



UNIVERSIDAD ANDINA DEL CUSCO  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



TESIS

PROPUESTA DE IMPLEMENTACIÓN DE UN MÉTODO DE  
PREDICCIÓN DEL RTT BASADO EN RNA ALTERNATIVO AL  
MÉTODO DE PREDICCIÓN ESTÁNDAR DEL PROTOCOLO  
SCTP PARA LA TRANSMISIÓN DE FLUJOS MULTIMEDIA

Presentado por: Bach. Escobedo Baca, Kyle

Para optar al Título Profesional de Ingeniero de  
Sistemas

Asesor: Mg. Edwin Carrasco Pobleto

CUSCO – PERÚ  
2022



## ÍNDICE GENERAL

<b>DEDICATORIA</b> .....	9
<b>AGRADECIMIENTOS</b> .....	10
<b>RESUMEN</b> .....	11
<b>ABSTRACT</b> .....	12
<b>INTRODUCCIÓN</b> .....	13
<b>CAPÍTULO I: ASPECTOS GENERALES</b> .....	14
1.1. Descripción de la situación actual .....	14
1.2. Formulación del problema .....	15
1.2.1. Formulación interrogativa del Problema General .....	15
1.2.2. Formulación interrogativa de los Problemas Específicos .....	15
1.3. Objetivos .....	16
1.3.1. Objetivo General .....	16
1.3.2. Objetivos Específicos .....	16
1.4. Hipótesis .....	16
1.5. Variables e indicadores .....	16
1.5.1. Variable dependiente .....	16
1.5.2. Indicadores de variables dependientes .....	16
1.5.3. Variables independientes .....	16
1.5.4. Indicadores de variables independientes .....	16
1.5.5. Cuadro de operacionalización de variables .....	17
1.6. Justificación de la investigación .....	18
1.7. Metodología de la tesis .....	18
1.7.1. Tipo de investigación .....	18
1.7.2. Nivel de investigación .....	18
1.7.3. Método de investigación .....	18
1.8. Matriz de Consistencia .....	20



<b>CAPÍTULO II: MARCO TEÓRICO</b> .....	21
2.1. Aspectos teóricos pertinentes .....	21
2.1.1. Redes informáticas.....	21
2.1.2. Modelo de referencia OSI.....	21
2.1.3. Protocolos de Red .....	22
2.1.4. La capa de transporte .....	23
2.1.5. Mecanismos del protocolo de transporte orientado a la conexión.....	23
2.1.5.1. Servicio de red confiable.....	24
2.1.5.2. Servicio de red no confiable.....	24
2.1.5.2.1. Estrategia de retransmisión.....	24
2.1.6. Stream Control Transmission Protocol (SCTP).....	25
2.1.7. Características Principales .....	26
2.1.8. Arquitectura SCTP.....	27
2.1.9. Estructura de segmento SCTP .....	28
2.1.9.1. Sección de encabezado común.....	28
2.1.9.2. Sección de fragmentos de datos .....	28
2.1.10. Paquetes SCTP .....	29
2.1.11. Establecimiento de la asociación.....	30
2.1.12. Transmisiones durante una asociación.....	31
2.1.13. Liberación de una asociación .....	32
2.1.14. Manejo de Retransmisiones .....	32
2.1.15. RTT .....	33
2.1.16. RTO.....	33
2.1.17. Redes Neuronales Artificiales (RNA).....	34
2.1.18. Neurona genérica.....	34
2.1.19. Topología de las RNA.....	35
2.1.20. Perceptron Multicapa (Multilayer Perceptron).....	35



2.1.21.	Funciones de activación .....	36
2.1.21.1.	Función Lineal.....	36
2.1.21.2.	Función sigmoidea .....	37
2.1.21.3.	Función tangente hiperbólica .....	38
2.1.21.4.	Función ReLU .....	39
2.1.22.	Algoritmo Gradiente Descendiente .....	40
2.1.23.	Algoritmo de Retropropagación (Backpropagation) .....	40
2.1.24.	Proceso de aprendizaje .....	41
2.1.25.	Predicción mediante RNA.....	41
2.1.26.	Metodología CRISP-DM.....	42
2.1.26.1.	Entendimiento el Negocio .....	43
2.1.26.2.	Entendimiento de los Datos.....	43
2.1.26.3.	Preparación de los Datos .....	43
2.1.26.4.	Modelamiento.....	43
2.1.26.5.	Evaluación.....	44
2.1.26.6.	Despliegue .....	44
2.2.	Antecedentes de la investigación.....	44
2.2.1.	Antecedentes a Nivel Internacional .....	44
2.2.2.	Antecedentes a Nivel Nacional .....	45
<b>CAPÍTULO III: METODOLOGÍA</b>	.....	<b>47</b>
3.1.	Tipo de investigación.....	47
3.2.	Diseño de la investigación.....	47
3.3.	Instrumentos .....	48
3.4.	Recolección y Análisis de Datos .....	48
<b>CAPÍTULO IV: DESARROLLO E IMPLEMENTACIÓN</b>	.....	<b>50</b>
4.1.	Entendimiento del Negocio .....	50
4.1.1.	Cronograma de trabajo.....	53



