



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES**

**“DESARROLLO DE UN ALGORITMO DE REDES NEURONALES  
ARTIFICIALES APLICADO A LA PREDICCIÓN DE TRÁFICO DE LA  
INFRAESTRUCTURA DE COMUNICACIONES DE REDES  
CORPORATIVAS”**

TRABAJO DE TITULACIÓN: **PROYECTO TÉCNICO**

Para optar al Grado Académico de:

**INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES**

**AUTORES:** BARRAZUETA LÓPEZ PAMELA LOURDES  
TIERRA AMAGUAYA LENNIN SANTIAGO

**TUTOR:** Ing. Alberto Arellano Aucancela

Riobamba-Ecuador

2018

**@2018, Pamela Lourdes Barraqueta López, Lennin Santiago Tierra Amaguaya.**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y  
REDES**

El Tribunal de trabajo de titulación certifica que: El trabajo de titulación: **DESARROLLO DE UN ALGORITMO DE REDES NEURONALES ARTIFICIALES APLICADO A LA PREDICCIÓN DE TRÁFICO DE LA INFRAESTRUCTURA DE COMUNICACIONES DE REDES CORPORATIVAS**, de responsabilidad de la señorita Pamela Lourdes Barrazueta López y el señor Lennin Santiago Tierra Amaguaya, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación quedando autorizado su presentación.

ING. JULIO SANTILLÁN

**VICEDECANO DE LA FACULTAD DE**

**INFORMÁTICA Y ELECTRÓNICA** \_\_\_\_\_

ING. FRANKLIN MORENO MONTENEGRO

**DIRECTOR DE LA ESCUELA DE**

**INGENIERÍA ELECTRÓNICA,**

**TELECOMUNICACIONES Y REDES** \_\_\_\_\_

ING. ALBERTO ARELLANO AUCANCELA

**DIRECTOR DE TESIS** \_\_\_\_\_

ING. VINICIO RAMOS VALENCIA

**MIEMBRO DEL TRIBUNAL** \_\_\_\_\_

Nosotros, Pamela Lourdes Barraqueta López y Lennin Santiago Tierra Amaguaya, somos responsables de las ideas, doctrinas y resultados expuestos en este trabajo de titulación y el patrimonio intelectual del Trabajo de Titulación, y el patrimonio de la misma pertenece a la Escuela Superior Politécnica de Chimborazo

Pamela Lourdes Barraqueta López

Lennin Santiago Tierra Amaguaya

## **DEDICATORIA**

A Dios por haberme bendecido cada momento de mi vida estudiantil, a mis padres quienes son pilares fundamentales de mi vida y han sido la guía y el camino para llegar a culminar mi carrera, a mis hermanos por compartir conmigo muchas alegrías y fracasos. Les dedico todo mi esfuerzo en reconocimiento a todo el sacrificio puesto en mí para cumplir esta meta, ustedes mi querida familia se merecen esto y mucho más, con amor

Lennin

Este trabajo se lo dedico a Dios quien supo guiarme, darme fuerza y nunca perder la fe en todo este tiempo. A mi familia quienes son mi mayor orgullo, a mis padres, Miguel y Mónica quienes siempre me apoyaron en todas mis decisiones para cumplir todas mis metas, a mis hermanas Ailin y Aneli por enseñarme que a pesar de la distancia el amor y cariño siempre prevalecerá.

Pamela

## **AGRADECIMIENTO**

A mi familia quienes siempre me brinda su amor y me impulsaron a cumplir día a día mis sueños y metas, primordialmente a mis padres los cuales confían plenamente en mí, ya que este éxito no solo es mío sino también de ellos, a mis amigos con quienes compartí en estos años universitarios tantas alegrías y tristezas siempre los recordare, a todos los ingenieros quienes me impartieron clases las cuales me ayudaron a formarme en la profesional que soy hoy en día.

Pamela

El más sincero agradecimiento a mis padres que a pesar de cualquier inconveniente, se han fortalecido para permitirnos estudiar a mis hermanos y a mí, además de brindarme su amor incondicional.

Lennin.

## TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS .....	x
ÍNDICE DE GRÁFICOS.....	xii
ÍNDICE DE ANEXOS .....	xiii
ANEXO A: TABLAS DE CAPTURAS .....	xiii
ANEXO B: CÓDIGO COMPLETO DE LA RED NEURONAL .....	xiii
RESUMEN .....	xiv
ABSTRACT .....	xv
INTRODUCCIÓN .....	1
CAPÍTULO I.....	8
1. MARCO TEÓRICO REFENCIAL .....	8
1.1. Redes neuronales artificiales .....	8
1.1.1. Definición .....	8
1.1.2. Componentes de las redes neuronales artificiales .....	8
1.1.3. Funciones de activación. ....	11
1.1.4. Proceso básico de la obtención de los valores para la salida de las rna's .....	12
1.1.5. Ventajas y desventajas de las redes neuronales artificiales .....	13
1.2. Clasificación de las redes neuronales artificiales .....	14
1.2.1. Redes neuronales artificiales por topología .....	14
1.2.1.1. Redes neuronales artificiales monocapa .....	15
1.2.1.2. Redes neuronales artificiales multicapa .....	15
1.2.2. Redes neuronales artificiales según el tipo de aprendizaje.....	17
1.3. Tipos de redes neuronales artificiales .....	18
1.3.1. Perceptrón multicapa .....	18
1.3.1.1. Arquitectura del perceptrón multicapa.....	18
1.4. Algoritmos de las redes neuronales artificiales .....	20
1.5. Aplicaciones de las redes neuronales artificiales.....	21
CAPÍTULO II .....	23
2. MARCO METODOLÓGICO.....	23
2.1. Desarrollo del trabajo de titulación. ....	23
2.1.1. Preparación de los datos de escenario de red .....	24
2.1.2. Pre procesamiento de datos. ....	30

2.1.3.	<i>Selección de protocolos para la predicción</i>	32
2.1.4.	<i>Transformación de los datos</i>	32
2.1.5.	<i>Selección del tipo de red neuronal artificial según los requerimientos necesarios</i>	38
2.1.6.	<i>Implementación de la red neuronal artificial seleccionada en la herramienta matlab.</i>	39
2.1.7.	<i>Implementación del algoritmo en la red neuronal artificial seleccionada.</i>	40
2.1.7.1.	<i>Creación de la red neuronal artificial</i>	41
2.1.7.2.	<i>Entrenamiento de la red neuronal</i>	42
2.1.7.3.	<i>Simulación de la red neuronal</i>	42
2.1.7.4.	<i>Interfaz gráfica de la red neuronal artificial</i>	43
<b>CAPÍTULO III</b>		<b>45</b>
<b>3.</b>	<b>EVALUACIÓN Y COMPARACIÓN DE RESULTADOS</b>	<b>45</b>
3.1.	<b>Simulación y análisis</b>	<b>45</b>
3.1.1.	<i>Simulación de la primera prueba</i>	45
3.1.2.	<i>Simulación de la segunda prueba</i>	58
3.1.3.	<i>Simulación de la tercera prueba</i>	70
<b>CONCLUSIONES</b>		<b>85</b>
<b>RECOMENDACIONES</b>		<b>86</b>
<b>BIBLIOGRAFÍA</b>		
<b>ANEXOS</b>		

## ÍNDICE DE TABLAS

<b>Tabla 1-1:</b> Comparación entre las redes neuronales.....	<b>19</b>
<b>Tabla2-1:</b> Aplicaciones de las redes neuronales artificiales en distintos campos.....	<b>22</b>
<b>Tabla1-2:</b> Características del servidor recolector de datos.....	<b>24</b>
<b>Tabla2-2:</b> Herramientas utilizadas para la implementación de la RNA.....	<b>25</b>
<b>Tabla3-2:</b> Tiempo y cantidad de tráfico capturado en la red de la FIE.....	<b>26</b>
<b>Tabla4-2:</b> Resumen del tiempo y cantidad de tráfico capturado en la red de la FIE.....	<b>30</b>
<b>Tabla5-2:</b> Inconsistencias y soluciones encontradas en la captura de tráfico de la red de la FIE.....	<b>32</b>
<b>Tabla 6-2:</b> Clasificación de los protocolos analizados de acuerdo al modelo OSI.....	<b>32</b>
<b>Tabla 7-2:</b> Trafico (Bytes) medido en 5 minutos segmentados por los protocolos de las capas de red y aplicación.....	<b>33</b>
<b>Tabla 8-2:</b> Tiempo y tráfico total (Gbytes) clasificado por los 6 protocolos.....	<b>33</b>
<b>Tabla 1-3:</b> Parámetros de simulación.....	<b>43</b>
<b>Tabla 2-3:</b> Parámetros de la primera prueba.....	<b>44</b>
<b>Tabla 3-3:</b> Resumen del error de la primera prueba.....	<b>58</b>
<b>Tabla 4-3:</b> Parámetros de la segunda prueba.....	<b>58</b>
<b>Tabla 5-3:</b> Error de la segunda prueba.....	<b>70</b>
<b>Tabla 6-3:</b> Parámetros de la tercera prueba.....	<b>70</b>
<b>Tabla 7-3:</b> Error de la tercera prueba.....	<b>82</b>
<b>Tabla 8-3:</b> Errores de las pruebas.....	<b>82</b>

## ÍNDICE DE FIGURAS

<b>Figura1-1:</b> Estructura básica de una neurona biológica.....	8
<b>Figura2-1:</b> Similitud entre neurona biológica y una neurona artificial.....	9
<b>Figura3-1:</b> Componentes una red neuronal artificial.....	9
<b>Figura4-1:</b> Función escalón y sigmoidea.....	10
<b>Figura5-1:</b> Modelo no-lineal de una red neuronal.....	11
<b>Figura6-1:</b> Organigrama de las redes neuronales artificial como criterio clasificar la topología.....	14
<b>Figura7-1:</b> Red neuronal artificial monocapa.....	15
<b>Figura8-1:</b> Red neuronal artificial multicapa.....	16
<b>Figura9-1:</b> Arquitectura de una red Perceptrón multicapa.....	19
<b>Figura1-2:</b> Script para la captura de tráfico de la infraestructura de la red de la FIE.....	26
<b>Figura2-2:</b> Protocolos en una captura de 5 minutos dentro de la red de infraestructura de la FIE.....	31
<b>Figura3-2:</b> Procesamiento de la computadora mientras realiza la predicción de los datos.....	39
<b>Figura4-2:</b> Red neuronal artificial backpropagation.....	40
<b>Figura5-2:</b> Líneas de código para cargar datos, crear un vector y visualización del tráfico.....	41
<b>Figura6-2:</b> Líneas de código para crear la RNA.....	41
<b>Figura7-2:</b> Líneas de código de los datos de entrada.....	42
<b>Figura8-2:</b> Líneas de código de entrenamiento de la red.....	41
<b>Figura9-2:</b> Líneas de código para simulación de la RNA.....	41
<b>Figura10-2:</b> Interfaz gráfica de la RNA.....	42
<b>Figura 1-3:</b> Simulación primera prueba protocolo IP.....	46
<b>Figura 2-3:</b> Simulación primera prueba protocolo IPv6.....	48
<b>Figura 3-3:</b> Simulación primera prueba protocolo ARP.....	50
<b>Figura 4-3:</b> Simulación primera prueba protocolo HTTP.....	52
<b>Figura 5-3:</b> Simulación primera prueba protocolo HTTPS.....	54
<b>Figura 6-3:</b> Simulación primera prueba protocolo ISCSI-TARGET.....	56
<b>Figura7-3:</b> Simulación de la segunda prueba protocolo IP.....	58
<b>Figura8-3:</b> Simulación de la segunda prueba protocolo IPv6.....	60
<b>Figura9-3:</b> Simulación de la segunda prueba protocolo ARP.....	62
<b>Figura10-3:</b> Simulación de la segunda prueba protocolo HTTP.....	64
<b>Figura11-3:</b> Simulación de la segunda prueba protocolo HTTPS.....	66

<b>Figura12-3:</b> Simulación de la segunda prueba protocolo ISCSI-TARGET.....	<b>68</b>
<b>Figura 13-3:</b> Simulación de la tercera prueba protocolo IP.....	<b>70</b>
<b>Figura 14-3:</b> Simulación de la tercera prueba protocolo IPv6.....	<b>72</b>
<b>Figura 15-3:</b> Simulación de la tercera prueba protocolo ARP.....	<b>74</b>
<b>Figura 16-3:</b> Simulación de la tercera prueba protocolo HTTP.....	<b>76</b>
<b>Figura 17-3:</b> Simulación de la tercera prueba protocolo HTTPS.....	<b>78</b>
<b>Figura 18-3:</b> Simulación de la tercera prueba protocolo ISCSI-TARGET.....	<b>80</b>

## ÍNDICE DE GRÁFICOS

<b>Gráfico1-1:</b> Clasificación de las conexiones de las Redes Neuronales Artificiales Multicapa.....	<b>16</b>
<b>Gráfico2-1:</b> Clasificación del aprendizaje supervisado de las RNA's.....	<b>18</b>
<b>Gráfico1-2:</b> Proceso para el desarrollo del proyecto.....	<b>23</b>
<b>Gráfico2-2:</b> Proceso para análisis de los datos.....	<b>24</b>
<b>Gráfico3-2:</b> Resumen del tráfico de la semana del 16-20 de octubre del 2017.....	<b>27</b>
<b>Gráfico4-2:</b> Resumen del tráfico de la semana del 23-27 de octubre del 2017.....	<b>28</b>
<b>Gráfico5-2:</b> Resumen del tráfico de la semana del 06-10 de octubre del 2017.....	<b>28</b>
<b>Gráfico6-2:</b> Resumen del tráfico de la semana del 06-10 de octubre del 2017.....	<b>29</b>
<b>Gráfico7-2:</b> Resumen del tráfico total capturado del 16 de octubre al 13 de noviembre del 2017.....	<b>29</b>
<b>Gráfico8-2:</b> Tráfico de los protocolos de capa de red de la semana del 16-20 de octubre.....	<b>34</b>
<b>Gráfico9-2:</b> Tráfico de los protocolos de capa de aplicación de la semana del 16-20 de octubre 2017.....	<b>35</b>
<b>Gráfico10-2:</b> Tráfico clasificado por protocolos de capa de red de la semana del 23-27 de octubre del 2017.....	<b>35</b>
<b>Gráfico11-2:</b> Tráfico de los protocolos de capa de aplicación de la semana del 23-27 de octubre 2017.....	<b>36</b>
<b>Gráfico12-2:</b> Tráfico clasificado por protocolos de capa de red de la semana del 06-10 de noviembre del 2017.....	<b>36</b>
<b>Gráfico13-2:</b> Tráfico clasificado por protocolos de capa de aplicación de la semana del 06-10 de noviembre del 2017.....	<b>37</b>
<b>Gráfico14-2:</b> Tráfico clasificado por protocolos de capa de red de la semana del 13-17 de noviembre del 2017.....	<b>37</b>
<b>Gráfico15-2:</b> Tráfico de los protocolos de capa de aplicación de la semana del 13-17 de noviembre 2017.....	<b>38</b>
<b>Gráfico16-2:</b> Diagrama de flujo para la elaboración de la RNA.....	<b>40</b>
<b>Gráfico17-2:</b> Matriz de entradas que ingresa en la RNA.....	<b>41</b>
<b>Gráfico1-3:</b> Comparación de error en protocolos IP, IPV6, ARP.....	<b>83</b>
<b>Gráfico2-3:</b> Comparación de error en protocolo HTTP, HTTPS, ISCSI-TARGET.....	<b>83</b>

## **ÍNDICE DE ANEXOS**

**ANEXO A: TABLAS DE CAPTURAS**

**ANEXO B: CÓDIGO COMPLETO DE LA RED NEURONAL**

## **RESUMEN**

Se desarrolló un algoritmo para predicción de tráfico de red basado en redes neuronales artificiales, en redes corporativas. Para obtener los datos del tráfico generado por la red de la Facultad de Informática y Electrónica (FIE), se instaló un servidor de recolección de datos con Linux, empleando TCPdump y un script para capturar automáticamente los datos durante monitoreos de 4 semanas entre las de 8am – 8pm, con intervalos de 5 minutos, los datos del tráfico recolectado contienen la información de protocolos seleccionados por ocupar mayor espacio en cada paquete: Protocolo de Internet (IP), Protocolo de Internet Versión 6 (IPV6), Protocolo de Resolución de Direcciones (ARP), Protocolo de Transferencia de Hipertexto (HTTP), Protocolo Seguro de Transferencia de Hipertexto (HTTPS) y ISCSI-TARGET. Se analizó el tamaño de cada uno de los protocolos que intervinieron en cada captura a través del programa SteelCentral Packet Analyzer, para el almacenamiento de los datos se utilizó Excel para la creación de una base de datos y posteriormente exportar los datos a Matlab donde se realizó la Red Neuronal Artificial con un algoritmo backpropagation para la predicción de datos. Se tomó una muestra de 2880 datos recolectados, analizados, ingresados y procesados en el algoritmo, el cual consta de 300 entradas, 1 capa oculta que consta de 17 neuronas y una única salida, todos los datos anteriormente mencionados se utilizaron para el entrenamiento de la red neuronal artificial. Se concluye que la Red Neuronal Artificial con un algoritmo backpropagation, luego de las pruebas con diferentes números de neuronas, obtuvo para cada uno de los protocolos los siguientes errores: protocolo IP el 9.50%, IPv6 el 15.38%, ARP el 10.24%, HTTP el 15.89%, HTTPS 12.12% y ISCSI- TARGET él 6.22%. Se recomienda para el pronóstico de tráfico de redes corporativas realizar una red neuronal artificial cuyas características sean mayores a 300 neuronas de entrada, una capa oculta con 17 o más neuronas intermedias.

**PALABRAS CLAVE:** <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <REDES DE COMPUTADORAS>, <ANÁLISIS DE TRAFICO>, <REDES NEURONALES ARTIFICIALES>, <RED PERCEPTRON MULTICAPA>, <BACKPROPAGATION>

## **ABSTRACT**

An algorithm for prediction of network traffic based on artificial neural networks in corporate networks was developed. In order to get the traffic data generated by the network of Computer Science and Electronics School (FIE), a data collection server with Linux was installed, using TCPdump and a script to capture automatically data during 4 weeks between 8am - 8pm, with intervals of 5 minutes. Traffic data collected contain the information of selected protocols due to they occupy more space in each package. Internet Protocol (IP), Internet Protocol Version 6 (IPV6), Address Resolution Protocol (ARP), Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), and ISCSI-TARGET. The size of each protocol involved in every scan through the Program SteelCentral Packet Analyzer for the storage of the data was analyzed. For the data storage Excel was used for the creation of a database and then export the data to Matlab where the Artificial Neural Network with a backpropagation algorithm for the prediction of data was performed. A sample of 2880 data was collected, analyzed, entered, and processed in the algorithm which consists of 300 entries, 1 hidden layer that consists of 17 neurons, and a single output. All data mentioned above were used for training the artificial neural network. It is concluded that the Artificial Neural Network with a backpropagation algorithm after the tests with different numbers of neurons obtained for each of the protocols the following errors, IP protocol 9.50%, IPv6 15.38%, ARP 10.24%, HTTP 15.89%, HTTPS 12.12%, and ISCSI-TARGET 6.22%. For the traffic prediction of corporate networks, it is recommended to make an artificial neural network whose characteristics are greater than 300 input neurons, a hidden layer with 17 or more intermediate neurons.

**KEYWORDS:** <TECHNOLOGY AND ENGINEERING SCIENCES>, <COMPUTER NETWORKS>, <TRAFFIC ANALYSIS>, <NETWORK LAYER PROTOCOLS>, <APPLICATION LAYER PROTOCOLS>, <ARTIFICIAL NEURAL NETWORKS>, <BACKPROPAGATION ALGORITHM>.

## INTRODUCCIÓN

Las redes neuronales artificiales, comúnmente denominado "redes neuronales", tiene sus comienzos cuando el hombre se cuestiona porqué el cerebro humano calcula de una manera totalmente diferente a la computadora convencional. El cerebro es una computadora altamente compleja, no lineal y paralelo (sistema de procesamiento de información). Este tiene la capacidad de organizar sus componentes estructurales, conocidos como neuronas, para realizar ciertos procesos (por ejemplo, reconocimiento de patrones, percepción y control de motores) muchas veces más rápido que la computadora más rápida existente en la actualidad(Haykin, 2001).

Las aplicaciones de las redes neuronales artificiales están presentes en una gran variedad de campos como son:

- espacio aéreo
- automotriz
- bancos
- electrónica
- financieros
- manufactura
- medicina
- telecomunicaciones
- transportes
- robótica
- seguridad
- voz entre otros

Debido al potencial que ofrece esta tecnología, el interés en esta nueva forma de realizar cálculos tras décadas de olvido se debe al extraordinario avance y éxito tanto en el aspecto teórico como de aplicación que se está, ha obtenido estos últimos años.(Basogain Olabe, 2005)

En este proyecto de investigación se realizó un análisis de las distintas redes neuronales artificiales y sus respectivos tipos de aprendizajes, utilizando la herramienta Matlab, de modo que se pueda determinar cuál es el algoritmo más eficiente y confiable para la predicción de los datos de la red. En el capítulo 1 se encuentra todo el marco teórico referente a las redes neuronales artificiales: características, componentes, clasificación y aplicaciones. En el capítulo 2 se detalla paso a paso la elaboración del algoritmo basado en los datos teóricos y todo el proceso de la captura de tráfico de la red de la FIE, finalmente en el capítulo 3 se presenta los resultados de las pruebas realizadas en la red de la FIE.

## **ANTECEDENTES**

### **REDES NEURONALES ARTIFICIALES**

Las redes neuronales artificiales (RNA) son una simulación abstracta de un sistema nervioso real que consiste en un conjunto de unidades neuronales conectadas entre sí a través de conexiones axiales que son muy similares a las dendritas y los axones en los sistemas nerviosos biológicos. (Muñoz, Castillo, & Melin, 2009)

Entre las aplicaciones más representativas desarrolladas con redes neuronales artificiales son:

- Detección de intrusos basado en las redes neuronales
- Detección de spam basado en redes neuronales
- Redes Neuronales Artificiales aplicadas al Análisis de Datos
- La aplicación de Redes Neuronales Artificiales (RNA) a la recuperación de la información

### **DETECCIÓN DE INTRUSOS BASADO EN LA REDES NEURONALES**

La aplicación de técnicas basadas en Inteligencia Artificial para la detección de intrusos (IDS), fundamentalmente las redes neuronales artificiales (ANN), están demostrando ser un enfoque muy adecuado para resolver muchos de los problemas abiertos en esta área. Sin embargo, el gran volumen de información que se requiere cada día para entrenar estos sistemas, junto con la necesidad exponencial de tiempo que requieren para asimilarlos, dificulta enormemente su puesta en marcha en escenarios reales. Se propone un método basado en la aplicación de una técnica para la reducción de características, denominada Análisis de Componentes Principales (PCA).

El PCA permite obtener un modelo para la reducción del tamaño de los vectores de entrada a la ANN, asegurando que la pérdida de información sea mínima y, en consecuencia, disminuyendo la complejidad del clasificador neuronal y manteniendo estables los tiempos de entrenamiento. Para validar la propuesta se ha diseñado un escenario de prueba mediante un IDS basado en ANN. Los resultados obtenidos a partir de las pruebas realizadas demuestran la validez de la propuesta y acreditan las líneas futuras de trabajo. (Fonseca, Pérez, Fernández, Mora Gimeno, & Martínez Abarca, 2008)

## DETECCIÓN DE SPAM BASADO EN REDES NEURONALES

El correo electrónico, es sin duda un medio que nos permite comunicar rápidamente ofreciéndonos reducción de tiempo y costo Sin embargo muchas personas aprovechan esto para utilizarlo de forma no legítima con fines publicitarios, ocasionando una serie de problemas a nivel personal como empresarial.

Como contramedida a esta acción se necesitan herramientas capaces de reducir el spam. De esta manera es muy importante la elaboración de anti-spams, ya que es la forma más viable de acabar con el spam y ofrecer a los usuarios seguridad y tranquilidad en los correos electrónicos, y por otra parte reducir los costos para las empresas ISP's y controlar la saturación de servidores de correo electrónico.

El desarrollo de una herramienta informática capaz de aminorar con los problemas que causa el spam, no es solamente capaz de ahorrar mucho dinero en aquellas empresas que suelen estar perjudicadas con el spam, sino también es capaz de permitir una mejor utilización y minimizar los dolores de cabeza a cualquier usuario del correo electrónico.(Galán Asencio & Martínez Bowen, 2010)

## IMPORTANCIA DEL MONITOREO DE TRÁFICO

Es fundamental tener en mente que la función de monitoreo de red debe ser una labor continua. La infraestructura de red es un organismo binario vivo, el cual requiere de una permanente supervisión de todos sus componentes, a fin de conocer oportunamente situaciones críticas como son las interrupciones de servicios, ataques a dispositivos, tráfico anómalo o comportamientos dentro de la red que requieren de la intervención del encargado para evitar colapsos o saturaciones que puedan poner en riesgo la continuidad de la operación de la infraestructura de TI.

Asegurarnos de buscar una solución que sea fácil de poner en operación, que le brinde acceso a conocer el estado de su infraestructura de red desde cualquier lugar, momento y dispositivo, con la cual supervise la disponibilidad y el desempeño de todos los componentes de su infraestructura de red, a un costo accesible, con soporte en su propio idioma. Lo mejor es ofrecer un esquema de funcionalidad “todo incluido” en su licenciamiento que permita evaluar a fondo y tomar la mejor decisión de compra con la información adecuada(Xujian, Hao, Dewen, & Haitao, 2013).

## **FORECAST NETWORK**

El forecast se basa en la predicción del uso de datos históricos para determinar la dirección de las tendencias futuras. Las empresas utilizan previsiones para determinar cómo asignar su presupuesto o planificar los gastos anticipados para un periodo de tiempo próximo. Se refiere a métodos estadísticos formales que emplean series temporales, datos transversales o longitudinales.

"El pronóstico de tráfico se está convirtiendo en el factor más importante en la prestación de la calidad de la experiencia para el usuario final", dice Stefano Savioli, jefe de optimización de la red y la garantía de Nokia Siemens Networks. Los modelos no se limitan a mirar a la cantidad de tráfico lleva una red. También examinan si el tráfico se enruta de manera eficiente. Muy poca capacidad y se le cae llamadas y alienar a los clientes(Joshi & Hadi, 2015).

## **FORMULACIÓN DEL PROBLEMA**

¿Cuál será el impacto en la gestión de la información de la comunicación de Redes Corporativas de la implementación de un Algoritmo de Redes Neuronales Artificiales aplicado a la Predicción de Tráfico?

## **SISTEMATIZACIÓN DEL PROBLEMA**

- ¿Cuál es el número adecuado de capas y de neuronas por capa para obtener una predicción de adecuada?
- ¿Qué tipo de red neuronal artificial y algoritmos de predicción se utiliza para la plataforma?
- ¿Cuál es el grado de precisión de la predicción?
- ¿Cuál es el tiempo de predicción que nos permitirá tener el algoritmo?

## **JUSTIFICACIÓN TEÓRICA**

Las redes neuronales artificiales (RNA) tienen la capacidad de aprender a realizar tareas basadas en un entrenamiento inicial (Aprendizaje adaptativo). De esta forma, la máquina puede aprender a llevar a cabo ciertas tareas mediante el entrenamiento con ejemplos ilustrativos, por lo que no es necesario elaborar modelos a priori. Las redes neuronales permiten organizar por sí mismas lo aprendido. Mientras que el aprendizaje es la modificación de cada elemento procesal, la auto

organización consiste en la modificación de la red neuronal completa para llevar a cabo un objetivo específico.

Además, las redes neuronales tienen la capacidad de ser tolerantes a fallos. Fueron los primeros métodos computacionales con esta capacidad inherente. De este modo, si se produce un fallo en un número no muy grande de neuronas, aunque el comportamiento del sistema se ve influenciado, no sufre una caída repentina. Una de las prioridades principales de las redes neuronales es que son capaces de operar en tiempo real. Las redes neuronales trabajan mediante conexiones en paralelo, lo que permite grandes velocidades de transmisión y respuesta casi al instante. Hay que destacar como ventaja la facilidad de inserción en la tecnología existente.

El crecimiento exponencial del tráfico en las redes medido por el forecast de CISCO dice que para el 2020 habrá 26.300 millones de dispositivos conectados a escala global y 4.100 millones de internautas, el 71 por ciento del tráfico IP procederá de dispositivos distintos al PC y el vídeo representará el 82 por ciento de dicho tráfico.

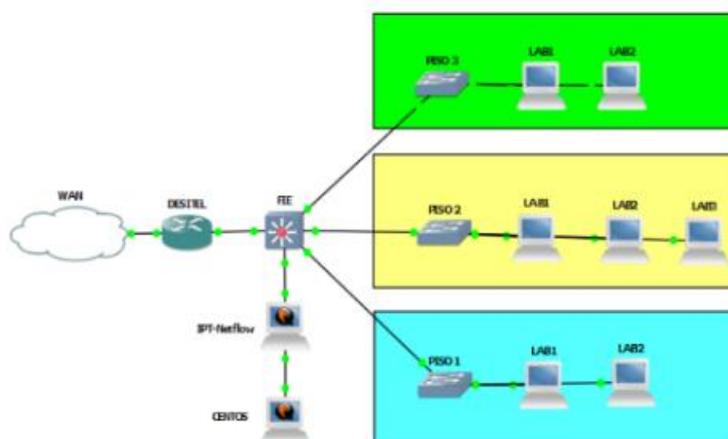
Se optó utilizar Linux el cual es un sistema operativo libre por lo que no necesitamos una licencia para instalarlo y utilizarlo en el servidor recolector de datos, además de ello es un sistema multitarea, multiusuario, compatible con UNIX, y nos proporciona una interfaz de comandos y una interfaz gráfica, que lo convierte en una de las mejores opciones para la expectativa perspectiva del futuro, este sistema consta con herramienta "tcpdump" la cual permite la captura automática de los datos y se encuentra en el Debían Linux Kernel 2.6.10.

Para el monitoreo del tráfico de la red; supervisara la herramienta tcpdump, que permite analizar el tráfico que circula por la red, su principal característica es la captura y muestra en tiempo real de los paquetes transmitidos y recibidos por la red a la cual el ordenador está conectado. El edificio principal de la FIE necesita en su infraestructura de red una plataforma de monitoreo la cual tendrá grandes ventajas como la planificación de red, planificación de hardware, evitar colapsos y saturaciones entre otras.

Para implementar el algoritmo de redes neuronales artificiales ya que van a costar de varias capas utilizaremos la herramienta Matlab ya que este es un lenguaje de programación cuya filosofía hace hincapié en una sintaxis muy limpia y un código legible. Es una buena alternativa para optar por este lenguaje en el mundo de la programación, puesto que es un lenguaje muy sencillo y fácil. Su sintaxis es fácil de entender puesto que es cercana al lenguaje natural.

## JUSTIFICACIÓN APLICATIVA

Para realizar la implementación de un algoritmo de redes neuronales artificiales la mejor opción es la herramienta Matlab como nos recomiendan los autores Torres, Hernández & Pedraza en sus trabajos sobre redes neuronales entre las ventajas que describe son que la herramienta es libre, lenguaje muy sencillo y fácil de entender para el usuario entre otras, Además se tiene información al respecto para realizar el algoritmo como en tesis de maestría, paper académicos entre otras fuentes bibliográficas. Tenemos toda la tecnología en mano la cual nos permitirá la realización del algoritmo para la predicción de tráfico dentro de la red de la FIE ya que gracias a su característica primordial que es la de trabajar con tareas de forma paralela y simultánea dándonos tiempos de respuesta casi inmediatos, también se analizó que es importante planificar los requisitos de hardware y la topología de red, para determinar los tipos de equipo que se necesitara y la distribución de dichos equipos en el sitio, ya que al diseñar la red, se debe decidir qué tipo de red se adapta mejor conforme a la necesidad que se tenga y esto incluye algunas decisiones de planificación relacionadas con el hardware como la necesidad de puentes o enrutadores que extiendan el medio o conexión la red local a redes externas, y si existe un requerimiento de interfaces adquiridas por separado además de sus interfaces integradas, todo esto nos permitirá que podamos tener ya una base inicial sólida para poder diseñar una red.



**Figura 1 Infraestructura de la Red a monitorear.**  
Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017

## **OBJETIVOS**

### **OBJETIVOS GENERALES**

- Desarrollar un algoritmo para la predicción de tráfico de red basado en las redes neuronales artificiales, en Redes Corporativas.

### **OBJETIVOS ESPECÍFICOS**

- Analizar las distintas redes neuronales Artificiales y analizar cada uno de los algoritmos de la red neuronal artificial que seleccionamos.
- Diseñar un algoritmo de predicción de tráfico para el edificio principal de la FIE.
- Implementar el sistema de monitoreo y predicción de tráfico.
- Verificar el correcto funcionamiento del algoritmo para la predicción de los datos y evaluar los resultados.

