) explican que el DENV es un flavivirus que ingresa por endocitosis y “se replica en el citoplasma de la célula infectada, originando tres proteínas estructurales y siete proteínas no estructurales, sobre las cuales se conocen sólo algunas de sus funciones en la replicación viral o en la infección”. (p.33).

El estudio de los aspectos virales del dengue, tratado como un polipéptido (molécula formada por aminoácidos), son abordados en el siguiente estudio. En efecto Silla, Nunes, Cormanich, Guerreiro, Ramalho y Freitas (2011) plantean la aplicación de modelos predictivos, intervalo PLS (iPLS), algoritmo genético (AG) y selección de predictores ordenados (OPS) para seleccionar los descriptores de MIA (Multivariate Image Analysis) apropiados para modelar constantes cinéticas, a saber, tasa de escisión del sustrato ( $K_{cat}$ ) y las constantes de Michaelis ( $K_m$ ), que se correlacionan con las bioactividades de los péptidos contra el Dengue

tipo 2 (DEN-2). Donde MIA-QSPR es el análisis de imagen multivariado de la QSPR (Quantitative Structure - Property Relationship), que es un proceso indirecto mediante el cual una estructura química (virus del dengue) se correlaciona cuantitativamente con cualquier propiedad o actividad biológica, ya sea mediante la medición o el cálculo de parámetros relacionados con la bioactividad, es decir, los llamados descriptores.

En el presente caso, las imágenes son estructuras químicas bidimensionales, que tienen correlación con las bioactividades pépticas y que, computacional y numéricamente, deben entenderse como un conjunto de píxeles negros (donde existe el dibujo) y píxeles blancos (espacio de trabajo en blanco, donde no existen los dibujos), es decir, son binarios.

Para evitar las variaciones espurias, en términos de Silla et al. (2011), se emplearon tres métodos de selección en este trabajo: Intervalo PLS (iPLS), algoritmo genético (AG) y selección de predictores ordenados (OPS). iPLS es un método basado en la división de todo el conjunto de variables independientes en intervalos más cortos, seguido de la obtención de modelos de regresión PLS para cada intervalo. El intervalo elegido será uno con el error más pequeño con respecto al modelo construido sin dividir el conjunto de datos. El método AG bien conocido se basa en la teoría evolutiva de Darwin. Las variables que producen modelos de calibración con errores menores en cada generación se seleccionan y representan a los individuos más apropiados. Las variables seleccionadas por AG se pueden usar para construir modelos de calibración PLS (AG-PLS) o como datos de entrada para iPLS (AG-iPLS) para obtener modelos de calibración con capacidad predictiva mejorada. El algoritmo OPS también ha emergido como una alternativa útil para este propósito. OPS consta de tres pasos: (1) obtención de un valor informativo; (2) orden decreciente de variables por este valor; (3) investigación de las variables ordenadas.

En general, los autores llegan a la conclusión de que:

Las predicciones de la constante de Michelis ( $k_m$ ) no fueron tan precisas como las de la tasa de escisión ( $k_{cat}$ ), lo que sugiere que estos parámetros no se correlacionan linealmente. Podemos encontrar que  $k_{cat}$  vs.  $K_m$  de hecho, no se correlacionan ( $r^2 = 0.27$ ), lo que indica que los mecanismos para la unión del sustrato y la escisión son independientes, pero se capturó una alta correlación con los descriptores de MIA seleccionados, especialmente para  $k_{cat}$ , por lo tanto, los mejores modelos MIA-QSPR se pueden usar de manera confiable para lograr  $k_{cat}$  y  $K_m$  de nuevos candidatos a fármacos. (Silla et al., 2011, p.149).

En este estudio, como se ha mencionado, el objeto de análisis no han sido los síntomas de la enfermedad del dengue o los factores ambientales, sino la evolución los aspectos genéticos del DENV considerando su estructura polipéptido y la forma de afrontarla.

Una aplicación de los algoritmos genéticos (AG) para la prevención del dengue en Indonesia lo observamos en el siguiente estudio, que trata sobre una estrategia de vacunación óptima utilizando una función objetiva no lineal. Widayani, Kallista, Nuraini y Sari (2014) plantean el uso de técnicas de programación de algoritmos genéticos que se combinan con el método de Runge - Kutta de cuarto orden para construir la vacunación óptima, con el fin de prevenir el dengue, así, los autores sostienen que “la creciente circulación de los cuatro serotipos del dengue significa que se necesita una vacuna que proteja contra los cuatro; por lo tanto, la vacuna debe ser tetravalente” (p.2). Las vacunas contra el dengue en desarrollo son de cuatro tipos: Virus vivos atenuados, virus vivos atenuados quiméricos, vacunas inactivadas o subunidades y vacunas basadas en ácidos nucleicos. En el futuro, la vacuna se producirá en gran número para minimizar la propagación de la infección por fiebre de dengue hemorrágico (FHD).

De acuerdo a los expuesto por Widayani et al. (2014), la propagación del modelo de infección:

Se puede construir simplemente utilizando el modelo SIR. En el modelo SIR, la población humana como huésped, se divide en 3 grupos: (Huésped sensible), el grupo de personas susceptibles, pero potencialmente afectadas por la enfermedad; (huésped infectado), los grupos de personas que están enfermas o infectadas con virus que causa la enfermedad, y, (huésped recuperado): los grupos de personas que se recuperan de la enfermedad. (p.2).

El modelo matemático se muestra a continuación:

$$\frac{dS_H}{dt} = \mu_H N_H - \beta_H b \frac{S_H}{N_H} I_V - \mu_H S_H - \tilde{\alpha}$$

$$\frac{dI_H}{dt} = \beta_H b \frac{S_H}{N_H} I_V - (\mu_H + \gamma_H) I_H$$

$$\frac{dR_H}{dt} = \gamma_H I_H - \mu_H R_H$$

$$\frac{dV_H}{dt} = \tilde{\alpha} - \mu_H V_H$$

$$\frac{dS_V}{dt} = \mu_V N_V - \beta_V b \frac{S_V}{N_V} I_H r - \mu_V S_V$$

$$\frac{dI_V}{dt} = \beta_V b \frac{S_V}{N_V} I_H r - \mu_V S_V$$

Se asumió que la población total es constante, es decir:

$$S_H + I_H + R_H + V_H = N_H \text{ y } S_V + I_V = N_V$$

De acuerdo a la información suministrada por los autores, la persona que ha sido vacunada pertenece al compartimento de vacunación ( $V_H$ ), por lo que el modelo se convierte en un SIRV. La suma de  $S_H$ ,  $I_H$ ,  $R_H$  y  $V_H$  es el número de población huésped en la muestra ( $N_H$ ). Al considerar al mosquito como un vector, la población de mosquitos se divide en dos compartimentos, es decir, vector susceptible ( $S_V$ ) y vector infectado ( $I_V$ ). Se asumió que no hay un compartimento recuperado en la población de mosquitos.

Los resultados de la aplicación del algoritmo genético, que se usa para generar un valor aleatorio como una solución para una función objetivo, y, el método Runge – Kutta para resolver numéricamente ecuaciones diferenciales, se observan en la Tabla 14, para diversos valores del parámetro  $d$ .

Tabla 14

*Comparación del host infectado total para cada valor del parámetro “d”.*

Month	I without vaccine	I with vaccine		
		$d$		
		0.02	0.03	0.04
4	0.0892	0.0862	0.0856	0.0868
8	0.0719	0.0657	0.0641	0.0667
12	0.0571	0.0481	0.0463	0.0487
16	0.0453	0.0342	0.0324	0.0335
20	0.0359	0.0236	0.0220	0.0219
24	0.0285	0.0156	0.0144	0.0136

Fuente: Widayani, 2014.

El algoritmo genético y el método Runge - Kutta se implementan para encontrar la solución al problema de la propagación de la infección por dengue, optimización la vacunación contra la enfermedad, que se puede formular a través de un modelo matemático. Widayani et al. (2014) sostienen que “la mejor solución con el costo mínimo de vacunación se obtiene cuando el parámetro  $d = 0.02$ . Además del costo, la reducción de casos de epidemia y el total de personas que se vacunan también se puede usar como una solución óptima para este problema” (p.4).

Este procedimiento, como se ha podido observar, centra su atención en la vacunación de las personas del área de influencia de la enfermedad en Indonesia para prevenir el dengue. Como se sabe, no existen vacunas que puedan curar la enfermedad, pero podemos entender que, si existen algunas que pueden prevenir la enfermedad, como ya se indicó líneas arriba de hasta cuatro tipos. De acuerdo a las conclusiones del estudio este fenómeno se puede modelar matemáticamente y a partir de ahí buscar respuestas positivas frente a la enfermedad, por lo menos aminorar su impacto negativo.

La siguiente investigación tiene por finalidad estudiar técnicas de optimización multiobjetivo para ayudar a resolver problemas relacionados con el control del mosquito que transmite la fiebre del dengue. Para tal efecto proponen un algoritmo genético multiobjetivo (MGA-DENGUE) para resolver el modelo de optimización y discutir los resultados computacionales obtenidos de la aplicación de esta técnica.

De acuerdo con Florentino, Cantane, Santos y Bannwart (2014), “es muy difícil controlar o eliminar los mosquitos *A. Aegypti* con estas medidas de control. Esto se debe al hecho de que tienen adaptaciones al entorno que las hacen altamente resistentes” (p.77). Por experiencia se sabe que el mosquito tiene la capacidad de recuperarse rápidamente a los números iniciales después de las perturbaciones resultantes de fenómenos naturales (por ejemplo, sequías) o intervenciones humanas como la fumigación.

En Brasil, los principales controles utilizados actualmente por la Superintendencia de Control de Enfermedades Endémicas (SUCEN) son:

Control físico, realizado por funcionarios de salud pública y residentes mediante la eliminación de la reproducción; y control químico, realizado mediante la aplicación de insecticidas. Los insecticidas desempeñan un papel importante en el control de los vectores del dengue. Sin embargo, con el tiempo, los vectores

adquieren resistencia, lo que requiere un aumento de los productos químicos, aumenta el costo de control y afecta la salud pública. (Florentino et al., 2014, p.77).

Como dicen los autores, en la investigación “se estudia la dinámica poblacional del mosquito que transmite la fiebre del dengue para comprender el fenómeno epidémico y sugerir nuevas estrategias de programación multiobjetivo para el control del mosquito” (Florentino et al., 2014, p.78). Asimismo, se propone un algoritmo genético multiobjetivo (MGA-DENGUE) para resolver el modelo de optimización y se discuten los resultados computacionales obtenidos de la aplicación de esta técnica.

En esta propuesta, el ciclo de vida del mosquito se consideraba dividido en dos fases: la fase acuática, cuando los mosquitos se encuentran en las etapas de huevo, larva y pupa; y la fase terrestre, cuando los mosquitos están en la etapa adulta. El tamaño de la población de mosquitos en la fase acuática en el momento  $t$  se denota por  $A(t)$ . Los mosquitos en la fase acuática entran en la fase terrestre a una tasa de  $\gamma$  per cápita, con  $r$  proporción de hembras y  $(1 - r)$  de machos. En la fase terrestre, los mosquitos adultos silvestres se dividieron en tres compartimentos: hembras inmaduras (antes del apareamiento),  $I$ ; insectos machos naturales (o salvajes),  $M$ ; y hembras fertilizadas,  $F$ . Al insertar mosquitos estériles en el sistema, se crean dos compartimentos más: insectos estériles (irradiados o transgénicos),  $S$ ; y hembras no fertilizadas (hembras eliminadas),  $U$ , que son las hembras que se aparean con machos estériles. (Florentino et al., 2014). El modelo matemático se muestra a continuación.

$$\begin{aligned}\frac{dA}{dt} &= \phi \left(1 - \frac{A}{C}\right)F - (\gamma + \mu_A)A \\ \frac{dI}{dt} &= r\gamma A - \left[\frac{\beta M}{M+S} + \frac{\beta_S S}{M+S} + \mu_1 + u_1\right]I \\ \frac{dF}{dt} &= \frac{\beta M I}{M+S} - (\mu_F + u_1)F \\ \frac{dM}{dt} &= (1-r)\gamma A - (\mu_M + u_1)M \\ \frac{dS}{dt} &= \mu_2 - (\mu_S + u_1)S\end{aligned}$$

La variable de estado  $U$ , relacionada con los mosquitos hembra eliminados, está desacoplada del sistema dinámico descrito por:

$$\frac{dU}{dt} = \frac{\beta_S SI}{M + S} - \mu_U U - u_1 U$$

Todo ello, tomando en cuenta a las variables de control  $u_1 = u_1(t)$  y  $u_2 = u_2(t)$ , asociadas respectivamente con el control químico y el control biológico, es decir, con controles de insecticida y mosquitos estériles.

Proponemos el algoritmo genético multiobjetivo (MGA-DENGUE), argumentan Florentino et al. (2014), para resolver el problema de optimización. En este algoritmo, se genera una población de soluciones (individuos)  $u_1(t)$ ,  $u_2(t)$  para el problema y esta población evoluciona según los operadores genéticos (selección, cruce y mutación) para promover una tendencia de las soluciones para representar soluciones eficientes o no dominadas que son cada vez más efectivas y más variadas a medida que continúa el proceso evolutivo.

En MGA\_DENGUE, el cruce se efectúa mediante la elección aleatoria de los dos individuos en una población intermedia y dos puntos de cruce ( $r_1$  y  $r_2$ ) (cruce de dos puntos) y, por lo tanto, las características de dos soluciones se combinan para formar dos descendientes (solución 3 y solución 4). Entre los cuatro individuos (solución 1, solución 2, solución 3 y solución 4), se eligen dos que tienen la mejor aptitud para el retorno de la población, y estos individuos reemplazan a los dos peores. (Florentino et al., 2014). Ver Figura 26.

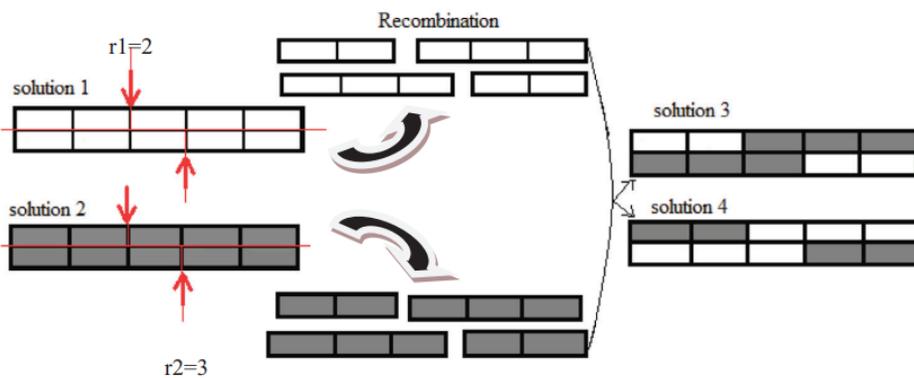


Figura 26. Operación de cruzamiento en individuos. Fuente: Florentino, 2014.

En general, los resultados sobre los casos evaluados indican que se debe aplicar una mayor cantidad de insecticida durante 10 ó 20 días, disminuyendo esta cantidad después de este período y luego se deben insertar mosquitos machos estériles hasta el día 70. En el siguiente período debe disminuir la cantidad de mosquitos estériles e insertar una pequeña cantidad de insecticida, como se muestra en la Figura 21 (a) y (b). Los valores pequeños de  $u_1(t)$  significan aplicar una pequeña cantidad de insecticida en el tiempo “t” y, el valor grande de  $u_2(t)$ , significa insertar una gran cantidad de densidad de mosquitos machos estériles en el tiempo “t”. Además, una comparación de los casos estudiados sugiere que cuando se necesita disminuir la cantidad de mosquitos estériles, se debe aumentar la cantidad de insecticida y viceversa, para que la población natural de mosquitos continúe disminuyendo con el tiempo.

Otro hecho es que la Figura 27 (de c a f) indica que el control con insecticida es más efectivo en términos de reducir el número de individuos en la población de mosquitos que el control con machos estériles. Pero, por otro lado, la combinación de estos dos controles tiene la ventaja de reducir el uso de insecticida y, en consecuencia, reduce el daño que puede causar en el ambiente, los animales y los seres humanos y no induce resistencia a los insecticidas en la próxima generación de mosquitos. (Florentino et al., 2014).