4.1.2.	Diagrama de Gantt .....	54
4.2.	Entendimiento de los Datos .....	55
4.2.1.	Captura de Datos .....	55
4.2.2.	Entorno de simulación .....	55
4.2.2.1.	Arquitectura lógica de la red .....	55
4.2.2.2.	Configuración de las máquinas .....	57
4.2.2.3.	Programas y librerías empleadas .....	58
4.2.2.4.	Arquitectura física de la red .....	59
4.2.2.5.	Características de los ordenadores .....	59
4.2.2.6.	Características de la Conexión .....	60
4.2.3.	Escenarios de prueba.....	60
4.2.3.1.	Escenario 1 .....	60
4.2.3.2.	Escenario 2 .....	60
4.2.3.3.	Descripción de Variables.....	61
4.3.	Preparación de los Datos: .....	62
4.3.1.	Limpieza de Datos .....	62
4.3.2.	Normalización de los datos .....	62
4.4.	Modelamiento .....	63
4.4.1.	Selección de predictores.....	63
4.4.2.	Criterios de Selección de Variables .....	64
4.4.3.	Estrategia de Inteligencia Artificial .....	68
4.4.4.	Diseño de la red neuronal.....	69
4.4.5.	Resultado de la RNA .....	70
4.5.	Evaluación .....	71
4.5.1.	Análisis comparativo de los métodos en el cálculo del RTT .....	71
4.6.	Despliegue .....	73
<b>CAPÍTULO V: RESULTADOS</b>	.....	<b>75</b>



5.1. Análisis de métodos en función al RTT .....	75
5.2. Análisis de métodos en función a la Predicción .....	77
5.3. Evolución del $R^2$ por el tiempo.....	78
5.4. Evolución del MAE en el tiempo .....	79
5.5. Evolución del RMSE en el tiempo .....	80
5.6. Análisis de Resultados.....	81
<b>CAPÍTULO VI: DISCUSIÓN</b> .....	<b>82</b>
<b>GLOSARIO</b> .....	<b>84</b>
<b>CONCLUSIONES</b> .....	<b>85</b>
<b>RECOMENDACIONES</b> .....	<b>86</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	<b>87</b>
<b>ANEXOS</b> .....	<b>90</b>
Topologías Diseñadas (Entrenamiento prueba - error):.....	90



## ÍNDICE DE TABLAS

<b>Tabla 1</b>	Cuadro de operacionalización de variables.....	17
<b>Tabla 2</b>	Matriz de consistencia.....	20
<b>Tabla 3</b>	Cronograma de trabajo.....	53
<b>Tabla 4</b>	Pesos sinápticos entre capa de entrada y capa oculta.....	70
<b>Tabla 5</b>	Pesos sinápticos entre la capa oculta y la capa de salida.....	71
<b>Tabla 6</b>	Análisis en función al R2 .....	75
<b>Tabla 7</b>	Análisis en función al MAE y RMSE .....	76
<b>Tabla 8</b>	Correlación de valores RTT estimados .....	77
<b>Tabla 9</b>	Análisis comparativo con otra investigación.....	82
<b>Tabla 10</b>	Análisis comparativo con otra investigación.....	83



## ÍNDICE DE FIGURAS

<b>Figura 1</b>	Modelo de referencia OSI.....	22
<b>Figura 2</b>	Comunicación en capa de transporte .....	23
<b>Figura 3</b>	Asociación SCTP.....	27
<b>Figura 4</b>	Estructura de segmentos SCTP.....	29
<b>Figura 5</b>	Establecimiento de asociación SCTP. ....	30
<b>Figura 6</b>	Cierre de asociación SCTP. ....	32
<b>Figura 7</b>	Neurona genérica .....	34
<b>Figura 8</b>	Perceptron Multicapa.....	36
<b>Figura 9</b>	Función de activación Lineal.....	37
<b>Figura 10</b>	Función de activación Sigmoidea.....	38
<b>Figura 11</b>	Función de activación Tangente Hiperbólica .....	39
<b>Figura 12</b>	Función de activación ReLU .....	40
<b>Figura 13</b>	Fases de la metodología CRISP-DM.....	42
<b>Figura 14</b>	Diagrama de Gantt.....	54
<b>Figura 15</b>	Arquitectura lógica del entorno de simulación.....	56
<b>Figura 16</b>	Arquitectura lógica y configuraciones.....	57
<b>Figura 17</b>	Arquitectura física de la red de simulación .....	59
<b>Figura 18</b>	Modelos generados acorde a la cantidad de variables .....	64
<b>Figura 19</b>	Criterios de selección para los posibles modelos.....	65
<b>Figura 20</b>	Correlación de variables en función del RTT.....	67
<b>Figura 21</b>	Topología de la RNA de estimación del RTT .....	69
<b>Figura 22</b>	Análisis comparativo de los valores del RTT.....	72
<b>Figura 23</b>	Correlación de variables bajo predicciones de los modelos .....	77
<b>Figura 24</b>	Evolución del R2 durante la transmisión.....	78
<b>Figura 25</b>	Evolución del MAE durante la transmisión.....	79
<b>Figura 26</b>	Evolución del RMSE durante la transmisión.....	80
<b>Figura 27</b>	Topología 1.....	90
<b>Figura 28</b>	Topología 2.....	90
<b>Figura 29</b>	Topología 3.....	91
<b>Figura 30</b>	Topología 4.....	91





## DEDICATORIA

*A Dios por darme el camino y a mi familia por caminarlo a mi lado.*



## AGRADECIMIENTOS

Gracias a Dios, por su maravilloso amor y porque con él todo es posible. Agradezco a mi familia por haber dedicado su vida para que me encuentre aquí.

También agradezco a mi asesor por todos los conocimientos y su inmenso apoyo. De la misma forma, a los dictaminantes por todo el esfuerzo puesto en este trabajo.



## RESUMEN

El objetivo central de la investigación es proponer la implementación de un método alternativo de predicción del valor del RTT (Round Trip Time o Tiempo de Ida y Vuelta), perteneciente al protocolo SCTP (Stream Control Transmission Protocol o Protocolo de Control de Transmisión de Flujos), cuando se encuentra frente a entornos con altas exigencias de red.