# CAPÍTULO I

## 1. MARCO TEÓRICO REFENCIAL

### 1.1. Redes neuronales artificiales

#### *1.1.1. Definición*

La Red Neuronal Artificial también llamadas redes neuronales o neuroredes; cuyas siglas son RNA's se las puede definir como todas aquellas redes en las que existen elementos procesadores de información de cuyas interacciones locales depende el comportamiento del conjunto del sistema(López & Fernandez, 2008). Las RNA's funcionan a diferencia de las computadoras comunes como un sistema de procesamiento que realiza múltiples tareas de forma paralela y simultánea.

Las características principales de las Redes Neuronales Artificiales son:

- Tienen una inclinación natural a adquirir el conocimiento a través de la experiencia, el cual es almacenado, al igual que en el cerebro, en el peso relativo de las conexiones interneuronales.(Haykin, 2001)
- Tienen una alta capacidad y gran adaptabilidad, son capaces de cambiar dinámicamente junto con el medio.
- Poseen un alto nivel de tolerancia a fallas, es decir, puede sufrir un daño considerable y continuar teniendo un buen comportamiento, al igual como ocurre en los sistemas biológicos.
- Tienen un comportamiento altamente no-lineal, lo que les permite procesar información procedente de otros fenómenos no-lineales.(Izaurieta & Saavedra, 1999)

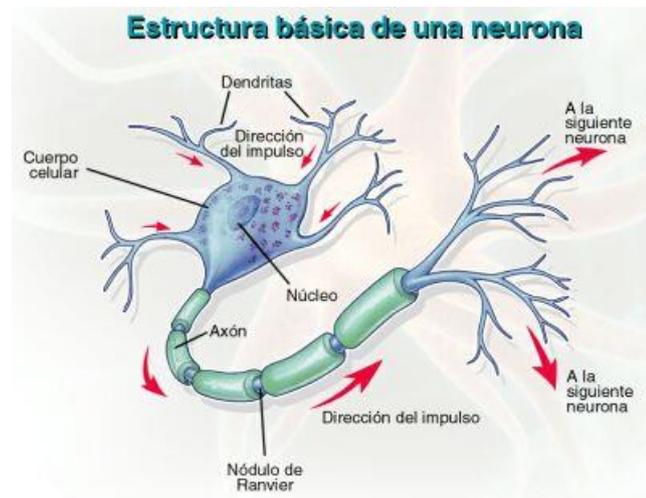
#### *1.1.2. Componentes de las redes neuronales artificiales*

Antes de conocer cada una de las partes de una Red Neuronal Artificial debemos conocer el sistema biológico del cual precede que es el cerebro humano, el cual consta de aproximadamente de 100000000000 (100 mil millones) neuronas interconectadas y las cuales realizan alrededor de 10000 (10 mil) conexiones por cada una de ellas. Las neuronas biológicas están constituidas por

3 partes primordiales las cuales son: las dendritas, el cuerpo de la célula o soma y el axón, como se puede observar en la figura 1-1. A continuación, se detalla cada una de ellas:

- ✓ Dendritas: Son fibras nerviosas encargadas de la recepción de las señales eléctricas, estas son cargadas al cuerpo de la neurona.
- ✓ Cuerpo celular o soma: Realiza la suma de las señales de entrada.
- ✓ Axón: Es una fibra larga que asume la función de llevar la señal desde el cuerpo de la célula hacia otras neuronas.

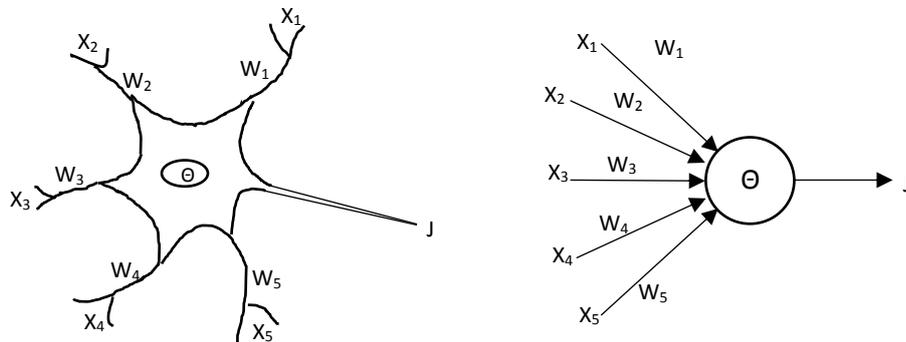
El proceso de conexión entre el axón de una neurona y las dendritas de otra se conoce como sinapsis, dependiendo de la extensión de la sinapsis se estabiliza la función de una Red Neuronal.(F. Tanco, C. Verrastro, D. Grinberg, 2004)



**Figura1-1: Estructura básica de una neurona biológica.**

Fuente: <https://residenciasalcalamahora.files.wordpress.com/2013/03/neurona.jpg>

En relación a lo antes mencionado del modelo biológico de una Red Neuronal, este se asemeja mucho al modelo de una Red Neuronal Artificial; como se puede observar en la figura 2-1 en la cual se denota los componentes de una Red Neuronal Artificial.

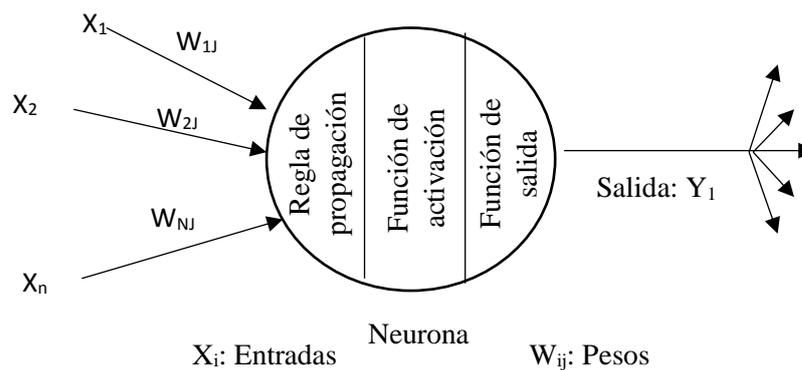


**Figura2-1: Similitud entre Neurona Biológica y una Neurona Artificial**

Fuente: Tanco, F. (2003).

De la observación realizada al proceso biológico de una neurona; esta tiene muchas similitudes con el sistema artificial, como se observa en la figura 3-1:

- Las entradas  $X_i$  representan las señales que provienen de otras neuronas y que son capturadas por las dendritas.
- Los pesos  $W_i$  son la intensidad de la sinapsis que conecta dos neuronas; tanto  $X_i$  como  $W_i$  son valores reales.
- Las salidas  $Y_i$  representan al valor resultante de la función de activación.
- $\theta$  es la función umbral que la neurona debe sobrepasar para activarse (función de activación); este proceso ocurre biológicamente en el cuerpo de la célula. (F. Tanco, C. Verrastro, D. Grinberg, 2004)

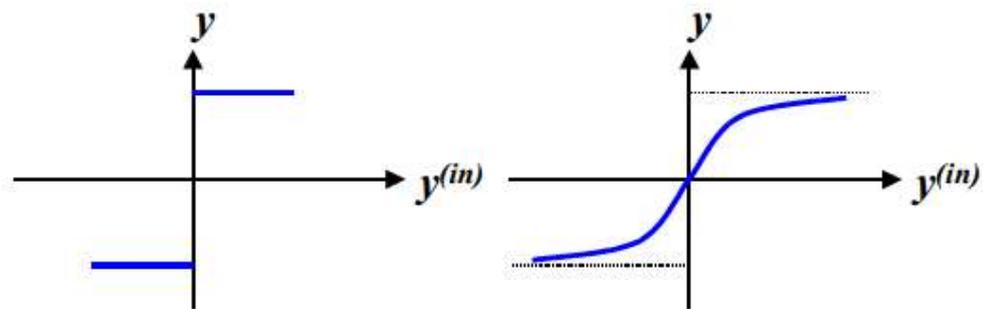


**Figura3-1: Componentes una Red Neuronal Artificial.**

Fuente: <https://gjorge.files.wordpress.com/2007/08/nueva-imagen.png>

### 1.1.3. Funciones de activación.

La función de activación establece el efecto de la entrada total  $\mu(t)$  en la activación de la unidad  $k$  (denotación función de activación). Cuando las funciones de activación de una neurona son lineales decimos que la neurona es lineal y se representa con un cuadrado, caso contrario es una neurona no-lineal y se representan con un círculo. (Izaurieta & Saavedra, 1999) Algunas de las funciones típicas de activación son: sigmoidea, tangente hiperbólica, escalón entre otras. Entre las funciones de activación no-lineales existen de dos tipos: escalón y sigmoidea tal es el caso de la tangente hiperbólica como se observar en la figura 4-1. (Ponce, 2010)



**Figura4-1: Función escalón y sigmoidea.**

Fuente: Izaurieta, F., & Saavedra, C. (2000).

Una condición para escoger estas funciones de activación, es obtener el cálculo de su derivada a partir de la propia función, para ejemplificar esto se escoge la función sigmoide de la ecuación 1-1 y se deriva para llegar a obtener la ecuación 1-2

$$f(x) = (1 + e^{-x})^{-1} \quad (1-1)$$

$$\frac{df(x)}{dx} = (1 + e^{-x})^{-2} \cdot e^{-x} \quad (1-2)$$

Se puede apreciar entonces que si utilizamos las ecuaciones 1-3 1-4 y las sustituimos en 1-2, se obtendrá la ecuación 1-5.

$$f(x)^2 = (1 + e^{-x})^{-2} \quad (1-3)$$

$$\frac{df(x)}{dx} = f(x)^2 \cdot e^{-x}$$

Despejando  $e^{-x}$  de la ecuación 1-1.

$$e^{-x} = \frac{1}{f(x)} - 1 \quad (1-4)$$

Entonces:

$$\frac{df(x)}{dx} = f(x)^2 \left[ \frac{1}{f(x)} - 1 \right]$$

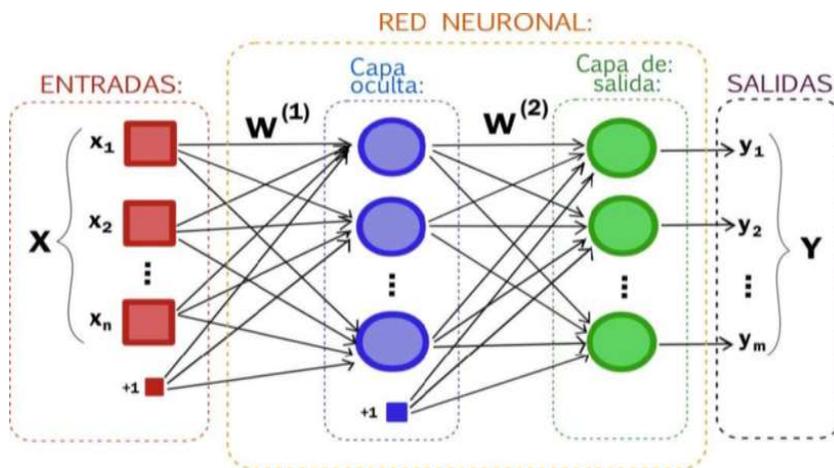
$$\frac{df(x)}{dx} = f(x)[1 - f(x)] \quad (1-5)$$

Como se puede observar en la ecuación 1-5, la derivada de una función sigmoidea se puede calcular a partir del valor de la propia función sigmoide, esto muestra una gran ventaja, y explica su uso en las RNA. Con todo lo anteriormente realizado se concluye que la función sigmoide es una de las mejores opciones para tomarle en cuenta como función de activación, pero otra opción es la función identidad. (La, Del, & Pleurotus, 2017)

#### 1.1.4. Proceso básico de la obtención de los valores para la salida de las RNA's

Como se observa en la figura 5-1, el proceso que realiza una RNA para obtener valores a la salida "Y" se basa en:

- Las "n" neuronas de "X" están enviando señales de entrada. Los valores numéricos que se asigna a "W" (pesos de las conexiones) son aleatorios, lo que realiza cada peso sináptico es multiplicar su entrada correspondiente y determinar el valor de cada entrada, haciendo analogía a la función del soma en la neurona biológica, así la Red Neuronal aprende a realizar una tarea específica. (Izaurieta & Saavedra, 1999)



**Figura5-1: Modelo no-lineal de una red neuronal.**

Fuente: [http://ceres.ugr.es/~alumnos/esclas/imagenes/index\\_img\\_4.jpg](http://ceres.ugr.es/~alumnos/esclas/imagenes/index_img_4.jpg)

En la ecuación 1-6, se muestra la obtención de la salida "Y".

$$Y_j^{(in)} = \sum_{i=1}^n W_{ji} X_i \quad (1-6)$$

Donde:

$X_i$ : Señal de salida de otros nodos o entradas externas.

$W_{ji}$ : Pesos de las conexiones.

$i$ : Entrada

Los pulsos que se obtiene a la entrada de la neurona se le asignan un peso específico, recuerde que el peso de las dendritas que ingresan debe superar el umbral establecido.

### ***1.1.5. Ventajas y desventajas de las redes neuronales artificiales***

Las Redes Neuronales Artificiales se han usado durante varios años en muchos de los campos de la ciencia como son la matemática, física teórica, psicología, ingeniería, medicina entre otras, a continuación presentaremos algunas de las ventajas y desventajas de trabajar con este sistema.(Serrano J, Soria, & Martín, 2009)

La principal ventaja que resalta en la red neuronal artificial es su sistema paralelo como por ejemplo una computadora común y corriente tardaría varias horas en realizar una simple actividad, está la realiza en unos pocos minutos; pero a pesar de ello existen otras ventajas que nos permiten determinar porque es realmente buenos los sistemas de redes neuronales artificiales:

- **Son sistemas distribuidos no lineales:** una neurona es una unidad no lineal y por consecuente el conjunto de ellas va a dar como resultado un sistema no lineal lo que favorece en la realización de procesos ya que no seguirán una secuencia, lo que sería imposible de realizar en un sistema lineal.
- **Son sistemas tolerantes a fallos:** Estos sistemas están compuestos por unidades independientes, pero a su vez complementarias, si llegara a fallar una de estas unidades no alteraría la respuesta final de un proceso, todo lo opuesto a los procesos que se tiene en las computadoras actuales las cuales son sistemas secuenciales, los cuales si llega a fallar algo dentro del proceso tendremos una respuesta totalmente opuesta a la que se desea.
- **Adaptabilidad:** Estos sistemas tienen la facilidad de ser modificados en sus parámetros ya sean a sus entradas o en otra parte de ellos, cabe recalcar que la capacidad de adaptabilidad tiene sus límites ya que si la excedemos esto provocará tener sistemas inestables que se dañaran con pequeñas perturbaciones. Esto se conoce como el dilema plasticidad-estabilidad, sabiendo que la plasticidad es la respuesta correcta a un estímulo nunca antes recibido del entorno.
- **Establece relaciones no lineales entre datos:** Los datos no necesariamente deben cumplir las condiciones de linealidad, gaussianidad y estacionaridad; además de ello las RNA's son capaces de relacionar dos conjuntos de datos.(Serrano J et al., 2009)

A continuación, se presenta las desventajas de las Redes Neuronales Artificiales las cuales son dominables se las presenta a continuación:

- **Complejidad de aprendizaje para grandes tareas:** Dependiendo de las tareas que realice la Red Neuronal, depende el aprendizaje de cada una de ellas; por ello el trabajo más complejo es enseñarle a la Red a realizar cada una de ellas.(Serrano J et al., 2009)
- **Tiempo de aprendizaje elevado:** Como sabemos cada uno de los patrones que reconoce la Red Neuronal se le debe colocar un peso determinado que representa lo que va a aprender la RNA, cuando existen patrones similares, es un arduo trabajo el de reconocer, separar e identificar lo que equivale a consumir mucho más tiempo.
- **No permite interpretaciones:** Las aplicaciones y los programadores de la RNA son quienes se encargan de darle una significativa validez a la salida de la Red Neuronal ya que es el único valor que no puede interpretar ella misma.
- **Elevada cantidad de datos para el entrenamiento:** Cuando la RNA tiene que aprender para un programa determinado se somete una preparación específica para él, para ello se debe enseñar una mayor cantidad de datos lo que implica mayor consumo de CPU; y es así como se prepara para identificar de forma correcta los patrones.(T Marino, victor Tomas ; Hernandes Hernandes, 2011)

## 1.2. Clasificación de las redes neuronales artificiales

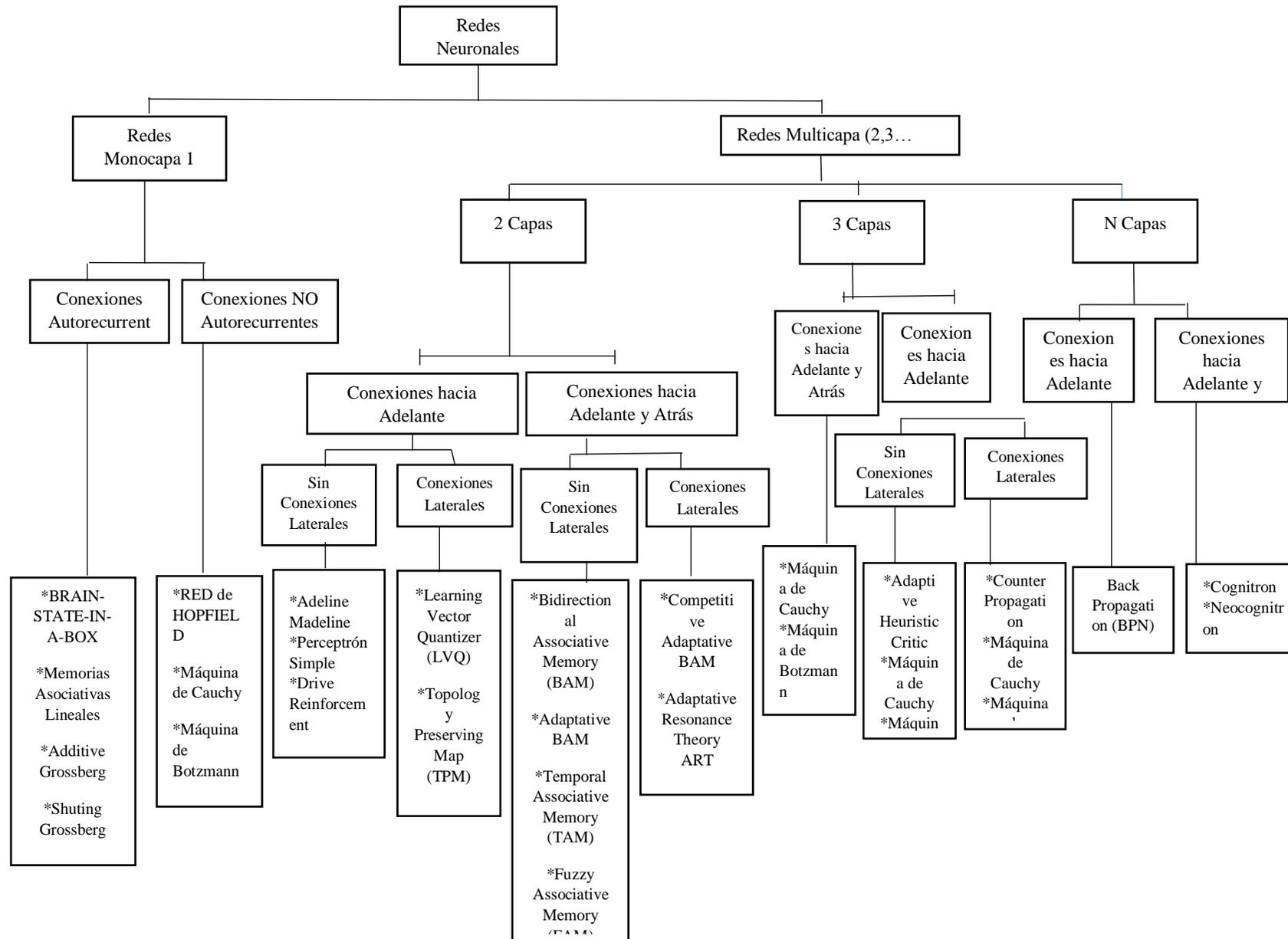
En este apartado, veremos dos clasificaciones de las redes neuronales en función de sus características más notables:

- Redes neuronales artificiales por topología.
- Redes neuronales artificiales según el tipo de aprendizaje.

### 1.2.1. *Redes neuronales artificiales por topología*

La arquitectura de una red neuronal consiste en organizar y disponer de las neuronas dentro de una red, estas se pueden agrupar formando capas, a su vez tienen diferentes características y se estructuran para formar la red.

A continuación, en la figura 6-1 se visualiza claramente este tipo de redes neuronales

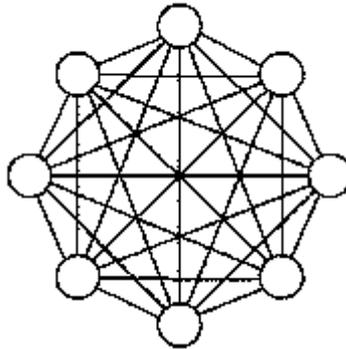


**Figura6-1: Organigrama de las Redes Neuronales Artificial como criterio clasificar la Topología.**

Fuente:<http://www.redes-neuronales.com.es/tutorial-redes-neuronales/Neural-Networks-Java/clasificacion-de-lar->

### 1.2.1.1. Redes neuronales artificiales monocapa

Tiene una sola capa de neuronas que proyecta las entradas a una capa de neuronas de salida, debido que las capas de entrada no realizan ningún cálculo no se las toma en cuenta de ahí su nombre monocapa, cuando existe una conexión entre las mismas neuronas esto se denomina como autorecurrente como se observa en la figura 7-1. Algunas de las aplicaciones de este tipo de red neuronal son las memorias asociativas, Hopfield, Brain-in-state-in-a-box y las maquinas estocásticas de Boltzmann y Cauchy. (Ballesteros, 2005)



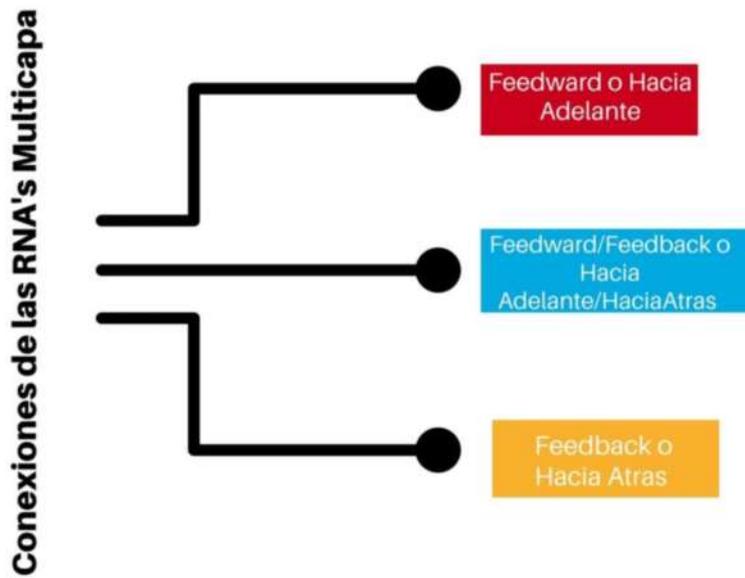
**Figura7-1: Red Neuronal Artificial Monocapa.**

**Fuente:**<http://www.redes-neuronales.com.es/tutorial-redes-neuronales/Neural-Networks/Java/monocapa.jpg>

### 1.2.1.2. Redes neuronales artificiales multicapa

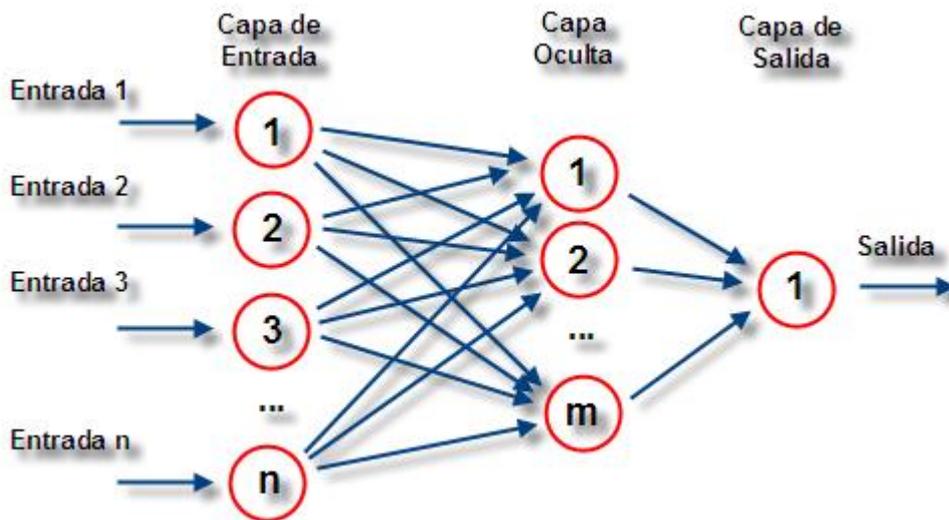
“Es un conjunto de capas intermedias entre la entrada y la salida (capas ocultas), estas redes pueden estar total o parcialmente conectadas” (Martin, Serrano y Soria, 2009, p.17); entre sus características se encuentran que estas pueden estar total o parcialmente conectadas, estas capas están conectadas desde su entrada hasta su salida, debido a esa orden de conexión estos dos tipos de conexiones:

En el siguiente grafico 1-1 se presenta la clasificación de las redes neuronales artificiales según su tipo de conexiones:



**Grafico1-1: Clasificación de las conexiones de las Redes Neuronales Artificiales Multicapa.**  
 Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

En la figura 8-1 se puede observar un ejemplo de red neuronal artificial multicapa con conexiones feedforward o hacia adelante.



**Figura8-1: Red Neuronal Artificial Multicapa.**  
 Fuente: <http://www.esacademic.com/pictures/eswiki/82/RedNeuronalArtificial.png>

### *1.2.2. Redes neuronales artificiales según el tipo de aprendizaje*

La red neuronal debe aprender a calcular la señal de salida correcta para el arreglo o matriz de señales de entrada de un conjunto de datos. Al proceso de aprendizaje se lo designa como: proceso de entrenamiento o acondicionamiento; este procedimiento se ejecutará en un conjunto de datos el cual se lo denomina: conjunto de datos de entrenamiento. (Serrano J et al., 2009)

En el aprendizaje de una red neuronal se realizan cambios en los pesos de las conexiones de la red; en palabras técnicas es el proceso mediante el cual se modifican los pesos de la red de acuerdo a las señales de entrada; estos cambios involucran la destrucción, alteración y creación de conexiones entre las neuronas. (Serrano J et al., 2009)

Se debe de tomar en cuenta que cuando se realiza una nueva conexión implica que el peso debe tener un valor distinto a cero, pero esta se destruye cuando el valor del peso pasa a ser cero. Cuando los pesos de las conexiones sufren alteraciones, la red a realizado su proceso de aprendizaje; matemáticamente la estabilidad de los pesos es  $\frac{dw_{ij}}{dt} = 0$ . (Basogain Olabe, 2005)

Durante la alteración de los pesos se toma como criterio la modificación de los valores para asignar un nuevo valor a las conexiones. Entonces se obtiene dos métodos de aprendizaje:

- Aprendizaje supervisado.
- Aprendizaje no supervisado.

#### *1.2.2.1. Aprendizaje supervisado*

Para realizar el aprendizaje de la red necesita asociarse cada señal de entrada con una señal de salida, las mismas que son aplicadas secuencialmente y de forma cíclica. El proceso de entrenamiento consiste en ingresar una señal de entrada, calcular la señal de salida, comparar la señal de salida deseada con la ya calculada por la red; entonces el error será utilizado para realimentarla y variar los pesos, con un algoritmo, que permita obtener un mínimo error el cual será aceptable y pequeño, finalizando el proceso de entrenamiento. (Basogain Olabe, 2005)

A continuación, se presenta en el gráfico 2-1 en la se observa la clasificación del aprendizaje supervisado.



**Grafico2-1: Clasificación del Aprendizaje Supervisado de las RNA's**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

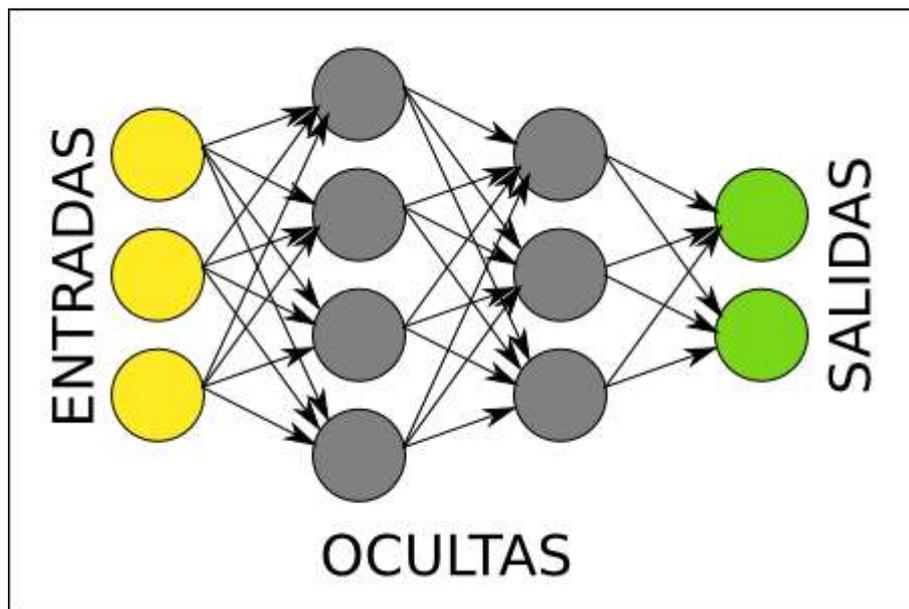
### 1.3. Tipos de redes neuronales artificiales

#### 1.3.1. *Perceptrón multicapa*

Este modelo es una ampliación del Perceptrón Simple a la cual añade una serie de capas que, básicamente, hacen una transformación sobre las variables de entrada, así permiten eludir el problema de las funciones linealmente no independientes en linealmente independientes gracias a la transformación de la capa oculta. Además, el Perceptrón multicapa admite valores reales. Además de ser un modelador de funciones universal.

##### 1.3.1.1. *Arquitectura del Perceptrón multicapa*

El Perceptrón multicapa está compuesta de una capa de entrada, una capa de salida y una o más capas ocultas. Dichas capas se unen de forma que, la capa entrada se une con la primera capa oculta y esta con la siguiente y la última capa oculta se une con la capa de salida. Los valores que el Perceptrón multicapa acepta son reales. Como se puede observar en la figura 9-1.



**Figura9-1: Arquitectura de una Red Perceptrón Multicapa**

**Fuente:** <https://rubenlopezg.files.wordpress.com/2014/04/neural-net1.png?w=640>

En la tabla 1-1 se muestra las ventajas y desventajas de 3 redes neuronales artificiales posibles opciones; pero la red Perceptrón Multicapa fue seleccionada ya que se acopla a las características del comportamiento del tráfico.

**Tabla 1-1:** Comparación entre las Redes Neuronales aptas para el trabajo de titulación.

RED NEURONAL	VENTAJAS	DESVENTAJAS
<b>Perceptrón</b>	<ul style="list-style-type: none"> <li>• Su estructura es monocapa unidireccional, con dos o más variables de entrada binaria o real, y una salida binaria.</li> <li>• Los pesos es el límite entre la toma de decisión los se les puede asignar un 0 y 1 dependiendo de sus salidas respectivas</li> </ul>	<ul style="list-style-type: none"> <li>• Sirve como clasificador para problemas con linealidad independiente.</li> </ul>
<b>Adaline</b>	<ul style="list-style-type: none"> <li>• Tiene salidas lineales con valores reales y con facilidad se logra saber el error que existe.</li> <li>• Su salida neta es igual a función de activación que esta a su vez es la suma equilibrada de las entradas.</li> </ul>	<ul style="list-style-type: none"> <li>• Asigna valores a la constante de error, existirá conflictos al encontrar el error con valores demasiados grandes o pequeños.</li> </ul>
<b>Perceptrón Multicapa</b>	<ul style="list-style-type: none"> <li>• Es catalogada como una técnica no paramétrica, evitando así el cumplimiento de algunos supuestos.</li> <li>• su adaptabilidad le permite procesar cualquier tipo de variables.</li> </ul>	<ul style="list-style-type: none"> <li>• El costo computacional; todo el proceso consume un número alto de iteraciones, lo que conllevó a un mayor consumo de tiempo.</li> </ul>

	<ul style="list-style-type: none"> <li>• Funciona como un aproximador universal de funciones.</li> </ul>	
--	--	--

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

#### 1.4. Algoritmos de las redes neuronales artificiales

La red BACKPROPAGATION tiene como característica describir la unión de varios perceptrones, utilizando procesos de Adeline dando solución al entrenamiento sin convergencia ideal, por consecuencia se obtiene la resta entre las salidas imprecisas y las salidas deseadas todo ello multiplicado por la derivada de la función de transferencia sigmoidea en cada nodo.