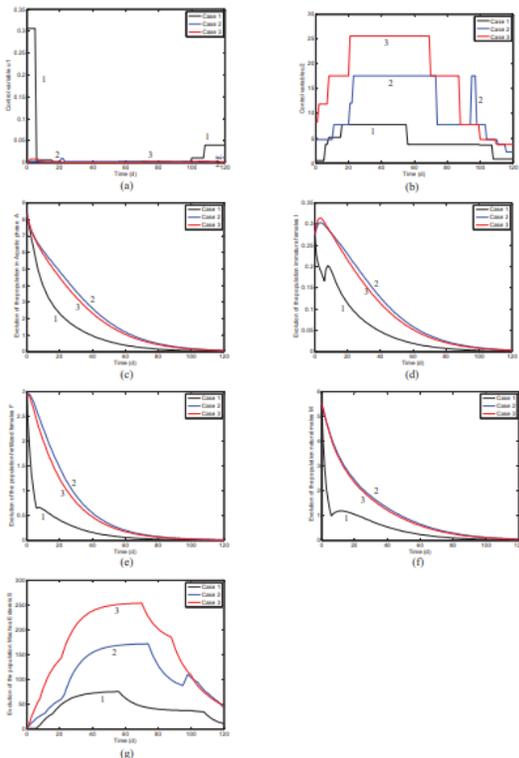


Figura 27. Evolución de la población de mosquitos con insecticida y esterilizados.

A modo de conclusión y en términos de Florentino et al (2014) podemos decir que el multiobjetivo propuesto:

Es una herramienta que ofrece una diversidad de soluciones, lo que brinda al responsable de la toma de decisiones una explicación de todas las posibles combinaciones optimizadas utilizando insecticidas y machos estériles que pueden fabricarse... En el presente trabajo, los problemas de optimización multiobjetivo se ejemplifican con dos funciones de rendimiento, pero MGA-DENGUE se puede implementar fácilmente con tres o más funciones, de acuerdo con el interés del usuario y sin ningún cambio en su estructura. El algoritmo genético es fácil de implementar y se ha demostrado que es una buena herramienta para determinar los controles óptimos para el problema en cuestión. (p.84).

A partir del largo proceso que hemos comentado, podemos concluir que los algoritmos genéticos multiobjetivo (MGA-DENGUE), constituyen una gran herramienta tecnológica e informática para predecir las poblaciones de dengue, tanto si han sido pasibles a los efectos de insecticidas o fueron parte de procesos de esterilización, y controlar su impronta, mitigando de esta manera sus efectos en el surgimiento del dengue en poblaciones vulnerables.

Otra aplicación de los algoritmos genéticos en el estudio del dengue es el realizado por Zhu, Saw, Nalaparaju, Grüber y Lu (2017), en Singapur. Al respecto, los autores expresan que:

El virus del dengue (DENV) tiene cuatro serotipos antigénicamente diferentes, DENV-1 a DENV-4, y causa enfermedades virales potencialmente mortales... La proteína 5 no estructural (NS5) es una enzima clave en la replicación viral, asociada a diversas actividades requeridas para la replicación del genoma, la limitación del genoma y la modulación de la respuesta inmune del huésped, por lo que se ha considerado un objetivo prometedor para el desarrollo de inhibidores antivirales” (p.3).

Se trata de una aplicación de una dispersión de rayos X de ángulo pequeño (SAXS por sus siglas en inglés, a la proteína no estructural 5 (NC5) del virus del dengue 2 (DENV 2), y su modelado por mínimos cuadrados no negativos (NNLS) o el método de optimización de conjunto (EOM) estándar basado en algoritmo genético. (Zhu et al., 2017).

En este estudio, se adoptó un modelo de CG (Coarse - grained o Grano Grueso) para estudiar la proteína de múltiples dominios DENV 2 NS5 en base de datos experimental SAXS.

El modelo CG se implementa aquí porque: i) es difícil modelar proteínas tan grandes como NS5 en simulaciones de todos los átomos que hace a la carga computacional costosa, y ii) SAXS ofrece información estructural en resolución media o baja en lugar de nivel atómico.

Zhu et al. (2014) manifiestan que la construcción del conjunto con regularización de baja energía se enfocó en el conjunto conformacional con energía favorable y alta consistencia con el experimento, “identificando exitosamente la interfaz de dominio principal y la interfaz de contacto de dominio de DENV 2 NS5” (p.33). Siendo que la estrategia de regularización de soluciones en este trabajo es general, ya que puede implementarse con diferentes algoritmos de adaptación y diversos enfoques de simulación.

Observamos pues, que el algoritmo genético es una herramienta poderosa para modelar fenómenos moleculares (peptídicos) como los del dengue, para predecir y controlar el virus del dengue en particular el serotipo DENV 2 NS5.

A modo de resumen presentamos la Tabla 15, que muestra cinco investigaciones realizadas utilizando los algoritmos genéticos (AG) como herramienta para el diagnóstico del dengue.

Tabla 15.

*Investigaciones sobre el dengue utilizando algoritmos genéticos.*

<b>Datos de entrada</b>	<b>Cantidad</b>
<b>Genéticos</b>	<b>4</b>
<b>Vacunación</b>	<b>1</b>
<b>Biológicos (esterilizaciones e insecticidas)</b>	<b>1</b>
<b>TOTAL</b>	<b>6</b>

Fuente: Propia.

Como se puede observar, en el estudio del dengue, los algoritmos genéticos básicamente se utilizan para analizar los efectos que tienen sobre la enfermedad la información genética contenida en los diferentes serotipos del virus del dengue (DENV). También se evalúan el uso de vacunas, aunque los autores no han sido específicos en cuanto a la composición química de éstas, y, el uso de hembras *Aedes Aegypti* esterilizadas, así como, de insecticida que, definitivamente, tienen efectos en el control de la población de mosquitos y, por ende,

en control de la enfermedad. No se han encontrado estudios que utilizan algoritmos genéticos que tomen en cuenta de manera exclusiva los síntomas clínicos del dengue.

## CAPÍTULO III

### MARCO METODOLÓGICO

#### Diseño de la investigación

Se realizó una investigación aplicada ex post facto, descriptiva y transversal. En términos de Padrón (2006) la investigación aplicada, aplicativa o simplemente “aplicaciones”, es aquel tipo de estudios científicos orientados a resolver problemas de la vida cotidiana o a controlar situaciones prácticas, dentro de las cuales se pueden distinguir dos sentidos específicos:

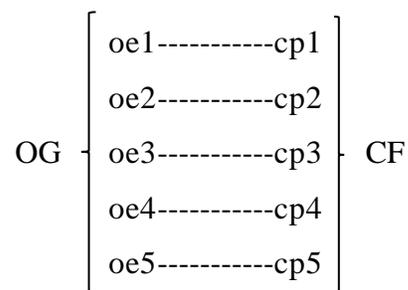
- a) Aquella que incluye cualquier esfuerzo sistemático y socializado por resolver problemas o intervenir situaciones, aunque no sea programático, es decir, aunque no pertenezca a una trayectoria de investigaciones descriptivas y teóricas. En ese sentido se concibe como *investigación aplicada* tanto la innovación técnica, artesanal e industrial como la propiamente científica.
- b) Aquella que sólo considera los estudios que explotan teorías científicas previamente validadas para la solución de problemas prácticos y el control de situaciones de la vida cotidiana. En este sentido sólo son *investigaciones aplicadas* las que se enmarcan dentro de una secuencia programática de búsquedas que tienen como origen el diseño de Teorías científicas.

Desde el punto de vista metodológico, Padrón propone la siguiente estructura general para las investigaciones aplicativas:

1. Se parte de una situación problemática, que requiere ser intervenida y mejorada. Se describe sistemáticamente esa situación, bajo criterios relevantes.
2. Se selecciona una TEORÍA, de carácter reconocido y suficientemente aceptada. Se expone sistemáticamente esa teoría, tanto en sus conceptos centrales como en sus rasgos contextuales.
3. Se examina la situación descrita en '1' a la luz de la teoría expuesta en '2'. Se deriva de esta teoría un sistema de acciones y de previsiones (prototipo) que, si se pone en práctica, resolvería favorablemente la situación problemática descrita en '1'. Se describe sistemáticamente este prototipo, tanto en sus secuencias de uso como en sus instrumentaciones asociadas.
4. Se ensaya y se prueba el prototipo que se obtuvo en el paso '3' (y que se obtuvo sobre la base de una aplicación de la teoría, '2', a la situación problemática '1'). Se determina la probabilidad que tiene el modelo aplicativo de resolver la clase de situaciones problemáticas del tipo descrito en '1'.

### **Prototipo de la investigación**

Para el diseño de la investigación, planteamos el esquema de una investigación por objetivos tal como se muestra a continuación:



Donde:

OG = Objetivo General

oe = Objetivo específico

cp = Conclusión parcial

CF = Conclusión Final

Para el primer objetivo específico, se realizó el estudio del arte para el diagnóstico de enfermedades virales producidas por el mosquito Aedes 2005 en, similares al dengue, utilizando las redes neuronales artificiales, algoritmos genéticos y lógica difusa; a fin de determinar el número de estudios, características de las tecnologías informáticas y logros obtenidos.

En el segundo objetivo específico, se trata de realizar un análisis de las técnicas informáticas evaluadas en el estudio del arte, con el propósito de determinar la técnica informática que no ha sido suficientemente estudiada o algún aspecto de ella, que requiere ser aplicada, para evaluar sus resultados en el diagnóstico del dengue en la región San Martín.

En cuanto al tercer objetivo específico, se busca identificar los síntomas del dengue sin alarma, con alarma y grave, en la región San Martín, con el propósito de analizar sus particularidades, así como, las coincidencias y diferencias respecto a otras enfermedades virales como la chicungunya y el zika. La información así obtenida sirvió para efectos de experimentación y simulación de la herramienta computacional elegida para estudiar el problema de investigación.

El cuarto objetivo específico, hace referencia al tratamiento y el análisis de datos estadísticos obtenidos de la Dirección Regional de Salud de San Martín, respecto a la incidencia del dengue en la región San Martín del Perú. En la práctica, se trata de aplicar el concepto de minería de datos, a fin de descubrir conocimientos y patrones de conducta de la enfermedad viral, para obtener el modelo de diagnóstico del dengue que se requiere.

Por último, en el quinto objetivo específico, se trata del desarrollo de un prototipo óptimo, obtenido a partir de la técnica informática de inteligencia artificial definida como la más adecuada para el diagnóstico del dengue. En el presente estudio, la técnica informática elegida es la Red Neuronal Artificial – Multi Layer Perceptron (RNA-MLP); siendo necesaria evaluar la arquitectura pertinente, el acondicionamiento de la red feedforward, su entrenamiento, validación y testeo utilizando como herramienta el MATLAB y su simulación.

### **Diagrama de flujo de diseño del prototipo**

A partir del Estudio del Arte se pudo conocer las diferentes investigaciones realizadas desde diferentes estrategias de inteligencia artificial (redes neuronales artificiales, lógica difusa y algoritmos genéticos), para diagnosticar el dengue, evaluando acto seguido los requerimientos necesarios para el diseño de un prototipo a partir de los síntomas clínicos del dengue, encontrando

que existen deficiencias en el diagnóstico del dengue utilizando redes neuronales artificiales con dichos síntomas. Esto nos lleva a la necesidad de implementar un prototipo de red neuronal artificial multicapa, backpropagation, que requiere de un proceso iterativo de entrenamiento, validación y test (prueba) para determinar la arquitectura de red de mejor performance para la simulación y operación del modelo. El diagrama de flujo se muestra en la Figura 28.

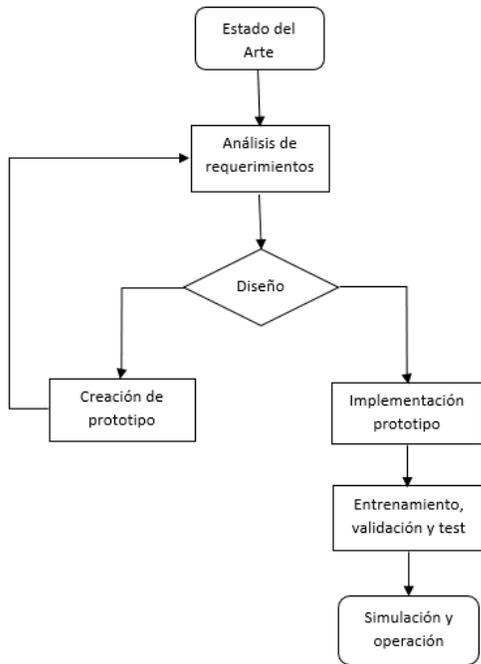


Figura 28.- Diagrama de flujo de diseño del prototipo.

### Selección de la muestra

Para el periodo en estudio, año 2017 al 2019, se cuenta con una población de 807 casos, distribuidos en 397 casos confirmados como dengue y 410 no confirmados o que corresponden a otras enfermedades virales. Debido a la situación de pandemia por el COVID -19 que afecta al mundo, América Latina y el Perú, no ha sido posible contar con información oficial sobre el dengue desde marzo del 2020 hasta la actualidad. La muestra para una población finita según Wittkoski (2010) es la que se muestra a continuación:

$$n = \frac{N \cdot Z^2 \cdot p \cdot q}{e^2 \cdot (N - 1) + Z^2 \cdot p \cdot q}$$

Donde:

n = Tamaño de la muestra.

N = Población o universo.

Z = Nivel de confianza (95%) = 1.96

p = Probabilidad a favor (50%) = 0.5.

q = Probabilidad en contra = (1-p) = 0.5

e = error muestral (5%) = 0.05.

En consecuencia, la muestra obtenida es: n = 261.

Considerando que la muestra obtenida es superior al 5% de la población de casos de dengue, se obtiene la muestra modificada ( $n'$ ) aplicando el factor de corrección (f) siguiente:

$$f = \sqrt{\frac{N - n}{N - 1}}$$

De donde se obtiene  $f = 0.823$ . En consecuencia, la muestra modificada es:

$n' = n(f) = 347 (0.823) = 214.8 = 215$  casos de dengue.

$n' = 215$ .

Finalmente, la muestra definitiva corresponde a  $n' = 215$  casos de dengue.

## Recolección de los datos

### Técnicas.

La recolección de datos se hizo en forma ordenada y sistemática, para lo cual se utilizó las siguientes técnicas:

- Análisis documental (revisión de la Data de la Dirección Regional de Salud - DIRESA).
- Análisis de resultados.

### **Instrumentos.**

Los instrumentos que se utilizaron para la recolección de datos fueron:

- Fichas de observación en Excel.
- Importación de datos de Excel a MATLAB.
- Exportación de datos de MATLAB a Excel.

### **Análisis de los datos**

El diseño de un modelo del diagnóstico del dengue basado en inteligencia artificial, requiere que los datos obtenidos contengan las características necesarias para su procesamiento y de ser el caso para su simulación en alguna herramienta de computación. En ese sentido, fue necesario realizar el pre procesamiento de los datos, la determinación de patrones para validación y la segmentación de los datos para su aplicación en MATLAB.

### **Pre procesamiento de datos.**

En el presente caso, fue necesario realizar el pre procesamiento de los datos obtenidos en Excel de la Dirección Regional de Salud (DIRESA) de la región San Martín, correspondiente a los años 2017, 2018 y 2019, toda vez que ellas contenían información del dengue sin signos de alarma con código de enfermedad A97.0, con signos de alarma (A97.1) y grave (A97.2), así como, de otras enfermedades virales similares, como la fiebre de la chikungunya (A92.0), fiebre amarilla selvática (A95.0) y zika (U06.9).

### **Determinación de patrones para validación.**

Como se indicó líneas arriba, se utilizó la técnica de minería de datos, con el propósito de conocer y determinar patrones de conducta de la enfermedad viral, necesarios para la aplicación de la técnica informática de inteligencia artificial correspondiente.

En ese sentido, se identificaron veinticuatro (24) síntomas en total, de los cuales once (11) corresponden al dengue sin síntomas de alarma, dieciséis (16) al dengue con síntomas de alarma, y, veinticuatro (24) que corresponden al dengue grave.

La determinación de patrones, que resultan de la combinación de síntomas en presencia y ausencia de la enfermedad viral, son de lo más importante, toda vez que nos permiten analizar la capacidad de procesamiento de datos del algoritmo correspondiente al sistema informático en uso.

### **Segmentación de datos.**

Para su aplicación en RNA MLP, utilizado en el presente estudio, se requiere la segmentación de los datos para su entrenamiento, validación y test. El comando feedforward nos permite hacer dicha operación en forma automática, haciendo que la data de Patrones se distribuya en partes disponibles para el entrenamiento, validación y test, como parte del proceso de aprendizaje de la red neural.