Para llevar a cabo esto se desplegó un entorno de red con exigencias elevadas de recursos, a través del cual se transmitieron flujos multimedia. La información relacionada a la asociación establecida entre los nodos, fue capturada, procesada y analizada. Estas acciones permitieron determinar las variables que definirían el valor del RTT, que son el Jitter y Packet Loss Rate.

Posteriormente, se recurrió al diseño e implementación de las RNA (Redes Neuronales Artificiales). Esta actividad exigió realizar la evaluación de varios modelos diseñados para la estimación del RTT, lo cual supuso una evaluación de modelos, prueba – error.

A fin de determinar el impacto de la estimación del método basado en RNA, se utilizaron indicadores para medir diferentes aspectos de los modelos. De esta forma, se puede asegurar que la evaluación comparativa se hace sometido a las mismas condiciones, para ambos métodos.



## ABSTRACT

The central objective of the research is to propose the implementation of an alternative method of predicting the RTT (Round Trip Time) value, belonging to the SCTP (Stream Control Transmission Protocol) protocol, when faced with environments with high network demands. To accomplish this, a network environment with high resource demands was deployed, through which multimedia streams were transmitted. The information related to the association established between the nodes was captured, processed and analyzed. These actions allowed to determine the variables that would define the value of the RTT, which are the Jitter and Packet Loss Rate.

Subsequently, the design and implementation of the ANN (Artificial Neural Networks) were used. This activity required the evaluation of several models designed for the estimation of the RTT, which involved an evaluation of models, trial - error.

In order to determine the impact of the estimation of the ANN-based method, indicators were used to measure different aspects of the models. In this way, it can be ensured that the comparative evaluation is carried out under the same conditions for both methods.



## INTRODUCCIÓN

La comunicación entre terminales de las redes informáticas, hace uso de una amplia variedad de protocolos, estos protocolos están segmentados de acuerdo a capas, en modelos conceptuales.

Los protocolos de transporte proporcionan un servicio de transferencia de datos extremo a extremo que aísla las capas superiores de los detalles de la red o redes subyacentes.

Es debido a las necesidades que surgen a partir de las mayores capacidades para transmitir información y hacerlo de la forma más rápida, que se busca experimentar con nuevas alternativas y formas de realizar la comunicación, a través de los protocolos de transporte.

Uno de los protocolos que ha demostrado un buen rendimiento, comparado con los protocolos de comunicación, UDP y TCP, es el protocolo de comunicación SCTP (Stream Control Transmission Protocol o Protocolo de Control de Transmisión de Flujos). Este protocolo de comunicación implementa características particulares; sin embargo, también toma algunas características e implementa algunos mecanismos de los otros protocolos. El algoritmo de ajuste del RTT (Round Trip Time o Tiempo de Ida y Vuelta) es uno de ellos, utilizado para estimar el tiempo de la confirmación del envío exitoso de segmentos. Además, de que el RTT es el que permite determinar el valor del RTO (Retransmission Timeout).

El algoritmo de estimación que implementa el protocolo, proporciona resultados sujetos a niveles de error considerables. El valor asignado al RTT repercutirá en el valor del RTO. Un tiempo elevado del RTO incrementa la latencia de la transmisión, y al contrario de ello, asignarle un tiempo demasiado bajo conlleva a retransmisiones innecesarias e incrementa la congestión del canal. Estos problemas se agravan en entornos con estrictos requerimientos de QoS (Quality of Service o Calidad de Servicio).

En este proyecto se propone modificar la función del algoritmo actual donde se realiza la predicción del RTT, buscando una alternativa a la estimación del tiempo que toma el envío de paquetes y la confirmación de los mismos, asignada a las transmisiones de una asociación SCTP, empleando RNA (Redes Neuronales Artificiales). Además de que estará enfocada a soportar entornos exigentes, por lo que se realizará la transmisión de flujos multimedia.

La primera sección del presente proyecto contempla los aspectos generales y problemática afín a la investigación; La segunda sección, abarca las bases teóricas y trabajos relacionados al protocolo de comunicación SCTP, y las RNA (Redes Neuronales Artificiales); en las secciones siguientes se trabaja aspectos relacionados al desarrollo del proyecto.



## CAPÍTULO I: ASPECTOS GENERALES

### 1.1. Descripción de la situación actual

En el año 2000, el grupo SIGTRAN definió un nuevo protocolo de capa de transporte conocido como Stream Control Transmission Protocol (SCTP) pensado, inicialmente, para transportar señalización telefónica SS7 (Sistema de Señalización N° 7) sobre el protocolo IP. La intención fue proveer algunas de las características de confiabilidad de SS7, sobre el protocolo IP. Por su versatilidad, más tarde, se ha propuesto utilizarlo en otras áreas (Oracle, 2018).

SCTP, es un protocolo de nivel de transporte orientado a la conexión que, a diferencia del protocolo de comunicación TCP y su concepto de conexión, introduce una nueva forma de comunicación entre sistemas denominada asociación. Este protocolo está especificado en los RFC's (Request For Comments, Peticiones de Comentarios) 3286 y 4960.

Permite multiplexar varios flujos de nivel de aplicación diferentes a través de una conexión SCTP única (Multi-Streaming). SCTP también permite que los datos se transfieran a través de dos rutas de salida cuando un host está conectado a dos o más redes, entrega opcional de datos fuera de orden y una serie de otras características. Los algoritmos de control de flujo y congestión de SCTP son esencialmente los mismos que en TCP (Kurose & Ross , Computer Networking a Top-Down Approach, 2021).