Para comenzar la retro propagación se obtiene los errores de cada uno de los nodos con la intención de cambiar los pesos, en este instante es cuando se encuentran los errores en las capas ocultas retro propagándose hacia la capa de entrada y cambiando los pesos en cada iteración utilizando como de denota en la ecuación 1-7.

$$\nabla W(t + 1) = W(t) + \alpha \epsilon x \quad (1-7)$$

Cada uno de los errores de los nodos de las capas ocultas se puede calcular como se denota en la ecuación 1-8

$$\epsilon_{no} = (\sum_{na}^n \epsilon w)(1 - Y_{na}) Y_{na} \quad (1-8)$$

Siendo:

$\epsilon_{no}$  : El error obtenido en las salidas de la red.

$\sum_{na}^n \epsilon w$  : La sumatorio de los errores de los nodos que están conectados al nodo de la capa oculta.

$(1 - Y_{na}) Y_{na}$  : La derivada de la función de transferencia que se obtuvo al derivar la salida de cada nodo  $Y_{na} = \frac{1}{1+e^{-NETI}}$

$NETI = \sum_{i=1}^n X_i W_i$  : Es la sumatorio de las entradas de cada nodo con sus pesos respectivos los cuales se encuentran conectados a los nodos de la siguiente capa.

Una de las desventajas principales que presenta la red BACKPROPAGATION es la demora de los ajustes de los pesos; al caer en el error de los mínimos locales por no poder definir los errores de nodos de las capas ocultas. Al asignar valores demasiados altos o bajos provoca una parálisis en el sistema por la lentitud de procesamiento por ello se recomienda un aprendizaje entre un intervalo de valores de -5 a 5. (Proaño Freire, 2012)

### 1.5. Aplicaciones de las redes neuronales artificiales.

Las aplicaciones de las redes neuronales artificiales están presentes en una gran variedad de campos como se observa en la tabla 2-1.

**Tabla2-1: Aplicaciones de las Redes Neuronales Artificiales en distintos campos**

CAMPOS	APLICACIONES
ESPACIO AÉREO	<ul style="list-style-type: none"><li>• Pilotos automáticos de alto desempeño.</li><li>• Simulaciones y Predicciones de trayectoria de vuelo.</li><li>• Sistemas de control de vuelo.</li><li>• Detección de fallas de componentes de la nave.</li></ul>
AUTOMOTRIZ	<ul style="list-style-type: none"><li>• Sistemas automáticos de navegación.</li><li>• Comando por voz.</li></ul>
BANCOS	<ul style="list-style-type: none"><li>• Lectores de documentos.</li><li>• Evaluadores de asignación de crédito.</li><li>• Identificador de firmas.</li></ul>
ELECTRÓNICA	<ul style="list-style-type: none"><li>• Predicción de secuencias de códigos.</li><li>• Control de procesos.</li><li>• Análisis de fallas de circuitos.</li><li>• Visión de máquina.</li><li>• Síntesis de voz.</li><li>• Modelado no lineal.</li></ul>
FINANCIEROS	<ul style="list-style-type: none"><li>• Evaluación de bienes raíces.</li><li>• Consultor de préstamos.</li><li>• Evaluación de bonos corporativos.</li><li>• Análisis del uso de la línea de crédito.</li><li>• Predicción de tipo de cambio.</li></ul>
MANUFACTURA	<ul style="list-style-type: none"><li>• Control de procesos de manufactura.</li><li>• Análisis y diseño de productos.</li><li>• Diagnóstico de máquinas y procesos.</li><li>• Identificación de partes en tiempo real.</li><li>• Sistemas de inspección de calidad.</li><li>• Predicción de fin de proceso.</li><li>• Análisis de mantenimiento de máquinas.</li></ul>
MEDICINA	<ul style="list-style-type: none"><li>• Detección de cáncer mamario o en la piel.</li><li>• Diseño de prótesis.</li><li>• Optimización de tiempos de trasplante.</li><li>• Reducción de gastos en hospitales.</li></ul>
TELECOMUNICACIONES	<ul style="list-style-type: none"><li>• Compresión de datos e imágenes.</li></ul>

	<ul style="list-style-type: none"> <li>• Servicios automáticos de información.</li> <li>• Traducción de lenguaje hablando en tiempo real.</li> </ul>
TRANSPORTES	<ul style="list-style-type: none"> <li>• Sistemas ruteadores.</li> <li>• Diagnóstico de motores.</li> <li>• Tiempos y movimientos.</li> </ul>
ROBÓTICA	<ul style="list-style-type: none"> <li>• Control de trayectorias.</li> <li>• Control de manipuladores.</li> <li>• Sistemas de visión.</li> </ul>
SEGURIDAD	<ul style="list-style-type: none"> <li>• Reconocimiento de rostros.</li> <li>• Identificación.</li> <li>• Acceso de personas.</li> </ul>
VOZ	<ul style="list-style-type: none"> <li>• Reconocimiento de voz.</li> <li>• Comprensión de voz.</li> <li>• Sintetizadores de texto a voz.</li> </ul>
OTROS	<ul style="list-style-type: none"> <li>• Oficinas postales.</li> <li>• Verificación remota.</li> <li>• Predicciones climatológicas.</li> <li>• Filtrado de ruido.</li> <li>• ServoControl.</li> </ul>

**Realizado por:** BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

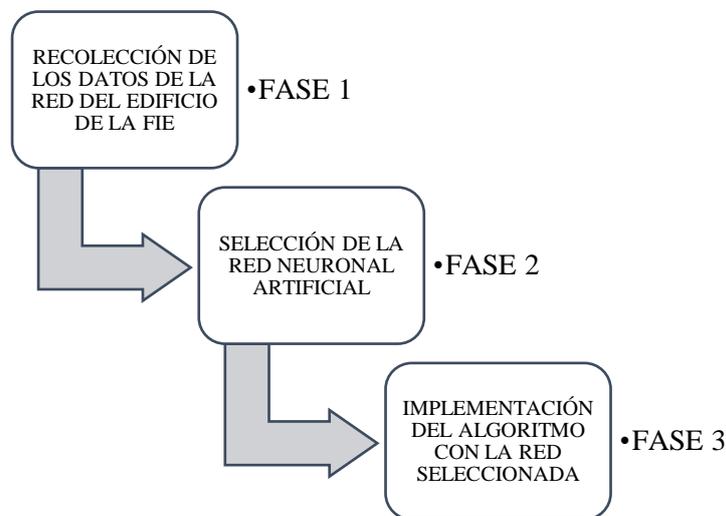
## CAPÍTULO II

### 2. MARCO METODOLÓGICO.

#### 2.1. Desarrollo del trabajo de titulación.

En este capítulo, se presenta la recolección y análisis del tráfico de la Red de la Infraestructura del Edificio de la FIE; en la cual se realizó el estudio de 6 protocolos, de los cuales 3 de ellos pertenecen a la capa de Red y los otros restantes a la capa de aplicación. En el gráfico 1-2, se describe el procedimiento a seguir para el desarrollo de la investigación.

La metodología utilizada para un proyecto es: análisis, diseño e implementación del mismo se ha decidido seguir este concepto, pero en vez de realizar esos pasos específicamente se optó por 3 fases que describa todo el proceso para llegar a nuestro fin.



**Gráfico1-2: Proceso para el desarrollo del proyecto.**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

#### FASE 1- RECOLECCIÓN DE LOS DATOS

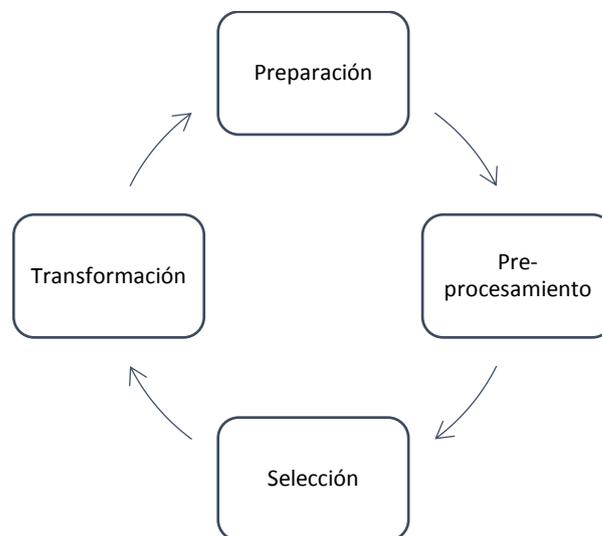
En esta fase, se realiza el análisis exploratorio el cual consiste en los cuatro pasos para la recolección de los datos: preparación, selección, pre-procesamiento y transformación de datos, como se observa en el gráfico 2-2.

## FASE 2-SELECCIÓN DE LA RED NEURONAL ARTIFICIAL

Luego de haber realizado un análisis minucioso en el capítulo anterior de 3 tipos de redes neuronales artificiales, en esta fase se estudió algunos parámetros entre ellos se encuentran: elevada cantidad de datos para el entrenamiento, complejidad de aprendizaje para grandes tareas entre otros.

## FASE 3 – IMPLEMENTACIÓN DEL ALGORITMO CON LA RED SELECCIONADA

Una vez seleccionada la red neuronal artificial se procede a implementar el algoritmo en la herramienta Matlab, para lo cual se define el número correcto de las capas de entrada, intermedias y de salida también el número de neuronas por capa y calcular de un mínimo error.



**Grafico2-2: Proceso para análisis de los datos.**  
Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

## FASE 1

### 2.1.1. Preparación de los datos de escenario de red

El primer paso consiste en la obtención e integración de los datos de la red de la infraestructura de la Facultad de Informática y Electrónica (FIE) a través del switch por el puerto g1/0/5 que se encuentra en modo espejo el cual reenvía toda la información de la red al servidor que se encuentra instalado en el datacenter de la FIE, las características del servidor recolector de los datos se encuentran en la tabla 1-2.

**Tabla1-2: Características del servidor recolector de datos**

DISPÓSITIVO	CARACTERÍSTICAS
COMPUTADOR Dual Core 4400 / 3.3 sk. 1151	<ul style="list-style-type: none"> <li>• Cases Q One slim de 600W.</li> <li>• Mainboard H110M-H Socket 1151/ hdmi/ 6ta generación</li> <li>• Memoria Kingston de 4gb pc 2133</li> <li>• Disco Duro de 2tb SATA WD. / BLUE</li> <li>• Óptico LG de 52x SATA.</li> <li>• Monitor LG de 19" LE</li> </ul>

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

A continuación, en la tabla 2-2 se indica las herramientas que se utiliza para la elaboración de este trabajo de titulación.

**Tabla2-2: Herramientas Utilizadas**

HERRAMIENTA	CARACTERÍSTICAS	APLICACIÓN
TCPdump	<ul style="list-style-type: none"> <li>• Permite al usuario capturar y mostrar en tiempo real los paquetes transmitidos y recibidos por la red a la cual el ordenador está conectado.</li> <li>• Funciona en la mayoría de los sistemas operativos UNIX.</li> </ul>	Captura del tráfico en tiempo real, de la red de la infraestructura de la FIE, el cual fue instalado en el servidor con sistema operativo Ubuntu 16.0.
SteelCentral Packet Analyzer	<ul style="list-style-type: none"> <li>• Se escogió esta herramienta ya que cuenta con alianzas de otras empresas como son: Microsoft y Zscaler; entre sus clientes se encuentran: ABB, Open Eyes Scientific, Michelin, Nation Instrument, T-Mobile entre otras.</li> <li>• Visualizar estadísticas de tráfico en directo de larga duración y sin conexión</li> </ul>	Análisis de cada uno de los paquetes de tráfico, esta herramienta permite observar las estadísticas de los protocolos que se encontraba trabajando en la red.

	retrocediendo en el tiempo a través de grandes conjuntos de datos.	
--	--	--

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

Como se explica en la tabla 2-2 el tráfico fue capturado a través de la herramienta TCPdump, con el script que se puede apreciar en la figura 1-2; el cual se programa para que las capturas se realicen cada cinco minutos en horarios de 8 am a 8 pm ya que es el tiempo el cual la red presenta mayor actividad. La muestra utilizada para este trabajo fue de 144 registros diarios de los paquetes asociados al tráfico que circula a través de la red de la FIE. Esta muestra se tomó en las fechas 16 de octubre del 2017 hasta 17 de noviembre del 2017, como se observa en la tabla 3-2.

```

for((sem=1; sem<=5; sem+=1)); do
    for((d=1; d<=5; d+=1)); do
        for((i=1; i<=144; i+=1)); do
            echo SEMANA $sem
            echo DIA $d
            echo CAPTURA $i
            sudo timeout 300 tcpdump -w TESISs$sem,d$d,c$i.pcap -s 0
        done
        sleep 39600
    done
done
sleep 172800
done

```

**Figura1-2: Script para la captura de tráfico de la infraestructura de la red de la FIE**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

El registro total de los datos son de 3600 capturas tomadas en 5 semanas, el tiempo de cada registro es de 5 minutos, por lo tanto en una hora se tiene 12 registros y por consiguiente en un día 144 registros como ya se dijo anteriormente, cada captura de tráfico varía entre 20 Mbytes – 27 Gbytes, cabe recalcar que la tabulación de los datos se realiza diariamente ya que se cuenta con un disco de 2 Terabytes de almacenamiento y no es suficiente espacio para la cantidad de tráfico que circula por la red.

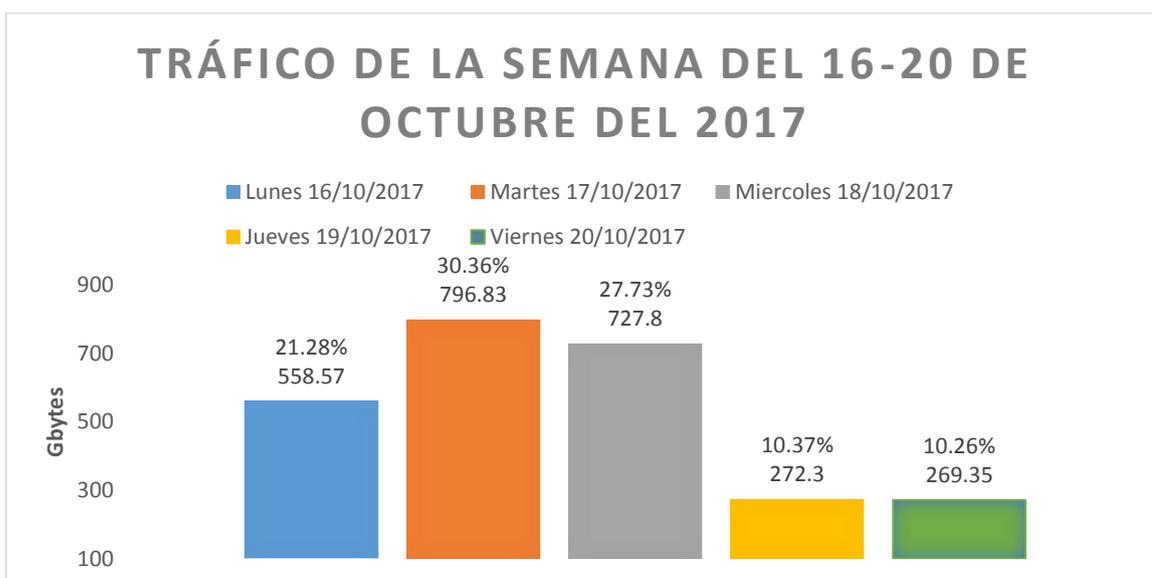
**Tabla3-2: Tiempo y Cantidad de Tráfico capturado en la Red de la FIE**

FECHAS	TRÁFICO CAPTURADO (Gbytes)
16-10-2017	558,57
17-10-2017	796,83
18-10-2017	727,8
19-10-2017	272,3
20-10-2017	269,35
23-10-2017	237,53

24-10-2017	190,13
25-10-2017	546,59
26-10-2017	803,24
27-10-2017	341,77
06-10-2017	404,04
07-10-2017	891,78
08-10-2017	801,69
09-10-2017	446,03
10-10-2017	527,17
13-10-2017	288,67
14-10-2017	388,87
15-10-2017	525,52
16-10-2017	821,82
17-10-2017	198,55

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

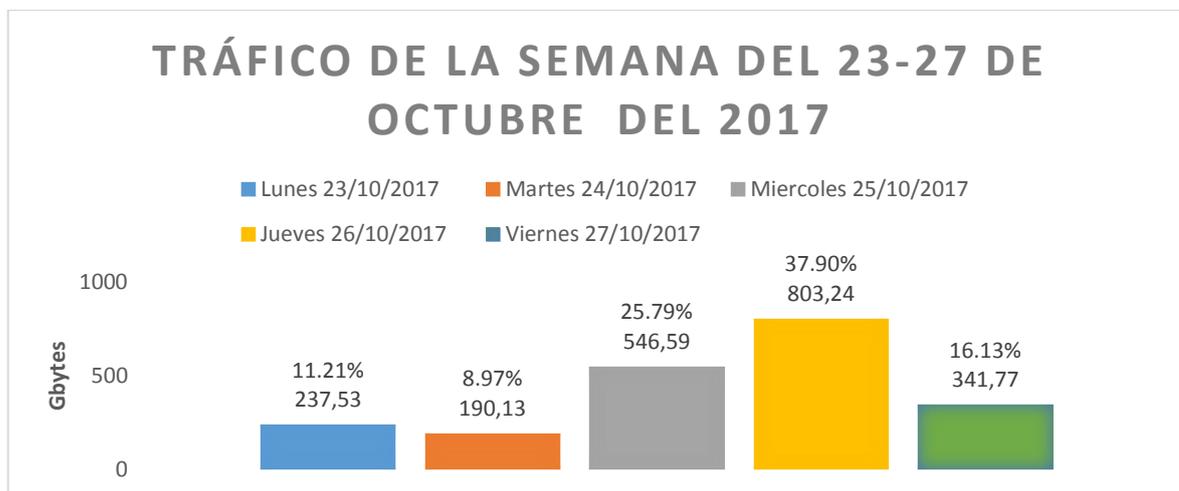
En el gráfico 3-2, se tiene un resumen semanal del tráfico que se generó en la semana del 16-20 de octubre del 2017; en el cual sobresale el día martes 17/10/2017 con mayor cantidad de tráfico de 796.83 Gbytes siendo el 30.36% del total del tráfico de la semana mientras que el día viernes 20/10/2017 se presenta la menor cantidad de tráfico de 269.35 Gbytes siendo el 10.26 % del tráfico total de la semana.



**Gráfico3-2: Resumen del tráfico de la semana del 16-20 de octubre del 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

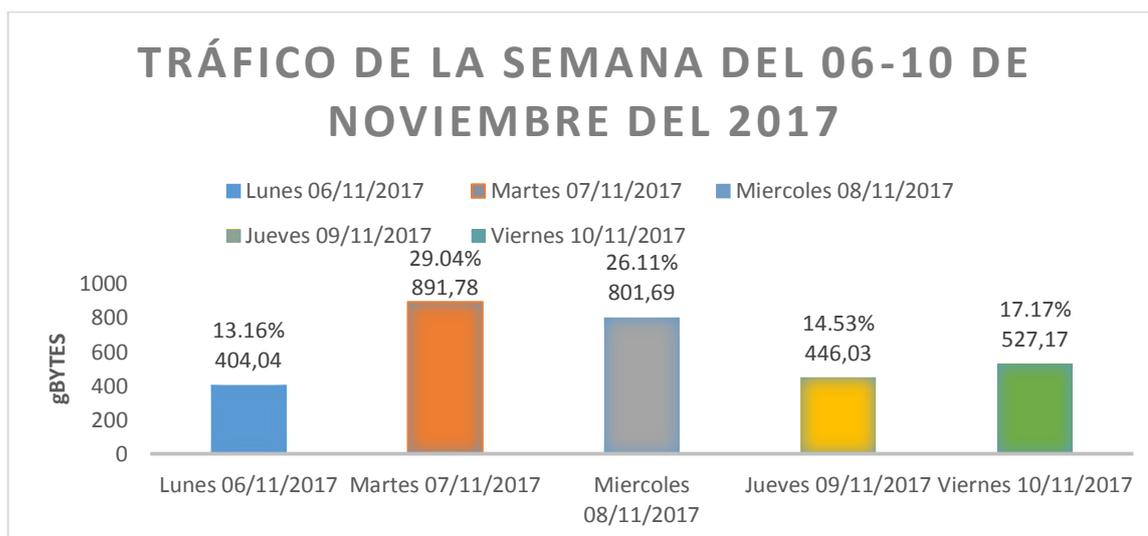
En el gráfico 4-2, se tiene un resumen semanal del tráfico que se generó en la semana del 23-27 de octubre del 2017; en el cual sobresale el día jueves 26/10/2017 con mayor cantidad de tráfico de 803.24 Gbytes siendo el 37.90% del total del tráfico de la semana mientras que el día martes 24/10/2017 se presenta la menor cantidad de tráfico de 190.13 Gbytes siendo el 8.97 % del tráfico total de la semana.



**Gráfico4-2: Resumen del tráfico de la semana del 23-27 de octubre del 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

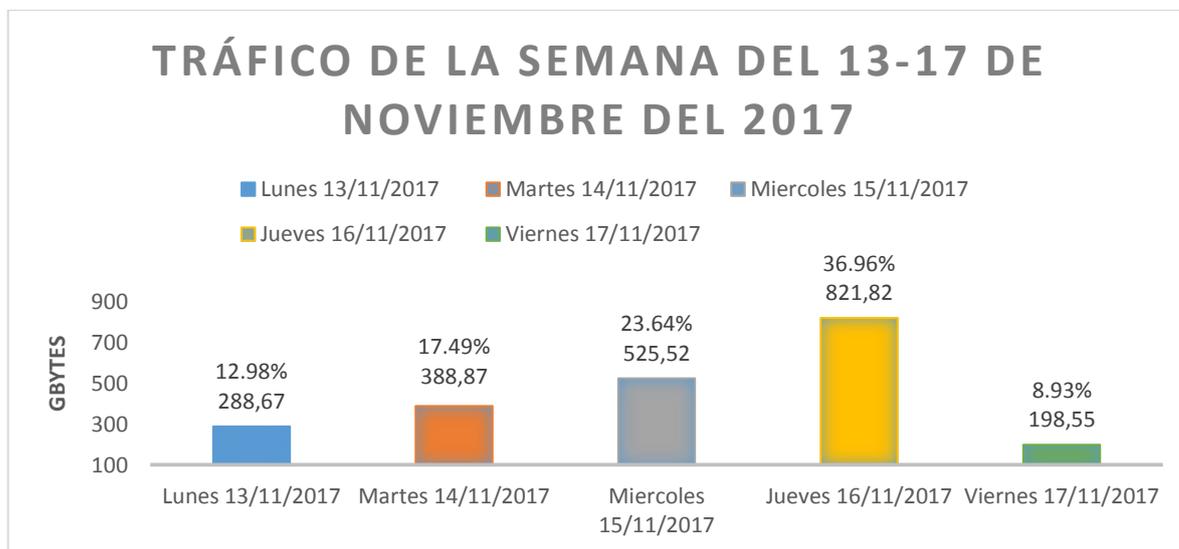
La semana de 30 de octubre al 3 de noviembre no se toma en cuenta por el feriado en los días 2 y 3 de noviembre provocando inactividad en la red, dentro del análisis de este trabajo de titulación no se tomó en cuenta este tráfico pero para el entrenamiento de la red si, los días del 30/10/2017 al 01/11/2017, porque entre más datos se ingresa a la red se tendrá una mayor precisión en los datos resultantes.



**Gráfico5-2: Resumen del tráfico de la semana del 06-10 de octubre del 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

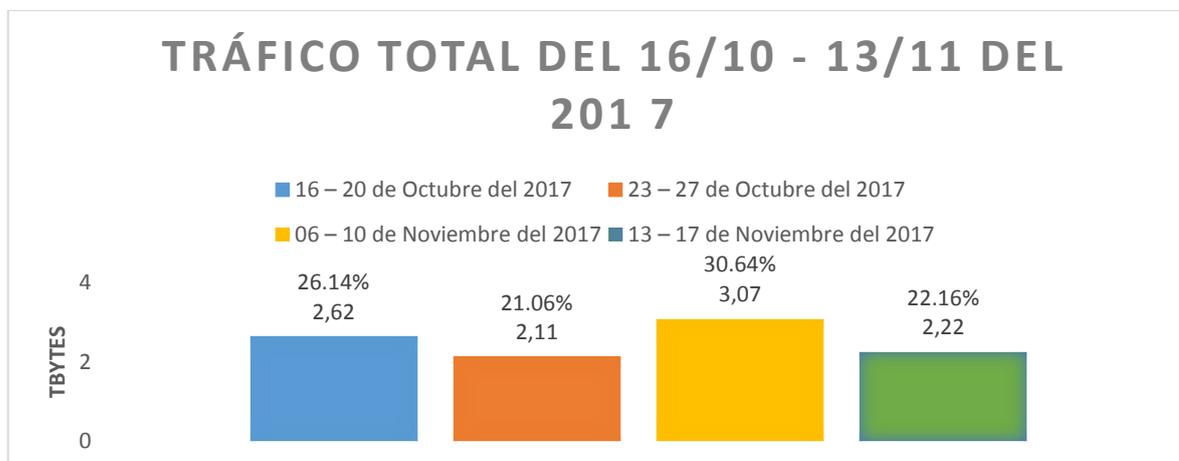
En el gráfico 5-2, se tiene un resumen semanal del tráfico que se generó en la semana del 6-10 de noviembre del 2017; en el cual sobresale el día martes 07/11/2017 con mayor cantidad de tráfico de 891.78 Gbytes siendo el 29.04 % del total del tráfico de la semana mientras que el día jueves 09/11/2017 se presenta la menor cantidad de tráfico de 446.03 Gbytes siendo el 14.53 % del tráfico total de la semana.



**Gráfico6-2: Resumen del tráfico de la semana del 06-10 de octubre del 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

En el gráfico 6-2, se tiene un resumen semanal del tráfico que se generó en la semana del 13-17 de noviembre del 2017; en el cual sobresale el día jueves 16/11/2017 con mayor cantidad de tráfico de 821.82 Gbytes siendo el 36.96% del total del tráfico de la semana mientras que el día viernes 17/11/2017 se presenta la menor cantidad de tráfico de 198.55 Gbytes siendo el 8.93 % del tráfico total de la semana.



**Gráfico7-2: Resumen del tráfico total capturado del 16 de octubre al 13 de noviembre del 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

En el gráfico 7-2, se tiene un resumen del total del tráfico capturado generado en las fechas del 16 de octubre del 2017 al 17 noviembre del 2017; en el cual sobresale la semana del 06-10/11/2017 con mayor cantidad de tráfico de 3.07 Terabytes siendo el 30.64% del total del tráfico mensual mientras que la semana del 23-27/10/2017 se presenta la menor cantidad de tráfico de 2.11 Terabytes siendo el 21.06 % del tráfico total del mes.

En la tabla 4-2 se puede observar un resumen por semanas del tráfico capturado en la red de la infraestructura de la Fie, como se puede ver el trafico semanalmente vario ente 2 Tbytes a 3 Tbytes.

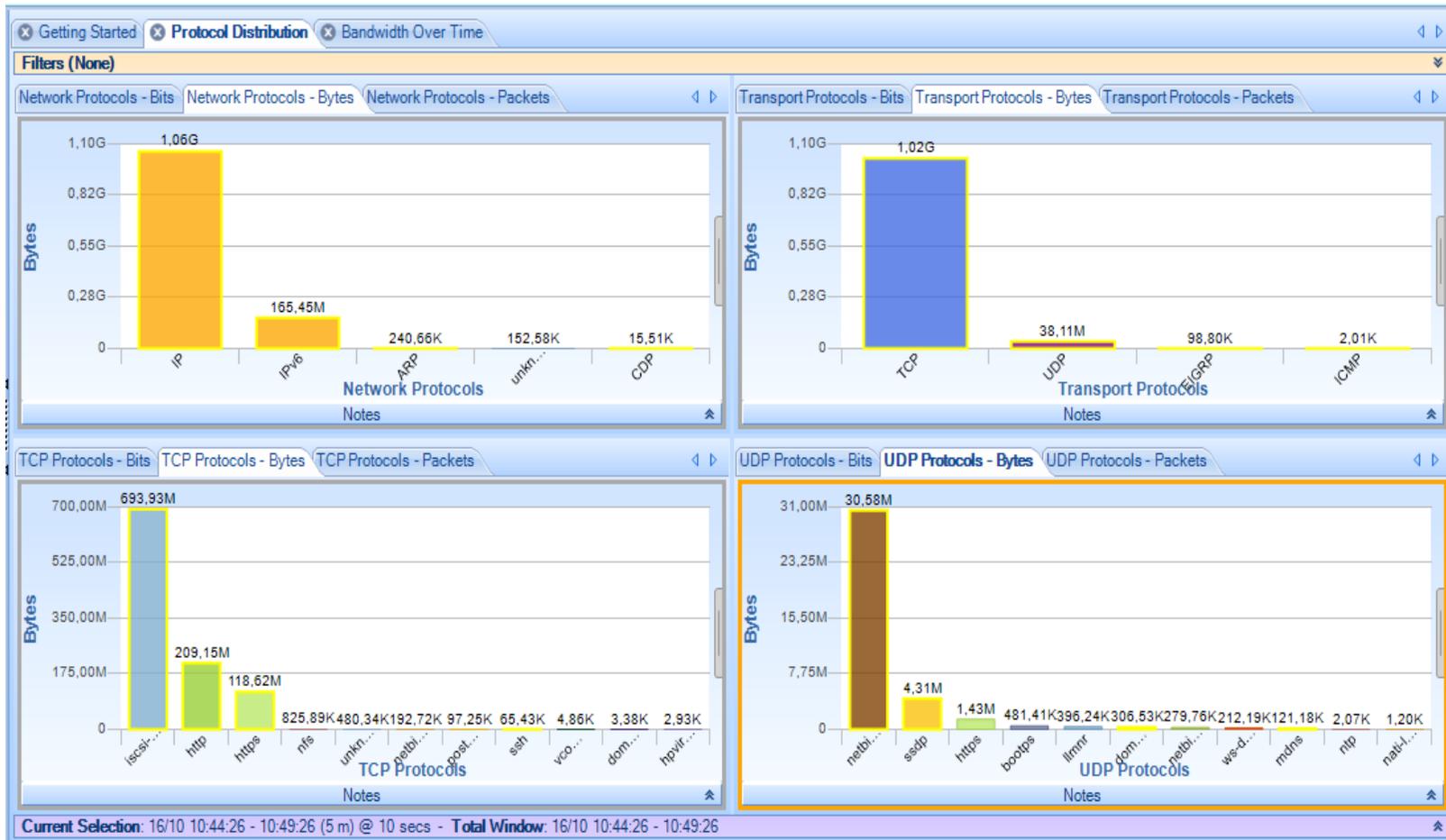
**Tabla4-2: Resumen del Tiempo y Cantidad de Tráfico capturado en la Red de la FIE**

<b>FECHAS DE LAS SEMANAS</b>	<b>TRÁFICO POR SEMANA (Gbytes)</b>
<b>16 – 20 de Octubre del 2017</b>	2624,91 ≈ 2,62 Tbytes
<b>23 – 27 de Octubre del 2017</b>	2119,26 ≈ 2,11 Tbytes
<b>06 – 10 de Noviembre del 2017</b>	3070,71 ≈ 3,07 Tbytes
<b>13 – 17 de Noviembre del 2017</b>	2223,43 ≈ 2,22 Tbytes

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

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

En esta etapa se realiza el pre procesamiento de los datos, el cual consiste en el análisis de cada uno de los paquetes a través de la herramienta SteelCentral Packet Analyzer, la cual permite al usuario conocer todos los protocolos que se encuentran en cada uno de los paquetes capturados que circulan a través de la red, como se observa en la figura 2-2 se puede ver la interfaz gráfica de la herramienta anteriormente mencionada.



**Figura2-2: Protocolos en una captura de 5 minutos dentro de la Red de Infraestructura de la FIE.**  
Fuente: BARRAZUETA, Pamela & TIERRA, Lennin, 2017

Se encontró ciertos inconvenientes dentro el Preprocesamiento de los datos los cuales se detallan en la tabla 5-2, con sus respectivas soluciones.

**Tabla5-2: Inconsistencias y Soluciones encontradas en la captura de tráfico de la red de la FIE**

INCONSISTENCIA	SOLUCIÓN
<b>Medidas del tráfico en Gbytes, Mbytes y Kbytes.</b>	Convertir todas las medidas de tráfico en Bytes.
<b>Encontrar paquetes de datos con protocolo TCP de 0 bytes.</b>	Separar estos paquetes de toda la muestra.

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

### **2.1.3. Selección de protocolos para la predicción**

Una vez realizado el análisis de cada una de las capturas de tráfico se procede a realizar la selección de los protocolos, como se muestra en esta etapa. En primera instancia se logró identificar un conjunto de 16 protocolos que puede explicar el comportamiento de la Red de la FIE, pero se tomó la decisión de simplemente analizar 6 de ellos, ya que representan la mayor cantidad de tráfico de la Red, como se muestra en la tabla 6-2.