### **Determinación de la arquitectura de red neural**

En este acápite se debe determinar el número de neuronas que inicialmente deben tener las capas de entrada, oculta y de salida de la RNA MLP elegida, en el presente caso la red feedforward. Es preciso señalar que, por defecto, la red feedforward establece que el número de neuronas en la Capa Oculta es 10. Considerando que para el estudio del dengue se cuenta con 24 síntomas, se requiere que en la Capa de Entrada existan 24 neuronas de entrada, una por cada síntoma detectado. Asimismo, tomando en cuenta que el resultado esperado es confirmar la presencia o ausencia del dengue, solo se requiere una neurona en la Capa de Salida. En consecuencia, la arquitectura de red neural asumida inicialmente fue la siguiente:

Arquitectura RNA: **24 – 10 – 1.**

Sabido es que, la RNA MLP feedforwardnet, por defecto, trabaja con funciones de transferencia y/o activación Sigmoidea Tangente Hiperbólica (tansig) para las capas ocultas y Función de Transferencia Lineal (purelin) para las capas de salida.

Sabido es que toda arquitectura de red neural requiere ser acondicionada para su funcionamiento, lo cual implica realizar pruebas con diferentes tipos de funciones de transferencia, algoritmos de entrenamiento y pruebas de rendimiento, con sus respectivos ajustes.

### **Entrenamiento y simulación de la red neural**

Finalmente, se necesita realizar pruebas de entrenamiento iterativas, muchas pruebas, la cual requiere a su vez ejecutar cambios recurrentes en el número de neuronas de la capa oculta, inclusive con más de una capa oculta, hasta encontrar la arquitectura que mejor responde al propósito de diagnosticar el dengue de manera óptima. El entrenamiento, validación, testeo y simulación de la RNA elegida se realiza utilizando como herramienta el MATLAB y el Simulink.

## CAPITULO IV

### RESULTADOS

#### Resultados del primero objetivo específico: elaborar el estado del arte del diagnóstico del dengue utilizando inteligencia artificial

Se ha elaborado el Estado del Arte del diagnóstico del dengue utilizando inteligencia artificial, el mismo que consta en el Capítulo II del presente estudio, cuyos resultados los podemos observar en la Tabla 16 que se muestra a continuación.

Tabla 16.

*Investigaciones realizadas utilizando estrategias de inteligencia artificial.*

Datos de entrada	RNA	LD	AG	TOTAL
Factores ambientales	4	4	0	8
Síntomas del dengue	1	2	0	3
Síntomas del dengue y otras variables	3	1	0	4
Genéticos	1	1	4	6
Imágenes fotográficas	1	0	0	1
Estadísticas y antecedentes dengue	0	2	0	2
Vacunación	0	0	1	1
Biológicas	0	0	1	1
<b>TOTAL</b>	<b>10</b>	<b>10</b>	<b>6</b>	<b>26</b>

Fuente: Propia.

Las técnicas de inteligencia artificial que fueron evaluadas son las redes neuronales artificiales (RNA), lógica difusa (LD) y algoritmos genéticos (AG). Se analizaron 26 investigaciones, obtenidas de diferentes revistas científicas y repositorios, de las cuales diez se realizaron utilizando redes neuronales artificiales (RNA), diez de lógica difusa (LD), y, seis de algoritmos genéticos (AG). Todos ellos rastreados entre los meses de enero a junio del 2019.

De las investigaciones obtenidas y analizadas sobre la fiebre del dengue sin síntomas de alarma, con síntomas de alarma y grave, ocho tuvieron como fuente de análisis a factores ambientales tales como la temperatura, humedad relativa, precipitaciones pluviales o el nivel del suelo; demostrando ser herramientas computacionales e informáticas poderosas para predecir la fiebre del dengue, a partir de la presencia de poblaciones de mosquitos *Aedes Aegypti* y su dependencia de factores climáticos. A más población de mosquitos mayor probabilidad de presencia del dengue.

Se analizaron siete estudios que consideraban como factor de análisis los síntomas clínicos y de laboratorio de la fiebre del dengue, tres de ellas con síntomas exclusivamente clínicos, y, cuatro que además consideraban pruebas de laboratorio como información complementaria, generalmente para comprobar la pertinencia del mal; las pruebas de laboratorio como se sabe requieren de contar con equipos adecuados para su realización, equipos de laboratorio con las que generalmente no se cuenta en los centros de salud de las zonas rurales de la selva del Perú. Eso implica que solo tres de las 26 investigaciones analizadas, es decir el 11%, se ejecutaron tomando en cuenta datos exclusivamente provenientes de los pacientes potenciales o afectados por la fiebre del dengue.

Seis del total de investigaciones evaluadas corresponden a aquellas tecnologías de inteligencia artificial que tomaron en cuenta, como datos de entrada, a factores genéticos del virus del dengue serotipos (DENV 1, DENV 2, DENV 3 y DENV 4); cuatro de ellas fueron abordadas desde los algoritmos genéticos; básicamente orientadas al control de las poblaciones de mosquitos *Aedes Aegypti*. Los algoritmos genéticos no se utilizaron para datos de entrada que consideran exclusivamente los síntomas de la fiebre del dengue, lo que demostraría que no es una tecnología adecuada para el diagnóstico del dengue.

Cinco de las 26 investigaciones analizadas tuvieron diversas fuentes de entrada de datos, como imágenes fotográficas, estadísticas y antecedentes (información de dengue de años pasados), vacunas, así como, biológicas como esterilizaciones y el uso de insecticidas. En todas ellas las

herramientas de inteligencia artificial demostraron ser versátiles para el diagnóstico de la fiebre del dengue.

### **Resultados del segundo objetivo específico: determinar la técnica informática de inteligencia artificial adecuada para el diagnóstico del dengue**

De acuerdo a Admadi et al. (2018), sólo el 9% de los casos analizados en su investigación utilizando la lógica difusa, se orientaron al estudio de enfermedades infecciosas; se dio cuenta de un solo estudio al respecto. Siendo las estrategias de inteligencia metodologías muy versátiles para la investigación de diversas enfermedades, sería importante ampliar el estudio no solo para la lógica difusa, sino también a las redes neuronales artificiales y los algoritmos genéticos, para conocer el impacto de la inteligencia artificial en las investigaciones de enfermedades como cardiología, neurología, nefrología, anestesiología, endocrinología, dermatología, etc.

Las redes neuronales artificiales han demostrado ser, según el presente estudio, muy versátiles para el diagnóstico del dengue desde diferentes fuentes o entrada de datos, como los factores ambientales, demográficos o genéticos; siendo necesario continuar con la investigación a partir de los síntomas clínicos de la fiebre del dengue, considerando las limitaciones técnicas, de laboratorio y equipos en las zonas rurales de la Amazonía peruana, donde se ubica la región San Martín.

La lógica difusa ha sido empleada en una de las investigaciones, para discriminar la fiebre del dengue de la varicela y la gripe. Considerando que en la Amazonía peruana su población también sufre los embates del Zika, Chicungunya, Malaria y otros tipos de enfermedades que tienen como origen a los aedes, sería pertinente investigar la posibilidad de distinguir la eficacia de la lógica difusa para otras enfermedades similares.

Otra alternativa de investigación proviene de los algoritmos genéticos, toda vez que, en el presente estudio, no se ha ubicado ningún caso orientado al diagnóstico de la fiebre del dengue tomando en cuenta los síntomas clínicos de la enfermedad. Los algoritmos genéticos analizados en el estudio del arte, seis de 26 casos, tuvieron como datos de entrada básicamente información genética, de vacunación y biológicas.

La combinación de estrategias inteligentes para el diagnóstico del dengue es una opción no trabajada suficientemente, como por ejemplo RNA-LD, RNA-AG, LD-AG o RNA-LD-AG, en otras circunstancias la lógica difusa (opinión de expertos) y las redes neuronales han sido utilizadas

con éxito, siendo interesante para el diagnóstico de enfermedades y particularmente de la fiebre del dengue y/o el dengue hemorrágico.

De los 26 casos analizados en el estado del arte, siete corresponden a datos de entrada que consideran a los síntomas del dengue sumadas con otras variables, de los cuales solo tres son estudiados tomando en cuenta exclusivamente los síntomas del dengue (uno utilizando redes neuronales artificiales y dos la lógica difusa), por lo que concluimos que el diagnóstico del dengue a partir de los síntomas clínicos de la enfermedad no ha sido suficientemente estudiado, menos aún, considerando los diferentes síntomas asumidos en la región San Martín.

Dado lo anterior, existe un vacío en el diagnóstico del dengue partiendo de los síntomas de los pacientes, antes que por otros factores como los ambientales, imágenes fotográficas, vacunación, etc., razón por la cual se optó realizar el presente estudio utilizando como técnica informática de inteligencia artificial a las Redes Neuronales Artificiales - Perceptrón Multicapa (MLP).

### **Resultados del tercer objetivo específico: identificar las características sintomáticas del dengue sin alarma, con alarma y grave**

En base a la información obtenida en formato Excel de la Dirección Regional de Salud (DIRESA) de la región San Martín, se encontró que la enfermedad viral del dengue se presenta en tres formas y grados, a decir, dengue sin signos de alarma, con signos de alarma y grave; cada una de ellas con su propia sintomatología. Los síntomas del dengue obtenidos de la DIRESA, corresponden a casos observados durante los años 2017, 2018 y 2019, se muestran en la Tabla 17.

Tabla 17

*Síntomas del dengue sin signos de alarma, con signos de alarma y grave.*

<b>Número</b>	<b>TIPO DE DENGUE</b>	<b>SINTOMA</b>	<b>ETIQUETA</b>
1	Dengue sin signos de alarma	Fiebre	Fiebre
2		Artralgias	Artralgias
3		Artralgias manos	Artralg_ma
4		Artralgias pies	Artralg_pi
5		Mialgias	Mialgias
6		Cefalea	Cefalea
7		Dolor ocular o retro ocular	Dolor_ocul
8		Dolor lumbar	Dolor_lumb

9		Rash/exantema	Rash_exant
10		Conjuntivitis no purulenta (ojos rojos)	Conjuntivi
11		Nauseas/vómitos	Nauseasvom
12	Dengue con signos de alarma	Dolor abdominal intenso y continuo	Dolor_abdo
13		Dolor torácico	Dolor_tora
14		Derrame seroso	Derrame_se
15		Vómito persistente	Vómitos
16		Ictericia	Ictericia
17	Dengue grave	Pulso débil	Pulso_debi
18		Extremidades frías o cianóticas	Extremidad
19		Diferencial de presión arterial < 20mmHg.	Difpresión
20		Sangrado grave	Sang_grave
21		Apertura ocular (1-4), escala Glasgow	Apert_ocul
22		Respuesta motora (1-6), escala Glasgow	Resp_motor
23		Respuesta verbal (1-5), escala Glasgow	Resp_verba
24	Otros	Diarrea	

Fuente: Propia.

Se identificaron los síntomas del dengue, de los cuales once (11) corresponden al dengue sin síntomas de alarma, dieciséis (16) al dengue con síntomas de alarma, y, veinticuatro (24) al dengue grave; incluyendo a la diarrea que es transversal a todos los tipos.

### **Resultados del cuarto objetivo específico: realizar el análisis de datos asociados al dengue sin síntomas de alarma, con síntomas y grave**

Se ha adecuado la información entregada por la DIRESA en formato Excel que, como ya se explicó, contenía abundante información sobre el dengue y otras enfermedades virales como la chikungunya, zika, fiebre amarilla selvática.

La información obtenida se ha organizado en forma matricial, considerando inicialmente 215 patrones, número obtenido de la selección de la muestra calculada en la metodología, ordenados por columnas, con 24 síntomas por cada patrón, organizados por filas, y, adjuntando el resultado para cada Patrón y sus respectivos síntomas, en una fila adicional, en la parte inferior de la Tabla en Excel, con los resultados reales sobre la presencia o ausencia de dengue para cada combinación.

Para efectos de análisis, se tomó como criterio asignar a cada síntoma presente el número uno (1) y a cada síntoma ausente el número cero (0), en la matriz de Excel. Del mismo modo, se asumió

como criterio asignar el número uno (1) al conjunto de síntomas que confirman la presencia del dengue y asignar el número cero (0) al conjunto de síntomas que no corresponden al dengue, en la última fila de la matriz. La Tabla correspondiente se muestra en el Anexo 1.

El comando *feedforwardnet* de la RNA MLP, segmenta automáticamente la data de patrones que se ingresan al sistema informático, en partes disponibles para el entrenamiento, validación y test, como parte del proceso de aprendizaje de la red neuronal. Esta distribución es aleatoria y se realiza de la siguiente manera (ver Tabla 18):

Tabla 18.

*Segmentación automática para entrenamiento, validación y test*

CONCEPTO	PORCENTAJE	CANTIDAD
Entrenamiento	70%	129
Validación	15%	43
Test	15%	43
TOTAL	100%	215

Fuente: Propia.

### **Resultados del quinto objetivo específico: proponer el prototipo óptimo para el diagnóstico del dengue en la región san martín**

La Arquitectura de RNA-MLP asumida inicialmente fue la siguiente: 24 capas de entrada, 10 capas ocultas y una capa de salida. Como se sabe, el comando *feedforwardnet*, por defecto, trabaja con funciones de transferencia y/o activación *Sigmoidea Tangente Hiperbólica* (*tansig*) para las capas ocultas y *Función de Transferencia Lineal* (*purelin*) para las capas de salida. A continuación, se expone paso a paso, el procedimiento seguido para la implementación del prototipo.

#### **Acondicionamiento de la red neuronal feedforward.**

Con el propósito de mejorar la convergencia de la salida de la red neural, se decidió poner a prueba algunos de los parámetros de funcionamiento de la red, como comparar las funciones de transferencia *tansig* – vs – *purelin* en la capa de salida, y, comparar los algoritmos de entrenamiento *trainlm* – vs – *trainbr*.

### **Funciones de transferencia tansig – vs – purelin.**

“Tansig” es la función de transferencia Tangencial Sigmoidal, cuya salida varía desde -1 hasta +1 cuando las entradas varían desde  $-\infty$  hasta  $+\infty$ , y, además, es derivable. Mientras la función de transferencia lineal, también conocida como “purelin”, la salida varía proporcionalmente respecto a la entrada en forma ilimitada.

La comparación de la performance entre estas dos funciones de transferencia, aplicadas en la capa de salida de la red neural, se observa en la Tabla 19:

Tabla 19

*Comparación funciones tansig y purelin en capa de salida*

<b>NUMERO NEURONAS CAPA OCULTA</b>	<b>TANSIG</b>	<b>PURELIN</b>
1	0.2046	0.2212
10	0.0911	0.1061
20	0.0606	0.1180
30	0.0418	0.0883
40	0.0531	0.0694
50	0.0735	0.1266
100	0.0867	0.1305

Fuente: Propia.

En todos los casos se observa que los errores en las funciones de transferencia tansig son menores que en las funciones de transferencia purelin, obteniendo el valor más bajo en el correspondiente a la función de activación tansig que contiene a 30 neuronas en la capa oculta.

### **Algoritmo de entrenamiento trainlm - vs – trainbr.**

“Trainlm” es una función de entrenamiento de retropropagación de redes neuronales MLP, conocida como Levenberg – Marquardt, que actualiza los valores de peso y sesgo (bías) de

acuerdo a los parámetros de optimización de Levenberg – Marquardt, sus creadores. El programa de ejecución se muestra a continuación:

```
Input = xlsread('inputDengue.xlsx','C6:PP29');
Target = xlsread('inputDengue.xlsx','C31:PP31')
Red = forwardnet([10]);
red.trainFcn = 'trainlm';
[red tr] = train(red,input,target);
Output = sim(red,input);
```

Del mismo modo, “trainbr” es otra función de entrenamiento de retropropagación de redes neuronales MLP, denominada de *regularización bayesiana*, que resulta ser una mejora de la función de entrenamiento Levenberg – Marquardt, que minimiza una combinación de errores al cuadrado y pesos, para producir una red que generalice bien. Un esquema del programa de ejecución se muestra a continuación:

```
Input = xlsread('inputDengue.xlsx','C6:PP29');
Target = xlsread('inputDengue.xlsx','C31:PP31')
Red = forwardnet([10]);
red.trainFcn = 'trainbr';
[red tr] = train(red,input,target);
Output = sim(red,input);
```

A continuación, se muestra un cuadro con los resultados obtenidos comparando el funcionamiento de la red feedforwardnet utilizando las funciones de entrenamiento trainlm y trainbr.