Un punto final SCTP utiliza un temporizador de retransmisión (T3-rtx) para garantizar la entrega de datos en ausencia de cualquier retroalimentación de su par. La duración de este temporizador se conoce como RTO (Retransmission Timeout o Tiempo de Espera de Retransmisión). El RTT es el periodo de tiempo asignado a la espera de la confirmación del envío exitoso de paquetes, lo cual implica el cálculo del RTO.

Cuando un par de un punto extremo es Multi-Homed, el punto extremo calculará un RTO separado para cada dirección de transporte de destino diferente de su punto extremo par.

El cálculo y la gestión de RTT en SCTP siguen de cerca cómo TCP gestiona su temporizador de retransmisión. Para calcular el RTT actual, un punto final mantiene dos variables de estado por dirección de transporte de destino: SRTT (tiempo de ida y vuelta suavizado) y RTTVAR (variación de tiempo de ida y vuelta) (Stewart, 2007).

El RTT, es resultado de un pronóstico estadístico, este cálculo está sujeto a niveles de error considerables, debido a que la forma en la que se calcula el valor del RTT se ve afectada por la constante  $\alpha$ , que por su naturaleza no sigue el comportamiento real del tráfico de la red. Junto



a ello, seguir el comportamiento de la red que presenta variaciones repentinas se hace difícil, pues las estimaciones que hace el algoritmo estándar tienen cambios graduales. Consecuentemente asignar al RTT un tiempo elevado incrementa la latencia de la transmisión, y al contrario de ello, asignarle un tiempo demasiado bajo conlleva a retransmisiones innecesarias e incrementa la congestión del canal (Dong, Du, & Yan, 2019).

Las consecuencias de este problema se ven reflejadas en ambientes donde se requiere de gran cantidad de recursos de red, como el despliegue de tecnologías basadas en teleprocesos o teleoperaciones, debido a que la demora en el control de señales sumado a las variaciones drásticas en los tiempos de demora, deterioran la capacidad de respuesta e impactan su estabilidad. Otro escenario es respecto a video almacenado o por streaming, que exige periodos de demora muy cortos. Como se puede observar la asignación del valor del RTT puede impactar negativamente en la calidad de servicio y en los costos operacionales.

Sin embargo, aún es posible modificar este protocolo, para optimizar el cálculo del tiempo estimado. En este contexto, se propone modificar la función del algoritmo actual donde se realiza la predicción del RTT, buscando una alternativa a la estimación del tiempo de espera, asignada a las transmisiones a través de una asociación SCTP, haciendo uso de las RNA.

Esta propuesta alternativa basada en RNA será construida en un entorno de pruebas que contemplará la transmisión de flujos multimedia, para replicar un entorno con estrictos requisitos de QoS.

## 1.2. Formulación del problema

### 1.2.1. Formulación interrogativa del Problema General

¿Puede un método de predicción del RTT basado en RNA ser una alternativa al método de predicción estándar del protocolo SCTP en la transmisión de flujos multimedia?

### 1.2.2. Formulación interrogativa de los Problemas Específicos

- ¿Qué variables permitirán calcular el valor asignado al RTT?
- ¿Qué modelo de predicción emplear para la estimación del valor del RTT utilizando RNA?
- ¿Puede modificarse el método de predicción del RTT en el protocolo de comunicación SCTP utilizando las RNA?
- ¿Cuál es la predicción del método estándar y el método propuesto?



### 1.3. Objetivos

#### 1.3.1. Objetivo General

Proponer la implementación de un método de predicción del RTT basado en RNA, alternativo al método de predicción estándar del protocolo SCTP en la transmisión de flujos multimedia.

#### 1.3.2. Objetivos Específicos

- Determinar variables que permitan calcular el valor asignado al RTT.
- Diseñar el modelo de predicción del cálculo del RTT, empleando las RNA.
- Implementar las RNA para la predicción del RTT en el protocolo SCTP.
- Evaluar la predicción del método estándar y el método propuesto.

### 1.4. Hipótesis

El método de predicción del RTT basado en RNA, es una alternativa al método de predicción estándar del protocolo SCTP en la transmisión de flujos multimedia.

### 1.5. Variables e indicadores

#### 1.5.1. Variable dependiente

Predicción: Estimación aproximada de valores futuros del RTT.

#### 1.5.2. Indicadores de variables dependientes

Se considera la correlación como el indicador de la capacidad predictiva de la RNA:

- CORR: Correlación entre los valores observados y pronosticados por el modelo.

#### 1.5.3. Variables independientes

RTT: El tiempo transcurrido entre el inicio de una solicitud de red y la recepción de una respuesta (Gartner, 2018).

#### 1.5.4. Indicadores de variables independientes

Para determinar los indicadores correspondientes a la variable RTT, se tomó como referencia la investigación realizada por Dong, Du, y Yan (2019), donde son consideradas las siguientes métricas:

- MAE: Mean Absolute Error o Error Medio Absoluto proporciona una magnitud promedio del error.
- RMSE: Root Squared Mean Error o Error Cuadrático Medio representa la desviación del modelo, le da más peso a la imprecisión de la estimación.
- $R^2$ : Multiple Coefficient of Determination o Coeficiente Múltiple de Determinación representa la variación de la data obtenida.