**Tabla 6-2:** Clasificación de los protocolos analizados de acuerdo al modelo OSI.

PROTOCOLOS	CAPA DEL MODELO OSI
<b>IPV6</b>	Capa de red
<b>IP</b>	Capa de red
<b>ARP</b>	Capa de red
<b>ISCSI-TARGET</b>	Capa de aplicación
<b>HTTP</b>	Capa de aplicación
<b>HTTPS</b>	Capa de aplicación

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

### **2.1.4. Transformación de los datos**

Una vez seleccionada los protocolos, se procede a realizar una base de datos de todas las capturas realizadas diariamente. En la tabla 7-2 se despliega el conjunto de protocolos que se analizó con sus respectivos porcentajes (cantidad) dentro del total del tráfico.

**Tabla 7-2:** Tráfico (Bytes) medido en 5 minutos segmentados por los protocolos de las capas de Red y Aplicación

PROTOCOLOS(BYTES)		Porcentajes referentes al tráfico total
IPV6	587.15M	10.9%
IP	1614.6M	29.98%
ARP	0.49M	0.013%
ISCSI-TARGET	20.98M	0.55%
HTTP	315.27M	8.36%
HTTPS	1227.65M	32.56%

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

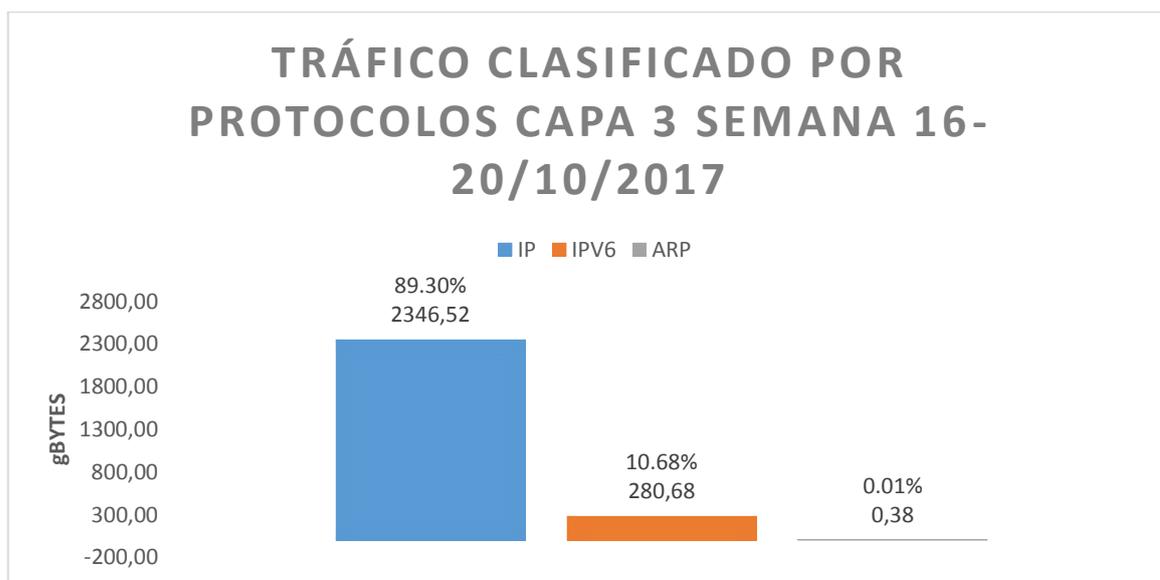
A continuación, en la tabla 8-2 se indica un resumen semanal de los protocolos, clasificados según las capas del modelo OSI con sus respectivas cantidades de tráfico y sus porcentajes referentes al tráfico total. Como se puede observar al sumar las cantidades de los protocolos de capa de aplicación se obtiene como resultado una cantidad cercana al total del protocolo IP de su semana respectiva, todo esto es debido a que este protocolo de Red contiene a los protocolos de capa de Aplicación.

**Tabla 8-2:** Tiempo y Tráfico Total (Gbytes) clasificado por los 6 protocolos seleccionados

SEMANAS	CAPA DEL MODELO OSI	PROTOCOLOS	CANTIDAD(Gbytes)	PORCENTAJES (%)
16-20 Octubre 2017	CAPA DE RED	IPV6	280.68	10.68
		IP	2346.52	89.30
		ARP	0.38	0.01
	CAPA DE APLICACIÓN	ISCSI-TARGET	1729.61	77.34
		HTTP	212.81	9.52
		HTTPS	293.88	13.14
23-27 Octubre 2017	CAPA DE RED	IPV6	236.91	11.17
		IP	1882.86	88.80
		ARP	0.54	0.03
	CAPA DE APLICACIÓN	ISCSI-TARGET	966.63	66.02
		HTTP	222.21	15.18
		HTTPS	275.32	18.80
06-10 Noviembre 2017	CAPA DE RED	IPV6	184.83	6.13
		IP	2830.52	93.86
		ARP	0.317	0.01
	CAPA DE APLICACIÓN	ISCSI-TARGET	2427.5	86.57
		HTTP	111.11	3.96
		HTTPS	265.41	9.47
13-17 Noviembre 2017	CAPA DE RED	IPV6	189.95	8.54
		IP	2033.04	91.44
		ARP	0.367	0.02
	CAPA DE APLICACIÓN	ISCSI-TARGET	1470.78	74.69
		HTTP	227.63	11.56
		HTTPS	270.8	13.75

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

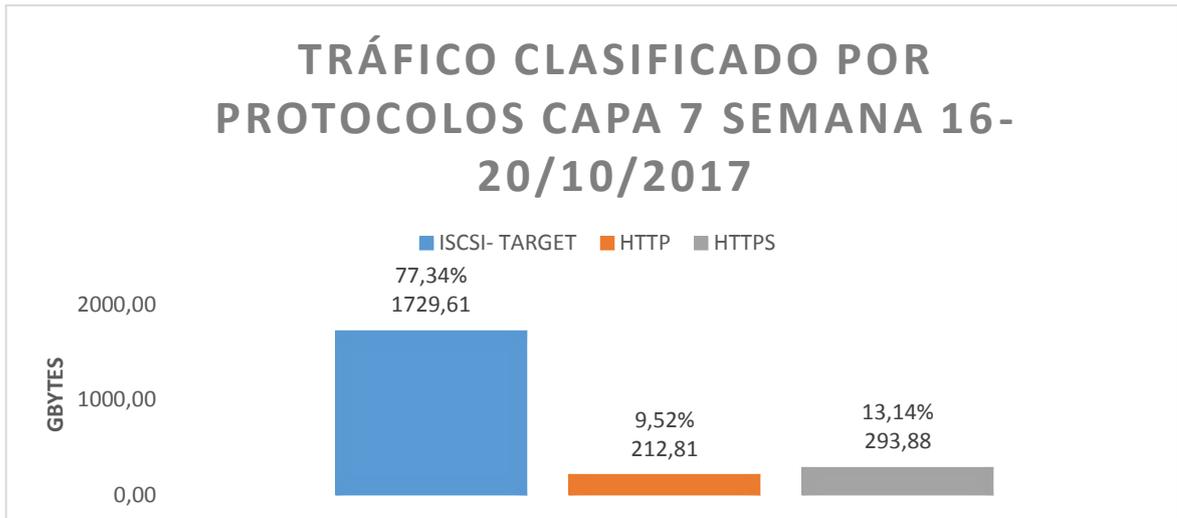
Como se observa en el gráfico 8-2 el tráfico se ha clasificado en 3 protocolos, los cuales son IP, IPV6 y ARP; que se encuentran presentes en la capa de red del modelo OSI, el protocolo IP se encuentra presente en el tráfico en gran cantidad con 2346,52 Gbytes que representa el 89.30% del tráfico total de la semana, mientras que el protocolo ARP el cual cuenta con el 0.01% del tráfico total de la semana algo insignificante en relación a los demás protocolos, todo ello en cuando a capa de red.



**Gráfico8-2: Tráfico de los protocolos de Capa de Red de la semana del 16-20 de octubre 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

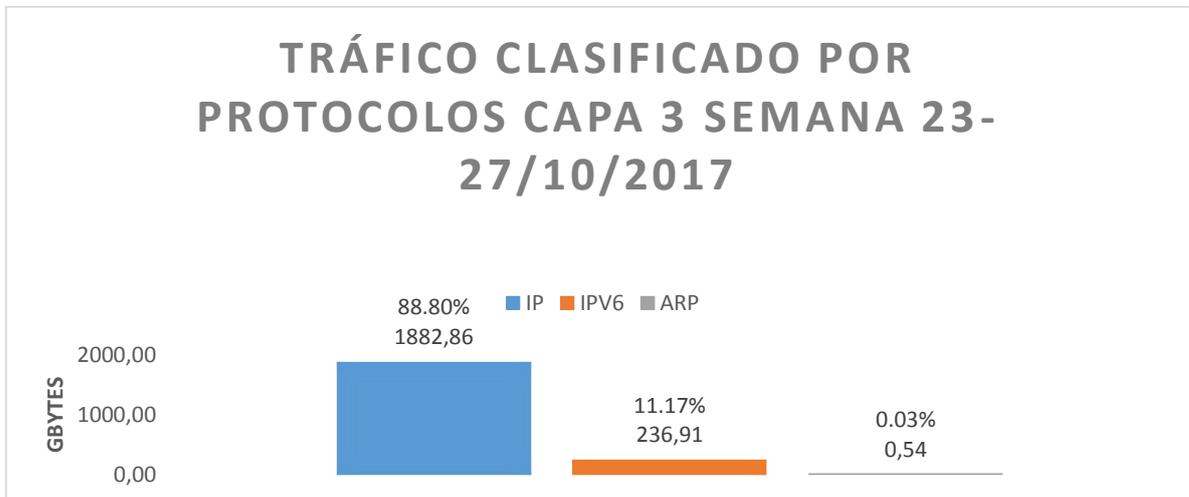
En cambio, como se indica en el gráfico 9-2 el tráfico se ha clasificado en 3 protocolos (ISCSI-TARGET, HTTP y HTTPS) los cuales pertenecen a la capa de aplicación del modelo OSI, la mayor cantidad de tráfico se encuentra centrada en el protocolo ISCSI-TARGET con 1729,61 Gbytes que representa el 77,34% del tráfico total semanal, otra observación que se pudo notar encontrar es, la diferencia mínima entre los protocolos HTTP y HTTPS, siendo de 81.08 Gbytes.



**Gráfico9-2: Tráfico de los protocolos de Capa de Aplicación de la semana del 16-20 de octubre 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

Como se observa en el gráfico 10-2, el protocolo IP se encuentra presente en el tráfico en gran cantidad con 1882,86 Gbytes que representa el 88.80% del tráfico total de la semana mientras que el protocolo ARP es el que se encuentre en una mínima cantidad en relación a los demás protocolos, siendo el 0.03% del tráfico semanal como se puede analizar que el protocolo IP disminuye su utilización en esta semana, todo ello en cuando a capa de red.

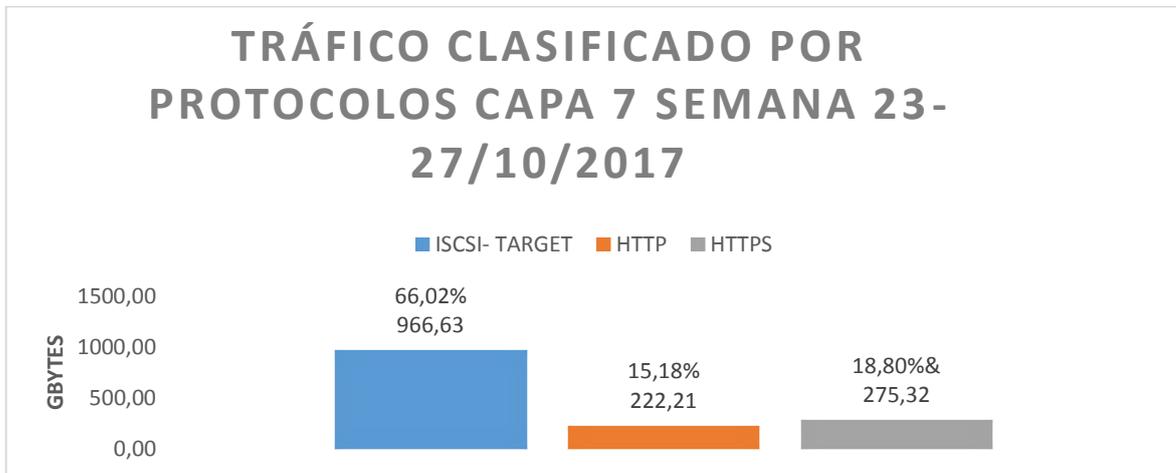


**Gráfico10-2: Tráfico clasificado por protocolos de Capa de Red de la semana del 23-27 de octubre del 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

En cambio, como se indica en el gráfico 11-2 el tráfico se ha clasificado en 3 protocolos (ISCSI-TARGET, HTTP y HTTPS) los cuales pertenecen a la capa de aplicación del modelo OSI, la mayor cantidad de tráfico se encuentra centrada en el protocolo ISCSI-TARGET con 966,63

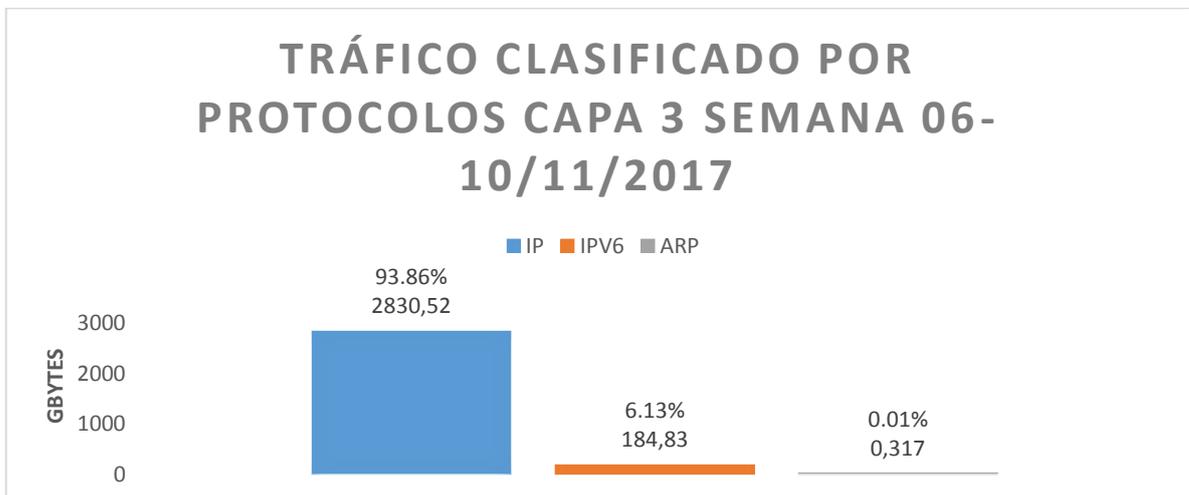
Gbytes que representa el 66,02% del tráfico total semanal, se puede ver que los protocolos HTTP y HTTPS está muy cercano el uno del otro, teniendo una diferencia entre ellos de 53.11 Gbytes.



**Gráfico11-2: Tráfico de los protocolos de Capa de Aplicación de la semana del 23-27 de octubre 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

Como se observa en el gráfico 12-2, el protocolo IP se encuentra presente en el tráfico en gran cantidad con 2830,52 Gbytes que representa el 93.86% del tráfico total de la semana mientras que el protocolo ARP es el que se encuentre en un mínimo cantidad en relación a los demás protocolos, todo ello sucede a nivel de capa de red.

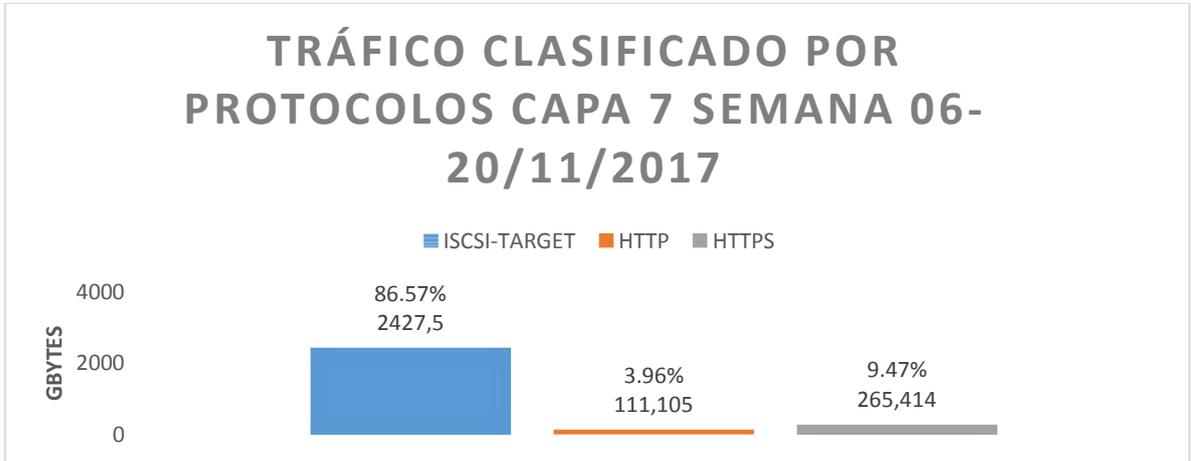


**Gráfico12-2: Tráfico clasificado por protocolos de Capa de Red de la semana del 06-10 de noviembre del 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

En cambio, como se indica en el gráfico 13-2 el tráfico se ha clasificado en 3 protocolos (ISCSI-TARGET, HTTP y HTTPS) los cuales pertenecen a la capa de aplicación del modelo OSI, la mayor cantidad de tráfico se encuentra centrada en el protocolo ISCSI-TARGET con 2427,5

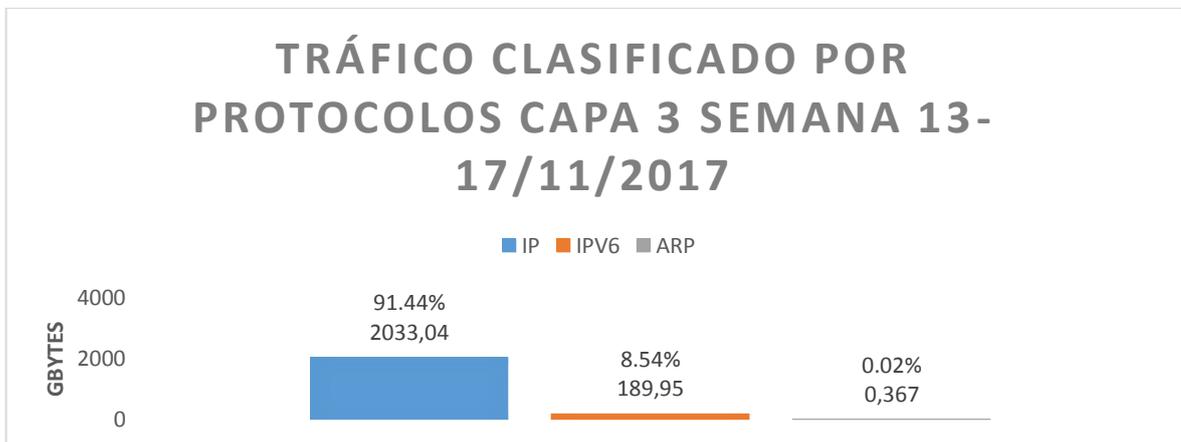
Gbytes que representa el 86,57% del tráfico total semanal, mientras que en capa de aplicación el protocolo HTTPS muestra mayor utilización con 265.41 Gbytes que representa el 9.47% del tráfico semanal.



**Gráfico13-2: Tráfico clasificado por protocolos de Capa de Aplicación de la semana del 06-10 de noviembre del 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

Como se observa en el gráfico 14-2, el protocolo IP se encuentra presente en el tráfico en gran cantidad con 2033,04 Gbytes que representa el 91.44% mientras que el protocolo ARP con 0.367 Gbytes es el que se encuentre en una mínima cantidad en relación a los demás protocolos, todo ello sucede a nivel de capa de red.

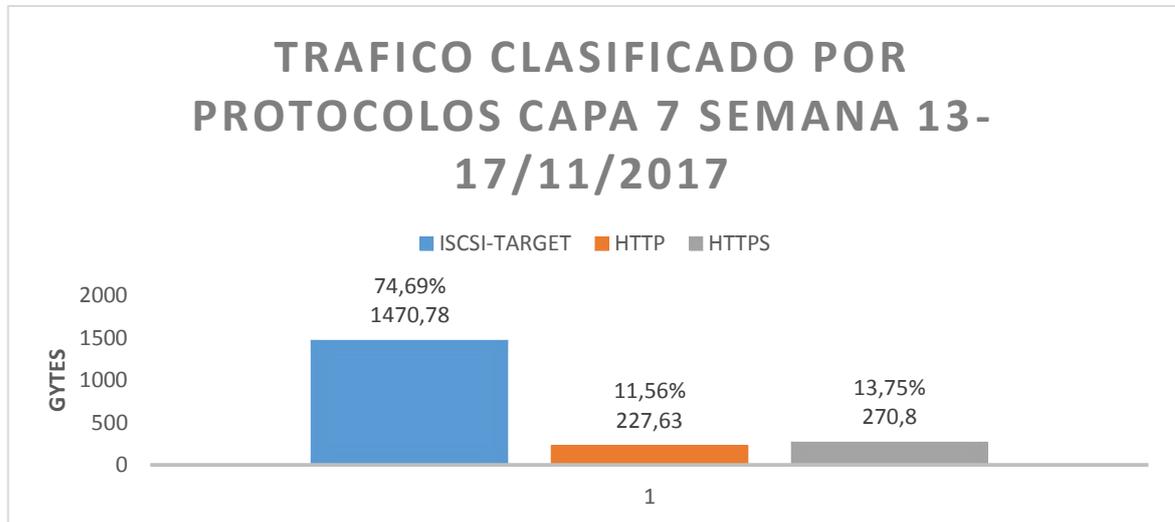


**Gráfico14-2: Tráfico clasificado por protocolos de Capa de Red de la semana del 13-17 de noviembre del 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

En cambio, como se indica en el gráfico 15-2 el tráfico se ha clasificado en 3 protocolos (ISCSI-TARGET, HTTP y HTTPS) los cuales pertenecen a la capa de aplicación del modelo OSI, la mayor cantidad de tráfico se encuentra centrada en el protocolo ISCSI-TARGET con 1270,78

Gbytes que representa el 74,69% del tráfico total semanal, mientras que entre los protocolos HTTP y HTTPS tienen una diferencia de 43.17 Gbytes



**Gráfico15-2: Tráfico de los protocolos de Capa de Aplicación de la semana del 13-17 de noviembre 2017**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

## FASE 2

### 2.1.5. Selección del tipo de red neuronal artificial según los requerimientos necesarios

Las redes neuronales se caracterizan por factores como los datos que estas puedan analizar y filtrar, el menor error cometido; lo que nos muestra su mayor ventaja en la tarea que se le asigna a la red( TORRES Á, NELSON S, HERNÁNDEZ C and PEDRAZA L, 2011).

- **Elevada cantidad de datos para el entrenamiento**

Debido a la gran cantidad de datos que son 2880 que se capturo, la red neuronal tiene la ventaja de contar con un mayor número de datos para su aprendizaje, con la facilidad de realizar el reconocimiento de cada uno de los protocolos, pero para ello se debe de tomar en cuenta que el aprendizaje conlleva un gran procesamiento del CPU ocupando el 78% de proceso de la computadora, como se observa en la figura 3-2.



**Figura3-2: Procesamiento de la computadora mientras realiza la predicción de los datos.**  
 Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

- **Complejidad de aprendizaje para grandes tareas**

Una vez recolectado todos los datos y estructurado la red neuronal se procede al aprendizaje para la predicción de los datos específicamente de estos 6 protocolos que se los selecciono anteriormente, por ello es importante que el aprendizaje de los datos sea lo más simple posible para que no se desarrolle ningún inconveniente con respecto a la tarea a realizar.

A continuación, se detalla los pasos que realiza el algoritmo backpropagation para el proceso de aprendizaje para la predicción de los datos:

- **Tiempo de aprendizaje elevado**

Debido a la gran cantidad de datos que se ingresan a la red neuronal, el tiempo de aprendizaje de este depende directamente del procesamiento de los datos en el CPU por ello es recomendable tener una computadora con un procesador suficientemente bueno para que realice de una manera rápida y correcta, el aprendizaje de los datos.

### ***2.1.6. Implementación de la red neuronal artificial seleccionada en la herramienta Matlab.***

De acuerdo a estudios previos se pudo concluir que el número correcto de capas ocultas es de 2 capas ya que no existe razón teórica para usar más que ese número. De acuerdo a la regla de la pirámide geométrica (Tsoukalas, Lefteri & E. Uhrig, Robert. 1997) que se muestra en la ecuación 1-9 se logra calcular el número de neuronas por cada oculta

$$h^2 = m * n \quad (1-9)$$

Donde:

h= es el número inicial de neuronas en la capa ocultas.

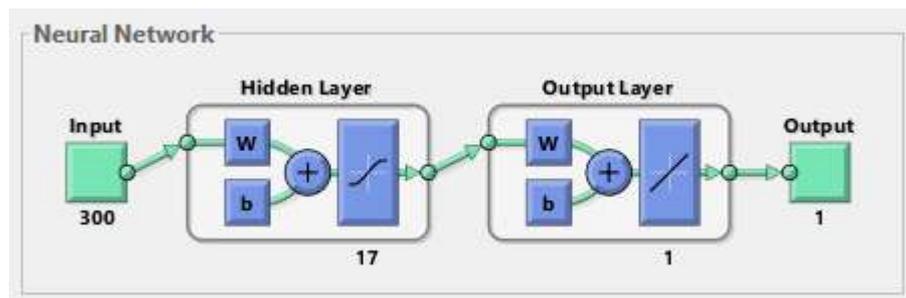
m= es el número de neuronas de salida.

n= es el número de neuronas de entrada.

Por consiguiente, luego de realizar el respectivo cálculo se obtuvo 17 neuronas por capa; si aumentamos el número de capas ocultas, el error no varía.

#### 2.1.6.1. Topología de la red neuronal artificial seleccionada.

Como se observa en la figura 4-2 se diseñó la red neuronal la cual está constituida por 300 entradas, 1 capa oculta con 17 neuronas y una única capa de salida.



**Figura4-2: Red Neuronal Artificial Backpropagation**  
Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

### FASE 3

#### 2.1.7. Implementación del algoritmo en la red neuronal artificial seleccionada.

Para realizar el desarrollo de la red neuronal artificial en la herramienta Matlab, se debe seguir el siguiente proceso como se muestra en la grafico 16-2.



**Grafico16-2: Diagrama de flujo para la elaboración de la RNA**  
Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

### 2.1.7.1. Creación de la red neuronal artificial

Para la creación de la red neuronal, hemos cargado los datos de tráfico de red hacia el programa almacenándolo en un vector llamado target (PONCOS, M, 2018), el cual se ejemplifica en el grafico 17-2, esta matriz de 7x7 es un ejemplo básico del ingreso de los datos hacia la red.

P1	P2	P3	P4	P5	P6
0	3470	447	0,10	0,08863	3456
0	0	3470	447	0,10	0,08863
0	0	0	3470	447	0,10
0	0	0	0	3470	447
0	0	0	0	0	3470
0	0	0	0	0	0

**Grafico17-2: Matriz de entradas que ingresa en la RNA.**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

En segundo lugar, generamos un vector de igual tamaño al anterior la cual llamaremos captura, como se observa en la figura 5-2.

```
target=[xlsread('DATOS.xlsx','TESIS','S2:S2331')]';%cargar datos
l=length(target); %calcula el largo del vector target
captura=1:1:1; %genera un vector de igual numero de datos ingresados

plot(captura,target);
xlabel ('Numero de Captura');
ylabel ('Tráfico en [Mbits]');
title ('Tráfico a Predecir');
%GRAFICO
```

**Figura5-2: Líneas de código para cargar datos, crear un vector y visualización del tráfico**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

A continuación, creamos la red neuronal con la función newff (Crear una red de retro propagación de feed-forward) indicándole que queremos 17 neuronas para la capa oculta (CZECH TECHNICAL UNIVERSITY, 2018). Listo, ya tenemos la red neuronal creada con los parámetros y configuración adecuada, número de capas, tamaño de la capa de entrada, tamaño de la capa de salida como se observa en la figura 6-2.

```
net=newff(IN,target,17); % Creacion de la red backpropagation
net.trainParam.epochs = 1000; % Número máximo de épocas para entrenar
net.trainParam.goal = 0.01; % Objetivo de rendimiento
net.trainParam.max_fail = 100 ;% Errores máximos de validación
```

**Figura6-2: Líneas de código para crear la RNA**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

### 2.1.7.2. Entrenamiento de la red neuronal

Una vez creada la red neuronal, y como ya tenemos los patrones de entrenamiento con sus respectivos objetivos cargados en Matlab, creamos una matriz que simulara las entradas de la red para su entrenamiento (CORCUERA P, 2013) con los datos ya cargados anteriormente. Tal como se observa en la figura 7-2.

```
IN=zeros(300,1); % creacion matriz 300xnumero de capturas
IN(1,2:1) = target(1,1:(1-1)); %datos de entrada
IN(2,3:1) = target(1,1:(1-2));
IN(3,4:1) = target(1,1:(1-3));
IN(4,5:1) = target(1,1:(1-4));
IN(5,6:1) = target(1,1:(1-5));
IN(6,7:1) = target(1,1:(1-6));
IN(7,8:1) = target(1,1:(1-7));
IN(8,9:1) = target(1,1:(1-8));
IN(9,10:1) = target(1,1:(1-9));
```

**Figura7-2: Líneas de código de los datos de entrada**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

Finalmente se llama a la función Train (MATHWORK, 2018) tal como se muestra en la figura 8-2. Con esto Matlab comenzará a entrenar la red neuronal hasta alcanzar el performance deseado, abriéndose una ventana con la información de dicho proceso.

```
net=train(net,IN,target); % Entrenamiento de la RNA
gensim(net) % diagrama de bloques en simulink |
```

**Figura8-2: Líneas de código de entrenamiento de la red**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

### 2.1.7.3. Simulación de la red neuronal

Cuando la red ya ha sido entrenada, ya podemos aplicar un patrón real a la entrada y ver en la salida. Para esto utilizamos la función sim (Simulate a neural network) que simula las salidas de la red neuronal (SAN DIEGO STATE UNIVERSITY, 2018) como se observa en la figura 9-2.

```
simu = sim (net, IN); % Simulacion
view(net)
```

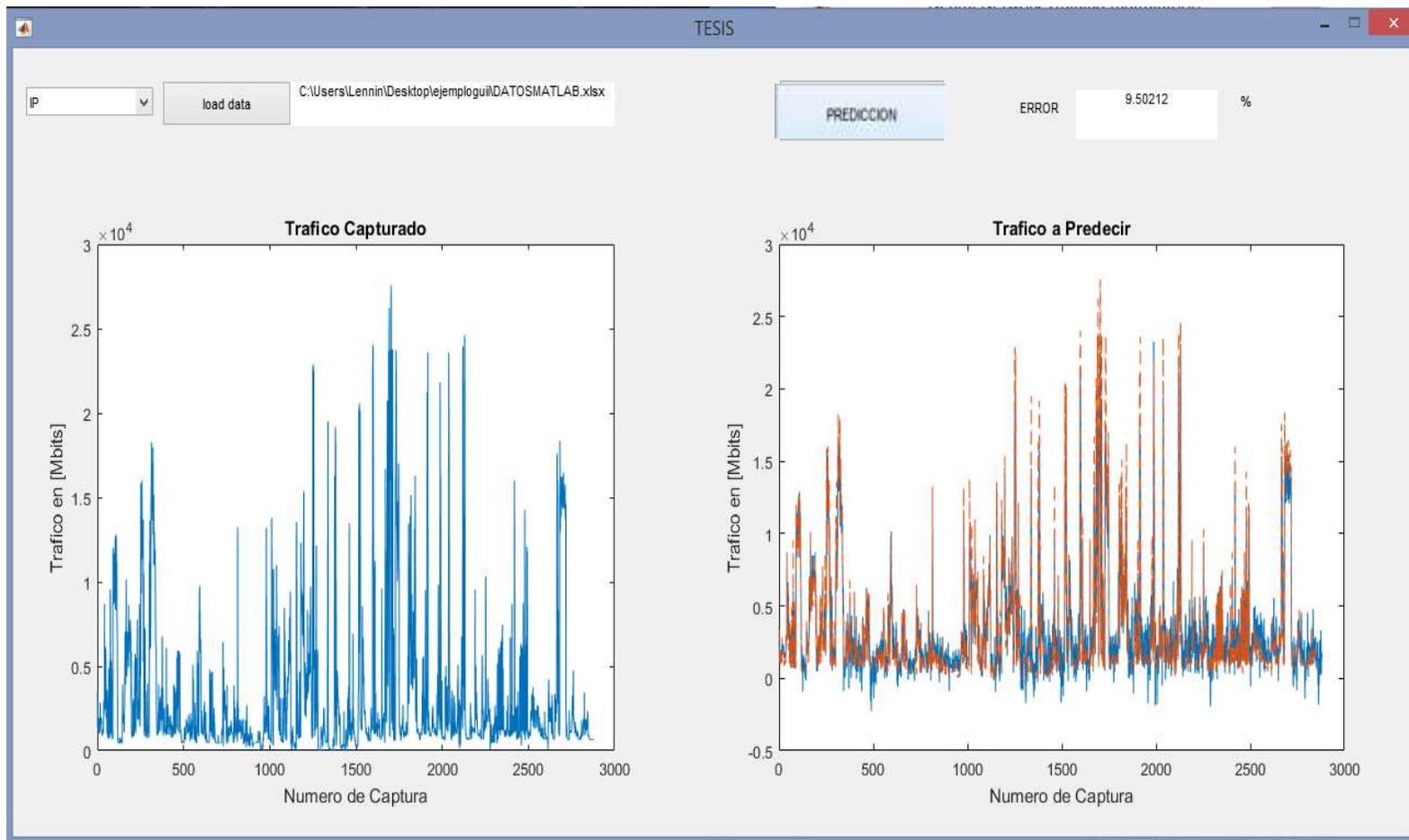
**Figura9-2: Líneas de código para simulación de la RNA**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

#### *2.1.7.4. Interfaz gráfica de la red neuronal artificial*

En la figura 10-2 podemos observar la interfaz gráfica ( MORENO A, 2009) con el análisis de cada uno de los protocolos pronosticados con sus respectivas gráficas. Realizamos el análisis del protocolo ARP. La interfaz gráfica funciona de la siguiente manera:

1. Extraer el archivo .xlsx para ello debemos dar clic en el botón **load data** y se cargara el archivo en nuestra red neuronal artificial.
2. Luego seleccionamos el protocolo que deseamos pronosticar entre las opciones tenemos: IP, IPV6, ARP, HTTPS, HTTP y ISCSI-target.
3. Finalmente se dará clic en el botón **PREDICCIÓN** y se mostrará como resultado las gráficas de las capturas realizas y las capturas pronosticadas, además de ello se muestra el error de cada protocolo.