Ver Tabla 20:

Tabla 20

*Comparación funciones entrenamiento trainlm y trainbr.*

EPOCAS	NUMERO NEURONAS CAPA OCULTA	TRAINLM			TRAINBR		
		mse	R	tiempo	mse	R	tiempo
1000	1	0.1830	0.4093	0.00	0.2029	0.3454	0.00
	10	0.0102	0.8887	0.00	0.0169	0.927	0.20
	20	0.0136	0.8944	0.00	0.0144	0.9504	1.38
	30	0.0162	0.8938	0.01	0.2052	0.3232	4.27
	40	0.0131	0.9038	0.04	0.2081	0.3287	8.33
500	50	0.0184	0.9066	0.05	0.1992	0.3319	7.36
100	60	0.0111	0.9441	0.09	0.2058	0.3209	2.22
	70	0.0106	0.8856	0.21	0.2024	0.3287	3.31
	80	0.0150	0.8671	0.14	0.2035	0.3361	5.25
	90	0.0100	0.9468	0.28	0.2257	0.3124	6.05
	100	0.0140	0.9265	1.30	0.1928	0.3394	10.03

Fuente: Propia.

En la Tabla 20 se puede observar que, en todos los casos el error cuadrático medio (mse) es menor en el algoritmo trainlm que en el trainbr, evidenciándose los menores valores con 10 y 20 neuronas en la capa oculta. Se puede ver también que en ambos casos el tiempo de iteración va en incremento conforme se incrementan el número de neuronas en la capa oculta. Sin embargo, los menores tiempos se observan en trainlm toda vez que en trainbr el algoritmo de entrenamiento se hace más complejo.

En general, la variable R de la curva de regresión, que mide la correlación entre las salidas (output) obtenidas y las salidas (targets) deseadas, tiene mejores resultados, cerca al valor “1” que es el ideal, con trainlm que con trainbr. De acuerdo a lo observado se decidió continuar la investigación haciendo uso como función de entrenamiento de la RNA MLP a la función “**trainlm**”.

### **Pruebas de rendimiento de la red feedforward.**

Con estos parámetros, tansig para las funciones de transferencia de las capas ocultas y purelin para la capa de salida, así como, definido el algoritmo de entrenamiento Levenberg – Marquardt (trainlm), se realizaron pruebas con los 215 patrones considerados,

asignándose automática y aleatoriamente los pesos y valor de las vías correspondientes a las neuronas. Sin embargo, no se logró la convergencia de la salida, es decir, el sistema no mostró un resultado satisfactorio de acuerdo a los estándares de las redes neuronales artificiales. Por lo que se decidió duplicar el número de patrones a 430, con los cuales el sistema alcanzó niveles de convergencia aceptables; siendo necesario hacer algunos ajustes al sistema a fin de lograr mejores resultados.

### ***Procedimiento de ajuste.***

El procedimiento seguido fue el siguiente:

1. Inicializar y entrenar de nuevo.- Se trata de reiniciar la asignación de pesos y bias a las neuronas, aleatoriamente, cada vez que ejecutamos un nuevo entrenamiento de la red, evitando que los pesos y bias de entrenamientos anteriores afecten los resultados. Para ello se hizo uso del siguiente juego de comandos:

```
red = init(red);  
red = train(red,input,output);
```

2. Aumentar el número de neuronas en la capa oculta.- El entrenamiento debe inicializarse considerando el menor número de neuronas en la capa oculta que sea posible y luego, progresivamente, ir incrementando este número hasta alcanzar uno que ofrezca los mejores resultados.
3. Cambiar *trainlm* por *trainbr*.- Este trámite no fue necesario, toda vez que, como hemos observado, al comparar la performance de ambas funciones de entrenamiento, en el caso particular de esta investigación, *trainlm* ofreció mejores resultados.
4. Usar datos de entrenamiento adicionales.- Inicialmente estuvimos trabajando con 215 datos de entrada (*input*) para entrenamiento, validación y test, luego se duplicó a 430 patrones. Ya se indicó que el comando *feedforward* automáticamente y de manera aleatoria selecciona el 70 % de los patrones para entrenamiento, 15 % para validación y 15 % para test.

### **Entrenamiento con 24 entradas, una capa oculta y una capa de salida.**

Se anotaron los cinco mejores resultados tanto del error cuadrático medio (mse) y del factor de relación R de las curvas de regresión, para efectos de comparación.

#### **Cambio por decenas del número de neuronas en la capa oculta.**

Se hicieron cambios en el número de neuronas de la capa oculta: 1, 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100. Se observó que los mejores resultados del error cuadrático medio y el factor de regresión R, fueron en las primeras decenas de neuronas ocultas, específicamente en el rango de 10 a 20 neuronas ocultas evaluadas (ver Anexo 1).

#### **Cambio por unidades del número de neuronas en la capa oculta.**

Se hicieron pruebas realizando cambios en el número de neuronas en la capa oculta, partiendo de 10 neuronas e incrementando el número de unidad en unidad hasta el límite de 20 neuronas. Los resultados se muestran en el Anexo 2.

Se concluye que los mejores resultados, es decir, los menores errores cuadráticos medios y los mayores factores R de regresión, se encuentran entre los números de neurona en capa oculta 11 y 13. Por extrapolación optamos por 12 neuronas en la primera capa oculta de la red feedforward.

### **Entrenamiento, validación y test con dos capas ocultas.**

Con el propósito de mejorar el comportamiento de la red neuronal feedforward, desde el punto de vista del menor error cuadrático medio y del mayor factor de regresión R, que nos permita una mayor convergencia entre los datos obtenidos en la salida output de la red y las salidas objetivo target, se decidió incrementar una capa oculta, con los resultados que se muestran en la Tabla 21.

Tabla 21

*Cuadro comparativo de la red feedforward con dos capas ocultas.*

Nº NEURONAS CAPA OCULTA 1	Nº NEURONAS CAPA OCULTA 2	ERROR Entrenamiento	ERROR Test	ERROR Validación	R Factor Regresión	Número Iteraciones
12	1	0.0145	0.1779	0.1191	0.87	40
12	2	0.0192	0.1135	0.1465	0.88	26
12	3	0.0042	0.1233	0.1385	0.90	18
12	4	0.0104	0.1308	0.0601	0.92	22
12	5	0.0133	0.1066	0.1038	0.90	34
12	6	0.0140	0.1297	0.0946	0.90	30
12	7	0.0103	0.2099	0.1926	0.87	24
12	8	0.0143	0.0936	0.0675	0.92	24
12	9	0.0097	0.1427	0.0932	0.88	28
12	10	0.0128	0.1085	0.0760	0.91	30
12	11	0.0185	0.0674	0.0635	0.92	38
12	12	0.0162	0.0916	0.1175	0.90	37

Fuente: Propia.

La Tabla 21 muestra que el menor error de validación corresponde a la capa oculta con cuatro neuronas, con un valor de 0.0601, en correspondencia con eso también se muestra el mayor factor de regresión  $R=0.92$ , para 22 iteraciones, uno de los valores más bajos de los considerados. Colateralmente corresponde un error de Entrenamiento de 0.0104 y otro de test equivalente a 0.1308, que son también valores bastante bajos de los obtenidos. En consecuencia, el mejor rendimiento se obtiene con una arquitectura de red de: **24 – 12 – 4 – 1**. Veinticuatro neuronas de entrada, doce neuronas en la capa oculta 1, cuatro neuronas en la capa oculta 2 y una neurona en la capa de salida.

### **Entrenamiento de la red feedforward versus cascadefeedforward.**

Siempre en la perspectiva de obtener un mejor resultado para nuestra red neuronal, se decidió comparar los resultados de la red feedforward con otra red denominada cascadefeedforward, que es una variante de la primera, manteniendo la misma arquitectura de red: 24 – 12 – 4 – 1.

Los resultados se muestran en la Tabla 22.

Tabla 22

*Comparación red feedforward versus cascadefeedforward.*

NUMERO DE PRUEBA	ERROR VALIDACION		FACTOR DE REGRESION	
	Feedforward	Cascade	Feedforward	Cascade
1	0.0785	0.0506	0.9034	0.9213
2	0.0806	0.0645	0.9069	0.9390
3	0.1008	0.0380	0.9070	0.9412
4	0.1239	0.0991	0.9079	0.9187
5	0.0699	0.0158	0.9167	0.9484

Fuente: Propia.

Cascadefeedforward, también conocida como red de avance en cascada, no es otra cosa que una variante de la red feedforward, que consiste en ingresar los datos de entrada, en el presente caso los 24 síntomas del dengue seleccionados, a todas y cada una de las neuronas de las capas posteriores incluida la capa de salida, y, de cada capa anterior a una capa posterior; esto permite a la red aprender cualquier relación entrada – salida finita con buenos resultados. Los comandos de MATLAB que nos permiten implementar esta red son las siguientes:

```
Input = xlsread('inputDengue.xlsx','C6:PP29');
Target = xlsread('inputDengue.xlsx','C31:PP31')
Red = cascadeforwardnet([12 4]);
red.trainFcn = 'trainlm';
[red tr] = train(red,input,target);
Output = sim(red,input);
```

Conforme se observa en la Tabla 24, los menores errores de validación se encuentran en los valores obtenidos para la red cascadefeedforward. Del mismo modo, los mejores valores del factor de regresión “R” se encuentran en la red cascadefeedforward.

Por lo expuesto, la red cascadefeedforward, con una arquitectura de red 24 – 12 – 4 – 1, es la que mejores resultados ofrece para el diagnóstico del dengue, utilizando como función de transferencia la función tansig en las neuronas de las capas ocultas y de salida, así como, la función trainlm (Levenberg – Marquardt) como algoritmo de aprendizaje Backpropagation.

## Entrenamiento haciendo uso de MATLAB

En el presente estudio, se ha utilizado el entorno MATLAB (Laboratory Mathematical) versión 2017b y su herramienta de simulación denominada SIMULINK. La pantalla general para la programación en código MATLAB se muestra en la Figura 29:

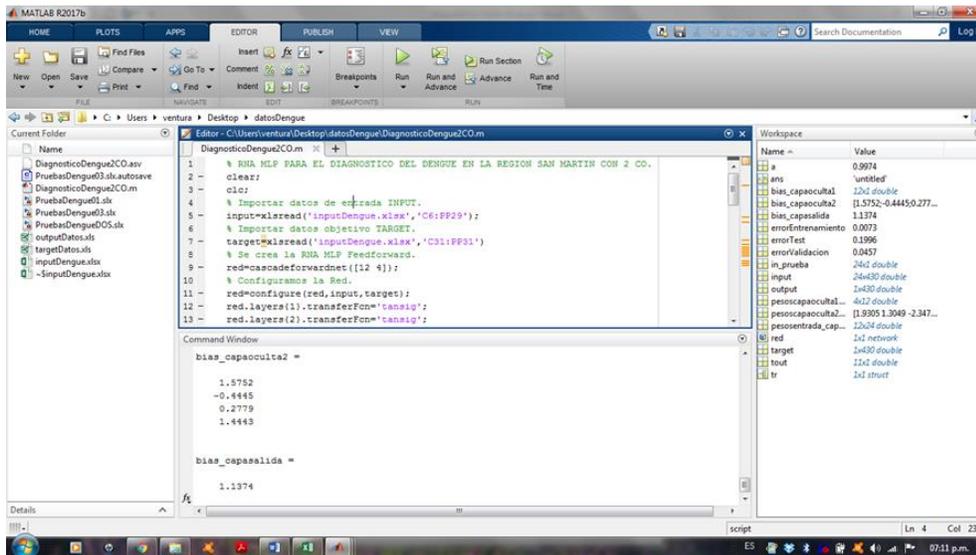


Figura 29. Ventanas del MATLAB.

Seguidamente, se diseñó la arquitectura de la RNA-MLP con 24 neuronas de entrada, 12 neuronas en la capa oculta número 1, 4 neuronas en la capa oculta número 2, y, una neurona en la capa de salida, como se muestra en la Figura 30.

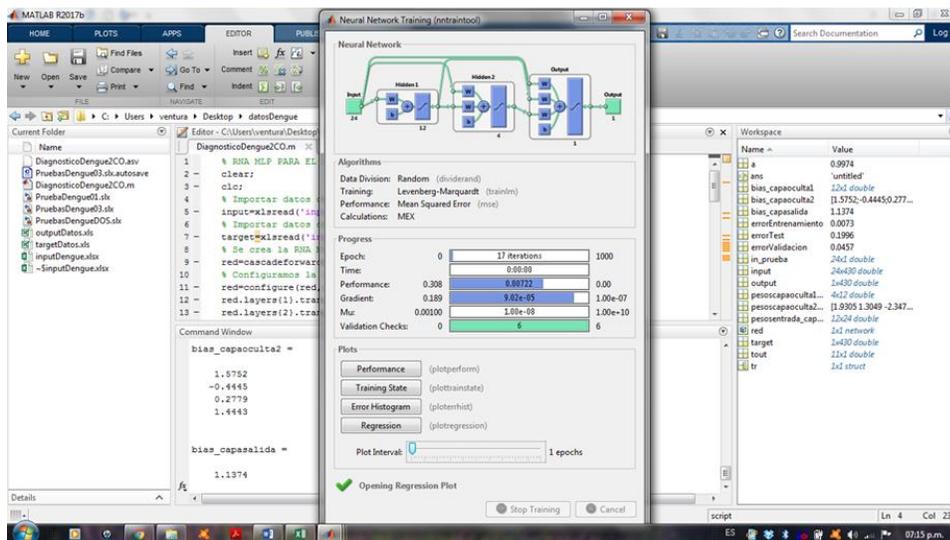


Figura 30. Arquitectura de la RNA MLP implementada.

Se muestra la arquitectura de red implementada cascadefeedforward: **24 – 12 – 4 – 1**; el algoritmo de entrenamiento trainlm (Levenberg – Marquardt), el tipo de error utilizado MSE (Error Cuadrático Medio), así como, el progreso de los diferentes parámetros del algoritmo epoch, times, performance, gradient, factor mu y validation cheks; a partir de los cuales convergió el sistema.

El código MATLAB utilizado, se muestra en el Anexo 3.

Los datos en formato Excel, fueron migrados al MATLAB. Se ingresaron 430 patrones de datos, extraídos al azar de la Base de Datos entregada por los funcionarios de la DIRESA, a través del Hospital II de la ciudad de Tarapoto; a cada patrón le correspondió 24 síntomas o entradas, las que se codificaron como “1” ante la presencia del síntoma y como “0” ante su ausencia. Los datos fueron organizados en una Matriz de 24 x 430, es decir, 24 filas por 430 columnas. Se agregó en la parte inferior una fila con los resultados esperados (target), constituyendo una Matriz 1 x 430. Ver Anexo 4.

Se seleccionaron 43 patrones de Test, de manera aleatoria, para la etapa de prueba de la red cascadefeedforward, con sus respectivas 24 entradas o síntomas. De esta forma se organizó una Matriz 24 x 43, es decir, 24 filas por 43 columnas. Igualmente, se agregó en la parte inferior una fila con los resultados esperados (target), constituyendo una Matriz 1 x 43. Ver Anexo 5.

Los síntomas, ordenados por sus denominaciones y etiquetas médicas, se muestran de manera integral en el Anexo 6.

Las características de la red cascadefeedforward de dos capas implementada, tales como: El nombre de la red neuronal “Cascade\_ Forward Neural Network”, sus dimensiones con una entrada, tres capas considerando dos ocultas y una de salida, 489 elementos de peso, conexiones, subobjetos, funciones como performFcn que hace referencia al error cuadrático medio (mse) o trainFcn que indica que la función de entrenamiento es trainlm (Levenberg – Marquardt), pesos y vías, métodos de adaptación, configuración, inicialización y entrenamiento, entre otros, se pueden observar en el Anexo 7.

Decidido el uso del algoritmo de entrenamiento trainlm, fue importante determinar los parámetros de entrenamiento (aprendizaje) de la red (ver Anexo 8). Ahí se puede observar que el número de iteraciones entre dos reportes es 25 (show), el número máximo de iteraciones previsto es 1 000 (epochs), el máximo error permitido (performance goal) equivalente a “0”, las

comprobaciones de validación máxima (maximum validation checks) de 6 errores consecutivos donde no mejora el error, entre otros.

### Entrenamiento de la red cascadefeedforward con dos capas ocultas

La red fue entrenada en múltiples ocasiones, buscando los mejores resultados, para lo cual se evaluaron diferentes parámetros tales como: El error de validación, el factor de regresión R y el nivel de convergencia gráfica observada entre la salida objetivo (target) y la salida real output. Para la arquitectura cascadefeedforward **24-12-4-1** considerada, el mejor resultado alcanzado corresponde a un error de Validación equivalente a 0.0457, con un factor de regresión  $R=0.90715$  que es bastante cercano al valor ideal de 1. La convergencia gráfica se puede observar en la Figura 31.