1.5.5. Cuadro de operacionalización de variables

Tabla 1

*Cuadro de operacionalización de variables.*

Variables	Dimensiones	Indicadores	Criterios de Valoración	Método de cálculo
<b>Variable dependiente</b>				
Predicción	CORR	Correlación entre los valores observados y pronosticados por el modelo.	Rango de 0 a 1; 0 mínima relación 1 máxima relación	$\frac{\sum_{i=1}^n (x_i - \hat{x}_i)(y_i - \hat{y}_i)}{(n-1)s_x s_y}$ ; Donde $\hat{x}$ y $\hat{y}$ son la media, $s_x$ y $s_y$ son la desviación estándar, y $x$ e $y$ son las variables actuales. $n$ es el número de elementos.
<b>Variable independiente</b>				
	R <sup>2</sup>	Variación de la data obtenida.	Rango de 0 a 1; 0 mínima variación 1 máxima variación	$1 - \left( \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \right)$ ; Donde $y_i$ y $\hat{y}_i$ son el valor actual y el predicho en el instante ( $i = 1, \dots, n$ ).
RTT	MAE	Proporciona una magnitud promedio del error.	Rango de 0 a más; 0 mínimo error >0 mayor error	$\frac{1}{n} \sum_{i=1}^n  y_i - \hat{y}_i $ ; Donde $y_i$ y $\hat{y}_i$ son el valor actual y el predicho en el instante ( $i = 1, \dots, n$ ). $n$ número de observaciones.
	RMSE	Desviación del modelo, le da más peso a la imprecisión de la estimación.	Rango de 0 a más; 0 mínima desviación >0 mayor desviación	$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$ ; Donde $y_i$ y $\hat{y}_i$ son el valor actual y el predicho en el instante ( $i = 1, \dots, n$ ). $n$ número de observaciones.

Nota. La tabla muestra las variables dependiente e independiente y sus correspondientes indicadores, con los cuales se hará la medición de las variables. Fuente de elaboración propia.



## 1.6. Justificación de la investigación

La importancia del presente proyecto radica en contribuir a los esfuerzos por mejorar los algoritmos estándar pertenecientes a los protocolos de comunicación, en búsqueda de un uso eficiente de los recursos computacionales disponibles, empleando Inteligencia Artificial para lograrlo.

Es por ello que, se pretende proponer una alternativa al algoritmo estándar para la estimación del RTT, debido a que asignar al RTT un tiempo elevado incrementa la latencia de la transmisión, y al contrario de ello, asignarle un tiempo demasiado bajo conlleva a retransmisiones innecesarias e incrementa la congestión del canal.

Además de lo mencionado, este proyecto documentará nuevas variables para el cálculo del RTT y nuevos datos, obtenidos a partir de los resultados de la ejecución de la parte experimental del proyecto, se suma el hecho de que contribuye a futuras investigaciones en la rama de experimentación sobre los protocolos de comunicación fusionados con las RNA. Una vez que se haya satisfecho el objetivo de la presente investigación, consecuentemente, se podrá analizar la probabilidad de poder darle alguna utilidad.

## 1.7. Metodología de la tesis

### 1.7.1. Tipo de investigación

El presente proyecto se enmarca en la investigación aplicada. Por cuanto pretende proporcionar una alternativa a la estimación estadística del RTT en una asociación SCTP.

Se plantea modificar el algoritmo de cálculo del periodo de tiempo asignado al RTT durante una transmisión SCTP, todo ello aplicando el conocimiento adquirido para luego, expresarlo en la identificación de las variables, puesta en ejecución, y finalmente sea propuesto como una alternativa al algoritmo estándar.

### 1.7.2. Nivel de investigación

La investigación expuesta pertenece a un estudio de alcance descriptivo, pues se especificará en el algoritmo que hace el cálculo del RTT del protocolo de comunicación SCTP. Para después ser sometido a un análisis y manipulación del mismo, haciendo uso de las RNA, que nos permita recoger información independiente sobre las variables alteradas.

### 1.7.3. Método de investigación

El diseño de la investigación será experimental, pues según Hernández Sampieri, Fernandez Collado y Baptista Lucio (2018) este diseño está definido como un estudio en el que se manipulan intencionalmente una o más variables independientes (supuestas causas antecedentes), para analizar las consecuencias que la manipulación tiene sobre una o más



variables dependientes (supuestos efectos consecuentes), dentro de una situación de control para el investigador. (p.129)

A partir de lo mencionado en este estudio, se manifiesta la realización de pruebas que muestren en los resultados, una aproximación del valor asignado al RTT por parte del algoritmo de estimación propuesto basado en el cálculo de tiempo empleando las RNA, y el algoritmo de estimación estadístico del protocolo de transporte SCTP estándar.



1.8. Matriz de Consistencia

Tabla 2

*Matriz de consistencia.*

Problema	Objetivo	Hipótesis	Variables
Problema general	Objetivo general		
¿Cómo implementar un método de predicción del RTT basado en RNA, alternativo al método de predicción estándar del protocolo SCTP en la transmisión de flujos multimedia?	Proponer la implementación de un método de predicción del RTT basado en RNA, alternativo al método de predicción estándar del protocolo SCTP en la transmisión de flujos multimedia.	El método de predicción del RTT basado en RNA, es una alternativa al método de predicción estándar del protocolo SCTP en la transmisión de flujos multimedia.	<ul style="list-style-type: none"> <li>• Variable dependiente: Predicción</li> </ul>
Problemas específicos	Objetivos específicos		
<ul style="list-style-type: none"> <li>• ¿Qué variables permitirán calcular el valor asignado al RTT?</li> <li>• ¿Qué modelo de predicción emplear para la estimación del valor del RTT utilizando RNA?</li> <li>• ¿Puede modificarse el método de predicción del RTT en el protocolo de comunicación SCTP utilizando las RNA?</li> <li>• ¿Cuál es la predicción del método estándar y el método propuesto?</li> </ul>	<ul style="list-style-type: none"> <li>• Determinar variables que permitan calcular el valor asignado al RTT.</li> <li>• Diseñar el modelo de predicción del cálculo del RTT, empleando las RNA.</li> <li>• Implementar las RNA para la predicción del RTT en el protocolo SCTP.</li> <li>• Evaluar la predicción del método estándar y el método propuesto.</li> </ul>		<ul style="list-style-type: none"> <li>• Variable independiente: RTT</li> </ul>

Nota. Fuente de elaboración propia.