**Figura10-2: Interfaz gráfica de la RNA**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

## CAPÍTULO III

### 3. EVALUACIÓN Y COMPARACIÓN DE RESULTADOS

#### 3.1. Simulación y análisis

A continuación, se presenta las pruebas y resultados de la red neuronal artificial y usando el modelo de predicción que se explicó anteriormente se procede a realizar la simulación. Los parámetros establecidos para realizar las pruebas dependieron del número de entradas, salidas y neuronas intermedias, como se puede observar en la tabla 1.3; en la cual se presenta un resumen de los parámetros utilizados en las pruebas posteriores.

**Tabla 1-3: Parámetros de simulación**

PRUEBAS		1	2	3
		PARÁMETROS		
NÚMERO DE NEURONAS	ENTRADA	100	200	300
	INTERMEDIA	10	14	17
	SALIDA	1	1	1
FUNCIÓN DE TRANSFERENCIA		Tangente sigmoidea		
FUNCIÓN DE ENTRENAMIENTO		Traingd <sup>1</sup>		

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

#### 3.1.1. Simulación de la primera prueba

La primera prueba se realizó con 100 capas de entradas, 10 neuronas intermedias y 1 capa de salida, donde se analizó el error de los protocolos con los parámetros que se detallan en la tabla 2-3.

---

<sup>1</sup> Función Matlab Retro producción de pendiente descendente

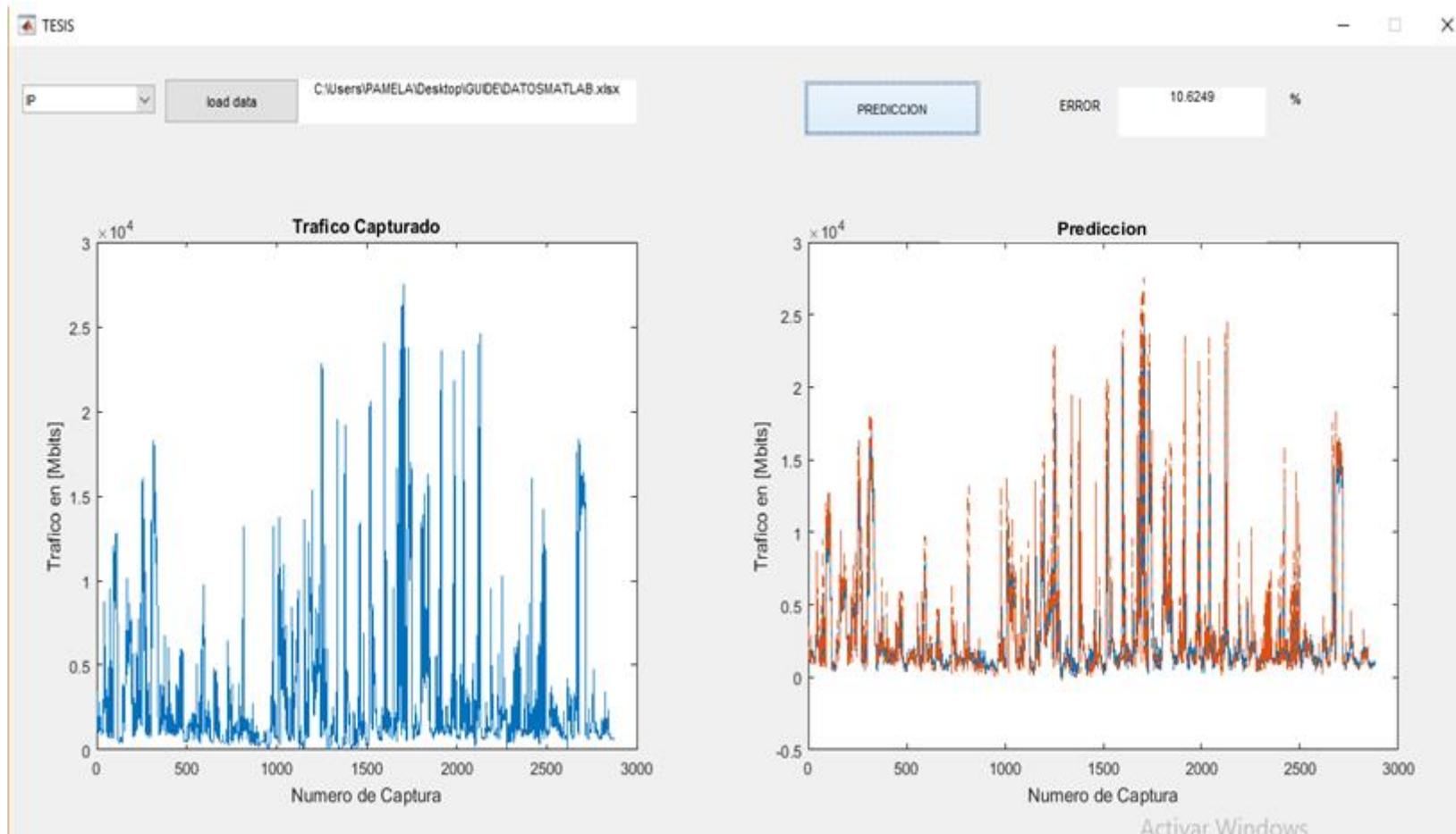
**Tabla 2-3: Parámetros de la Primera Prueba**

<b>PARÁMETROS</b>	<b>ENTRADA</b>	<b>OCULTA</b>	<b>SALIDA</b>
<b>NÚMERO DE NEURONAS</b>	100	10	1
<b>FUNCIÓN DE TRANSFERENCIA</b>	tansig	tansig	tansig
<b>FUNCIÓN DE ENTRENAMIENTO</b>	Traingd		

**Realizado por:** BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

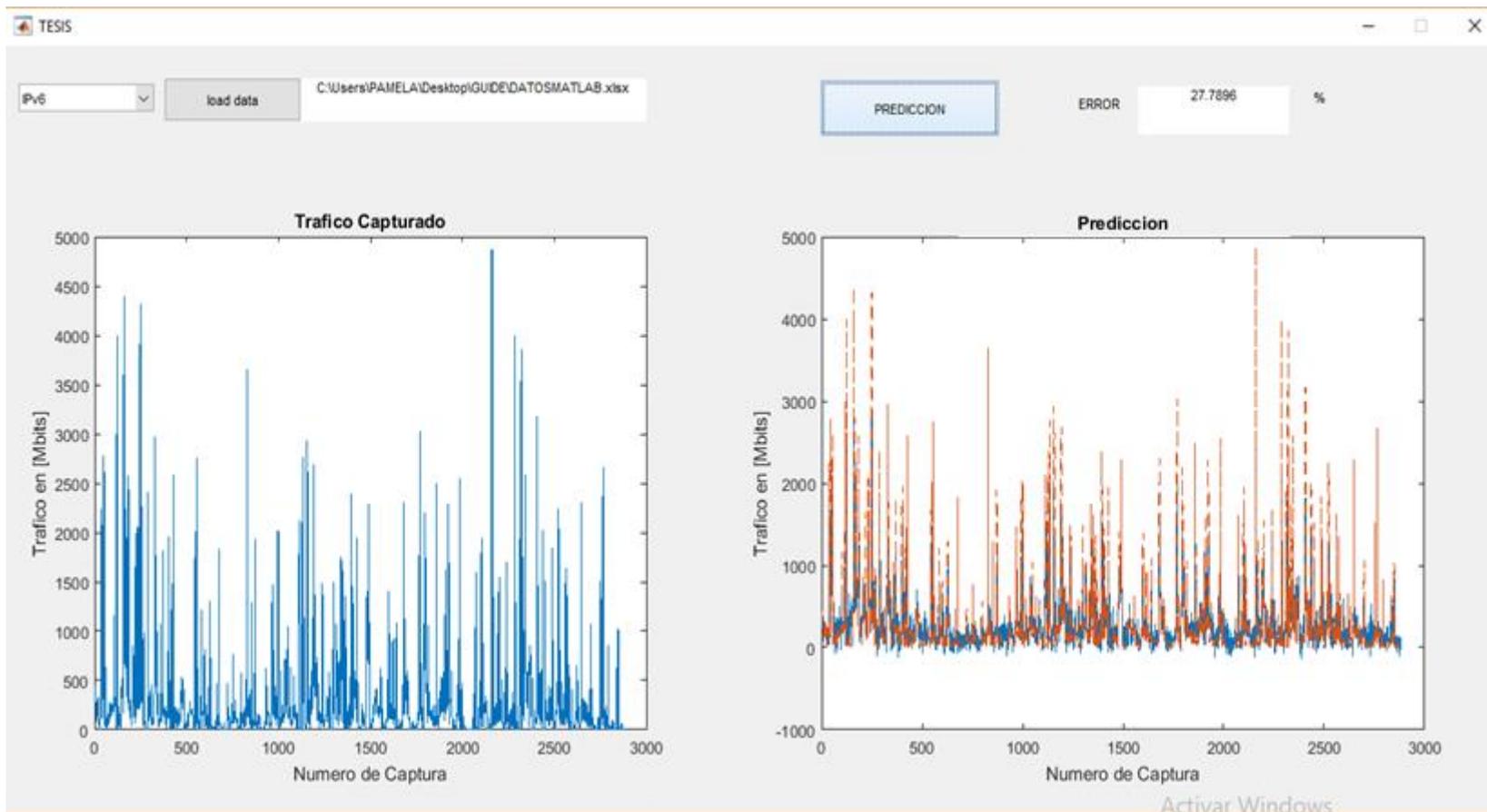
En la figura 1.3 se observa la simulación del protocolo IP, obteniendo una gráfica del tráfico real y del tráfico simulado, con una red neuronal artificial cuyos parámetros se encuentran en la tabla 2.3.

Como se observa en la figura el tráfico del protocolo IP es muy parecido al tráfico total de la red, ya que este abarca gran cantidad del tráfico utilizado. Después de analizar las 2 gráficas obtenidas se pudo concluir que se tiene una predicción muy cercana a la real con un error 10.62%.



**Figura 1-3: Simulación Primera Prueba Protocolo IP**  
 Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018

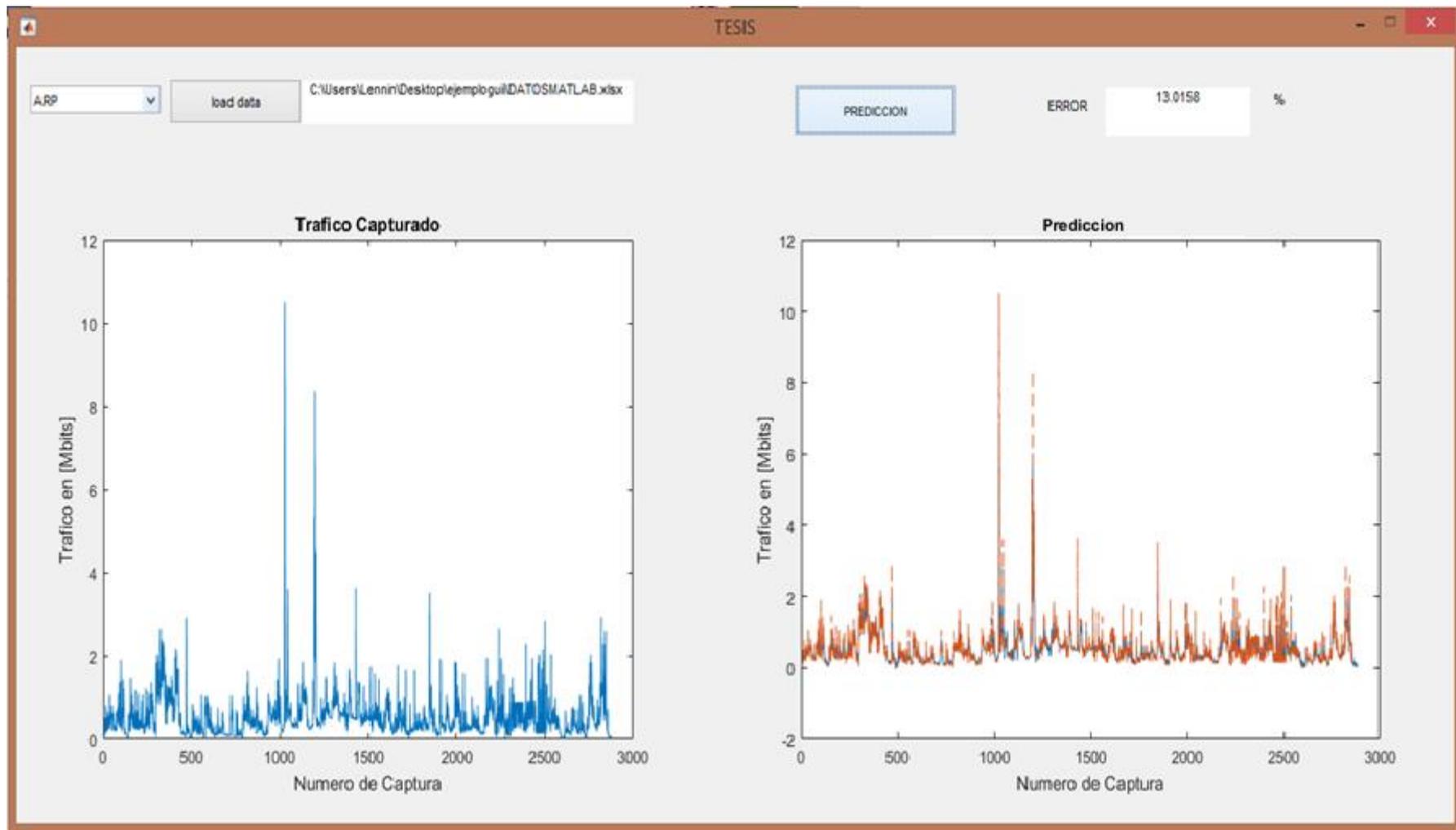
En la figura 2.3 se observa la simulación del protocolo IPv6, después de realizar el entrenamiento con los datos de 4 semanas dentro de la red neuronal artificial cuyos parámetros se encuentran en la tabla 2.3. Posterior al análisis de las 2 gráficas obtenidas se pudo concluir que se tiene una predicción con un error del 27.78%.



**Figura 2-3: Simulación Primera Prueba Protocolo IPv6**  
 Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018

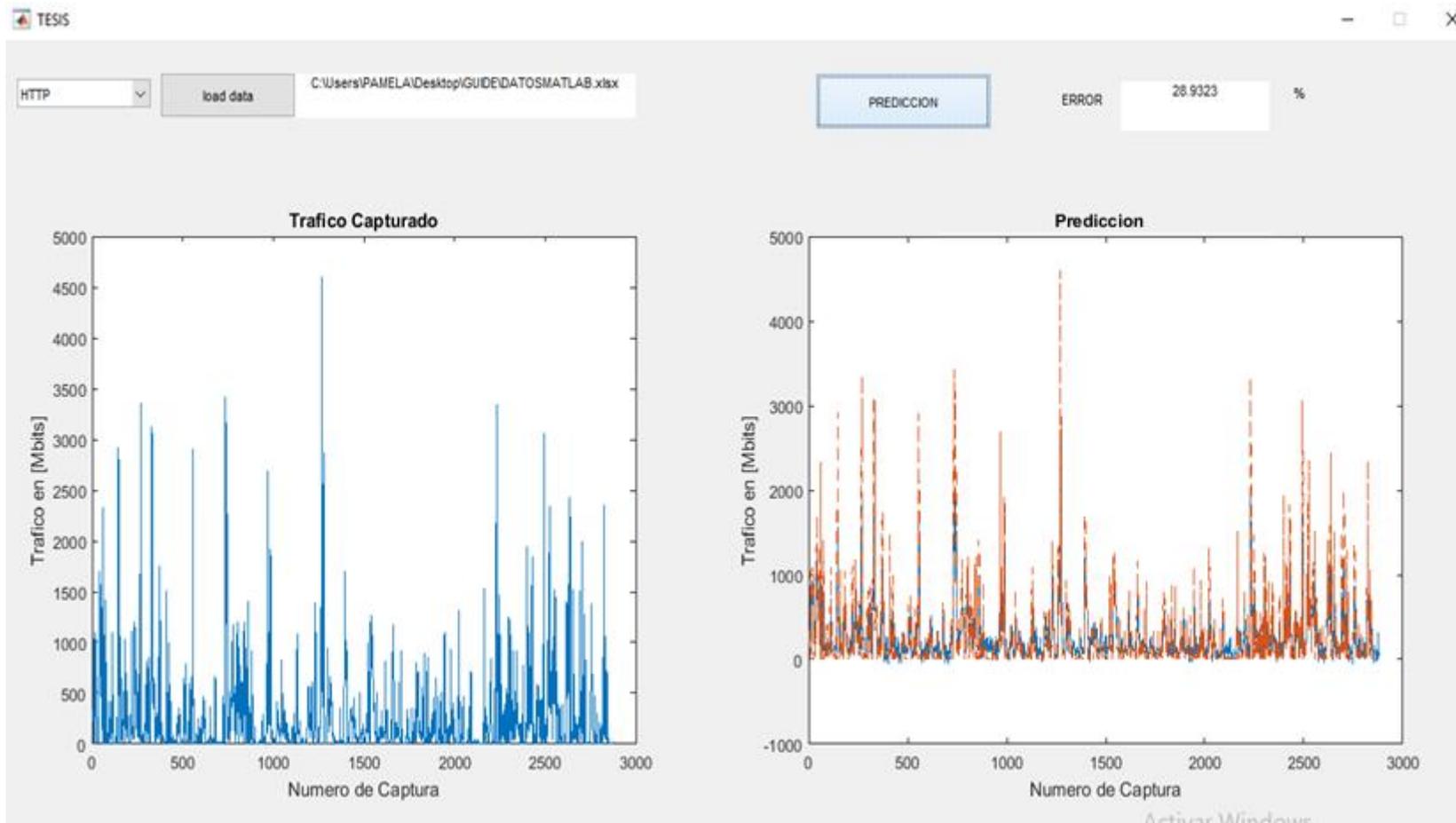
En la figura 3.3 se observa la simulación del protocolo ARP, obteniendo una gráfica del tráfico real y del tráfico simulado, con una red neuronal artificial cuyos parámetros se encuentran en la tabla 2.3.

Después de analizar las 2 graficas se pudo concluir que se tiene una predicción muy cercana a la real obteniendo un error de 13.01%.



**Figura 3-3: Simulación Primera Prueba Protocolo ARP**  
Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018

En la figura 4.3 se observa la simulación del protocolo HTTP, después de realizar el entrenamiento con los datos de 4 semanas dentro de la red neuronal artificial cuyos parámetros se encuentran en la tabla 2.3. Posterior al análisis de las 2 graficas obtenidas se pudo concluir que se tiene una predicción con un error del 28.93%

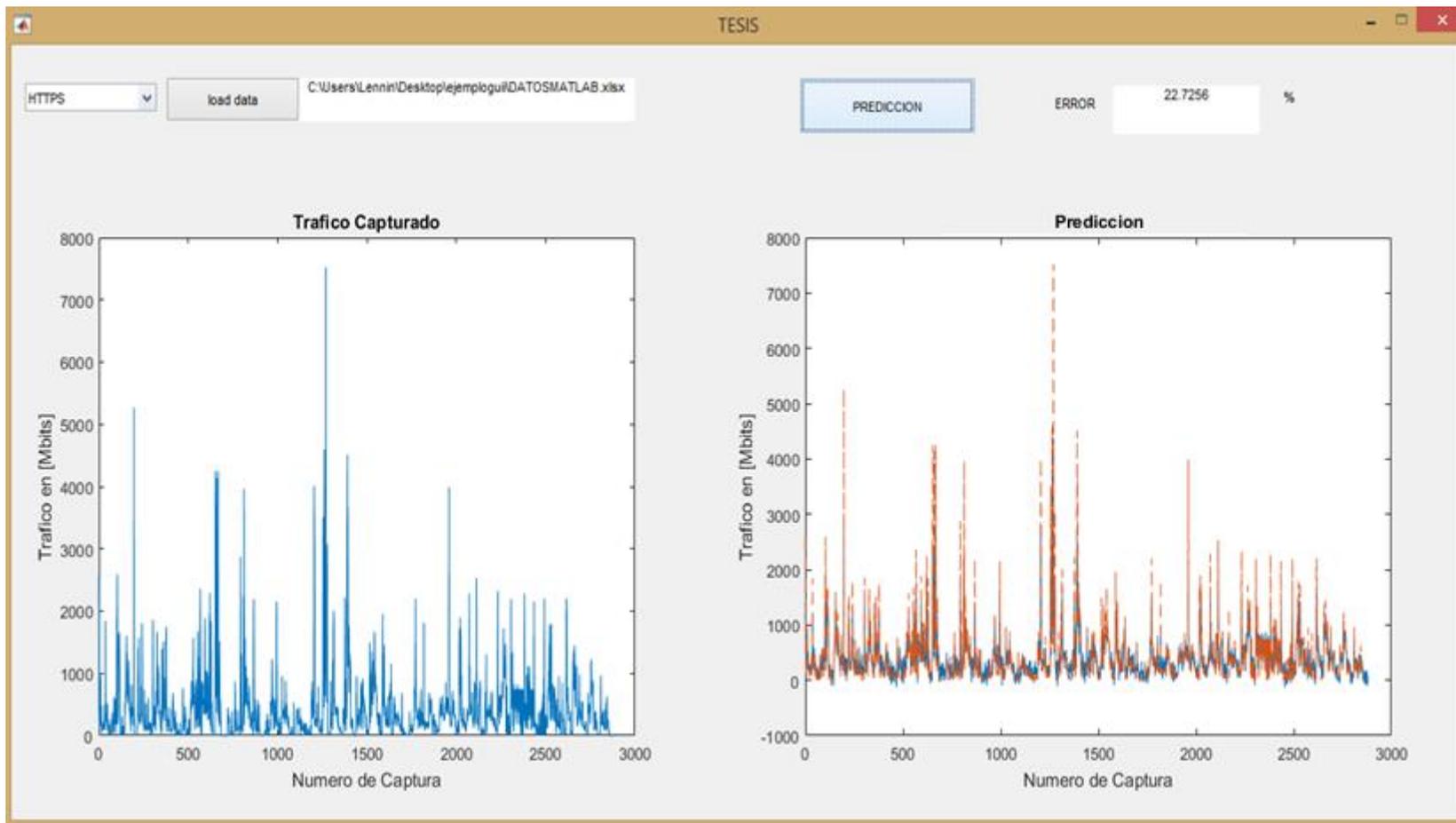


**Figura 4-3: Simulación Primera Prueba Protocolo HTTP**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018

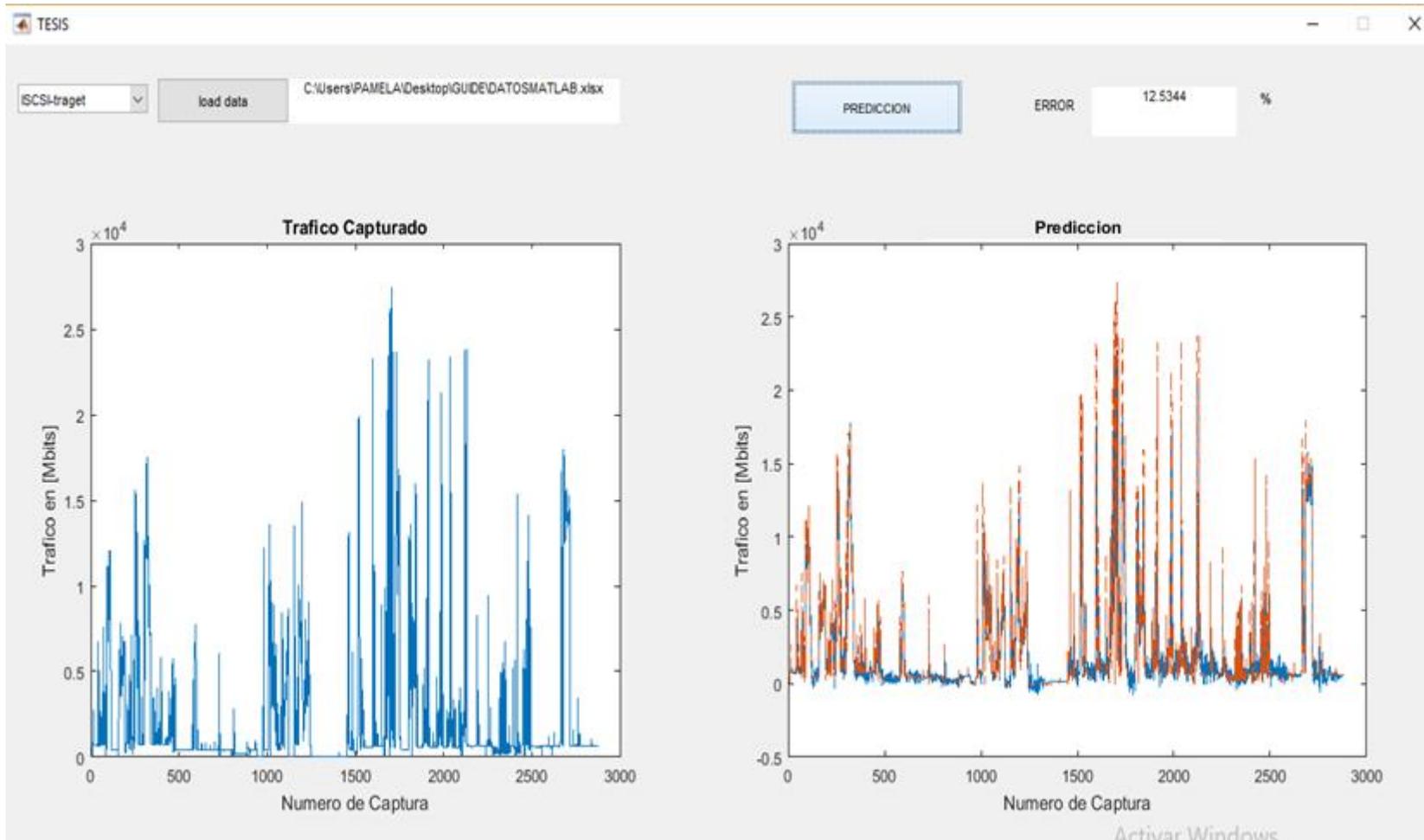
En la figura 5.3 se observa la simulación del protocolo HTTPS, la cual muestra el tráfico capturado y el tráfico a predecir, con una red neuronal artificial cuyos parámetros se encuentran en la tabla 2.3.

Después de analizar las 2 graficas obtenidas se pudo concluir que se tiene una predicción con un error 22.72 %.



**Figura 5-3: Simulación Primera Prueba Protocolo HTTPS**  
Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018

En la figura 6.3 se observa la simulación del protocolo ICSSI-TARGET, obteniendo una gráfica del tráfico real y del simulado, con una red neuronal artificial cuyos parámetros se encuentran en la tabla 2.3. Analizando las 2 graficas se pudo concluir que este protocolo tiene una predicción muy cercana a la real obteniendo un error de 12.53%.



**Figura 6-3: Simulación Primera Prueba Protocolo ISCSI-TARGET**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018.

A continuación, se presenta en la tabla 3.3 un resumen del error que se obtuvo en cada uno de los protocolos, con los parámetros establecidos en la tabla 3-3.

**Tabla 3-3: Resumen del Error de la Primera Prueba**

<b>PROTOCOLO</b>	<b>ERROR</b>
<b>IP</b>	10.6249%
<b>IPV6</b>	27.7896%
<b>ARP</b>	13,0158%
<b>HTTP</b>	28,9323%
<b>HTTPS</b>	22.7256%
<b>ISCIS-TARGET</b>	12.5344%

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

Después de realizar un análisis detallado con los parámetros anteriormente mencionados se tiene como resultados que con estos parámetros no se obtendrá una predicción correcta por lo que se procede a realizar una segunda prueba aumentando el número de neuronas de entrada, intermedia y su única salida.

### **3.1.2. Simulación de la segunda prueba**

En esta prueba los parámetros utilizados fueron 200 capas de entradas, 14 neuronas intermedias y 1 capa de salida, donde se analizó el error en cada protocolo, los parámetros se detallan en la tabla 4-3.