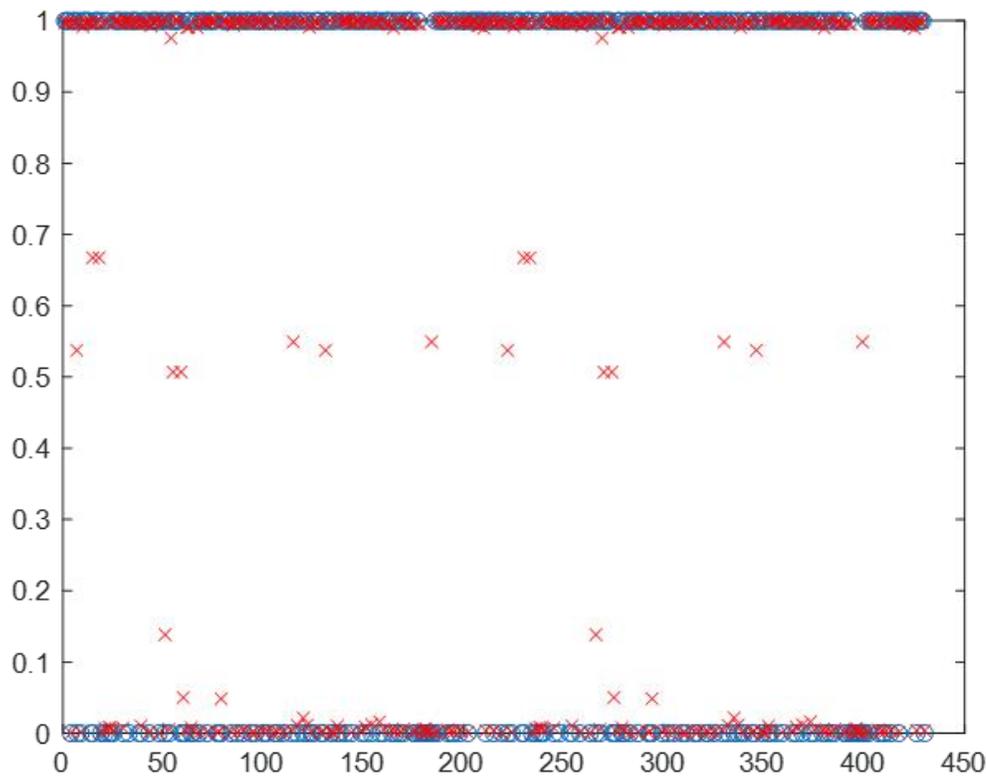


Figura 31. Convergencia de datos target (azul) y datos output (rojo).

### Comportamiento de la curva de validación.

La curva de validación (color verde) alcanza su mejor performance con un valor mínimo equivalente a 0.045682 al cumplirse once épocas o iteraciones. Como se verá las curvas de validación y prueba (color rojo) son muy similares, no presentándose entre ambas una separación en aumento, durante su evolución, que hagan suponer la existencia de un posible sobreajuste. La curva de entrenamiento (color azul) disminuye rápidamente en las primeras épocas y luego se estabiliza a partir de la época 11.

Lo que nos lleva a concluir que hemos obtenido un buen resultado. Ver Figura 32.

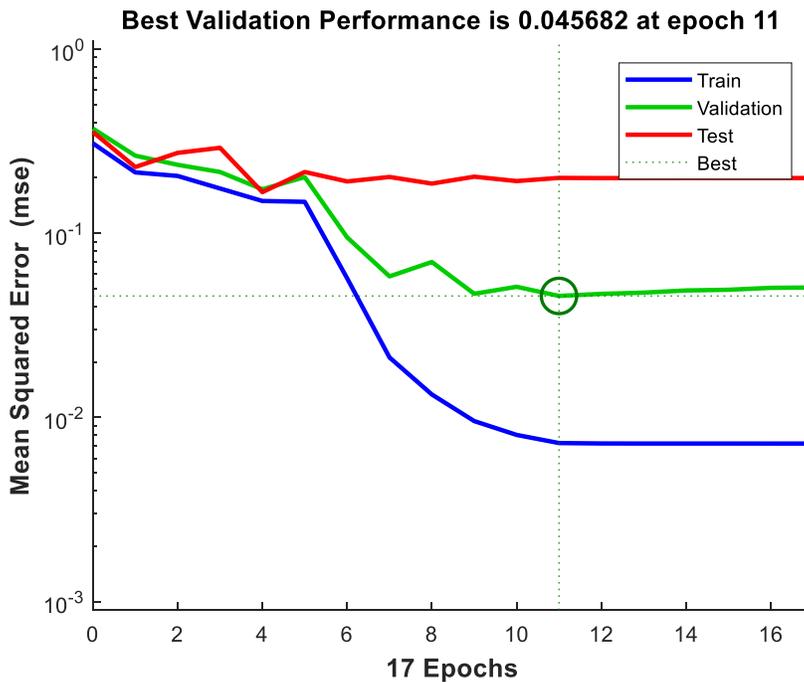


Figura 32. Mejor curva de validación obtenida para un error de 0.045682.

### Estado del entrenamiento.

El estado del entrenamiento (training state) se puede observar en la Figura 33. Nótese que el gradiente, luego de alcanzar un valor máximo en la época 4, disminuye sostenidamente hasta llegar a un valor mínimo en la época 17, con un valor equivalente a  $9.0163 \times 10^{-5}$ .

Igualmente, el factor de corrección “mu” se mantiene con una tendencia a disminuir hasta llegar a un valor mínimo de  $1 \times 10^{-8}$ , en la época 17.

Por último, la Validación Checks es un factor que por defecto vale 6, que detiene el entrenamiento cuando se produce una degradación del error de validación seis veces de manera consecutiva, como se produjo en la tercera gráfica, lo cual se produjo a partir de la época 11 y se detuvo en la época 17.

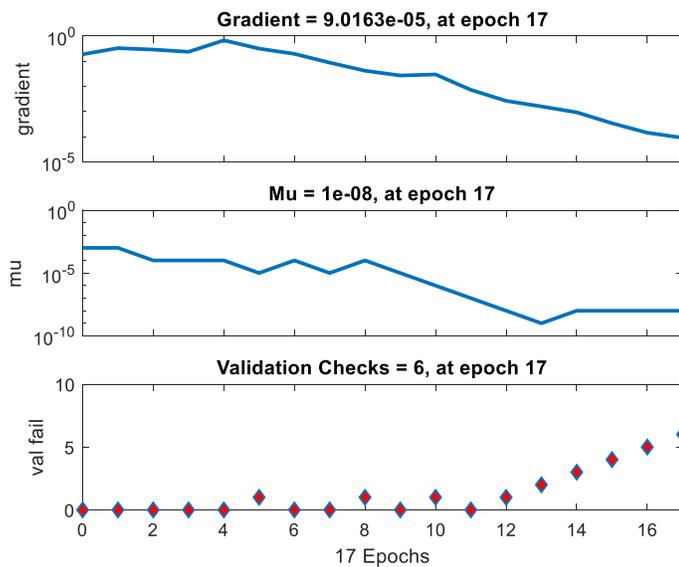


Figura 33. Estado del entrenamiento de la red cascadedefeedforward.

### Histograma del error.

En el histograma de error se puede observar que tanto los datos de entrenamiento, validación y prueba se concentran alrededor del error, lo que significa que no hay mayor dispersión y que existe una gran convergencia entre los datos objetivo (target) y las salidas realmente obtenidas u output.

Ver Figura 34.

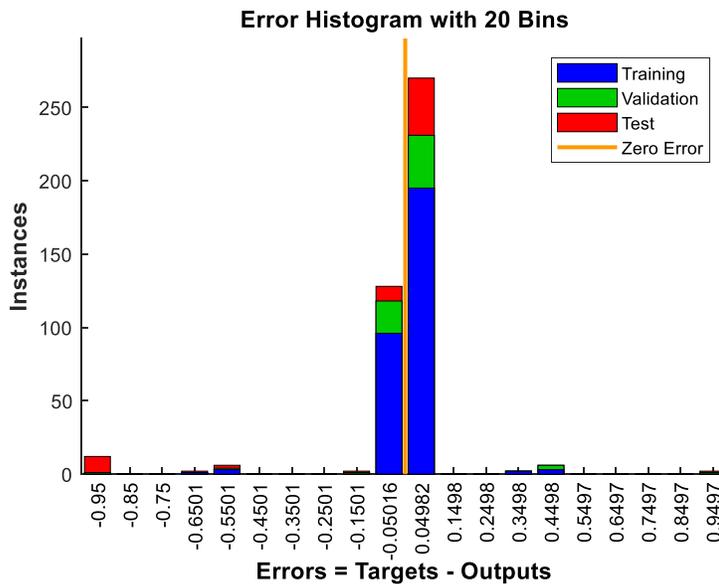


Figura 34. Histograma de error, muestra la convergencia de los resultados obtenidos.

### Gráfica de regresión.

Las gráficas de regresión muestran la correlación existente entre la salida objetivo (target) y la salida (output) obtenida luego del entrenamiento de la red cascadedfeedforward. Un valor de  $R=1$  representa una relación lineal perfecta o exacta, por el contrario, un valor de  $R=0$  significa que entre los dos resultados no existe relación lineal. Se puede observar en la Figura 35, que la correlación en el bloque de entrenamiento fue de  $R=0.98392$ , un valor muy bueno, próximo al ideal de 1. Igualmente, la correlación en el bloque de validación fue de  $R=0.90179$ , un valor muy bueno, y, en el bloque de test el valor de  $R=0,5811$  lo cual es indicativo.

El valor que tiene mayor relevancia, de acuerdo a la experiencia, es el cuarto factor de correlación "All", que combina todas las anteriores, y nos muestra una  $R=0.90715$  relevante; lo que significa que en general hay una importante correlación de 90.715%.

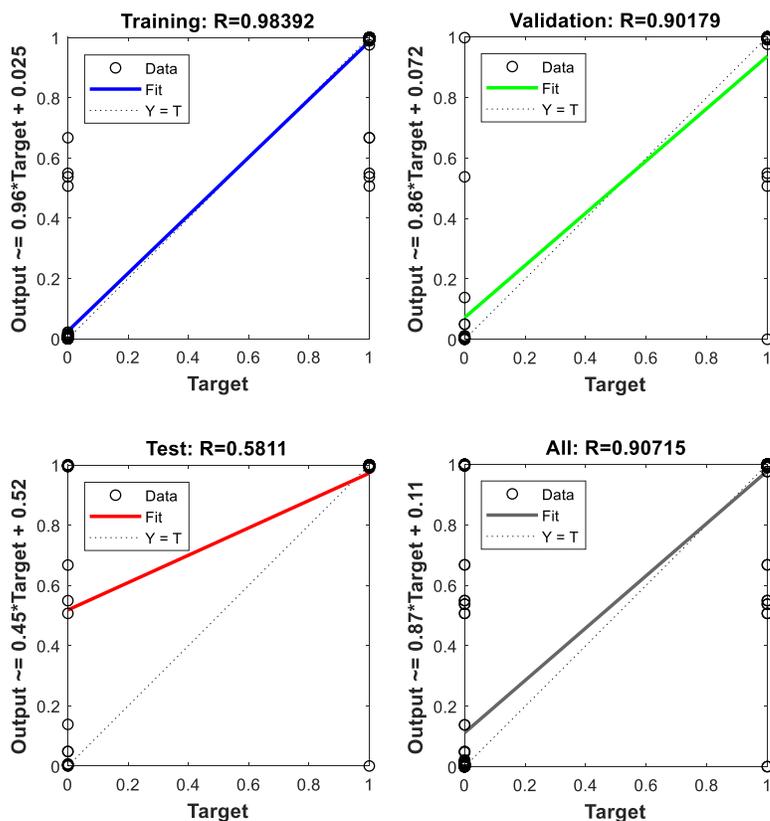


Figura 35. Mejor factor de correlación de la red cascadedefeedforward es  $R=0.90715$ .

Los datos objetivo, también conocidos como target, son los datos de salida conocidos, contenida en la información entregada por la Dirección Regional de Salud de San Martín, ingresados al sistema para el entrenamiento de la red. Las salidas output son las obtenidas como resultado del entrenamiento de la red cascadedefeedforward. Ambos resultados los podemos visualizar en el Anexo 9, en las que podemos verificar la aproximación de valores existentes de manera biunívoca, existiendo convergencia en la absoluta mayoría de los datos.

Los pesos y bías que, son valores de las conexiones internas de la red cascadedefeedforward, entre las capas de entrada, ocultas y de salida de la red, producto del proceso de aprendizaje y de la aplicación del algoritmo trainlm, utilizados en el presente estudio, se pueden observar en el Anexo 10.

## Exportación al simulink de matlab

Simulink es la librería de MATLAB que nos permite simular procesos diversos en forma gráfica, de modo que se puede visualizar a través de diagramas de bloque, los resultados de una serie de entradas cuando se aplican a un bloque NNET (Neural Network).

En el presente caso, al lado izquierdo en verde, se presentan 24 bloques de entrada, que representan los 24 síntomas del dengue sin alarma, con alarma y grave, así como, de otras enfermedades similares como la chikungunya, zika y fiebre amarilla de la selva, que tienen el mismo origen viral, es decir, el vector *Aedes Aegypti*.

En el centro, en color azul, se ubica la red cascadedfeedforward (NNET) entrenada, con los pesos y bías correspondientes, que emulan el comportamiento de la red neural artificial.

En el extremo derecho, en verde, se muestra un bloque Scope, que nos permite visualizar en pantalla la respuesta de la red ante cualquier combinación de entradas, presentando un escalón de valor “1” ante la presencia del dengue y “0” ante la ausencia del mismo. Ver Figura 36.

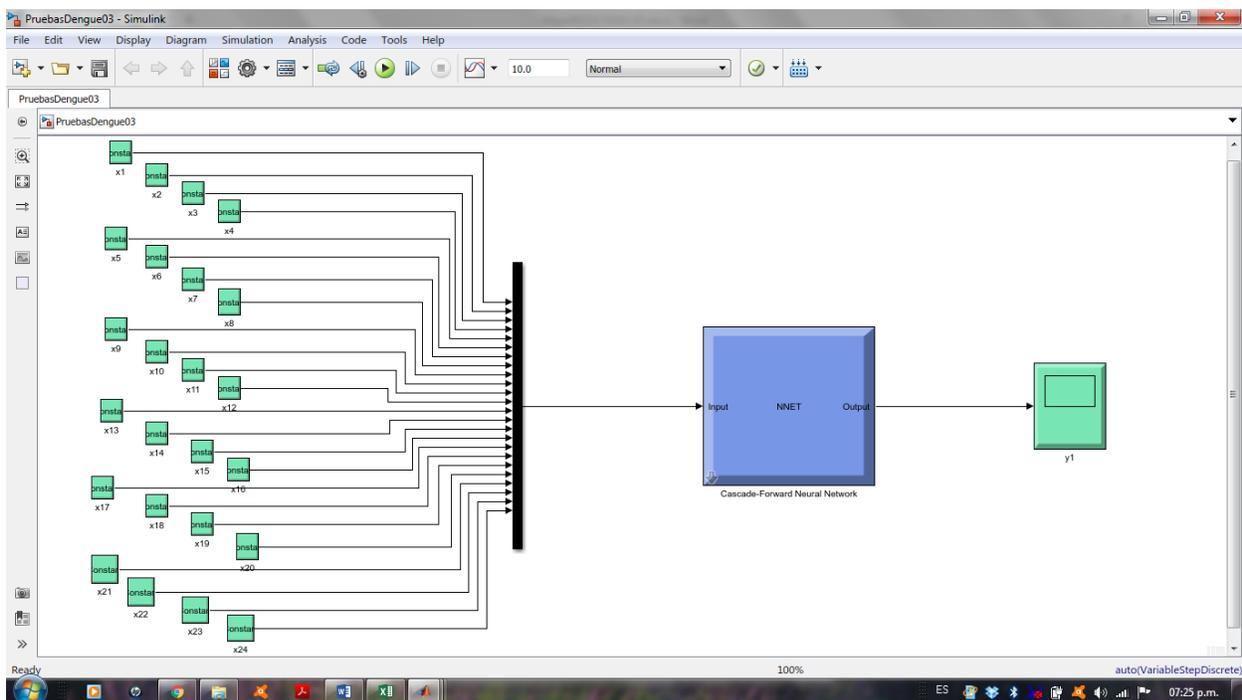


Figura 36. Diagrama de bloques en Simulink de la red cascadedfeedforward para el diagnóstico del dengue en la región San Martín.