## CAPÍTULO II: MARCO TEÓRICO

### 2.1. Aspectos teóricos pertinentes

#### 2.1.1. Redes informáticas

Se le llama red de computadoras, a un conjunto de computadoras autónomas interconectadas mediante una sola tecnología. Se dice que dos computadoras están interconectadas si pueden intercambiar información. La conexión no necesita ser a través de un cable de cobre; también se puede utilizar fibra óptica, microondas, infrarrojos y satélites de comunicaciones (Tanenbaum & Wetherall, Computer Networks, 2021).

Internet es una red informática que interconecta cientos de millones de dispositivos informáticos en todo el mundo (Kurose & Ross, Computer Networking a Top-Down Approach, 2021).

La complejidad de las redes de sistemas modernas ha dado origen a varios modelos conceptuales para explicar cómo funcionan las redes.

Entre estos modelos, uno de los más comunes es el Modelo de referencia OSI (Open Systems Interconnection o Interconexión de sistemas abiertos) de la International Standards Organization (Organización Internacional de Normas o ISO), que también se conoce como modelo de siete capas OSI.

Cada capa describe una función determinada que se produce en las comunicaciones de datos. Las siete capas funcionan del nivel más bajo (nivel de máquina) al nivel más alto (el nivel en el que tiene lugar la mayor parte de la interacción de las personas) (IBM, 2021).

#### 2.1.2. Modelo de referencia OSI

También conocido como el modelo OSI. Este modelo se basa en una propuesta desarrollada por la Organización Internacional de Normas, como el primer paso hacia la estandarización internacional de los protocolos utilizados en las diversas capas (Day y Zimmerman, 1983). Este modelo se revisó en 1995 (Day, 1995) y se le llama Modelo de referencia OSI, puesto que se ocupa de la conexión de sistemas abiertos; esto es, sistemas que están abiertos a la comunicación con otros sistemas (Tanenbaum & Wetherall, Computer Networks, 2021).

El modelo OSI está conformado por siete capas:

- Capa de Aplicación.
- Capa de Presentación.
- Capa de Sesión.
- Capa de Transporte.

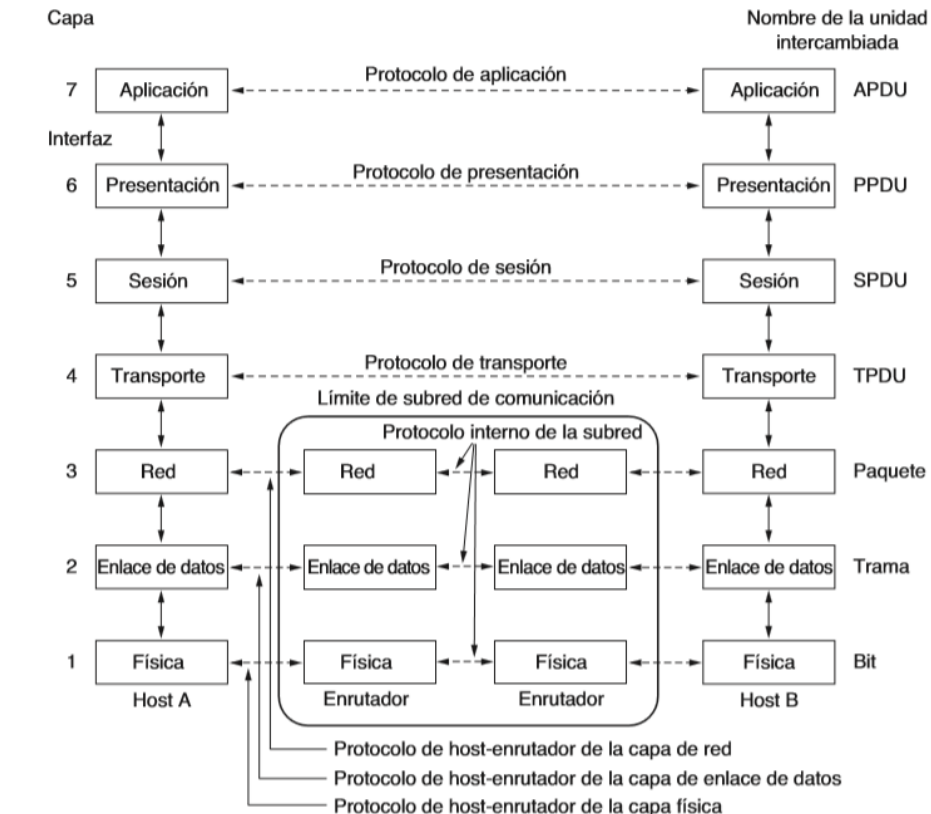


- Capa de Red.
- Capa de Enlace de Datos.
- Capa Física.

Se puede observar la disposición de las siete capas que forman parte de este modelo de referencia en la Figura 1 mostrada a continuación.

Figura 1

Modelo de referencia OSI



Nota. Arquitectura de redes propuesta a través de siete capas. Tomado de *Computer Networks* por A. S. Tanenbaum, y D. J. Wetherall, 2021, (edición n° 6, p. 36) <https://www.studocu.com/en-us/document/california-university-of-pennsylvania/computer-programming-i/computer-networks-6th-edition-by-andrew-s-tanenbaum/23608956> . Derechos reservados 2021 por Pearson Education.

### 2.1.3. Protocolos de Red

Un protocolo define el formato y el orden de los mensajes intercambiados entre dos o más entidades que se comunican, así como las acciones tomadas en la transmisión y / o recepción de un mensaje u otro evento. Toda actividad en Internet que involucre dos o más entidades remotas que se comunican se rige por un protocolo.