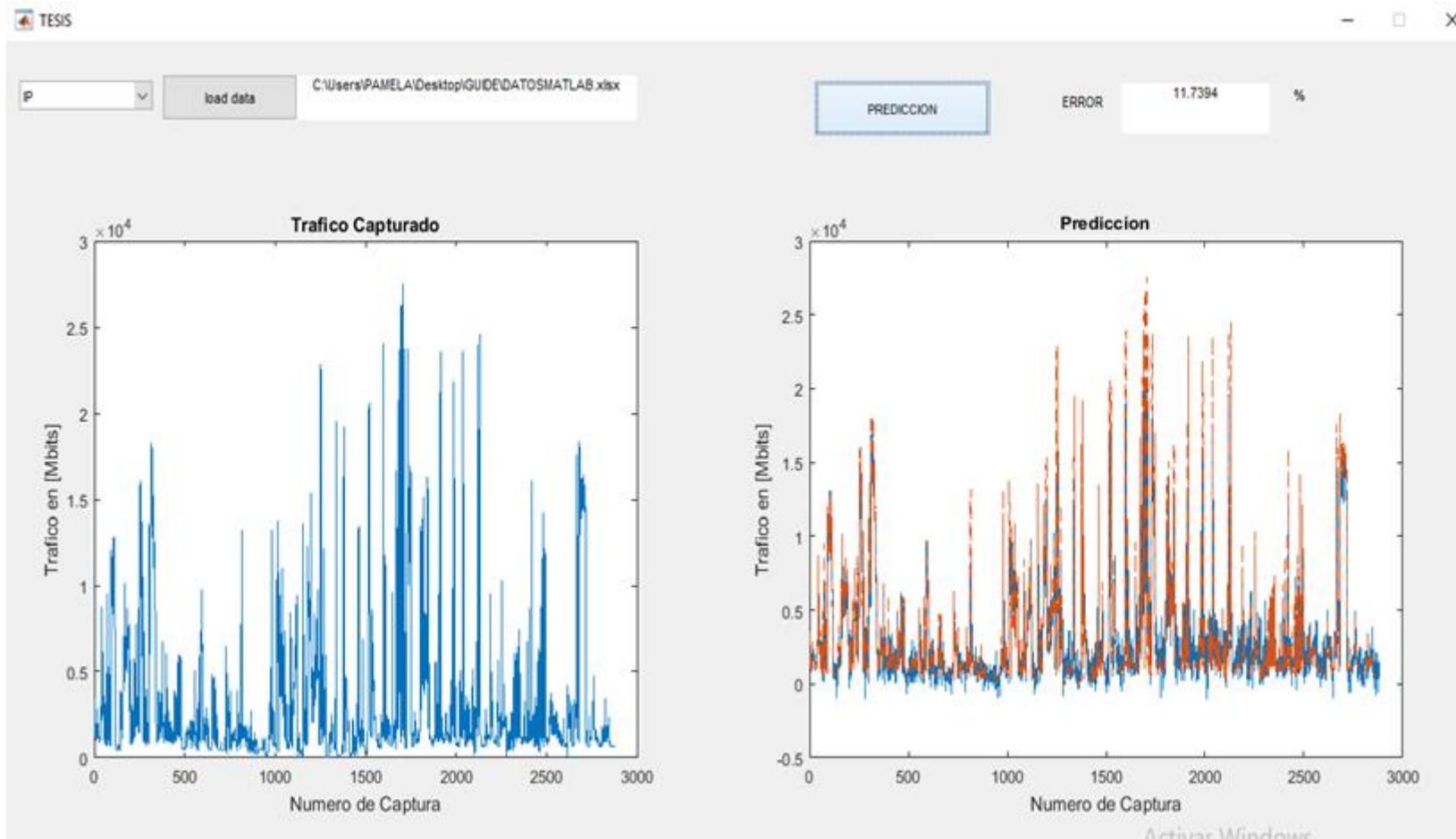
**Tabla 4-3: Parámetros de la Segunda Prueba**

<b>PARÁMETROS</b>	<b>ENTRADA</b>	<b>OCULTA</b>	<b>SALIDA</b>
<b>NUMERO DE NEURONAS</b>	200	14	1
<b>FUNCIÓN DE TRANSFERENCIA</b>	tansig	tansig	tansig
<b>FUNCIÓN DE ENTRENAMIENTO</b>	Traingd		

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

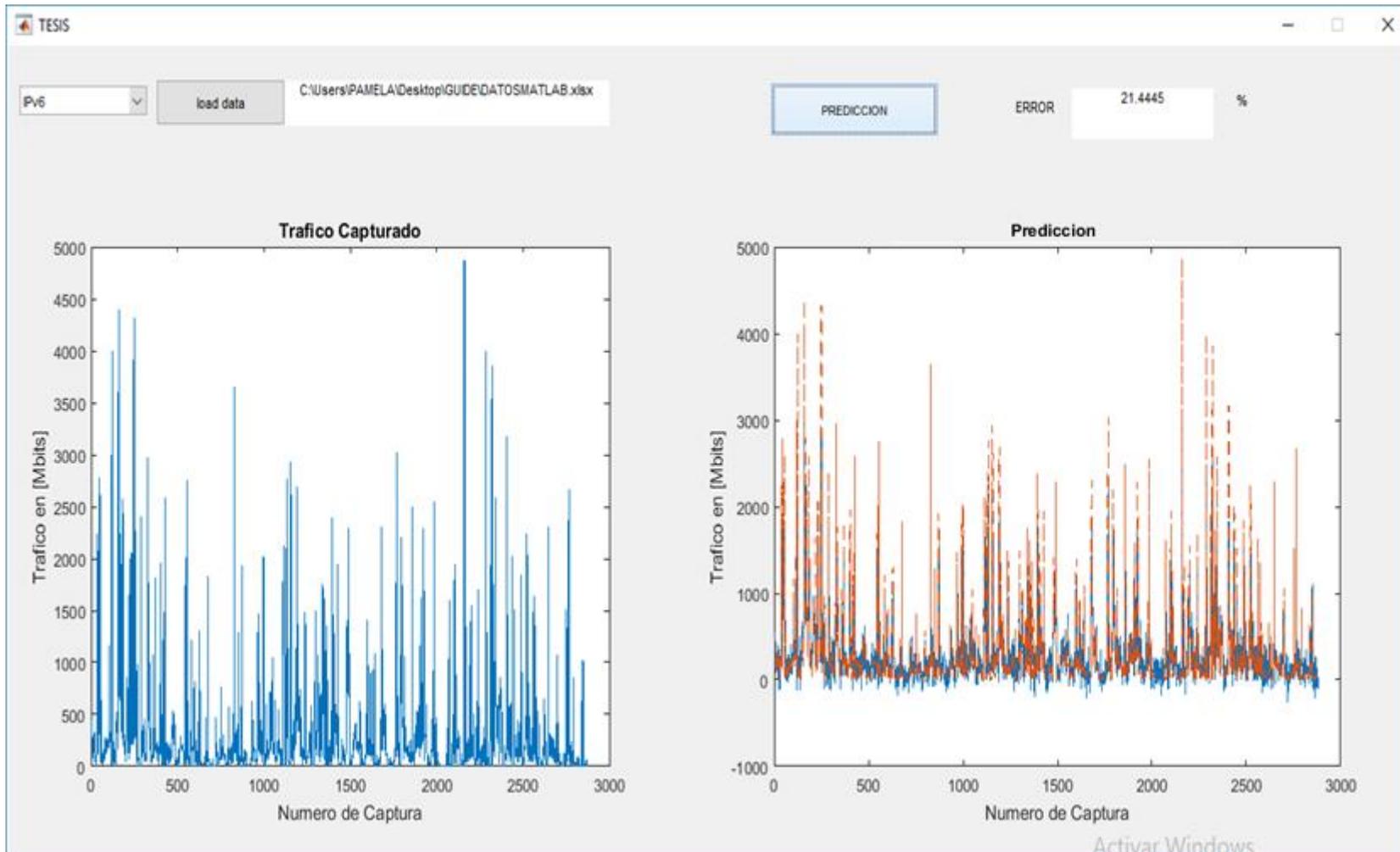
En la figura 7.3 se observa la simulación del protocolo IP, obteniendo una gráfica del tráfico real y del simulado con una red neuronal artificial cuyos parámetros se encuentran en la tabla 4.3.

Después de analizar las 2 graficas obtenidas se pudo concluir que se tiene una predicción con un error 11.73 %, que en comparación con la primera prueba aumento el error un 1.11%.



**Figura7-3: Simulación de la Segunda Prueba Protocolo IP**  
 Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018.

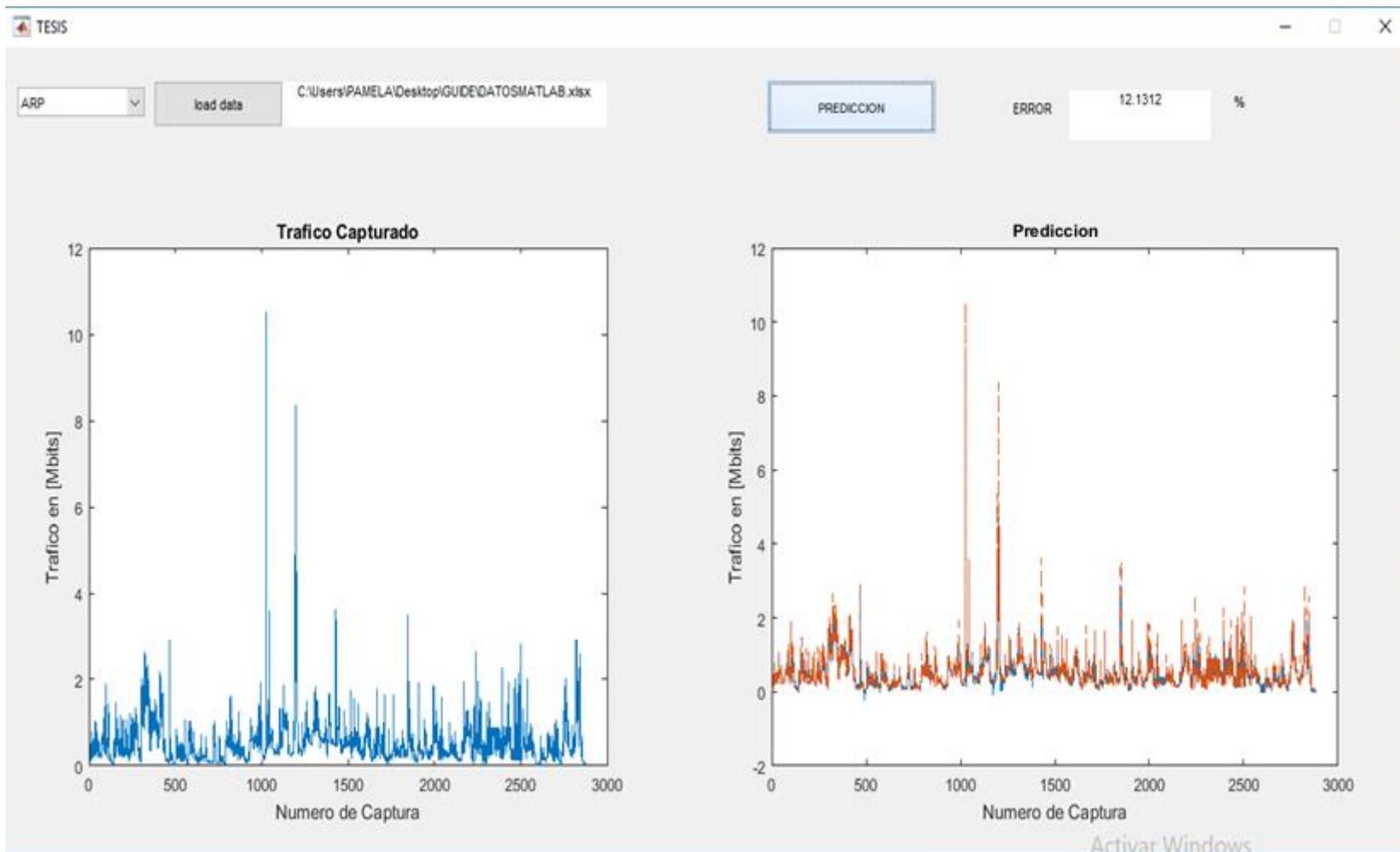
En la figura 8.3 se observa la simulación del protocolo IPv6, después de realizar el entrenamiento con los datos de 4 semanas dentro de la red neuronal artificial cuyos parámetros se encuentran en la tabla 4.3. Posterior al análisis de las 2 graficas obtenidas se pudo concluir que se tiene una predicción con un error del 21.44%, comprando los resultados de la primera prueba tenemos un descenso en el error de 6.34%.



**Figura8-3: Simulación de la Segunda Prueba Protocolo IPV6**

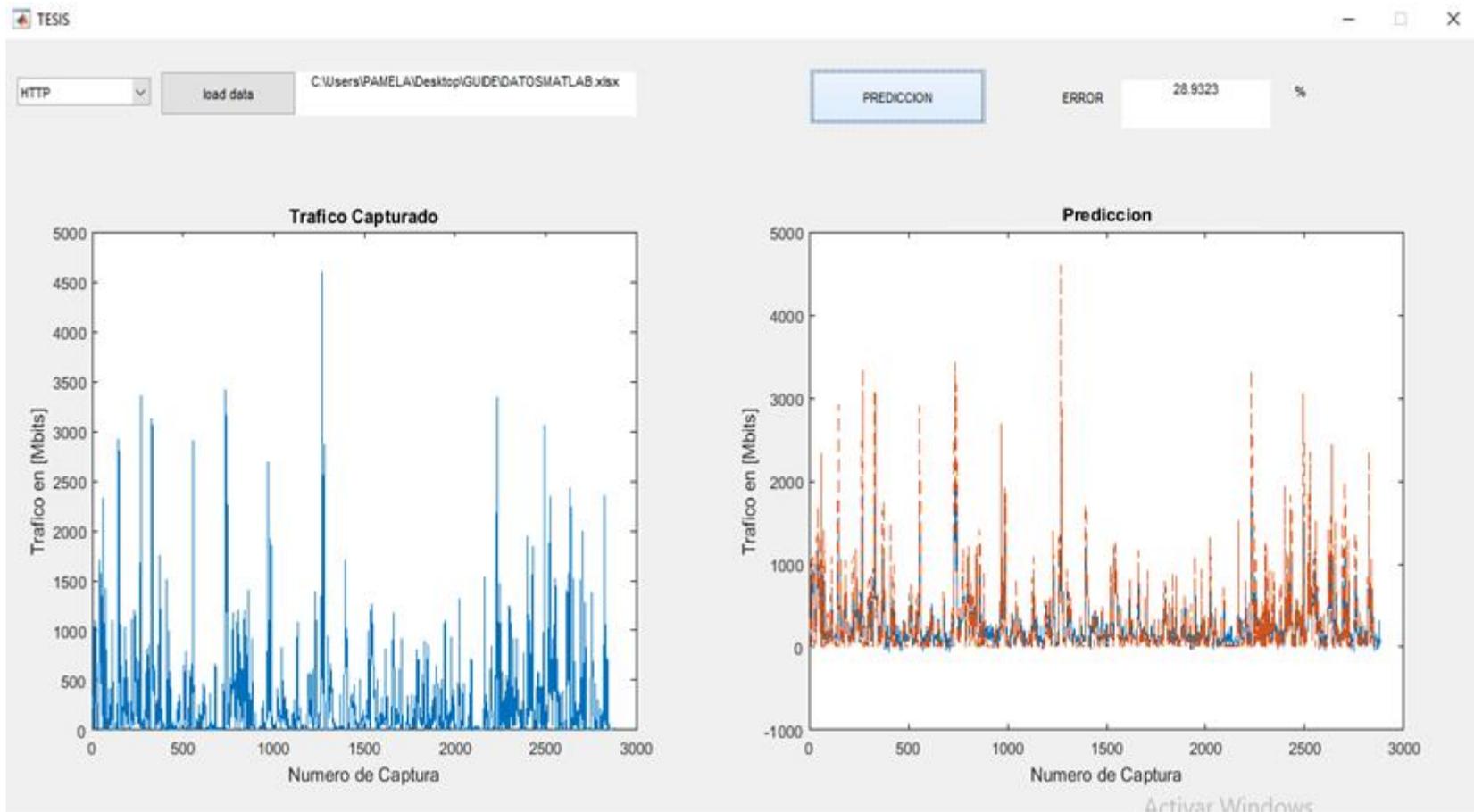
Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018

En la figura 9.3 se observa la simulación del protocolo ARP, obteniendo una gráfica del tráfico real y del tráfico simulado, con una red neuronal artificial cuyos parámetros se encuentran en la tabla 4.3. Después de analizar las 2 graficas se pudo concluir que se tiene una predicción muy cercana a la real obteniendo un error de 12.13%, relacionando estos resultados con los de la primera prueba el error bajo un 0.88%.



**Figura9-3: Simulación de la Segunda Prueba Protocolo ARP**  
Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018

En la figura 10.3 se observa la simulación del protocolo HTTP, después de realizar el entrenamiento con los datos de 4 semanas dentro de la red neuronal artificial cuyos parámetros se encuentran en la tabla 4.3. Posterior al análisis de las 2 graficas obtenidas se pudo concluir que se tiene una predicción con un error del 21.24%, examinado con la simulación de la primera prueba vemos que error bajo un 7.69%.

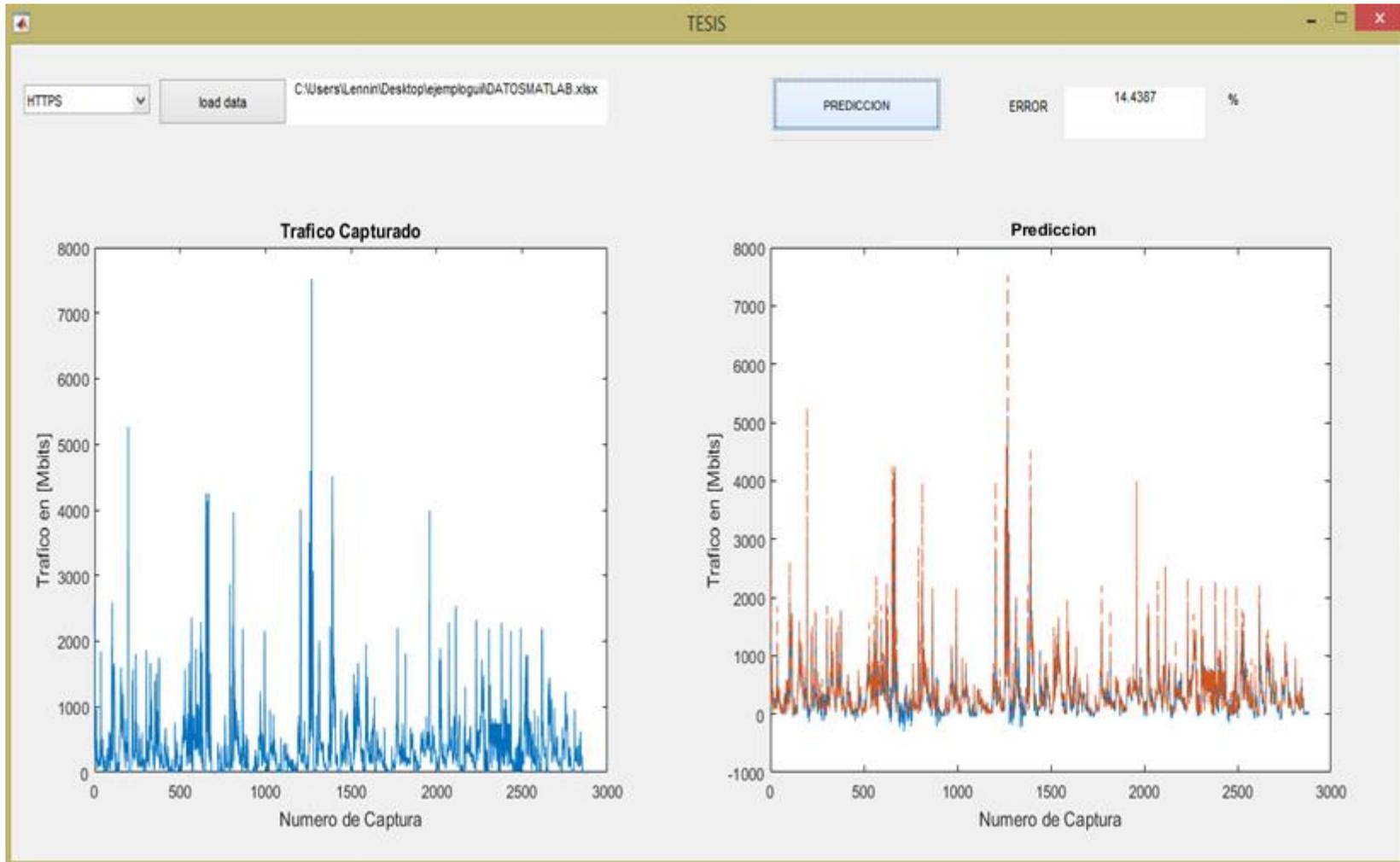


**Figura10-3: Simulación de la Segunda Prueba Protocolo HTTP**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018.

En la figura 11.3 se observa la simulación del protocolo HTTPS, la cual muestra el tráfico capturado y el tráfico a predecir, con una red neuronal artificial cuyos parámetros se encuentran en la tabla 4.3.

Después de analizar las 2 graficas obtenidas se pudo concluir que se tiene una predicción con un error 14.43 %, comparando con los resultados dela primera prueba tenemos un descenso en el error del 8.29%

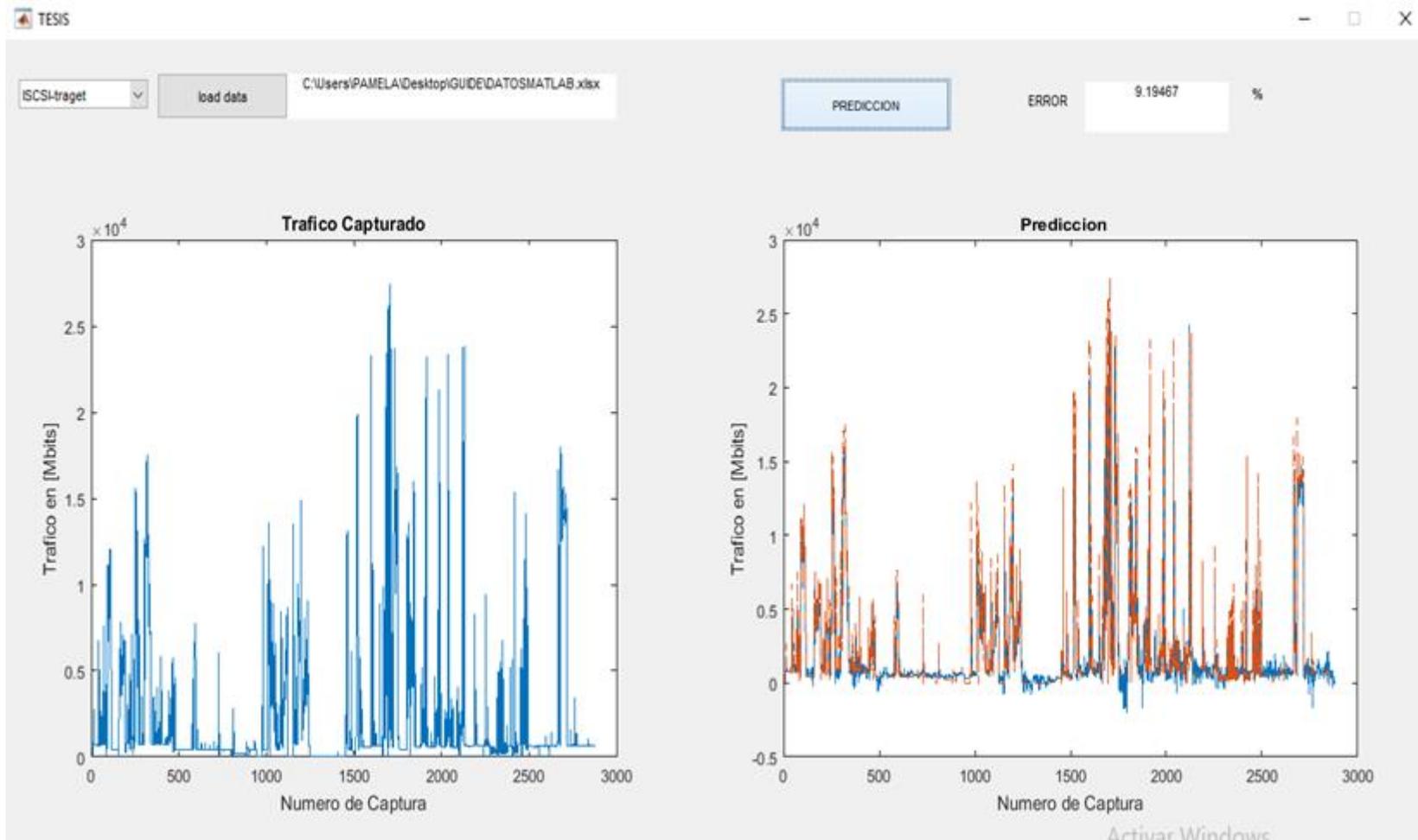


**Figura11-3: Simulación de la Segunda Prueba Protocolo HTTPS**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018.

En la figura 12.3 se observa la simulación del protocolo ICSSI-TARGET, obteniendo una gráfica del tráfico real y del simulado, con una red neuronal artificial cuyos parámetros se encuentran en la tabla 4.3.

Analizando las 2 graficas se pudo concluir que este protocolo tiene una predicción muy cercana a la real obteniendo un error de 9.19%, en comparación con la primera prueba tenemos un descenso error del 3.34%.



**Figura12-3: Simulación de la Segunda Prueba Protocolo ISCSI-TARGET**  
 Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018

A continuación, se presenta en la tabla 5-3 un resumen del error que se obtuvo en cada uno de los protocolos, con los parámetros establecidos en la tabla 4.3.

**Tabla 5-3: Error de la Segunda Prueba**

<b>PROTOCOLO</b>	<b>ERROR</b>
<b>IP</b>	11.7394%
<b>IPV6</b>	21.4445%
<b>ARP</b>	12.1312%
<b>HTTP</b>	21.2489%
<b>HTTPS</b>	14.4387%
<b>ISCIS-TARGET</b>	9.19467%

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

Después de analizar detalladamente los parámetros anteriormente mencionados se tienen como resultados un descenso del error comparando la primera y la segunda prueba con sus respectivos parámetros, pero la predicción aún sigue teniendo un nivel medianamente bajo de error por lo que se procede a realizar una tercera prueba aumentando el número de neuronas de entrada, intermedia y su única salida.

### 3.1.3. Simulación de la tercera prueba

En esta prueba los parámetros utilizados fueron 300 capas de entradas, 17 neuronas intermedias y 1 capa de salida, donde se analizará el error de cada protocolo, los parámetros se pueden observar en la tabla 6-3.

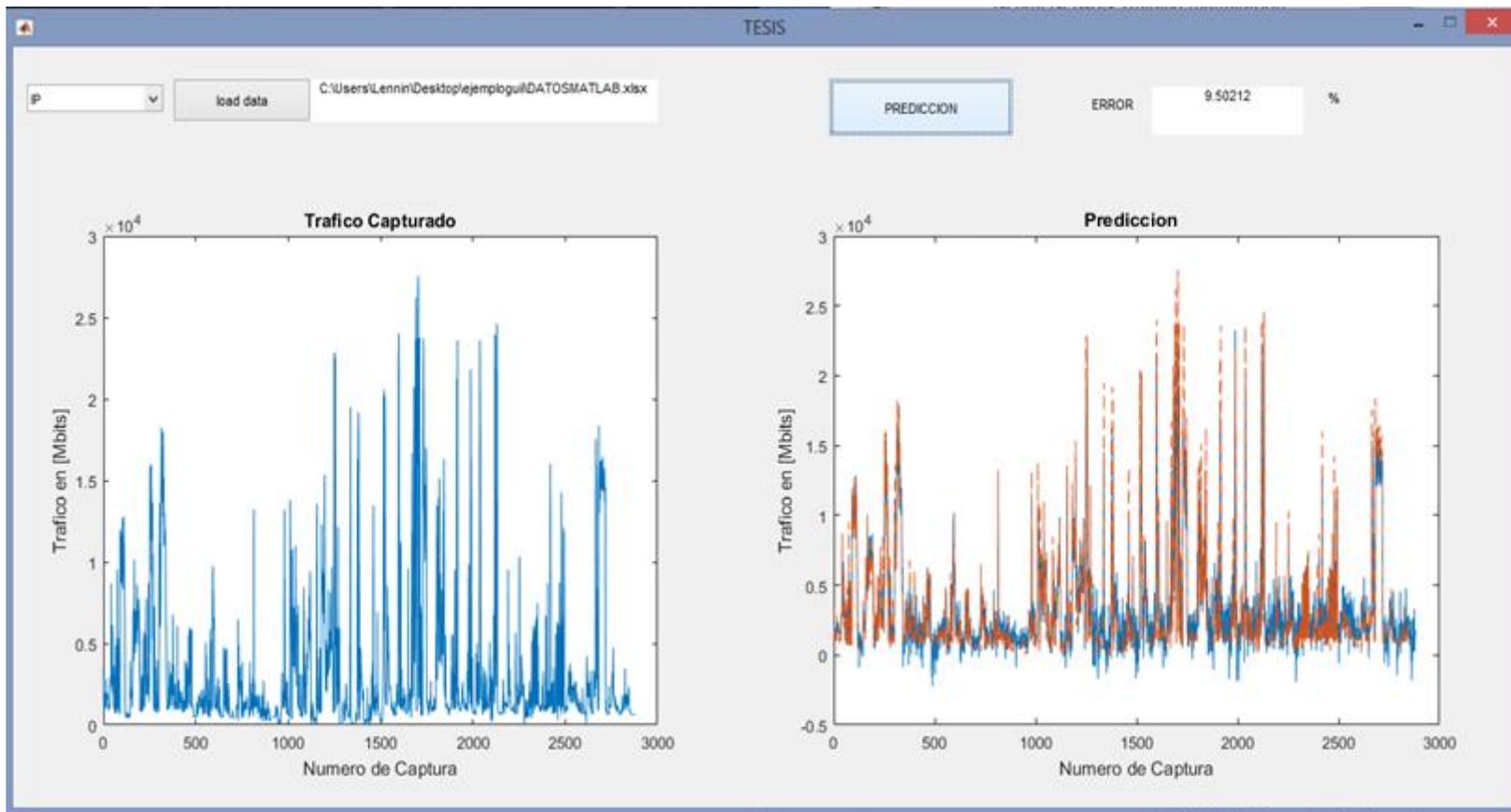
**Tabla 6-3: Parámetros de la Tercera Prueba**

<b>PARÁMETRO</b>	<b>ENTRADA</b>	<b>OCULTA</b>	<b>SALIDA</b>
<b>NÚMERO DE NEURONAS</b>	300	17	1
<b>FUNCIÓN DE TRANSFERENCIA</b>	tansig	tansig	tansig
<b>FUNCIÓN DE ENTRENAMIENTO</b>	Traingd		

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

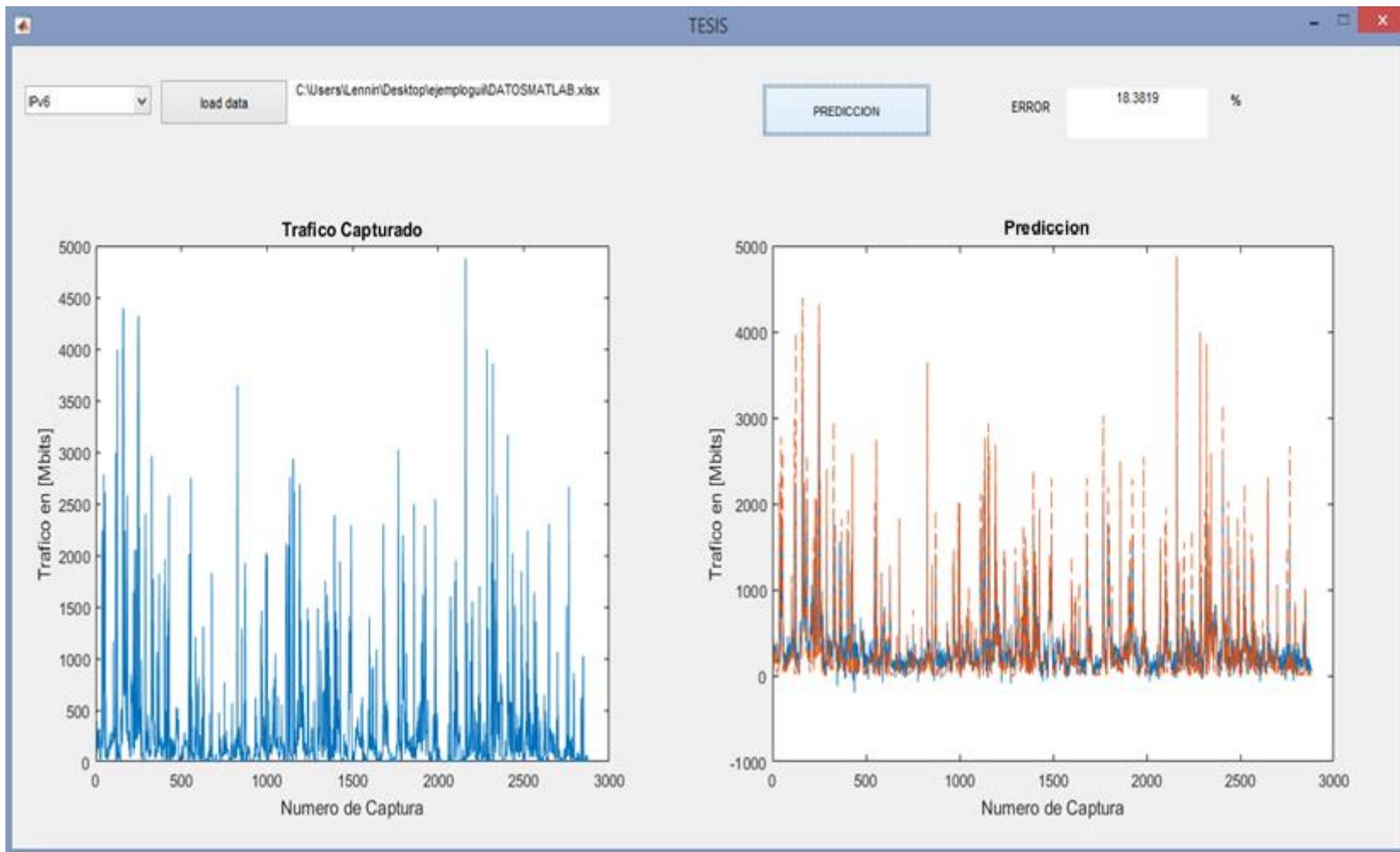
En la figura 13.3 se observa la simulación del protocolo IP, obteniendo una gráfica del tráfico real y del simulado con una red neuronal artificial cuyos parámetros se encuentran en la tabla 6.3.

Después de analizar las 2 graficas obtenidas se pudo concluir que se tiene una predicción con un error 9.5%, que en comparación con la primera prueba decreció error un 1.12% y con la segunda un error 2.23%



**Figura 13-3: Simulación de la Tercera Prueba Protocolo IP**  
 Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018.