A modo de ejemplo presentamos la siguiente combinación de unos y ceros, que corresponden al test 25 con 24 síntomas, que aparece en el Anexo 2: 0000 1101 0011 0010 0000 0000. El resultado es un escalón unitario que confirma el dengue para el patrón ingresado (ver Figura 37).

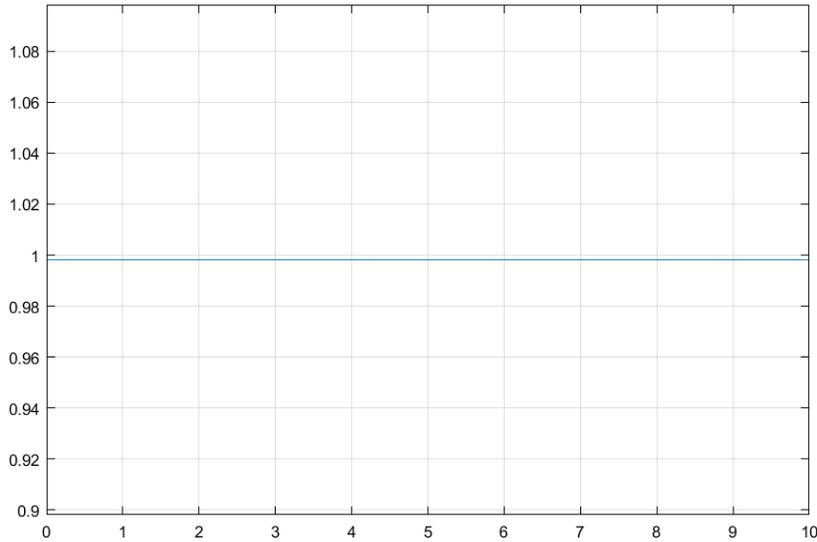


Figura 37. Respuesta de prueba de la red cascadedefeedforward con presencia de dengue.

### Aplicación del bloque de test para efectos de validación de la red neural

Se aplicó al bloque Simulink los 43 patrones de entrada del grupo de test (prueba), con el fin de medir el rendimiento de la red cascadedefeedforward y determinar si las redes neuronales artificiales, como parte de una tecnología de inteligencia artificial, es capaz de diagnosticar el dengue. Los resultados los observamos en la Tabla 23.

Tabla 23

*Resultados del rendimiento de la red neural cascadedefeedforward.*

TEST	TARGET	OUTPUT	RESULTADO
T1	1	1	v+
T2	1	1	v+
T3	0	0	v-
T4	0	0	v-
T5	1	1	v+
T6	0	1	F+
T7	0	0	v-

T8	0	0	v-
T9	0	0	v-
T10	0	0	v-
T11	0	0	v-
T12	0	0.54	F+
T13	0	0	v-
T14	1	1	v+
T15	0	0	v-
T16	1	1	v+
T17	1	1	v+
T18	1	1	v+
T19	1	1	v+
T20	0	0	v-
T21	1	1	v+
T22	0	0	v-
T23	0	0	v-
T24	1	1	v+
T25	1	1	v+
T26	0	0	v-
T27	1	1	v+
T28	1	1	v+
T29	1	1	v+
T30	0	0	v-
T31	1	1	v+
T32	1	1	v+
T33	1	1	v+
T34	1	1	v+
T35	1	1	v+
T36	1	1	v+
T37	1	1	v+
T38	1	1	v+
T39	0	0	v-
T40	1	1	v+
T41	1	1	v+
T42	1	1	v+
T43	0	0	v-

Fuente: Propia.

Se identificaron 25 casos de verdaderos positivos (v+), es decir, pacientes con dengue que fueron confirmados como positivos. Igualmente, se presentaron 16 casos de verdaderos negativos (v-), es decir, pacientes que no tienen dengue y que fueron confirmados como negativos. Solo se

presentaros dos casos, el correspondiente a los T6 y T12, de falsos positivos, o sea, personas que no tienen dengue pero que dieron como resultado positivo.

Es preciso señalar que todos los datos de entrada, tanto para entrenamiento, validación y test, son datos reales provenientes de la Dirección Regional de Salud de San Martín, que han sido adecuados para su tratamiento a través de los algoritmos de la Red Neuronal Artificial.

En suma, considerando el número de aciertos como verdaderos positivos y verdaderos negativos, que en conjunto suman 41 de 43 casos, podemos afirmar que la red neuronal artificial cascadefeedforward es capaz de diagnosticar el dengue con un 95.35 % de veracidad; lo que significa que, la red neuronal artificial es una herramienta informática óptima de la Inteligencia Artificial para diagnosticar el dengue en la región San Martín de la Amazonía del Perú.

## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

#### Conclusiones

1. Se ha logrado diseñar un modelo de diagnóstico del dengue, basado en inteligencia artificial, para ser aplicada en la región San Martín del Perú, con la implementación de una Red Neuronal Artificial – Perceptrón Multicapa (RNA – MLP) *cascafeedforward* de dos capas ocultas, con arquitectura **24 – 12 – 4 - 1**, que nos permitió diagnosticar el dengue con un 95.35% de efectividad.
2. Se elaboró el Estado del Arte para el diagnóstico del dengue utilizando Inteligencia Artificial. Se analizaron 26 investigaciones: Diez (10) de redes neuronales artificiales, diez (10) de lógica difusa y seis (6) algoritmos genéticos; con los que se demuestra que las técnicas de informática de inteligencia artificial, constituyen importantes herramientas para el diagnóstico de enfermedades virales y muy en particular del dengue sin alarma, con alarma y grave.
3. Las redes neuronales artificiales han demostrado ser, según el presente estudio, muy versátiles para el diagnóstico del dengue, desde diferentes fuentes o entrada de datos, como los factores ambientales, demográficos o genéticos, sin embargo, no ha sido suficientemente estudiada a partir de los síntomas clínicos de la fiebre del dengue, por lo que, considerando que es una técnica informática que funciona bien para el tratamiento de patrones, se determinó que constituye la técnica más adecuada para el diagnóstico del dengue.

4. Se identificó las características sintomáticas del dengue, encontrándose veinticuatro (24) síntomas, de los cuales once (11) corresponden al dengue sin síntomas de alarma, dieciséis (16) al dengue con síntomas de alarma, y, 23 al dengue grave. Anotando que la diarrea actúa como un síntoma transversal a todos ellos.
5. Se realizó el análisis de los datos asociados al dengue sin alarma, con alarma y grave, discriminando los síntomas con respecto a otras enfermedades virales como la chicungunya, zika y la fiebre amarilla amazónica; las que fueron organizadas en patrones de 24 síntomas como datos de entrada de la red neural artificial, formando una matriz de datos de 24 x 430, para ser entrenadas, validadas y testeadas, para validar el sistema.
6. Se determinó que el mejor prototipo para el diagnóstico del dengue en la región San Martín, es la Red Neuronal Artificial Backpropagation, *cascadefeedforward*, con arquitectura 24 – 12 – 4 – 1, es decir, 24 neuronas en la capa de entrada, 12 en la primera capa oculta, 4 en la segunda capa oculta y una (1) en la capa de salida, con funciones de transferencia Tangencial – Sigmoidea (tansig) en las capas ocultas, así como, en la capa de salida, utilizando como algoritmo de entrenamiento Levenberg – Marquardt (Trainlm); con los cuales se ha obtenido, el mejor resultado, presentando un error de validación equivalente a 0.0457, factor de regresión  $R=0.90715$  y 95.35% de efectividad.
7. Con el nivel de error de validación, factor de regresión y nivel de efectividad anotados, se demuestra que se puede diagnosticar el dengue utilizando una RNA – Backpropagation, de manera automática, rápida y sin el requerimiento de pruebas de laboratorio inexistentes en los centros de salud y postas sanitarias de las zonas rurales de la región San Martín. Para dicho efecto, se requiere contar con una computadora personal, laptop o una Tablet programada con el algoritmo adecuado.

## Recomendaciones

1. Ampliar la investigación de las redes neuronales artificiales MLP, tipo feedforward (fitnet, patternnet o cascadefeedforward), para determinar qué tipo de enfermedad viral afecta al paciente a partir de los síntomas estudiados en la presente investigación, es decir, que tenga la capacidad no solo de distinguir al dengue de otras enfermedades virales, sino que además diagnostique si el paciente está afectado por el dengue, chicungunya, zika o fiebre amarilla de la selva.
2. Diseñar e implementar un prototipo en hardware, como la plataforma Arduino o un dispositivo móvil, para diagnosticar el dengue, que sirva de herramienta de apoyo al personal médico, los pacientes y particularmente los centros de salud de la Amazonía y la región San Martín.
3. Diseñar e implementar un aplicativo móvil para diagnosticar el dengue, transformando el código en MATLAB de la presente investigación a un código aplicable en tablets, teléfonos móviles o una simple PC; permitiendo a la población amazónica, peruana y latinoamericana contar con un dispositivo ágil que le permita diagnosticar el dengue y, eventualmente, otras enfermedades virales, en forma oportuna y tomar medidas preventivas para la protección de su salud.
4. Estudiar la posibilidad de aplicar los algoritmos genéticos, como herramienta de inteligencia artificial, para el diagnóstico del dengue y otras enfermedades virales.

## Referencias bibliográficas

- Aburas, H., Cetiner, B. y Sari, M. (2010). *Dengue confirmed-cases prediction: A neural network model*. Elsevier Ltd.
- Ahmadi, H., Gholamzadeh, M., Shahmoradi, L., Nilashi, M. y Rashvand, P. (2018). *Diseases Diagnosis Using Fuzzy Logic Methods: A Systematic and Meta-Analysis Review*. Computer Methods and Programs in Biomedicine. DOI: 10.1016/j.cmpb.2018.04.013.
- Amador (1996). *Inteligencia Artificial y Sistemas Expertos*. Editorial Universidad de Córdoba. Córdoba, Argentina.
- Aswi, A., Cramb, S., Duncan, E., Hu, W., White, G. y Mengersen, K. (2021). *Variabilidad climática y fiebre del dengue en Makassar, Indonesia: Modelado espacio-temporal bayesiano*. Revista Epidemiología Espacial y Espacio-temporal; Volumen 33 , junio de 2020. Doi.org/10.1016/j.sste.2020.100335
- Barsante, L., Cardoso, R. y Acebal, J. (2011). *Otimizando custos no combate da dengue a través de Algoritmos Genéticos*. Centro Federal de Educacao Tecnológica de Minas Gerais. Bello Horizonte, Brasil.
- Boden M. (1984). *Inteligencia Artificial y hombre natural*. Editores Tecnos. Madrid.
- Cabezas, C. (2005). *Dengue en el Perú: Aportes para su Diagnóstico y Control*. Rev. Perú. med. exp. Salud pública, 22(3), 212-228.
- Caicedo, E. López, J. (2005). *Entrenamiento Bayesiano para Redes Neuronales Artificiales*. Cali: Universidad del Valle. Programa Editorial.
- Caicedo, E. López, J. (2017). *Una Aproximación Práctica a las Redes Neuronales Artificiales*. Cali: Universidad del Valle. Programa Editorial.
- Carlos, G. y Jesús, B. (2015). *Sistema Experto para la detección de Influenza y Dengue desarrollado sobre plataformas móviles*. Centro de Enseñanza Técnica Industrial. Guadalajara, México.
- Carretero L. (1989). *Consideraciones en torno a las posibilidades de Implantación de Sistemas Expertos en las decisiones organizativas*. Esic Market. México.
- Castillo, M. (2009). *Diseño e Implementación de un Sistema para Predicción de Enfermedades Tropicales transmitidas por Vectores Artrópodos en la Ciudad de Guayaquil*. Recuperado de: <http://www.dspace.espol.edu.ec/bitstream/123456789/604/1/1114.pdf>.
- Chatterjee, S., Dey, N., Shi, F., Ashour, A., Fong, S., y, Sen, S. (2017). *Clinical application of modified bag-of-features coupled with hybrid neural-based classifier in dengue fever classification using gene expression data*. Revista, Medical Biologic Engineer Computer. DOI: 10.1007/s11517-017-1722-y.
- Delgado, L. (2003). *Sensores Remotos, Modelos Digitales de Terreno y SIG Herramientas para el Control de la Malaria en el Estado Sucre, Venezuela*. Acta Científica Estudiantil 2003, 1(4), 143-151.
- Espinal, M. (2015). *Guías para la atención de enfermos en la Región de la Américas*. Organización Panamericana de la Salud.
- Faisal, T., Taib, M. y Ibrahim, F. (2010). *Neural network diagnostic system for dengue patients risk classification*. Journal of Medical Systems. DOI: 10.1007/s10916-010-9532-x.
- Florentino, H., Cantane, D., Santos, F. y Bannwart, B. (2014). *Multiobjective Genetic Algorithm applied to dengue control*. Mathematical Biosciences. DOI: 10.1016/j.mbs.2014.08.013.

- Gambhira, S., Malika, S., y Kumar, Y. (2017). *PSO-ANN based diagnostic model for the early detection of dengue disease*. New Horizons in Translational Medicine. DOI: 10.1016/j.nhtm.2017.10.001.
- Gerardi, D. y Monteiro, L. (2011). *System Identification and Prediction of Dengue Fever Incidence in Rio de Janeiro*. Hindawi Publishing Corporation, Mathematical Problems in Engineering, Volume 2011. DOI: 10.1155/2011/720304.
- Gurtler, R. y Vezzani, D. (2009). *El Dengue, el mosquito Aedes Aegypti y la prevención*. CONICET: Facultad de Ciencias Exactas y Naturales – UBA.
- Hangeland J. (1988). *La Inteligencia Artificial*. Siglo XXI Editores. México.
- Hayes P. (1973). *Some comments on Sir James Lighthill's Report on Artificial Intelligence*. Esic Market. Londres.
- Hilera, J. Martínez, V. (1995). *Redes Neuronales Artificiales: Fundamentos, modelos y aplicaciones*. Madrid: RA-MA Editorial.
- Husin, N., Alharogi, A., Mustapha, N., Hamdan, H. y Husin, U. (2018). *Early Self-Diagnosis of Dengue Symptoms Using Fuzzy and Data Mining Approach*. American Institute of Physics (2016). DOI: 10.1063/1.5055450.
- Ibrahim, F., Nasir, M., Abu, W., Chong, Ch. y Sulaiman, S. (2005). *A novel dengue fever (DF) and dengue hemorrhagic fever (DHF) analysis using artificial neural network (ANN)*. Elsevier, Malaysia.
- Ibrahim, F., Faisal, T., Salim, M., y Taib, M. (2010). *Non-invasive diagnosis of risk in dengue patients using bioelectrical impedance analysis and artificial neural network*. Medical and Biological Engineering. DOI: 10.1007/s11517-010-0669-z.
- INEI (2018). Censos Nacionales 2017: XII de Población y VII de Vivienda. Recuperado de: [https://www.inei.gov.pe/media/MenuRecursivo/publicaciones\\_digitales/Est/Lib1530/libro.pdf](https://www.inei.gov.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1530/libro.pdf).
- Jayasundara, P., Perera, S., Malavige, G. y Jayasinghe, S. (2017). *Mathematical modelling and a systems science approach to describe the role of cytokines in the evolution of severe dengue*. BMC Systems Biology. DOI: 10.1186/s12918-017-0415-3.
- Juárez, J., Soto, P. Bernuy, G. y otros (2005). Evaluación de la definición de caso probable de dengue clásico durante el brote de dengue en Lima, 2005. Lima: Revista Peruana de Medicina Experimental.
- Koh, Y., Spindler, R., Sandgren, M., y Jiang, J. (2018). *A model comparison algorithm for increased forecast accuracy of dengue fever incidence in Singapore and the auxiliary role of total precipitation information*. Revista: International Journal of Environmental Health Research. DOI: 10.1080/09603123.2018.1496234.
- Laureano-Rosario, A., Duncan, A., Mendez-Lazaro, P., Garcia-Rejon, J., Gomez-Carro, S., Farfán-Ale, J., Savic, D., y Muller-Karger, F. (2018). *Application of Artificial Neural Networks for Dengue Fever Outbreak Predictions in the Northwest Coast of Yucatan, Mexico and San Juan, Puerto Rico*. MDPI. DOI: 10.3390/tropicalmed3010005.
- Lyra, L., Florentino, H., Santos, F. y Cantane, D. (2013). *Dengue: Otimizacao via Algoritmo Genético*. CMAC Sudeste 2013. Sao Paulo, Brasil.
- Mahdi, A., Razali, A., y AlWakil, A. (2011). *Comparison of Fuzzy Diagnosis with K-Nearest Neighbor and Naïve Bayes Classifiers in Disease Diagnosis*. BRAIN: Broad Research in Artificial Intelligence & Neuroscience. <http://web.b.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=4&sid=993b6a26-e69c-4448-90d4-8e98cc3d796d%40sessionmgr120>.