Internet y las redes de computadoras en general hacen un uso extenso de los protocolos. Se usan diferentes protocolos para realizar diferentes tareas de comunicación (Kurose & Ross , *Computer Networking a Top-Down Approach*, 2021).

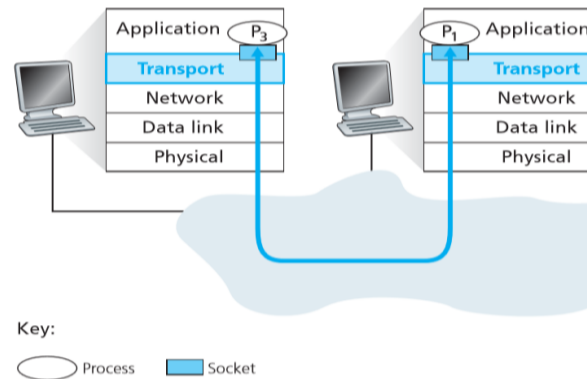


#### 2.1.4. La capa de transporte

Para Kurose y Ross (2021) la capa de transporte tiene la función fundamental de proporcionar servicios de comunicación directamente a los procesos de aplicación que se ejecutan en diferentes hosts, explicado gráficamente en la Figura 2.

Figura 2

Comunicación en capa de transporte



Nota. La imagen mostrada expresa la comunicación en capa de transporte. Tomado de *Computer Networking a Top-Down Approach* por J. F. Kurose, y K. W. Ross, 2021, (edición n° 8, p. 192) <https://www.pearson.com/us/higher-education/program/Kurose-Pearson-e-Text-Computer-Networking-Access-Card-8th-Edition/PGM2877610.html> . Derechos reservados 2021 por Pearson Education.

Las aplicaciones hacen uso de la comunicación a nivel lógico que provee la capa de transporte, para enviarse mensajes entre los extremos en comunicación. Todo ello sin preocuparse por los detalles de las redes intermedias (Kurose & Ross , 2021).

Existen dos tipos básicos de servicio de transporte: orientado a la conexión y sin conexión o servicio de datagrama. Un servicio orientado a la conexión proporciona el establecimiento, mantenimiento y terminación de una conexión lógica entre usuarios TS (Transport Service users o usuarios del Servicio de Transporte). El servicio orientado a la conexión generalmente implica que el servicio es confiable (Kurose & Ross , 2021).

#### 2.1.5. Mecanismos del protocolo de transporte orientado a la conexión

Los mecanismos requeridos para el transporte de datos entre dos puntos terminales son directamente dependientes de la confiabilidad que presenta el servicio de red, según ello pueden ser disgregados como: Servicio de red confiable o Servicio de red no confiable.

Sin embargo, se profundizará en el servicio de red no confiable, en vista de que bajo ese contexto es donde se emplean los conceptos del RTT.



#### 2.1.5.1. Servicio de red confiable

Se asume que el servicio de red acepta mensajes con longitud arbitraria y con una confiabilidad total, entrega los mensajes en orden secuencial al destino. Para este tipo de servicio solo es necesario abordar cuatro mecanismos:

- Direccionamiento.
- Multiplexado.
- Control de Flujo.
- Establecimiento y terminación de la conexión.

#### 2.1.5.2. Servicio de red no confiable

Este contexto implica que el problema no es únicamente el de los segmentos ocasionalmente perdidos, estos segmentos pueden llegar fuera de secuencia junto con tiempos de demora variables. Para abordar estas deficiencias fue necesario elaborar algunos nuevos mecanismos como se verá a continuación:

- Entrega ordenada
- Estrategia de retransmisión.
- Detección de duplicados.
- Control de Flujo.
- Establecimiento de la conexión.
- Terminación de la conexión.
- Recuperación de fallas.

De los mecanismos mencionados, únicamente para este proyecto será tomado el de estrategia de retransmisión, debido a que está enmarcado dentro de él.

##### 2.1.5.2.1. Estrategia de retransmisión

La retransmisión de segmentos se da bajo dos circunstancias. Primero, cuando un segmento puede ser dañado en tránsito, pero llega al destino y la segunda contingencia es cuando el segmento no llega al destino.

Para afrontar estas contingencias es usado el esquema de confirmación de recibo (acknowledgment o ACK). El terminal que recibe el segmento debe confirmar la recepción exitosa empleando segmentos que contengan un ACK con número de secuencia correspondiente. Si un segmento no es transmitido con éxito, ningún ACK será enviado y la retransmisión estará en orden.

Para abordar esta situación, se tiene un Timer (Temporizador) asociado con cada segmento enviado, si el Timer expira antes de que el segmento sea confirmado, se hace una retransmisión.





Otro problema, derivado de este mecanismo es el tiempo que se asignará al Timer. Si el valor es muy pequeño, habrá transmisiones innecesarias, desperdiciando la capacidad del canal, en el caso contrario, el protocolo será lento respondiendo a un segmento perdido.

Existen dos estrategias para determinar el tiempo asignado al Timer, un valor fijo o un valor adaptativo. Establecer el tiempo bajo un esquema adaptativo se basa en un promedio de las demoras observadas, este valor puede no ser real porque las condiciones de la red pueden cambiar espontáneamente.

#### 2.1.6. Stream Control Transmission Protocol (SCTP)

El protocolo SCTP, es un protocolo orientado a la conexión, similar a TCP, pero proporciona transferencia de datos orientada a mensajes, similar a UDP.