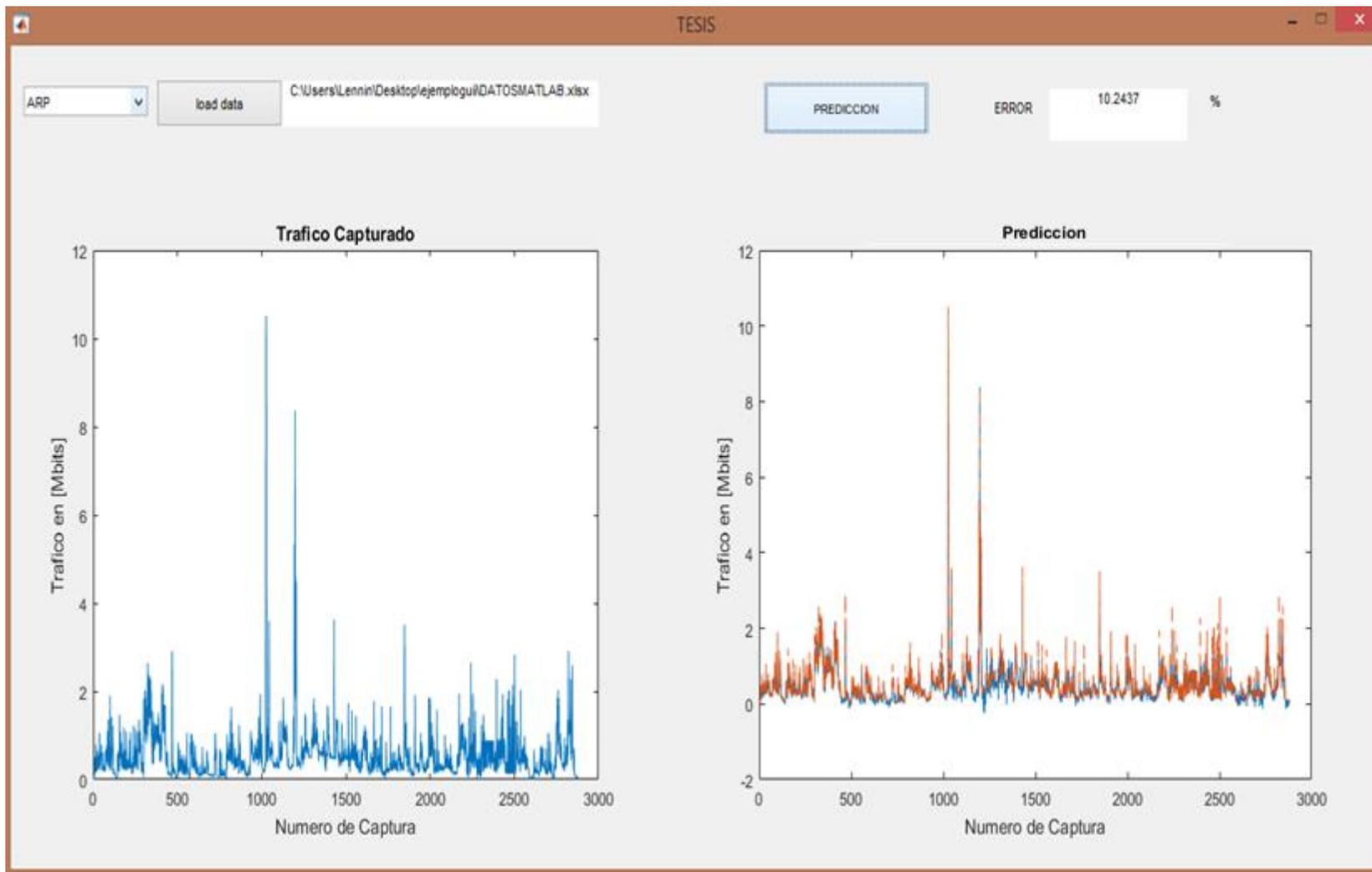
En la figura 14.3 se observa la simulación del protocolo IPv6, después de realizar el entrenamiento con los datos de 4 semanas dentro de la red neuronal artificial cuyos parámetros se encuentran en la tabla 6.3. Posterior al análisis de las 2 graficas obtenidas se pudo concluir que se tiene una predicción con un error del 15.38%, comprobando los resultados de la primera prueba tenemos un descenso en el error de 12.4% y con la segunda prueba un error 6.06%.



**Figura 14-3: Simulación de la Tercera Prueba Protocolo IPV6**  
 Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018.

En la figura 15.3 se observa la simulación del protocolo ARP, obteniendo una gráfica del tráfico real y del tráfico simulado, con una red neuronal artificial cuyos parámetros se encuentran en la tabla 6.3.

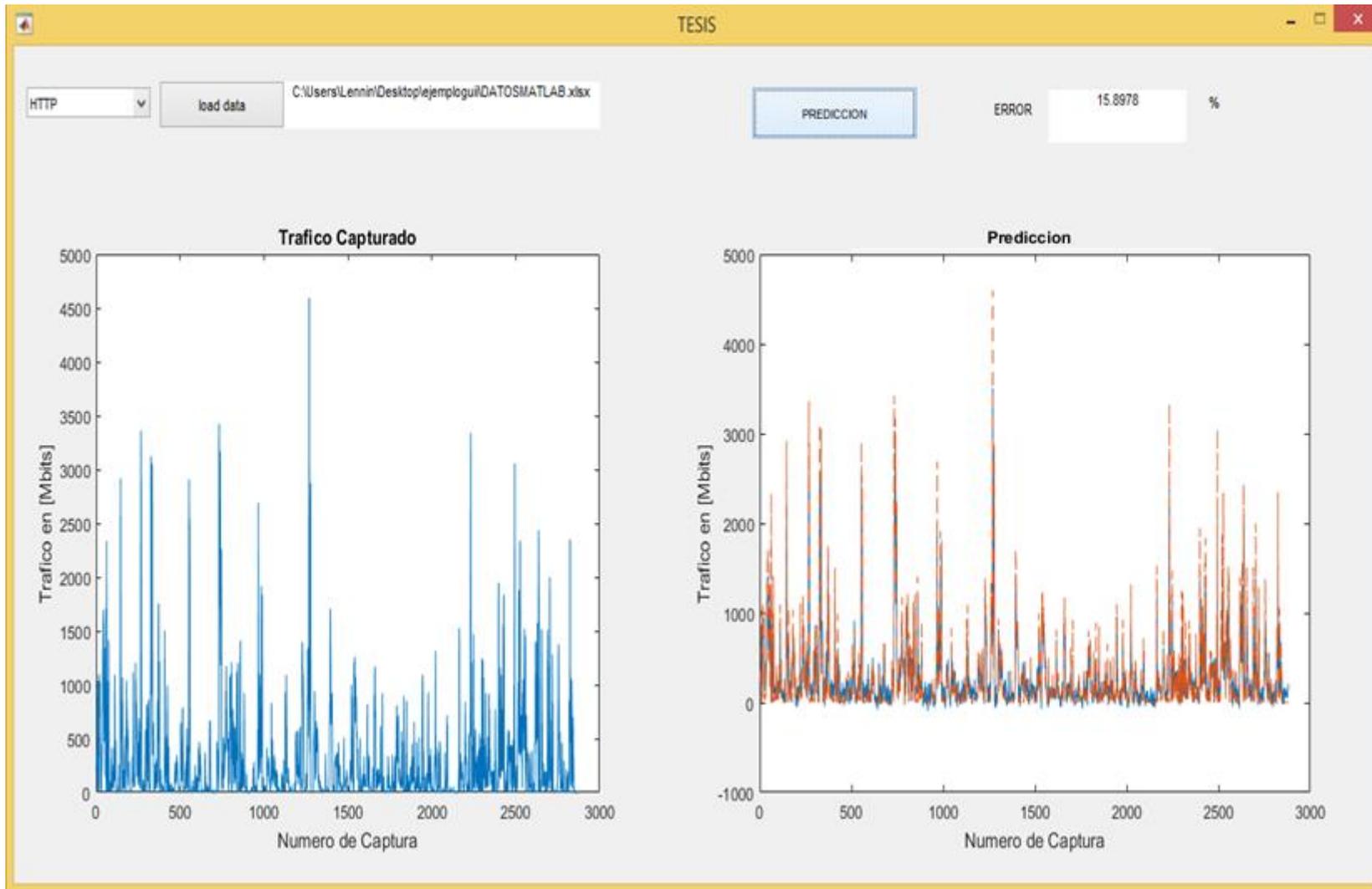
Después de analizar las 2 graficas se pudo concluir que se tiene una predicción muy cercana a la real obteniendo un error de 10.24%, relacionando estos resultados con los de la primera prueba el error bajo un 2.77% y con la segunda prueba un error de 1.89%



**Figura 15-3: Simulación de la Tercera Prueba Protocolo ARP**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018

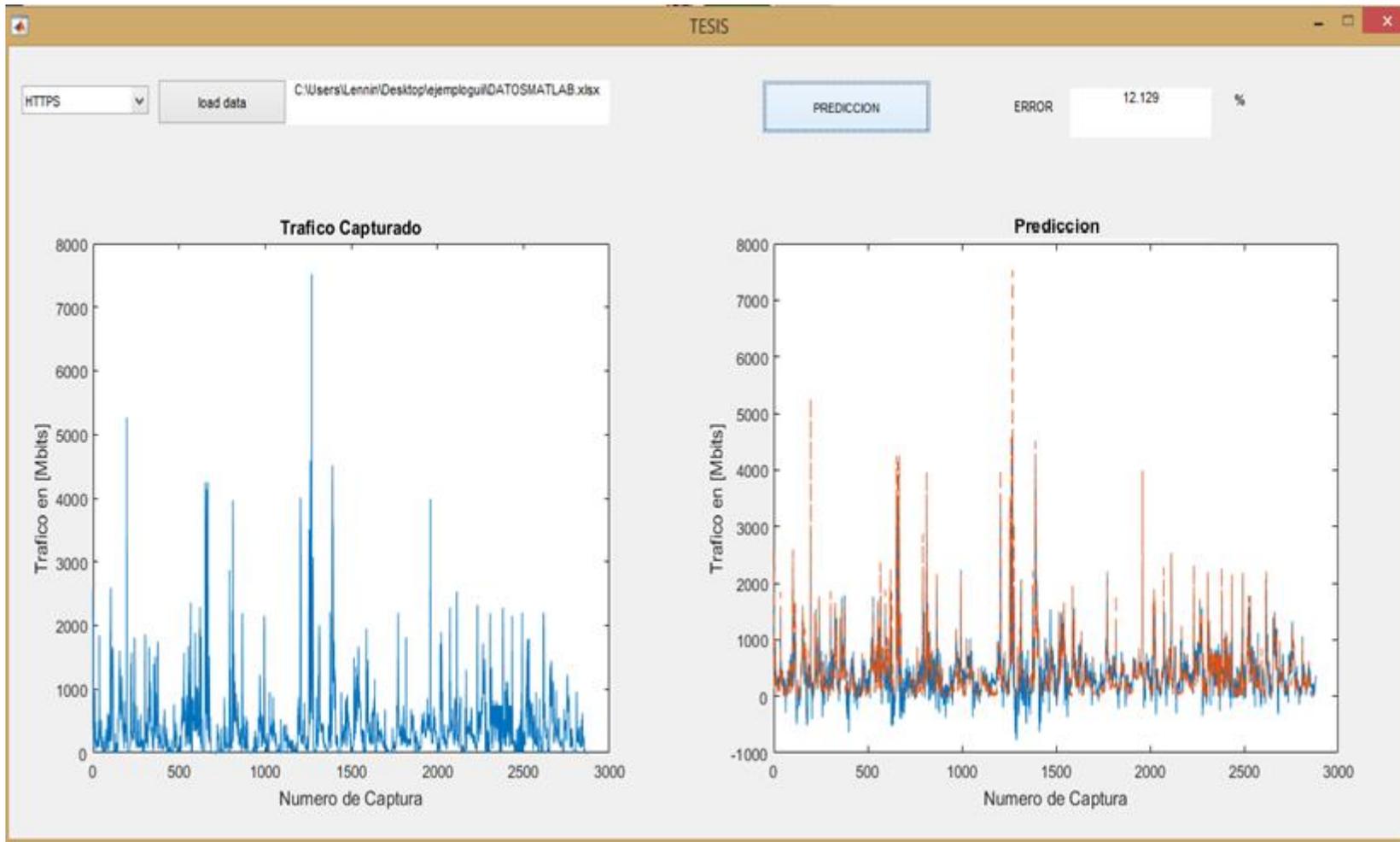
En la figura 16.3 se observa la simulación del protocolo HTTP, después de realizar el entrenamiento con los datos de 4 semanas dentro de la red neuronal artificial cuyos parámetros se encuentran en la tabla 6.3. Posterior al análisis de las 2 graficas obtenidas se pudo concluir que se tiene una predicción con un error del 15.89%, examinado con la simulación de la primera prueba vemos que error bajo un 13.55% y con la segunda prueba un error de 5.35%.



**Figura 16-3: Simulación de la Tercera Prueba Protocolo HTTP**  
 Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018

En la figura 17.3 se observa la simulación del protocolo HTTPS, la cual muestra el tráfico capturado y el tráfico a predecir, con una red neuronal artificial cuyos parámetros se encuentran en la tabla 6.3.

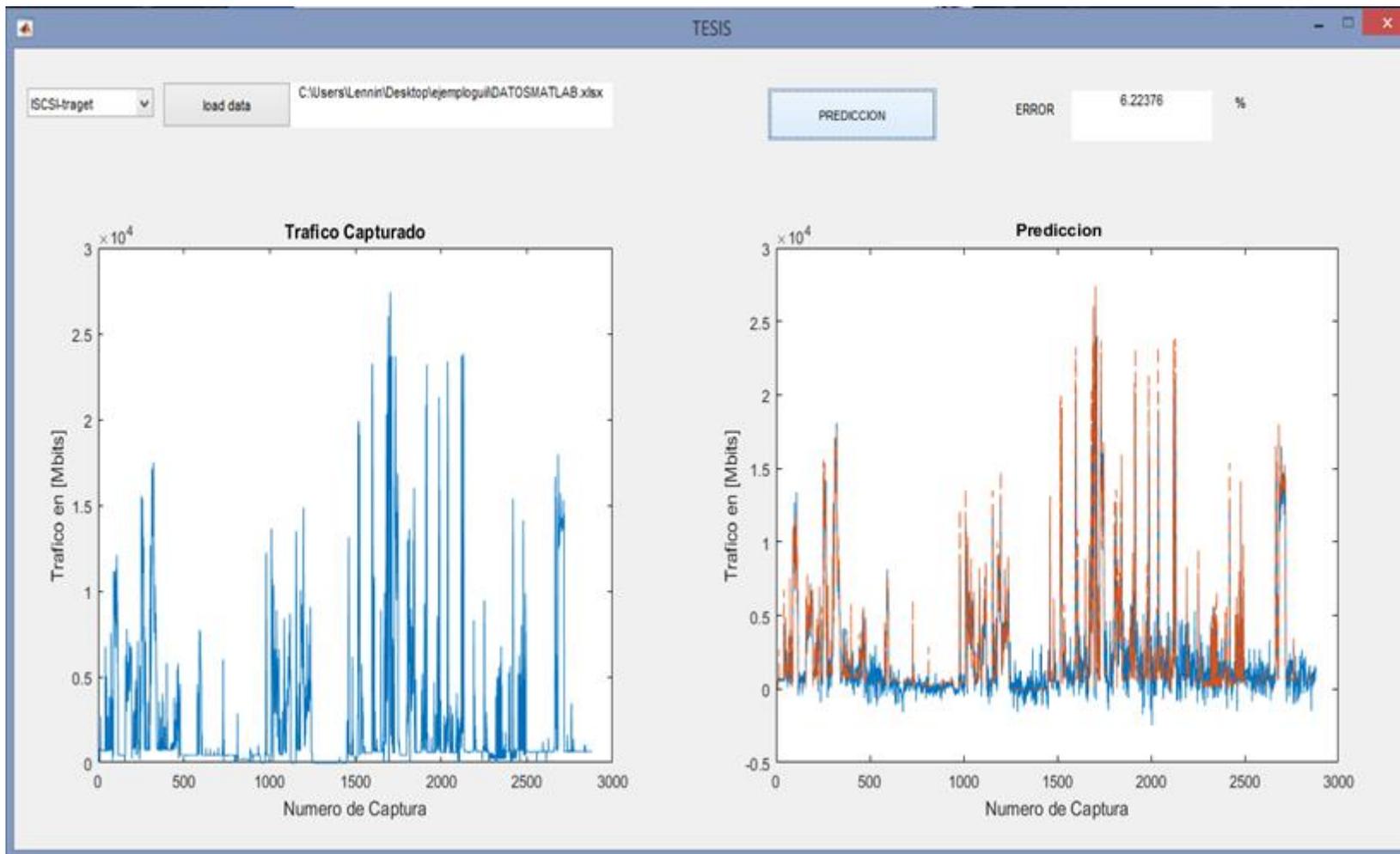
Después de analizar las 2 graficas obtenidas se pudo concluir que se tiene una predicción con un error 12.12%, comparando con los resultados de la primera prueba tenemos un descenso en el error del 10.6% y con la segunda prueba un error de 2.31%.



**Figura 17-3: Simulación de la Tercera Prueba Protocolo HTTPS**  
Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018.

En la figura 18.3 se observa la simulación del protocolo ICSSI-TARGET, obteniendo una gráfica del tráfico real y del simulado, con una red neuronal artificial cuyos parámetros se encuentran en la tabla 6.3.

Analizando las 2 graficas se pudo concluir que este protocolo tiene una predicción muy cercana a la real obteniendo un error de 6.22%, en comparación con la primera prueba tenemos un descenso error del 6.31% y con la segunda con un error de 2.97%



**Figura 18-3: Simulación de la Tercera Prueba Protocolo ISCSI-TARGET**  
Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2018.

A continuación, se presenta en la tabla 7-3 un resumen del error que se obtuvo en cada uno de los protocolos, con los parámetros establecidos en la tabla 6.3.

**Tabla 7-3: Error de la Tercera Prueba**

<b>PROTOCOLO</b>	<b>ERROR</b>
<b>IP</b>	9.50212%
<b>IPV6</b>	15.3819%
<b>ARP</b>	10.2437%
<b>HTTP</b>	15.8978%
<b>HTTPS</b>	12.129%
<b>ISCSI-TARGET</b>	6.22376%

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

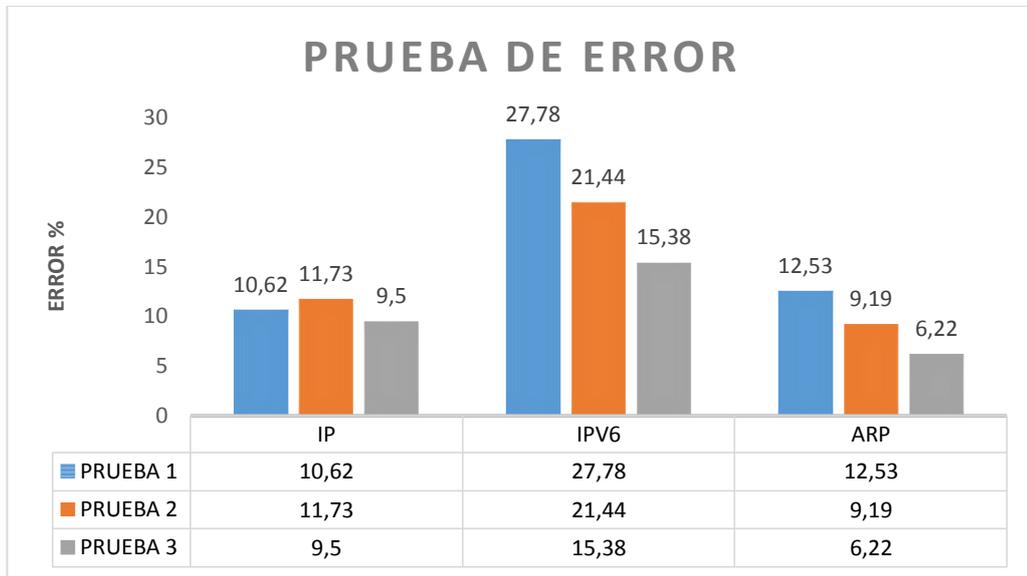
Posteriormente analizando detalladamente los parámetros anteriormente mencionados se tienen como resultados un descenso del error analizando la primera, la segunda y la tercera prueba, cada uno con sus respectivos parámetros tal como se observa en la tabla 8-3.

**Tabla 8-3: Errores de las Pruebas**

<b>PROTOCOLO</b>	<b>PARÁMETROS PRUEBAS</b>	<b>ENTRADA</b>	<b>INTERMEDIA</b>	<b>SALIDA</b>	<b>ERROR</b>
<b>IP</b>	1	100	10	1	10.6249
	2	200	14	1	11.7394
	3	300	17	1	9.50212
<b>IPV6</b>	1	100	10	1	27.7896
	2	200	14	1	21.4445
	3	300	17	1	15.3819
<b>ARP</b>	1	100	10	1	13,0158
	2	200	14	1	12.1312
	3	300	17	1	10.2437
<b>HTTP</b>	1	100	10	1	28,9323
	2	200	14	1	21.2489
	3	300	17	1	15.8978
<b>HTTPS</b>	1	100	10	1	22.7256
	2	200	14	1	14.4387
	3	300	17	1	12.129
<b>ISCSI-TARGET</b>	1	100	10	1	12.5344
	2	200	14	1	9.19467
	3	300	17	1	6.22376

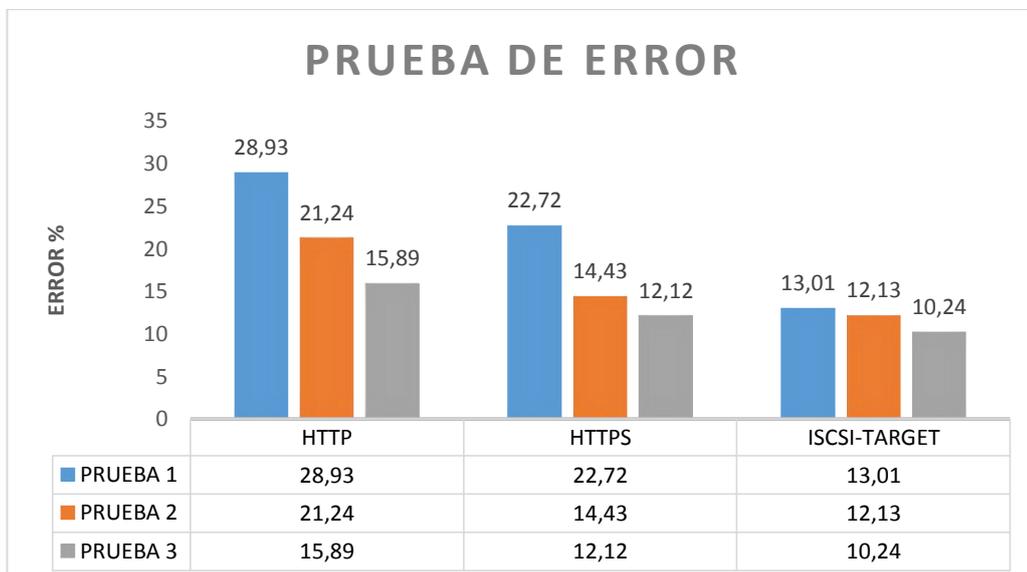
Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

Como se puede observar en el gráfico 19.3 y 20.3, en la tercera prueba los niveles de error en cada protocolo son bajos con respecto a la primera y segunda prueba, debido a que este diseño de red neuronal artificial puede ser el más óptimo y estable para la predicción de tráfico.



**Gráfico1-3: Comparación de Error en Protocolos IP, IPV6, ARP**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.



**Gráfico2-3: Comparación de Error en Protocolos HTTP, HTTPS, ISCSI-TARGET**

Realizado por: BARRAZUETA, Pamela & TIERRA, Lennin, 2017.

Luego del análisis de los resultados se determinó que el impacto en la gestión de la información de la comunicación de Redes Corporativas es positivo ya que presenta grandes beneficios entre ellos la planificación en varios aspectos como:

Físicos (hardware): La creación de una red debe ser organizada y rentable, por ello se debe determinar algunos factores para obtener esta organización los cuales son: planificación de los tipos de equipos que se necesitaran y su distribución en el sitio donde serán instalados, tipo de equipamientos que se necesitara entre otros.

Soporte (software): Dentro de los factores a tomar en cuenta en la parte lógica de la red son los siguientes: determinar el número de direcciones IP que se necesitara para una red determinada, determinar subdivisiones administrativas dentro de la red entre otros factores.

Económico: Comprobar si la empresa está preparada para la adquisición de nuevos equipos los que permitirán a la red tener soporte a las nuevas tecnologías y el incremento de los usuarios en el caso de darse.

Todo ello permitirá conocer a los administradores de red a futuro, los factores anteriormente mencionados y así comprobar el correcto funcionamiento de la red, evitando colapsos y saturaciones en la red, además de ello les permitirá reconocer si la red soportara nuevas tecnologías que se presentaran más adelante.

## CONCLUSIONES

- Después de realizar un estudio minucioso de las Redes Neuronales Artificiales, la mejor opción para la predicción de datos de la red corporativa de la FIE con un mínimo error es el algoritmo backpropagation.
- Basado en el cálculo de la pirámide geométrica los parámetros para la implementación de la Red Neuronal Artificial con el algoritmo backpropagation fueron: 300 neuronas de entrada, 17 neuronas en una única capa oculta y una única capa de salida para esta red específicamente.
- La recolección de los datos fue realizada durante 4 semanas desde 16 de octubre hasta 17 de noviembre del 2017, en los horarios de 8 am a 8 pm, obteniendo un total de 3312 datos de los cuales tomamos una muestra de 2880.
- En el tráfico de las capturas se observó, que en cada una de ellas se encontraban 50 protocolos, de los cuales se seleccionó seis de ellos, de los cuales 3 de ellos pertenecen a la capa de red (IP, IPV6, ARP) y los otros 3 en la capa de aplicación (HTTP, HTTPS Y ISCSI- TARGET), en conjunto representan al tráfico total.
- Luego de realizar las pruebas se obtuvo un error diferente para cada uno de los protocolos los cuales de detallan a continuación para el protocolo IP el 9.50%, IPV6 el 15.38%, ARP el 10.24%, HTTP el 15.89%, HTTPS 12.12% y ISCSI- TARGET el 6.22%.

## RECOMENDACIONES

- Se recomienda que para la predicción de datos en redes corporativas específicamente la red de la FIE utilizar redes neuronales que tengan características cuyas entradas sean mayores a 300 neuronas en la capa de entrada, una única capa oculta con 17 o más neuronas intermedias y una única salida
- Otra técnica para realizar la predicción de datos podría ser las siguientes Machine Learning y Deep Learning entre otras las cuales permitirán realizar de una manera más rápida el procesamiento de los datos.
- Otra propuesta de investigación específicamente para la predicción de datos será la implementación con otras técnicas de inteligencia artificial como las anteriormente mencionadas para hacer comparaciones entre los resultados arrojados por estos y nuestra investigación.
- Finalmente se recomienda evaluar otros parámetros que influyen en el rendimiento RNA's, como la función de activación, número de capas ocultas, número de iteraciones, tasa de aprendizaje, y otros no abarcados por esta tesis

## **BIBLIOGRAFÍA**

**ASQUISANTILLAN, G**, Predicción del nivel de agua del embalse, basado en redes neuronales, para la mejora de la planificación de producción de energía en la Central Hidroeléctrica Agoyán (Tesis) (Maestría) [en línea]. 2017 Riobamba: Escuela Superior Politécnica de Chimborazo. [Consulta: 1 septiembre 2017]. Disponible en: <http://dspace.espech.edu.ec/handle/123456789/6159>

**BALLESTEROS, A**, Neuronal Network Framework, Las Redes Monocapa [en línea]. 2005. [Consulta: 7 agosto 2017]. Disponible en: <http://www.redes-neuronales.com.es/tutorial-redes-neuronales/Las-redes-neuronales-monoCapa.html>.

**BASOGAIN OLABE, X**, Redes Neuronales Artificiales Y Sus Aplicaciones [en línea]. 2005. [Consulta: 17 agosto 2017]. Disponible en: <http://www.sciencedirect.com/science/article/pii/S021056910574198X?via%3DIhub>.

**CORCUERA, PEDRO**, Creación de interfaces de usuario con MATLAB [en línea]. 2013. Cantabria. [Consulta: 18 agosto 2017]. Disponible en: [http://www.utm.mx/~vero0304/HCPM/GUI\\_Matlab.pdf](http://www.utm.mx/~vero0304/HCPM/GUI_Matlab.pdf)

**CZECH TECHNICAL UNIVERSITY**, newff (Neural Network Toolbox). Radio.feld.cvut.cz [en línea]. 2018. [Consulta: 15 agosto 2017]. Disponible en: <http://radio.feld.cvut.cz/matlab/toolbox/nnet/newff.html>

**FLÓREZ LÓPEZ, RAQUEL and FERNÁNDEZ FERNÁNDEZ, JOSÉ M**, Las redes neuronales artificiales. 2008. Oleiros, La Coruña: Netbiblo.

**GALÁN ASENCIO, H and MARTÍNEZ BOWEN, A**, xInteligencia artificial. Redes neuronales y Aplicaciones [en línea]. 2010. Madrid: UC3M-DIT. [Consulta: 21 agosto 2017]. Disponible en: <http://www.it.uc3m.es/jvillena/irc/practicas/10-11/06mem.pdf>.

**HAYKIN, S**, Neural Networks [desconectado]. 2001. 5. [Consulta: 22 agosto 2017]. Disponible en: <https://doi.org/10987654321>.

**IZAURIETA, F and SAAVEDRA, C**, Redes Neuronales Artificiales [en línea]. 1999. Concepción: UdeC-DF. [Consulta: 29 agosto 2018]. Disponible en: <http://vipo.uta.cl/charlas/volumen16/Indice/Ch-csaavedra.pdf>.

**JOSHI, M and HADI, T**, A Review of Network Traffic Analysis and Prediction Techniques [en línea]. 2015. Jalgaon: School of Computer Sciences, North Maharashtra University. [Consulta: 30 agosto 2017]. Disponible en: <https://arxiv.org/ftp/arxiv/papers/1507/1507.05722.pdf>.

**LORENZO FONSECA, I, MACIÁ PÉREZ, F, LAU FERNÁNDEZ, R, MORA GIMENO, F and GIL MARTÍNEZ-ABARCA, J**, Método para la Detección de Intrusos mediante Redes Neuronales basado en la Reducción de Características [en línea]. 2008. Alicante: GrupoM. [Consulta: 18 agosto 2017]. Disponible en: <http://rua.ua.es/dspace/handle/10045/15687>.

**MARINO, T**, Introducción a las Redes Neuronales [en línea]. 2011. Huejutla. [Consulta: 14 septiembre 2017]. Disponible en: [https://www.uaeh.edu.mx/docencia/P\\_Presentaciones/huejutla/sistemas/redes\\_neuronales/introduccion.pdf](https://www.uaeh.edu.mx/docencia/P_Presentaciones/huejutla/sistemas/redes_neuronales/introduccion.pdf).

**MATHWORK**, Gradient descent backpropagation - MATLAB traingd - MathWorks United Kingdom. La.mathworks.com [en línea]. 2018. [Consulta: 17 agosto 2017]. Disponible en: <https://la.mathworks.com/help/nnet/ref/traingd.html>

**MORENO RODRÍGUEZ, ALFONSO**, Desarrollo de una interfaz gráfica de redes neuronales usando Matlab [en línea]. 2009. Madrid. [Consulta: 16 agosto 2017]. Disponible en: <https://orff.uc3m.es/bitstream/handle/10016/8488/Proyecto%20Redes%20neuronales%20GUI.pdf;jsessionid=E684C4A2ECB7514123A9A66CB5A5E5D6?sequence=1>

**MUNOZ, R, CASTILLO, O and MELIN, P**, Optimization of fuzzy response integrators in modular neural networks with hierarchical genetic algorithms: The case of face, fingerprint and voice recognition. [en línea]. 2009. Berlín. [Consulta: 4 septiembre 2017]. Disponible en: [https://link.springer.com/chapter/10.1007/978-3-642-04514-1\\_7](https://link.springer.com/chapter/10.1007/978-3-642-04514-1_7).

**PAUCAR GUAMÁN, F**, Estudio de un sistema de prevención de intrusos basado en redes neuronales; Propuesta de diseño Unidad Educativa Brethren [en línea]. 2012. Quito. [Consulta: 7 septiembre 2017]. Disponible en: <http://repositorio.puce.edu.ec/handle/22000/12328>.

**PONCOS, M**, Matlab: ejemplo backpropagation. Mi rincocito [en línea]. 2018. [Consulta: 17 agosto 2017]. Disponible en: <https://poncos.wordpress.com/2009/01/15/matlab-ejemplo-backpropagation/>

**SAN DIEGO STATE UNIVERSITY**, sim (Neural Network Toolbox). Edoras.sdsu.edu [en línea]. 2018. [Consulta: 15 agosto 2017]. Disponible en: <https://edoras.sdsu.edu/doc/matlab/toolbox/nnet/sim.html>

**SERRANO, J, SORIA, E and MARTIN, J**, Redes Neuronales [en línea]. 2009. Valencia. [Consulta: 12 septiembre 2017]. Disponible en: [http://ocw.uv.es/ingenieria-y-arquitectura/1-2/Course\\_listing](http://ocw.uv.es/ingenieria-y-arquitectura/1-2/Course_listing).