- Minski M. (1988). *Practical Applications of Expert Systems*. Wellesley.
- MINSA (2011). *Situación del Dengue en Región San Martín*. Reporte del Centro de Operaciones de Emergencias – OGDN del Ministerio de Salud – Perú.
- MINSA (2017). *Casos de Dengue por Departamentos – 2017*. Centro Nacional de Epidemiología, Prevención y Control de Enfermedades.
- MINSA (2020). *Casos de Dengue por Departamentos -2020*. Centro Nacional de Epidemiología, Prevención y Control de Enfermedades.  
<https://www.dge.gob.pe/portal/docs/vigilancia/sala/2020/SE01/dengue.pdf>.
- Montaño, J. (2002). *Redes neuronales artificiales aplicadas al análisis de datos*. Palma de Mallorca, Universitat de les Illes Balears.
- Motta, D., Bandeira, A., Winkler, I., Souza, B., Dias, D., Morais, A., Lins, E., Kirchner, F., y Badaro, R. *Application of convolutional neural networks for classification of adult mosquitoes in the field*. México, Instituto Nacional de Salud Pública. DOI: org/10.1371/journal.pone.0210829.
- Muñoz, E. Espinoza, J. y Rivera, L. (2012). *Un sistema basado en redes neuronales artificiales para diagnóstico de Anemia Ferropénica*. Lima: Universidad Nacional Mayor de San Marcos.
- Mussemeci, E. y Codeco, F. (2020). *Modelos de pronóstico multivariados a gran escala para el dengue: LSTM versus regresión forestal aleatoria*. Revista, Spatial and Spatio - temporal Epidemiology. Volumen 35, noviembre 2020. Doi.org/10.1016/j.sste.2020.100372.
- OPS. (2009). *Dengue, Información General*. Organización Panamericana de la Salud.  
 Recuperado de:  
[https://www.paho.org/hq/index.php?option=com\\_content&view=article&id=4493:2010-informacion-general-dengue&Itemid=40232&lang=es](https://www.paho.org/hq/index.php?option=com_content&view=article&id=4493:2010-informacion-general-dengue&Itemid=40232&lang=es).
- OPS. (2011). *Number of Reported Cases of Dengue and Severe Dengue (SD) in The Americas, by Country*. Organización Panamericana de la Salud.
- OPS. (2014). *Alerta Epidemiológica. Fiebre por Chikungunya y Dengue en las Américas*. Organización Panamericana de la Salud.
- OPS. (2016). *Number of Reported Cases of Dengue and Severe Dengue (SD) in the Americas, by Country*. Organización Panamericana de la Salud: Epidemiological Week / EW 41.
- Padrón, J. (2006). *Bases del concepto de Investigación Aplicada*. Caracas. Recuperado de:  
<http://padron.entretemas.com.ve/InvAplicada/index.htm>.
- Paletta, M. (2007). *¿Sistemas Inteligentes, realidad o fantasía?* Venezuela, Universidad Católica Andrés Bello: Revista GUAYANA. Recuperado de:  
[http://books.google.com.pe/books?id=BCJm9jigHr4C&pg=PA77&lpg=PA77&dq=sistema+inteligente+mauricio+paletta&source=bl&ots=3UJ5FX-ZNJ&sig=\\_Ju3dnlfXEBCZfH7P8nKdbBIJ-s&hl=es&ei=O-xFTsX7Jajm0QHwrLHxBw&sa=X&oi=book\\_result&ct=result&resnum=1&ved=0CB8Q6AEwAA#v=onepage&q&f=false](http://books.google.com.pe/books?id=BCJm9jigHr4C&pg=PA77&lpg=PA77&dq=sistema+inteligente+mauricio+paletta&source=bl&ots=3UJ5FX-ZNJ&sig=_Ju3dnlfXEBCZfH7P8nKdbBIJ-s&hl=es&ei=O-xFTsX7Jajm0QHwrLHxBw&sa=X&oi=book_result&ct=result&resnum=1&ved=0CB8Q6AEwAA#v=onepage&q&f=false).
- Paletta, M. (2010). *Inteligencia Artificial Básica*. Puerto Ordaz: Fondo Editorial UNEG.
- Quispe, J. (2 de marzo del 2020). Epidemia: 15 fallecidos por dengue en dos meses. *La República*. Recuperado de: <https://larepublica.pe/sociedad/2020/03/02/dengue-en-peru-2020-quince-fallecidos-en-dos-meses-segun-minsa/>.
- Rich E. (1985). *Artificial Intelligense*. Mc Graw Hill. New York.
- Saikia, D. y Dutta, J. (2016). *Early Diagnosis of Dengue Disease Using Fuzzy Inference System*. IEEE. DOI: 100.1109/microcosm.2016.7522513.

- San Martín, J. (2010). *The Epidemiology of Dengue in the Americas Over the Last Three Decades: A Worrying Reality*. Organización Panamericana de la Salud. Recuperado de: [http://new.paho.org/hq/index.php?option=com\\_content&task=view&id=264&Itemid=363&lang=es](http://new.paho.org/hq/index.php?option=com_content&task=view&id=264&Itemid=363&lang=es).
- Santos, O., Reis, L., y Chiaravalloti-Neto, F. (2018). *Dengue forecasting in São Paulo city with generalized additive models, artificial neural networks and seasonal autoregressive integrated moving average models*. PLOS ONE. DOI: 10.1371/journal.pone.0195065.
- Sasongko, P., Wibawa, H., Maulana, F., y Bahtiar, N. (2017). *Performance Comparison of Artificial Neural Network Models for Dengue Fever Disease Detection*. IEEE, First International Conference on Informatics and Computational Sciences (ICICoS). DOI: 10.1109/ICICOS.2017.8276359.
- Shahzad, M., Tripathi, N., Faruque, F., Bajwa, U., Kitamoto, A. y Souris, M. (2014). *Mapping urban and peri-urban breeding habitats of Aedes mosquitoes using a fuzzy analytical hierarchical process based on climatic and physical parameters*. Revista, Geospatial Health. DOI: 10.4081/gh.2014.297.
- Silla, J., Nunes, C., Cormanich, R., Guerreiro, M., Ramalho, T. y Freitas, M. (2011). *MIA-QSPR and effect of variable selection on the modeling of kinetic parameters related to activities of modified peptides against dengue type 2*. Revista, Chemometrics and Intelligent Laboratory Systems. DOI: 10.1016/j.chemolab.2011.06.009.
- Silveira, G. y de Barros, L. (2013). *Numerical methods integrated with fuzzy logic and stochastic method for solving PDEs: An application to dengue*. Fuzzy sets and Systems; UNICAMP. DOI: org/10.1016/j.fss.2013.04.003.
- Silveira, G. y de Barros, L. (2015). *Analysis of the dengue risk by means of a Takagi–Sugeno style model*. Revista, Fuzzy Sets and Systems. DOI: 10.1016/j.fss.2015.03.003.
- Torres, J. (2011). *Procesamiento y Análisis de Imágenes Satelitales utilizando Lógica Difusa para apoyar en el estudio de casos de Dengue*. Universidad de Colima. México.
- Torres, C., Barguil, S., Melgarejo, M., y Olarte, A. (2014). *Fuzzy model identification of dengue epidemic in Colombia based on multiresolution analysis*. Artificial Intelligence in Medicine. DOI: 10.1016/j.artmed.2013.11.008.
- Trujillano, J. (2004). *Aproximación metodológica al uso de redes neuronales artificiales para la predicción de resultados en medicina*. España: Universitat De Lleida. Recuperado de: <http://web.udl.es/Biomath/Group/Treballs/2004MedClin.pdf>.
- Velandia, M. y Castellanos, J. (2011). *Virus del dengue: estructura y ciclo viral*. Asociación Colombiana de Infectología. <http://www.scielo.org.co/pdf/inf/v15n1/v15n1a06.pdf>.
- Wickramaarachchi, WPTM. y Jayasinghe, S. (2015). *Investigating the impact of climate on dengue disease transmission in urban Colombo: A Fuzzy logic model*. 4ta. Conferencia Internacional Anual sobre Matemática Computacional, Geometría y Estadística Computacional (CMCGS 2015). DOI: 10.5176/2251-1911\_CMCGS15.10.
- Widayani, H., Kallista, M., Nuraini, N. y Sari, M. (2014). *Optimization Model of Vaccination Strategy for Dengue Transmission*. AIP Conference Proceedings. DOI: 10.1063/1.4866548.
- Wittkoski D. (2010). *Manual para la Formulación y Evaluación de Proyectos de Inversión*. Cooperación Técnica del BID. Lima.
- Zhu, G., Saw, G., Nalaparaju, A., Grüber, G., y Lu, L. (2017). *Coarse-Grained Molecular Modeling of the Solution Structure Ensemble of Dengue Virus Nonstructural Protein 5 with*

*Small-Angle X-ray Scattering Intensity*. Revista, Journal of Physical Chemistry B. DOI: 10.1021/acs.jpcc.7b00051.

## Anexo 1

### Mejor performance *mse* y **R** con una capa oculta.

NUMERO NEURONAS	MSE	R
CAPA OCULTA	Error Cuadrático Medio	Factor de Regresión
1	0.2135	0.4376
	0.2159	0.2965
	0.2000	0.3847
	0.1907	0.2517
	0.2287	0.1573
10	0.1537	0.8298
	0.0411	0.9305
	0.1001	0.9117
	0.1092	0.8686
	0.0997	0.7896
20	0.1215	0.8882
	0.0482	0.8984
	0.1529	0.5673
	0.2029	0.8435
	0.1650	0.8170
30	0.1034	0.9059
	0.1490	0.9125
	0.0621	0.9224
	0.1731	0.8521
	0.0787	0.9090
40	0.1484	0.8644
	0.0560	0.9109
	0.0956	0.8983
	0.1135	0.8917
	0.1065	0.9179
50	0.1238	0.8950
	0.1476	0.8777
	0.1408	0.7322
	0.1225	0.8931
	0.0706	0.9105
60	0.0963	0.9062
	0.0579	0.9163
	0.1095	0.8885
	0.1053	0.8999
	0.0861	0.9014

70	0.0587	0.9084
	0.0986	0.8963
	0.0755	0.9276
	0.0722	0.9307
	0.0557	0.9277
80	0.0674	0.8779
	0.0779	0.8551
	0.0976	0.9174
	0.0660	0.9117
	0.0915	0.9199
90	0.1429	0.8791
	0.1735	0.8823
	0.0995	0.8809
	0.1070	0.8744
	0.0506	0.9079
100	0.0933	0.9054
	0.1168	0.8870
	0.1298	0.9194
	0.0722	0.9085
	0.1488	0.8742

Fuente: Propia.

## Anexo 2

Mejor performance *mse* y *R* variando el número de neuronas.

NUMERO NEURONAS	MSE	R
CAPA OCULTA	Error Cuadrático Medio	Factor de Regresión
10	0.1537	0.8298
	0.0411	0.9305
	0.1001	0.9117
	0.1092	0.8686
	0.0997	0.7896
11	0.0289	0.8961
	0.1588	0.8823
	0.0691	0.9107
	0.0876	0.8953
	0.1216	0.8744
12	0.1150	0.9010
	0.0562	0.9148
	0.0829	0.8958
	0.0808	0.8709
	0.0911	0.9024
13	0.0384	0.9305
	0.0705	0.8921
	0.1134	0.8779
	0.1213	0.8849
	0.0580	0.8747
14	0.1144	0.8953
	0.1134	0.8798
	0.0796	0.8851
	0.0645	0.9249
	0.0984	0.9045
15	0.1721	0.8682
	0.0855	0.9247
	0.1605	0.8393
	0.0720	0.9133
	0.0942	0.9069
16	0.1085	0.9199
	0.0775	0.8812
	0.1591	0.8703
	0.1151	0.8738
	0.0828	0.9171

17	0.1502	0.8596
	0.1803	0.8665
	0.0633	0.9066
	0.0818	0.9089
	0.1360	0.8716
18	0.1332	0.8634
	0.0917	0.8952
	0.0905	0.9129
	0.1567	0.8808
	0.0861	0.8936
19	0.1164	0.8780
	0.0805	0.8820
	0.0469	0.9360
	0.1607	0.8929
	0.1818	0.8532
20	0.1215	0.8882
	0.0482	0.8984
	0.1529	0.5673
	0.2029	0.8435
	0.1650	0.8170

Fuente: Propia.



```
% Visualizamos los errores en Workspace.
errorEntrenamiento=[];
errorValidacion=[];
errorTest=[];
errorEntrenamiento(end+1)=tr.best_perf;
errorValidacion(end+1)=tr.best_vperf;
errorTest(end+1)=tr.best_tperf;

% Exportamos los resultados al ambiente de trabajo SIMULINK.
gensim(red);
```





## Anexo 5

### Datos de test

43 patrones y resultados deseados.

	CODIGO MINSA	8	41	57	74	89	124	142	160	187	195	209	243	269	284	297	313	317	345	372	394	401	410	426	431	449	476	494	507	518	535	544	561	585	621	630	647	676	689	717	725	743	768	796					
Número	SINTOMAS/PATRONES	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27	T28	T29	T30	T31	T32	T33	T34	T35	T36	T37	T38	T39	T40	T41	T42	T43					
1	Fiebre	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0				
2	Artralgias	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	1	0	0	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1			
3	Artralgias_manos	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	1	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1			
4	Artralgias_pies	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	1	1	1	1	0	0	0	1	1	1	1	1	0	0	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1		
5	Mialgias	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	0	1	1	1	1	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0			
6	Cefalea	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0			
7	Dolor_ocular	1	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	1	0	0	1	1	0	1	0	1	1	0	1	0	1	1	1	0	1	1	1	0	1	1	0	1	0	1	0	1	0		
8	Dolor_lumbar	1	0	0	1	0	0	1	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0		
9	Rash_exantema	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	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	
10	Conjuntivitis	0	0	0	0	0	0	0	0	0	0	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	
11	Nauseas_vomitos	0	1	1	1	1	1	1	1	0	0	1	0	1	0	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0		
12	Dolor_abdominal	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	1	1	1	1	0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1	1	1	0		
13	Dolor_torácico	0	0	0	0	0	0	0	0	0	0	0	0	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	
14	Derrame_seroso	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	
15	Vómito_persistente	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	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
16	Ictericia	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	
17	Pulso_débil	0	0	0	0	0	0	0	0	0	0	0	0	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	
18	Extremidad	0	0	0	0	0	0	1	0	0	0	0	0	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
19	Difresión	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	
20	Sangrado_grave	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	
21	Apertura_ocular	1	0	0	0	1	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
22	Respuesta_motora	1	0	0	0	1	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
23	Respuesta_verbal	1	0	0	0	1	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	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
24	Diarrea	1	0	0	1	0	0	0	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	
<b>Resultados Esperados</b>		<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>																				

## Anexo 6

### Síntomas del dengue y sus respectivos códigos.

NUMERO	SINTOMA	ETIQUETA
1	Fiebre	Fiebre
2	Artralgias	Artralgias
3	Artralgias manos	Artralgias_ma
4	Artralgias pies	Artralgias_pi
5	Mialgias	Mialgias
6	Cefalea	Cefalea
7	Dolor ocular o retro ocular	Dolor_ocul
8	Dolor lumbar	Dolor_lumb
9	Rash/exantema o sarpullido	Rash_exant
10	Conjuntivitis no purulenta (ojos rojos)	Conjuntivi
11	Nauseas/vómitos	Nauseasvom
12	Dolor abdominal intenso y continuo	Dolor_abdo
13	Dolor torácico o disnea	Dolor_tora
14	Derrame seroso	Derrame_se
15	Vómitos persistentes	Vómitos
16	Ictericia (coloración amarillenta de piel)	Ictericia
17	Pulso débil e indetectable	Pulso_debi
18	Extremidades frías o cianóticas	Extremidad
19	Diferencial de presión arterial (<20mmHg)	Difpresión
20	Sangrado grave	Sang_grave
21	Escala de Glasgow: Apertura ocular (1-4)	Apertura_ocul
22	Escala de Glasgow: Respuesta motora (1-6)	Resp_motor
23	Escala de Glasgow: Respuesta verbal (1-5)	Resp_verba
24	Diarrea	Diarrea

Fuente: Propia.

## Anexo 7

### Cascade-forward neural network.

```
>> red
```

```
red =
```

```
Neural Network
```

```
  name      : 'Cascade-Forward Neural Network'  
  userdata: (your custom info)
```

```
dimensions:
```

```
  numInputs: 1  
  numLayers: 3  
  numOutputs: 1  
  numInputDelays: 0  
  numLayerDelays: 0  
  numFeedbackDelays: 0  
  numWeightElements: 489  
  sampleTime: 1
```

```
connections:
```

```
  biasConnect   : [1; 1; 1]  
  inputConnect  : [1; 1; 1]  
  layerConnect  : [0 0 0; 1 0 0; 1 1 0]  
  outputConnect : [0 0 1]
```

```
subobjects:
```

```
  input : Equivalent to inputs{1}  
  output : Equivalent to outputs{3}  
  
  inputs : {1x1 cell array of 1 input}  
  layers : {3x1 cell array of 3 layers}  
  outputs : {1x3 cell array of 1 output}  
  biases : {3x1 cell array of 3 biases}  
  inputWeights : {3x1 cell array of 3 weights}  
  layerWeights : {3x3 cell array of 3 weights}
```

```
functions:
```

```
  adaptFcn : 'adaptwb'
```

```

    adaptParam      : (none)
      derivFcn     : 'defaultderiv'
      divideFcn    : 'dividerand'
    divideParam    : .trainRatio, .valRatio, .testRatio
    divideMode     : 'sample'
      initFcn     : 'initlay'
      performFcn  : 'mse'
    performParam   : .regularization, .normalization
      plotFcns    : {'plotperform', plottrainstate, ploterrhist, plotregression}
      plotParams  : {1x4 cell array of 4 params}
      trainFcn    : 'trainlm'
      trainParam  : .showWindow, .showCommandLine, .show, .epochs, .time, .goal,
.min_grad, .max_fail, .mu, .mu_dec, .mu_inc, .mu_max

```

weight and bias values:

```

    IW : {3x1 cell} containing 3 input weight matrices
    LW : {3x3 cell} containing 3 layer weight matrices
    B  : {3x1 cell} containing 3 bias vectors

```

methods:

```

    adapt      : Learn while in continuous use
    configure  : Configure inputs & outputs
    genism     : Generate Simulink model
    init       : Initialize weights & biases
    perform    : Calculate performance
      sim      : Evaluate network outputs given inputs
      train    : Train network with examples
      view     : View diagram
    unconfigure : Unconfigure inputs & outputs

    evaluate :   outputs = red(inputs)

```

>>

## Anexo 8

### Función parámetros para trainlm

```
>> red.trainparam
```

```
ans =
```

```
Function Parameters for 'trainlm'
```

Show Training Window Feedback	showWindow: true
Show Command Line Feedback	showCommandLine: false
Command Line Frequency	show: 25
Maximum Epochs	epochs: 1000
Maximum Training Time	time: Inf
Performance Goal	goal: 0
Minimum Gradient	min_grad: 1e-07
Maximum Validation Checks	max_fail: 6
Mu	mu: 0.001
Mu Decrease Ratio	mu_dec: 0.1
Mu Increase Ratio	mu_inc: 10
Maximum mu	mu_max: 10000000000

```
>>
```

## Anexo 9

### Salida objetivo (target).

Target =

Columns 1 through 16

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

Columns 17 through 32

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

Columns 33 through 48

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

Columns 49 through 64

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

Columns 65 through 80

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

Columns 81 through 96

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

Columns 97 through 112

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

Columns 113 through 128

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

Columns 129 through 144

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

Columns 145 through 160

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

Columns 161 through 176

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

Columns 177 through 192

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

Columns 193 through 208

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

Columns 209 through 224

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

Columns 225 through 240

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

Columns 241 through 256

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

Columns 257 through 272

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

Columns 273 through 288

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

Columns 289 through 304

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

Columns 305 through 320

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

Columns 321 through 336

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

Columns 337 through 352

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

Columns 353 through 368

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

Columns 369 through 384

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

Columns 385 through 400

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

Columns 401 through 416

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

Columns 417 through 430

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

**Salida output después del entrenamiento.**

Output =

Columns 1 through 9

0.9968 0.9981 0.9980 0.0003 0.9993 0.9974 0.5374 0.9950 0.0020

Columns 10 through 18

0.9919 0.9966 0.9973 0.9979 0.9979 0.6675 0.9967 0.9984 0.6675

Columns 19 through 27

0.0007 0.9967 0.0071 0.9974 0.0083 0.0073 0.0046 0.9992 0.9993

Columns 28 through 36

0.9975 0.9989 0.0073 1.0000 1.0000 0.9976 0.9990 0.9959 0.9991

Columns 37 through 45

0.9970 1.0000 0.0103 0.9978 0.9968 0.9966 0.0018 0.9928 0.9998

Columns 46 through 54

0.9998 1.0000 0.0000 0.9978 0.9992 0.1384 0.9995 0.0054 0.9758

Columns 55 through 63

0.5070 1.0000 1.0000 0.9993 0.5070 0.0502 0.0038 0.9906 0.9907

Columns 64 through 72

0.0078 0.0042 0.9999 0.9904 0.0002 0.9996 0.9977 0.9985 0.0017

Columns 73 through 81

0.9999 0.9994 0.9987 0.9989 0.0010 0.0024 0.0484 0.9992 0.9977

Columns 82 through 90

0.9974 0.9975 0.0021 0.9939 0.9996 0.0012 0.9985 1.0000 0.0041

Columns 91 through 99

0.9985 0.9991 0.9956 0.0002 1.0000 0.0000 1.0000 0.9995 0.9998

Columns 100 through 108

0.0028 0.9956 0.0033 0.9989 0.9998 0.9981 0.0001 0.0025 0.9989

Columns 109 through 117

0.9948 0.0012 0.9988 0.9984 0.9985 0.0044 0.5493 0.9988 0.0110

Columns 118 through 126

0.9974 0.9994 0.0216 0.9994 0.0116 0.9911 0.0009 0.9947 0.9997

Columns 127 through 135	0.9952	0.9992	0.0014	0.0033	0.5374	0.9974	1.0000	0.0003	0.0003
Columns 136 through 144	0.9999	0.0101	0.0045	0.9989	0.9994	1.0000	0.9996	0.9998	0.0012
Columns 145 through 153	0.9961	0.9998	0.9988	0.9986	0.0040	0.9975	0.0086	0.9982	1.0000
Columns 154 through 162	0.0122	0.9982	0.9973	0.9974	0.0162	1.0000	0.0052	0.9975	0.9947
Columns 163 through 171	0.9948	0.0045	0.9899	0.0026	0.9999	0.0015	1.0000	0.0056	0.9977
Columns 172 through 180	0.9992	0.9947	0.9953	0.0009	0.0034	0.9956	0.9953	0.0002	0.0051
Columns 181 through 189	0.0068	0.0018	0.0026	0.5493	0.0004	1.0000	0.0011	0.9986	1.0000
Columns 190 through 198	0.9985	1.0000	0.0048	0.9966	0.0036	0.0019	0.9973	1.0000	0.0037
Columns 199 through 207	0.9973	0.9981	0.9992	0.0011	0.9964	0.9986	0.9998	0.9995	0.9938
Columns 208 through 216	0.9928	1.0000	0.9895	0.0038	1.0000	1.0000	0.9974	0.0036	0.9968
Columns 217 through 225	0.9981	0.9980	0.0003	0.9993	0.9974	0.5374	0.9950	0.0020	0.9919
Columns 226 through 234	0.9966	0.9973	0.9979	0.9979	0.6675	0.9967	0.9984	0.6675	0.0007
Columns 235 through 243	0.9967	0.0071	0.9974	0.0083	0.0073	0.0046	0.9992	0.9993	0.9975
Columns 244 through 252	0.9989	0.0073	1.0000	1.0000	0.9976	0.9990	0.9959	0.9991	0.9970
Columns 253 through 261	1.0000	0.0103	0.9978	0.9968	0.9966	0.0018	0.9928	0.9998	0.9998

Columns 262 through 270	1.0000	0.0000	0.9978	0.9992	0.1384	0.9995	0.0054	0.9758	0.5070
Columns 271 through 279	1.0000	1.0000	0.9993	0.5070	0.0502	0.0038	0.9906	0.9907	0.0078
Columns 280 through 288	0.0042	0.9999	0.9904	0.0002	0.9996	0.9977	0.9985	0.0017	0.9999
Columns 289 through 297	0.9994	0.9987	0.9989	0.0010	0.0024	0.0484	0.9992	0.9977	0.9974
Columns 298 through 306	0.9975	0.0021	0.9939	0.9996	0.0012	0.9985	1.0000	0.0041	0.9985
Columns 307 through 315	0.9991	0.9956	0.0002	1.0000	0.0000	1.0000	0.9995	0.9998	0.0028
Columns 316 through 324	0.9956	0.0033	0.9989	0.9998	0.9981	0.0001	0.0025	0.9989	0.9948
Columns 325 through 333	0.0012	0.9988	0.9984	0.9985	0.0044	0.5493	0.9988	0.0110	0.9974
Columns 334 through 342	0.9994	0.0216	0.9994	0.0116	0.9911	0.0009	0.9947	0.9997	0.9952
Columns 343 through 351	0.9992	0.0014	0.0033	0.5374	0.9974	1.0000	0.0003	0.0003	0.9999
Columns 352 through 360	0.0101	0.0045	0.9989	0.9994	1.0000	0.9996	0.9998	0.0012	0.9961
Columns 361 through 369	0.9998	0.9988	0.9986	0.0040	0.9975	0.0086	0.9982	1.0000	0.0122
Columns 370 through 378	0.9982	0.9973	0.9974	0.0162	1.0000	0.0052	0.9975	0.9947	0.9948
Columns 379 through 387	0.0045	0.9899	0.0026	0.9999	0.0015	1.0000	0.0056	0.9977	0.9992
Columns 388 through 396	0.9947	0.9953	0.0009	0.0034	0.9956	0.9953	0.0002	0.0051	0.0068
Columns 397 through 405									

0.0018 0.0026 0.5493 0.0004 1.0000 0.0011 0.9986 1.0000 0.9985

Columns 406 through 414

1.0000 0.0048 0.9966 0.0036 0.0019 0.9973 1.0000 0.0037 0.9973

Columns 415 through 423

0.9981 0.9992 0.0011 0.9964 0.9986 0.9998 0.9995 0.9938 0.9928

Columns 424 through 430

1.0000 0.9895 0.0038 1.0000 1.0000 0.9974 0.0036

## Anexo 10

### Pesos y bías entre capas

#### Pesos de las conexiones entre la capa de entrada y la capa oculta número uno.

Este arreglo de pesos corresponde a una Matriz 12 x 24, es decir, 12 filas por 24 columnas. Haciendo un total de 288 pesos.

pesosentrada\_capaoculta1 =

Columns 1 through 9

-1.1211	0.3220	-0.5029	-0.3756	0.7895	-1.1505	0.6682	0.1022	-0.3029
-0.2571	-0.0373	-0.2790	-0.5048	-0.4133	0.8316	0.1405	-1.3468	-1.1225
-0.6055	-0.6226	0.3336	-0.6723	0.4907	-0.5221	0.2058	-0.0395	-0.9239
-0.2659	-0.8195	0.9077	0.7004	-0.1756	0.1608	-1.1743	1.2437	0.3559
1.4580	-1.4808	-0.2945	0.5860	0.7627	-0.0058	0.3445	0.5159	0.1345
1.0772	-1.0983	-0.5014	-0.1859	0.1780	-0.7250	-0.8393	-0.0354	0.8041
-1.6293	-1.1311	-1.2407	-0.4251	0.1232	0.1134	0.3781	0.5939	-0.7057
-0.1248	0.7298	0.2243	-0.5604	-0.5316	1.0473	0.0211	0.4717	1.3033
-0.5456	0.3210	-0.5087	-0.7277	-0.0145	0.7884	0.4253	-0.1128	-0.0116
0.8374	-1.1872	0.3630	-0.5757	-1.2434	0.1602	-0.9334	0.2187	0.7756
-0.1427	-0.0600	0.6617	-0.0842	0.8084	0.3833	0.9438	1.1921	0.2952
-1.2863	1.7949	0.4198	-0.2810	1.0373	1.2234	0.2796	0.1762	-0.7404

Columns 10 through 18

0.9101	0.2420	-0.5760	-0.1856	-0.1649	0.5120	-0.4119	-0.6349	0.2358
-0.4084	0.4233	0.1901	-0.1072	1.2414	0.0860	0.1273	-0.7571	-0.4158
0.2170	-0.2859	-0.4392	-0.2908	-0.2823	0.1279	-0.0500	0.3084	0.2776
-0.6850	1.6148	0.5133	-0.1391	-0.2464	-1.3757	0.6466	1.0294	0.0828
-0.1107	1.0389	0.4586	0.4433	0.4267	-0.3243	-0.1421	0.1885	1.1105

0.0320	0.1538	-0.7527	0.4579	0.2203	0.1028	-0.4603	-0.2153	0.2927
-0.6035	0.0123	1.3371	-0.4376	-0.5119	-0.2750	-0.1044	-0.0125	-0.3059
1.0394	-1.6458	-0.7954	-0.3758	0.4086	-0.2969	-0.4097	0.1026	-0.2391
0.0712	-0.7031	-0.8806	-0.7129	-0.1637	0.3421	-0.1017	0.2755	0.1178
-0.6540	1.1585	-0.5136	-0.5837	-0.2560	1.5714	-0.1788	-0.4471	0.2748
0.6990	-0.8388	1.4426	0.3762	0.3538	-1.0107	-0.0603	0.5721	0.1009
0.1432	0.3390	0.0485	-0.3901	-0.2913	0.5489	0.0758	-0.3479	-0.6647

Columns 19 through 24

-0.6672	-0.0340	-0.8847	-0.8979	-0.9744	0.4059
0.1560	-0.7373	0.6741	0.8516	0.5764	-0.1875
0.6405	0.4628	-0.5995	-0.1584	-0.4894	-0.0385
0.7137	-0.0784	-0.0258	-0.2193	-0.1740	0.0089
-0.2640	-0.0579	0.6943	0.6703	-0.0943	1.2768
-0.1277	0.0430	-0.3958	-0.4587	-0.9742	0.4044
-0.0984	0.1117	0.4117	0.3043	0.3565	0.8533
-0.3467	0.5228	-0.1275	-0.3708	-0.3236	0.6202
0.0703	-0.5573	0.0870	1.1810	0.0579	-0.5245
-0.2689	0.0436	-0.2673	-0.4579	-0.6191	0.5945
0.0087	0.0064	-0.0863	-0.7719	0.3175	-0.0858
-0.3871	-0.2157	0.3019	0.1944	0.1789	1.6737

**Pesos de las conexiones entre la capa oculta número uno y la capa oculta número dos.**

Este arreglo representa una Matriz de 4 x 12, es decir, 4 filas por 12 columnas. Haciendo un total de 48 pesos.

pesoscapaoculta1\_capaoculta2 =

Columns 1 through 9

-1.4183	-0.1864	0.2983	0.2115	-0.2770	0.0546	-1.1075	0.1097	0.3026
---------	---------	--------	--------	---------	--------	---------	--------	--------

-0.0118 0.5459 -0.2248 0.3331 -0.3950 -0.2415 -0.2119 -0.5229 -0.4126  
-0.0911 0.1833 -0.1533 -0.3355 0.2523 0.5515 0.4819 -0.4223 -0.1397  
0.6202 -0.3310 -0.1521 0.0090 0.2008 -0.2577 -0.4700 0.7449 0.5482

Columns 10 through 12

0.2182 0.3213 0.2293  
0.4477 -0.7314 -0.3015  
-0.9957 -0.0290 0.9832  
-0.2650 0.0260 0.4347

### **Pesos de las conexiones entre la capa oculta número dos y la capa de salida.**

El arreglo representa una Matriz 1 x 4, es decir, una fila por cuatro columnas, haciendo un total de cuatro pesos.

pesoscapaoculta2\_salida =

1.9305 1.3049 -2.3471 -1.7074

### **Bías de la capa oculta número uno.**

Este arreglo es una Matriz 12 x 1, es decir, doce filas por una columna; haciendo un total de 12 bías.

bias\_capaoculta1 =

1.7078  
-1.0514  
0.8034  
-0.4356

-0.5256  
0.0515  
-0.3764  
-0.4066  
-0.4817  
1.4406  
1.5109  
-1.5872

**Bías de la capa oculta número dos.**

Este arreglo es una Matriz 4 x 1, es decir, cuatro filas por una columna; haciendo un total de cuatro vías.

bias\_capaoculta2 =

1.5752  
-0.4445  
0.2779  
1.4443

**Bías de la capa de salida.**

Este arreglo es una Matriz 1 x 1, es decir, con un solo elemento.

bías\_capasalida =

1.1374