**TANCO, F, VERRASTRO, C, GRINBERG, D and ROITMAN, J**, Implementación de redes neuronales artificiales en hardware para aplicación en detección automática de fulguraciones solares [en línea]. 2004. Buenos Aires: UTN-FRBA. [Consulta: 15 agosto 2017]. Disponible en: [https://www.researchgate.net/profile/Claudio\\_Verrastro/publication/266500702\\_IMPLEMENTACION\\_DE\\_REDES\\_NEURONALES\\_ARTIFICIALES\\_EN\\_HARDWARE\\_PARA\\_APLICACION\\_EN\\_DETECCION\\_AUTOMATICA\\_DE\\_FULGURACIONES\\_SOLARES/links/545d283c0cf2c1a63bfa604e/IMPLEMENTACION-DE-REDES-NEURONALES-ARTIFICIALES-EN-HARDWARE-PARA-APLICACION-EN-DETECCION-AUTOMATICA-DE-FULGURACIONES-SOLARES.pdf](https://www.researchgate.net/profile/Claudio_Verrastro/publication/266500702_IMPLEMENTACION_DE_REDES_NEURONALES_ARTIFICIALES_EN_HARDWARE_PARA_APLICACION_EN_DETECCION_AUTOMATICA_DE_FULGURACIONES_SOLARES/links/545d283c0cf2c1a63bfa604e/IMPLEMENTACION-DE-REDES-NEURONALES-ARTIFICIALES-EN-HARDWARE-PARA-APLICACION-EN-DETECCION-AUTOMATICA-DE-FULGURACIONES-SOLARES.pdf).

**TORRES ÁLVAREZ, NELSON STIVET, HERNÁNDEZ, CÉSAR and PEDRAZA, LUIS**, Redes neuronales y predicción de tráfico [en línea]. 2011. Bogota. [Consulta: 15 agosto 2017]. Disponible en: <https://dialnet.unirioja.es/descarga/articulo/3802210.pdf>

Tsoukalas, Lefteri & E. Uhrig, Robert. 1997. Fuzzy Neural Approaches in Engineering. Neural Networks. [Consulta: 15 agosto 2017]. Disponible en: [https://www.researchgate.net/publication/234773126\\_Fuzzy\\_Neural\\_Approaches\\_in\\_Engineering](https://www.researchgate.net/publication/234773126_Fuzzy_Neural_Approaches_in_Engineering)

**XUJIAN, F, DEWEN, S and HAITAO, X**, Research Article Traffic flow data mining based on cloud computing. Journal of Chemical and Pharmaceutical Research [en línea]. 2013. Hangzhou. [Consulta: 18 septiembre 2017]. Disponible en: <https://pdfs.semanticscholar.org/76cd/893a1314cfad94cfe569d0721ef68237c3b0.pdf>.

## ANEXOS

### ANEXO A: TABLAS DE CAPTURAS

Numero captura	IP	IPv6	ARP	HTTPS	HTTP	ISCSI-traget
1	3470247901	447246350	109320	2570924698	215653175	669367898
2	1303865169	443104066	67200	317035552	270846828	697660440
3	1285036192	436669100	57660	229569720	335743144	700345332
4	1156118396	394462260	64080	112224192	316846260	716215686
5	1156761792	383547495	65160	108379023	336919081	700869904
6	1325847736	389132547	52080	363202664	234503000	715276300
7	1019699622	76855309	124560	280918913	374808568	336286192
8	1046335301	246837682	89280	267195563	502181091	263037734
9	1956060190	251412541	503100	160752044	1037314589	688870670
10	1131050096	111549487	157380	966964480	139923681	15342012
11	1042795058	131648633	170280	217963732	5037049	811671290
12	1069203568	165704546	149700	226976739	5146081	829081764
13	1051145879	157804788	163920	225595735	298637	817236850
14	1779637171	300096434	499680	479742182	532667678	685521188
15	1177337957	212331679	271080	246985265	158530000	712180934
16	2835051675	316541716	800400	65079471	16444965	2722494186
17	1895038820	132925465	444360	65054827	1067386942	705564276
18	1896342359	134779199	523020	50441226	1098091882	692929328
19	1572246382	148972320	421080	108070618	697927106	706330114
20	1667838255	164780231	180060	160164378	757054787	692466582
21	1925842220	156655600	357300	116788319	1034850590	693497092
22	1956060190	251412541	503100	160752044	1037314589	688870670
23	1935108446	323404664	176340	172992240	1031189452	688200740
24	1847601801	78089703	379140	193355178	925919159	689152808
25	1198721755	43279700	551640	144081974	316992792	693513670
26	1173818402	73778534	653220	229323021	183598068	716656198
27	895876732	9518886	221640	129393262	29014609	700449050
28	875830942	5721086	271860	96878815	26988247	712727488
29	967323468	52133299	319440	120718398	77354334	694124640
30	1009092920	67866390	385500	189641946	68951427	704126116
31	1023002835	40891505	375600	266312062	13211948	695560338
32	1266304078	136212192	545280	492667166	11728465	708742782
33	1061584725	165451912	240660	118622117	209147013	693927870
34	1282903869	246245960	335820	89292446	455433101	698827836
35	1042744004	156261964	514380	87253819	185847922	695234412
36	1526339537	269741095	426420	482865011	238862974	707219380
37	2970658269	1796214087	1042500	1846663017	349424856	640230784
38	2155881278	2240828402	1059900	691098034	622419391	667555942

39	2209157981	876313038	589560	422910859	983475558	688783436
40	2415033972	745101898	454020	330501122	1335493762	671304720
41	2850419035	345542649	342240	236311672	1193907469	1333019842
42	8702681081	124948257	333240	243073399	1547611773	6753256744
43	3750764143	311828217	527400	241671662	1697488562	1709052774
44	2640174926	285238803	217140	374230226	1502494983	693378368
45	2461417803	338809457	431100	186800444	1573241058	658564082
46	5598330159	2781019072	629220	201531141	739770544	4487142226
47	5948807147	1673680384	520740	148329631	815495751	4253435916
48	6135302525	2293175864	588900	518585457	467812534	5042295772
49	5826410925	912965214	612600	88159836	763997283	4918354040
50	5216906633	2016950600	799020	127362449	888330867	4157783542
51	2000496226	652578667	500640	157202493	982429454	821949180
52	1993353807	850971355	478200	70441580	965115481	920660772
53	1897581422	1268091051	502560	75242697	1084202474	695633208
54	2180230482	1483088730	500040	131655363	1344673631	663172186
55	2222684913	2611188350	370860	178751167	676055838	1330867834
56	1014594850	39369805	191340	6704722	145251023	852485580
57	3593216076	644675915	616320	258812546	577424952	2708196312
58	1125624639	595685815	311880	62102155	314679176	718958934
59	2467984131	599715146	290520	20686313	1712058390	714799022
60	1213360863	259514258	291540	34891005	457094939	705132580
61	774548493	69484913	350940	33309342	5036956	723333254
62	940303735	136704321	265260	179558443	47172850	702380966
63	3042846991	28638945	219660	5480364	2334406502	692943148
64	716411796	83177884	213660	2755588	2042761	701003492
65	1292498679	130549179	227940	11890421	557133246	713065518
66	4889643470	90290303	296940	3477896	1033538125	3841807588
67	1014594850	39369805	191340	6704722	145251023	852485580
68	772807962	19751387	264360	9783568	26683694	726324062
69	750977744	38192379	226140	2519777	35624171	702730390
70	728548587	60202141	255720	1016800	1080616	717711484
71	795358017	58233861	301920	21376260	49496860	710904066
72	2780085501	45079398	507720	274179636	308444480	2151597066
73	5332206335	63836772	674280	275531472	1029495195	3885347054
74	1014594850	39369805	191340	6704722	145251023	852485580
75	9541671582	144775666	518700	373309141	1417663843	7564314874
76	5207365368	166478197	496440	372592523	959750446	3729127952
77	1486900603	121942947	645000	182641244	392698312	828113592
78	1049842143	61480248	658800	257183526	9777138	743315068
79	1025886959	119354481	584040	215474844	63357828	703862980
80	3499456732	117912042	525420	151067548	17901558	3174051174
81	5315985337	183660913	765000	87368389	20432230	3713617262



.....	.....	.....	.....	.....	.....	.....
2800	675100639	5790775	597000	3079736	7534423	638710558
2801	889269756	4502855	364080	22925259	215321910	626713464
2802	759280094	4227586	325980	4105646	100825293	629239718
2803	682563640	2337913	309720	15890123	269334	641737092
2804	667642578	1190074	295800	17249236	251251	627671954
2805	705744454	5547355	426960	28382469	321226	649161060
2806	686397619	20990770	435900	22815759	226949	640145596
2807	860310518	34319277	472020	153048652	36228371	627981626
2808	1159617449	32400642	382200	350912863	69579101	642183504
2809	1808164665	1238674	615720	965215063	155897427	624286350
2810	1516541757	5345876	776040	727218362	114944895	623689584
2811	1149202595	5245480	389340	349347760	123692171	644330788
2812	966449622	1651616	487620	159759355	89422959	648222904
2813	1127115343	15659079	311520	295491817	191557002	611955818
2814	890310058	81552802	506340	205378723	5481282	646467760
2815	867541311	78471120	373200	187375137	23429793	623420566
2816	904022710	73748980	469260	227195634	20361543	625857594
2817	1067240990	96695160	1359600	279685865	81653541	657106592
2818	1191649105	34804680	531720	304258490	208362802	637363466
2819	1881206090	38763660	556740	367603608	848804712	616275032
2820	1756984186	16752988	815280	221745599	850172808	637684116
2821	1325222933	2258249	1229220	159146523	466853867	639767324
2822	1518440655	10677916	2922840	232313153	501262278	635674946
2823	1466493171	3365795	1853460	279827481	487555735	638699470
2824	3453249491	72691758	1171560	413345751	2350584694	624246744
2825	1735151222	6245102	999840	152969246	890499714	635348600
2826	1541838719	28804934	1366560	130965181	675237574	662835670
2827	1551707960	43892681	1221900	218464493	642083100	632696868
2828	928980100	3413109	1152060	98061903	104677021	671075422
2829	936730920	1908678	1085580	158147862	58679584	666247210
2830	952527925	2992755	1263720	142640557	117010833	633188108
2831	1404334203	8534169	1187640	145077586	575059056	622313260
2832	1825579572	9118870	1125060	78998508	1053678736	630123014
2833	1826175837	9836027	1054500	115693713	1034684651	615692292
2834	1765132810	14035883	1076580	240480379	851747415	618843778
2835	1885361457	17674574	2280420	419946616	696524304	620060258
2836	1172936652	621248020	603540	81667718	37675031	1031542344
2837	1227177847	603264137	435060	187151152	162709036	812857626
2838	1275671036	498315689	853200	320707318	124783435	794761732
2839	1610293983	373179893	1185540	520493541	253270033	800950978
2840	1545255311	372495252	583740	303703303	174663161	1037174892
2841	1289335578	215018258	789480	329886961	102150733	824524848

2842	1871150222	19996594	1481100	458641975	701686437	625339138
2843	2235048291	12322955	1521420	625830878	352775663	629530900
2844	2346047989	3349294	2596620	400275774	336245202	617242368
2845	2018268896	8619924	677580	182502014	290627146	619764710
2846	1183542943	713622218	592740	104743688	24149203	625770778
2847	848950965	1023692223	595620	172278562	8828464	627076308
2848	963322810	1009291294	580620	245691717	59299481	623765686
2849	820579445	999491894	526320	166806362	601146	619439442
2850	894694013	165158897	523620	210091058	2459693	645106324
2851	914525770	1620845	293280	259908018	1016988	618190696
2852	813372558	1556648	323340	135888371	381331	646073648
2853	732218079	1145657	199380	61438921	260612	648202178
2854	663663038	1023111	177240	5270742	79966	649509902
2855	702708675	5093309	716040	58815183	40185	633725192
2856	732680138	2433762	129420	73592514	106150	649359442
2857	644836656	3055850	116460	5368767	255452	630443076
2858	637092850	20700729	142140	2348665	11308	626470170
2859	643358857	10940029	93720	252275	25114	634821108
2860	659987853	2888923	104160	9586969	7030	642087978
2861	644231939	2961312	83040	1822261	1961	627120172
2862	646772719	62884436	116940	2759149	393223	634475340
2863	646880968	36841687	94200	2079763	0	639300106
2864	636968158	30057331	36060	6072161	8445	627033384
2865	649263969	2068370	17460	2759176	0	642683780
2866	648464593	1082670	19800	0	4400572	640098572
2867	636012230	26562641	30540	1342031	3266424	627755166
2868	651101073	14041990	43500	52848	103704	647217488
2869	649911685	9532907	44820	45114	5877	646166388
2870	645574887	62484410	49440	355150	2610225	638838668
2871	643410283	31526899	45960	7881209	252868	631435556
2872	659893240	43194365	47040	721087	15241879	640138592
2873	653587482	3969030	51600	3239092	555118	645851100
2874	630166788	3891584	89400	1965950	34322	624525708
2875	642571565	4447581	66960	47478	182803	638587300
2876	662815614	3333836	38040	1353728	15372145	642160728
2877	646936209	13531314	15480	5922848	404120	636784808
2878	634545776	823707	16740	747	0	630856714
2879	654273617	900971	32640	49693	35274	650536758
2880	641808081	924521	39720	276542	13715	637583642
	9068.8847	876.744775	1.5275201	1092.86524	774.770264	6594.96244

## ANEXO B: CÓDIGO COMPLETO DE LA RED NEURONAL

```
function varargout = TESIS(varargin)
% TESIS MATLAB code for TESIS.fig
% TESIS, by itself, creates a new TESIS or raises the existing
% singleton*.
%
% H = TESIS returns the handle to a new TESIS or the handle to
% the existing singleton*.
%
% TESIS('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in TESIS.M with the given input
arguments.
%
% TESIS('Property','Value',...) creates a new TESIS or raises the
% existing singleton*. Starting from the left, property value
pairs are
% applied to the GUI before TESIS_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property
application
% stop. All inputs are passed to TESIS_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help TESIS

% Last Modified by GUIDE v2.5 19-Dec-2017 15:11:38

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @TESIS_OpeningFcn, ...
                  'gui_OutputFcn',  @TESIS_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before TESIS is made visible.
function TESIS_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
```

```

% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to TESIS (see VARARGIN)

% Choose default command line output for TESIS
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes TESIS wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = TESIS_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
[handles.filename pathname]= uigetfile('*.xlsx');
guidata(hObject, handles);
setPopupmenuString(handles.popupmenu1, eventdata,handles);
set(handles.popupmenu1, 'callback',
'TESIS('updateAxes',gcbo,[],guidata(gcbo))');

fullpathname = strcat(pathname, handles.filename);
set(handles.text2, 'String', fullpathname );

function setPopupmenuString(hObject, eventdata, handles)
filename = handles.filename;
[num, col]= xlsread(filename);
set(hObject, 'string', col);

function [x,target] = readExcelColumns(filename, ycol)
a = xlsread(filename);
x = [a(:,1)]';
target = [a(:,ycol)]';
l=length(target);

function updateAxes(hObject, eventdata, handles)
ycol = get(handles.popupmenu1, 'Value');
filename = handles.filename;
[x,target]= readExcelColumns(filename, ycol);

plot (handles.axes1, x, target)
xlabel (handles.axes1,'Numero de Captura');
ylabel (handles.axes1,'Trafico en [Mbits]');
title (handles.axes1,'Trafico Capturado');

```

```

% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject      handle to popupmenu1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1
contents as cell array
%           contents{get(hObject,'Value')} returns selected item from
popupmenu1

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to popupmenu1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)

ycol = get(handles.popupmenu1, 'Value');
filename = handles.filename;
[captura,target,simu,e]= readExcelColumns1(filename, ycol);

plot(handles.axes2, captura,simu,captura,target,'--');
xlabel (handles.axes2,'Numero de Captura');
ylabel (handles.axes2,'Trafico en [Mbits]');
title (handles.axes2,'Prediccion');

set(handles.text4, 'String', e );

function [captura,target,simu,e] = readExcelColumns1(filename, ycol)
a = xlsread(filename);
captura = [a(:,1)]';
target = [a(:,ycol)]';
l=length(target);

IN=zeros(100,1); % creacion matriz 300xnumero de capturas
IN(1,2:1) = target(1,1:(l-1)); %datos de entrada
IN(2,3:1) = target(1,1:(l-2));

```

```
IN(3,4:1) = target(1,1:(1-3));
IN(4,5:1) = target(1,1:(1-4));
IN(5,6:1) = target(1,1:(1-5));
IN(6,7:1) = target(1,1:(1-6));
IN(7,8:1) = target(1,1:(1-7));
IN(8,9:1) = target(1,1:(1-8));
IN(9,10:1) = target(1,1:(1-9));
IN(10,11:1) = target(1,1:(1-10));
IN(11,12:1) = target(1,1:(1-11));
IN(12,13:1) = target(1,1:(1-12));
IN(13,14:1) = target(1,1:(1-13));
IN(14,15:1) = target(1,1:(1-14));
IN(15,16:1) = target(1,1:(1-15));
IN(16,17:1) = target(1,1:(1-16));
IN(17,18:1) = target(1,1:(1-17));
IN(18,19:1) = target(1,1:(1-18));
IN(19,20:1) = target(1,1:(1-19));
IN(20,21:1) = target(1,1:(1-20));
IN(21,22:1) = target(1,1:(1-21));
IN(22,23:1) = target(1,1:(1-22));
IN(23,24:1) = target(1,1:(1-23));
IN(24,25:1) = target(1,1:(1-24));
IN(25,26:1) = target(1,1:(1-25));
IN(26,27:1) = target(1,1:(1-26));
IN(27,28:1) = target(1,1:(1-27));
IN(28,29:1) = target(1,1:(1-28));
IN(29,30:1) = target(1,1:(1-29));
IN(30,31:1) = target(1,1:(1-30));
IN(31,32:1) = target(1,1:(1-31));
IN(32,33:1) = target(1,1:(1-32));
IN(33,34:1) = target(1,1:(1-33));
IN(34,35:1) = target(1,1:(1-34));
IN(35,36:1) = target(1,1:(1-35));
IN(36,37:1) = target(1,1:(1-36));
IN(37,38:1) = target(1,1:(1-37));
IN(38,39:1) = target(1,1:(1-38));
IN(39,40:1) = target(1,1:(1-39));
IN(40,41:1) = target(1,1:(1-40));
IN(41,42:1) = target(1,1:(1-41));
IN(42,43:1) = target(1,1:(1-42));
IN(43,44:1) = target(1,1:(1-43));
IN(44,45:1) = target(1,1:(1-44));
IN(45,46:1) = target(1,1:(1-45));
IN(46,47:1) = target(1,1:(1-46));
IN(47,48:1) = target(1,1:(1-47));
IN(48,49:1) = target(1,1:(1-48));
IN(49,50:1) = target(1,1:(1-49));
IN(50,51:1) = target(1,1:(1-50));
IN(51,52:1) = target(1,1:(1-51));
IN(52,53:1) = target(1,1:(1-52));
IN(53,54:1) = target(1,1:(1-53));
IN(54,55:1) = target(1,1:(1-54));
IN(55,56:1) = target(1,1:(1-55));
IN(56,57:1) = target(1,1:(1-56));
IN(57,58:1) = target(1,1:(1-57));
IN(58,59:1) = target(1,1:(1-58));
IN(59,60:1) = target(1,1:(1-59));
IN(60,61:1) = target(1,1:(1-60));
IN(61,62:1) = target(1,1:(1-61));
```

```
IN(62,63:1) = target(1,1:(1-62));
IN(63,64:1) = target(1,1:(1-63));
IN(64,65:1) = target(1,1:(1-64));
IN(65,66:1) = target(1,1:(1-65));
IN(66,67:1) = target(1,1:(1-66));
IN(67,68:1) = target(1,1:(1-67));
IN(68,69:1) = target(1,1:(1-68));
IN(69,70:1) = target(1,1:(1-69));
IN(70,71:1) = target(1,1:(1-70));
IN(71,72:1) = target(1,1:(1-71));
IN(72,73:1) = target(1,1:(1-72));
IN(73,74:1) = target(1,1:(1-73));
IN(74,75:1) = target(1,1:(1-74));
IN(75,76:1) = target(1,1:(1-75));
IN(76,77:1) = target(1,1:(1-76));
IN(77,78:1) = target(1,1:(1-77));
IN(78,79:1) = target(1,1:(1-78));
IN(79,80:1) = target(1,1:(1-79));
IN(80,81:1) = target(1,1:(1-80));
IN(81,82:1) = target(1,1:(1-81));
IN(82,83:1) = target(1,1:(1-82));
IN(83,84:1) = target(1,1:(1-83));
IN(84,85:1) = target(1,1:(1-84));
IN(85,86:1) = target(1,1:(1-85));
IN(86,87:1) = target(1,1:(1-86));
IN(87,88:1) = target(1,1:(1-87));
IN(88,89:1) = target(1,1:(1-88));
IN(89,90:1) = target(1,1:(1-89));
IN(90,91:1) = target(1,1:(1-90));
IN(91,92:1) = target(1,1:(1-91));
IN(92,93:1) = target(1,1:(1-92));
IN(93,94:1) = target(1,1:(1-93));
IN(94,95:1) = target(1,1:(1-94));
IN(95,96:1) = target(1,1:(1-95));
IN(96,97:1) = target(1,1:(1-96));
IN(97,98:1) = target(1,1:(1-97));
IN(98,99:1) = target(1,1:(1-98));
IN(99,100:1) = target(1,1:(1-99));
IN(100,101:1) = target(1,1:(1-100));
IN(101,102:1) = target(1,1:(1-101));
IN(102,103:1) = target(1,1:(1-102));
IN(103,104:1) = target(1,1:(1-103));
IN(104,105:1) = target(1,1:(1-104));
IN(105,106:1) = target(1,1:(1-105));
IN(106,107:1) = target(1,1:(1-106));
IN(107,108:1) = target(1,1:(1-107));
IN(108,109:1) = target(1,1:(1-108));
IN(109,110:1) = target(1,1:(1-109));
IN(110,111:1) = target(1,1:(1-110));
IN(111,112:1) = target(1,1:(1-111));
IN(112,113:1) = target(1,1:(1-112));
IN(113,114:1) = target(1,1:(1-113));
IN(114,115:1) = target(1,1:(1-114));
IN(115,116:1) = target(1,1:(1-115));
IN(116,117:1) = target(1,1:(1-116));
IN(117,118:1) = target(1,1:(1-117));
IN(118,119:1) = target(1,1:(1-118));
IN(119,120:1) = target(1,1:(1-119));
IN(120,121:1) = target(1,1:(1-120));
```

```
IN(121,122:1) = target(1,1:(1-121));
IN(122,123:1) = target(1,1:(1-122));
IN(123,124:1) = target(1,1:(1-123));
IN(124,125:1) = target(1,1:(1-124));
IN(125,126:1) = target(1,1:(1-125));
IN(126,127:1) = target(1,1:(1-126));
IN(127,128:1) = target(1,1:(1-127));
IN(128,129:1) = target(1,1:(1-128));
IN(129,130:1) = target(1,1:(1-129));
IN(130,131:1) = target(1,1:(1-130));
IN(131,132:1) = target(1,1:(1-131));
IN(132,133:1) = target(1,1:(1-132));
IN(133,134:1) = target(1,1:(1-133));
IN(134,135:1) = target(1,1:(1-134));
IN(135,136:1) = target(1,1:(1-135));
IN(136,137:1) = target(1,1:(1-136));
IN(137,138:1) = target(1,1:(1-137));
IN(138,139:1) = target(1,1:(1-138));
IN(139,140:1) = target(1,1:(1-139));
IN(140,141:1) = target(1,1:(1-140));
IN(141,142:1) = target(1,1:(1-141));
IN(142,143:1) = target(1,1:(1-142));
IN(143,144:1) = target(1,1:(1-143));
IN(144,145:1) = target(1,1:(1-144));
IN(145,146:1) = target(1,1:(1-145));
IN(146,147:1) = target(1,1:(1-146));
IN(147,148:1) = target(1,1:(1-147));
IN(148,149:1) = target(1,1:(1-148));
IN(149,150:1) = target(1,1:(1-149));
IN(150,151:1) = target(1,1:(1-150));
IN(151,152:1) = target(1,1:(1-151));
IN(152,153:1) = target(1,1:(1-152));
IN(153,154:1) = target(1,1:(1-153));
IN(154,155:1) = target(1,1:(1-154));
IN(155,156:1) = target(1,1:(1-155));
IN(156,157:1) = target(1,1:(1-156));
IN(157,158:1) = target(1,1:(1-157));
IN(158,159:1) = target(1,1:(1-158));
IN(159,160:1) = target(1,1:(1-159));
IN(160,161:1) = target(1,1:(1-160));
IN(161,162:1) = target(1,1:(1-161));
IN(162,163:1) = target(1,1:(1-162));
IN(163,164:1) = target(1,1:(1-163));
IN(164,165:1) = target(1,1:(1-164));
IN(165,166:1) = target(1,1:(1-165));
IN(166,167:1) = target(1,1:(1-166));
IN(167,168:1) = target(1,1:(1-167));
IN(168,169:1) = target(1,1:(1-168));
IN(169,170:1) = target(1,1:(1-169));
IN(170,171:1) = target(1,1:(1-170));
IN(171,172:1) = target(1,1:(1-171));
IN(172,173:1) = target(1,1:(1-172));
IN(173,174:1) = target(1,1:(1-173));
IN(174,175:1) = target(1,1:(1-174));
IN(175,176:1) = target(1,1:(1-175));
IN(176,177:1) = target(1,1:(1-176));
IN(177,178:1) = target(1,1:(1-177));
IN(178,179:1) = target(1,1:(1-178));
IN(179,180:1) = target(1,1:(1-179));
```

IN(180,181:1) = target(1,1:(1-180));  
IN(181,182:1) = target(1,1:(1-181));  
IN(182,183:1) = target(1,1:(1-182));  
IN(183,184:1) = target(1,1:(1-183));  
IN(184,185:1) = target(1,1:(1-184));  
IN(185,186:1) = target(1,1:(1-185));  
IN(186,187:1) = target(1,1:(1-186));  
IN(187,188:1) = target(1,1:(1-187));  
IN(188,189:1) = target(1,1:(1-188));  
IN(189,190:1) = target(1,1:(1-189));  
IN(190,191:1) = target(1,1:(1-190));  
IN(191,192:1) = target(1,1:(1-191));  
IN(192,193:1) = target(1,1:(1-192));  
IN(193,194:1) = target(1,1:(1-193));  
IN(194,195:1) = target(1,1:(1-194));  
IN(195,196:1) = target(1,1:(1-195));  
IN(196,197:1) = target(1,1:(1-196));  
IN(197,198:1) = target(1,1:(1-197));  
IN(198,199:1) = target(1,1:(1-198));  
IN(199,200:1) = target(1,1:(1-199));  
IN(200,201:1) = target(1,1:(1-200));  
IN(201,202:1) = target(1,1:(1-201));  
IN(202,203:1) = target(1,1:(1-202));  
IN(203,204:1) = target(1,1:(1-203));  
IN(204,205:1) = target(1,1:(1-204));  
IN(205,206:1) = target(1,1:(1-205));  
IN(206,207:1) = target(1,1:(1-206));  
IN(207,208:1) = target(1,1:(1-207));  
IN(208,209:1) = target(1,1:(1-208));  
IN(209,210:1) = target(1,1:(1-209));  
IN(210,211:1) = target(1,1:(1-210));  
IN(211,212:1) = target(1,1:(1-211));  
IN(212,213:1) = target(1,1:(1-212));  
IN(213,214:1) = target(1,1:(1-213));  
IN(214,215:1) = target(1,1:(1-214));  
IN(215,216:1) = target(1,1:(1-215));  
IN(216,217:1) = target(1,1:(1-216));  
IN(217,218:1) = target(1,1:(1-217));  
IN(218,219:1) = target(1,1:(1-218));  
IN(219,220:1) = target(1,1:(1-219));  
IN(220,221:1) = target(1,1:(1-220));  
IN(221,222:1) = target(1,1:(1-221));  
IN(222,223:1) = target(1,1:(1-222));  
IN(223,224:1) = target(1,1:(1-223));  
IN(224,225:1) = target(1,1:(1-224));  
IN(225,226:1) = target(1,1:(1-225));  
IN(226,227:1) = target(1,1:(1-226));  
IN(227,228:1) = target(1,1:(1-227));  
IN(228,229:1) = target(1,1:(1-228));  
IN(229,230:1) = target(1,1:(1-229));  
IN(230,231:1) = target(1,1:(1-230));  
IN(231,232:1) = target(1,1:(1-231));  
IN(232,233:1) = target(1,1:(1-232));  
IN(233,234:1) = target(1,1:(1-233));  
IN(234,235:1) = target(1,1:(1-234));  
IN(235,236:1) = target(1,1:(1-235));  
IN(236,237:1) = target(1,1:(1-236));  
IN(237,238:1) = target(1,1:(1-237));  
IN(238,239:1) = target(1,1:(1-238));

IN(239,240:1) = target(1,1:(1-239));  
IN(240,241:1) = target(1,1:(1-240));  
IN(241,242:1) = target(1,1:(1-241));  
IN(242,243:1) = target(1,1:(1-242));  
IN(243,244:1) = target(1,1:(1-243));  
IN(244,245:1) = target(1,1:(1-244));  
IN(245,246:1) = target(1,1:(1-245));  
IN(246,247:1) = target(1,1:(1-246));  
IN(247,248:1) = target(1,1:(1-247));  
IN(248,249:1) = target(1,1:(1-248));  
IN(249,250:1) = target(1,1:(1-249));  
IN(250,251:1) = target(1,1:(1-250));  
IN(251,252:1) = target(1,1:(1-251));  
IN(252,253:1) = target(1,1:(1-252));  
IN(253,254:1) = target(1,1:(1-253));  
IN(254,255:1) = target(1,1:(1-254));  
IN(255,256:1) = target(1,1:(1-255));  
IN(256,257:1) = target(1,1:(1-256));  
IN(257,258:1) = target(1,1:(1-257));  
IN(258,259:1) = target(1,1:(1-258));  
IN(259,260:1) = target(1,1:(1-259));  
IN(260,261:1) = target(1,1:(1-260));  
IN(261,262:1) = target(1,1:(1-261));  
IN(262,263:1) = target(1,1:(1-262));  
IN(263,264:1) = target(1,1:(1-263));  
IN(264,265:1) = target(1,1:(1-264));  
IN(265,266:1) = target(1,1:(1-265));  
IN(266,267:1) = target(1,1:(1-266));  
IN(267,268:1) = target(1,1:(1-267));  
IN(268,269:1) = target(1,1:(1-268));  
IN(269,270:1) = target(1,1:(1-269));  
IN(270,271:1) = target(1,1:(1-270));  
IN(271,272:1) = target(1,1:(1-271));  
IN(272,273:1) = target(1,1:(1-272));  
IN(273,274:1) = target(1,1:(1-273));  
IN(274,275:1) = target(1,1:(1-274));  
IN(275,276:1) = target(1,1:(1-275));  
IN(276,277:1) = target(1,1:(1-276));  
IN(277,278:1) = target(1,1:(1-277));  
IN(278,279:1) = target(1,1:(1-278));  
IN(279,280:1) = target(1,1:(1-279));  
IN(280,281:1) = target(1,1:(1-280));  
IN(281,282:1) = target(1,1:(1-281));  
IN(282,283:1) = target(1,1:(1-282));  
IN(283,284:1) = target(1,1:(1-283));  
IN(284,285:1) = target(1,1:(1-284));  
IN(285,286:1) = target(1,1:(1-285));  
IN(286,287:1) = target(1,1:(1-286));  
IN(287,288:1) = target(1,1:(1-287));  
IN(288,289:1) = target(1,1:(1-288));  
IN(289,290:1) = target(1,1:(1-289));  
IN(290,291:1) = target(1,1:(1-290));  
IN(291,292:1) = target(1,1:(1-291));  
IN(292,293:1) = target(1,1:(1-292));  
IN(293,294:1) = target(1,1:(1-293));  
IN(294,295:1) = target(1,1:(1-294));  
IN(295,296:1) = target(1,1:(1-295));  
IN(296,297:1) = target(1,1:(1-296));  
IN(297,298:1) = target(1,1:(1-297));

```
IN(298,299:1) = target(1,1:(1-298));
IN(299,300:1) = target(1,1:(1-299));
IN(300,301:1) = target(1,1:(1-300));

for j=1:1
    net=newff(IN,target,1); % Creacion de la red backpropagation
    net.trainParam.epochs = 1000; % Número máximo de épocas para
entrenar
    net.trainParam.goal = 0.05; % Objetivo de rendimiento
    net.trainParam.max_fail = 100 ;% Errores máximos de validación

    net=train(net,IN,target); % Entrenamiento de la RNA
    %gensim(net) % diagrama de bloques en simulink

    simu = sim (net, IN); % Simulacion
    %view(net)
end
```