En general, SCTP puede brindar más flexibilidad para ciertas aplicaciones, como Voice Over IP (VoIP), que requieren una transferencia de datos confiable pero orientada a mensajes. Para esta categoría de aplicaciones, lo más probable es que SCTP sea más adecuado que TCP o UDP. TCP proporciona una entrega de datos en orden de transmisión estricto y confiable. Para las aplicaciones que necesitan confiabilidad, pero que pueden tolerar la entrega de datos desordenados o parcialmente ordenados, TCP puede causar demoras innecesarias debido al bloqueo de cabecera de línea. Con el concepto de múltiples flujos dentro de una sola conexión, SCTP puede proporcionar una entrega estrictamente ordenada dentro de un flujo mientras aísla lógicamente los datos de diferentes flujos.

SCTP está orientado a mensajes, a diferencia de TCP, que está orientado a bytes. Debido a la naturaleza orientada a bytes de TCP, la aplicación debe agregar su propia marca de registro para mantener los límites del mensaje.

SCTP proporciona cierto grado de tolerancia a fallas mediante el uso de la función Multihoming. Un host se considera multitarjeta cuando tiene más de una interfaz de red conectada, ya sea en la misma red o en redes diferentes. Se puede establecer una asociación SCTP entre dos hosts multitarjeta. En este caso, todas las direcciones IP de ambos extremos se intercambian al inicio de la asociación; esto permite que cada extremo use cualquiera de estas direcciones durante la vida útil de la conexión si una de las interfaces está inactiva por algún motivo, siempre que se pueda acceder al par a través de las interfaces alternativas.

SCTP proporciona funciones de seguridad adicionales que TCP y UDP no ofrecen. En SCTP, la asignación de recursos durante la configuración de la asociación se retrasa hasta que se puede verificar la identidad del cliente mediante un mecanismo de intercambio de cookies, lo que reduce la posibilidad de ataques de denegación de servicio (IBM, 2022).



Este protocolo tiene como último documento referido a sus especificaciones el RFC 4960 (IBM, 2022).

#### 2.1.7. Características Principales

Según IBM (2022), el protocolo presenta las siguientes características generales:

- Entrega secuenciada dentro de flujos:  
Un flujo en el contexto SCTP se refiere a una secuencia de mensajes de usuario que se transfieren entre puntos finales. Una asociación SCTP puede admitir varios flujos. Dentro de cada flujo se mantiene estrictamente el orden de entrega de datos. Sin embargo, a través de los flujos, la entrega de datos es independiente. Por lo tanto, la pérdida de datos de un flujo no impide que los datos se entreguen en otro flujo.
- Fragmentación de datos de usuario:  
SCTP puede fragmentar los mensajes de usuario para garantizar que el tamaño del paquete pasado a la capa inferior no supere la MTU de la ruta. En el momento de la recepción, los fragmentos se vuelven a ensamblar en un mensaje completo y se pasan al usuario.
- Acknowledgment y Control de Congestión:  
El reconocimiento de paquetes es necesario para una entrega de datos confiable. Cuando SCTP no obtiene un reconocimiento de un paquete que envía dentro de un tiempo específico, desencadena una retransmisión del mismo paquete. SCTP sigue algoritmos de control de congestión similares a los utilizados por TCP. Además de utilizar reconocimientos acumulativos como TCP, SCTP utiliza el mecanismo de reconocimiento selectivo (SACK) que le permite reconocer paquetes de forma selectiva.
- Agrupación de Chunks:  
Un Chunk puede contener datos de usuario o información de control SCTP. Se pueden agrupar varios fragmentos bajo el mismo encabezado SCTP. La agrupación de fragmentos requiere el ensamblaje de fragmentos en un paquete SCTP en el extremo de envío y, posteriormente, el desensamblaje del paquete en fragmentos en el extremo del receptor.
- Validación de paquetes:  
Cada paquete SCTP tiene un campo de etiqueta de verificación que se establece durante el tiempo de inicio de la asociación por parte de cada extremo. Todos los paquetes se envían con la misma etiqueta de verificación durante el tiempo de vida de la asociación. Si, durante la vigencia de la asociación, se recibe un paquete con una etiqueta de



verificación inesperada, el paquete se descarta, lo mismo en el caso de una suma de comprobación CRC-32 no valida.

- Gestión de rutas:

En el momento de la configuración de la asociación, cada punto final puede anunciar la lista de direcciones de transporte que tiene. Sin embargo, solo se define una ruta principal para la asociación SCTP y se utiliza para la transferencia de datos normal. En caso de que la ruta principal se caiga, se utilizan las otras direcciones de transporte. Durante el tiempo de vida de la asociación, se envían HEARTBEATS a intervalos regulares a través de todas las rutas para monitorear el estado de la ruta.

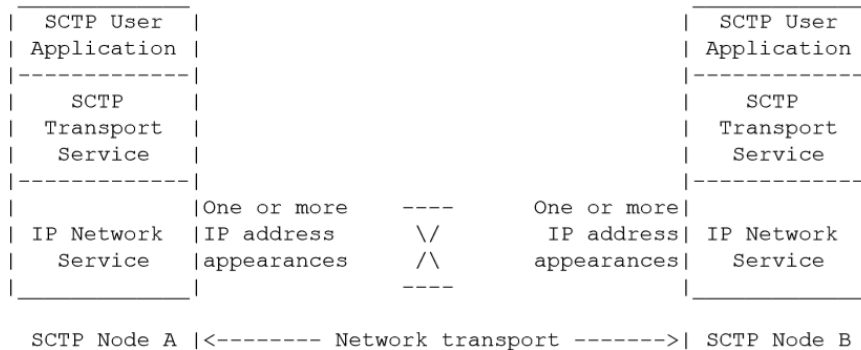
### 2.1.8. Arquitectura SCTP

SCTP es visto como una capa entre la aplicación de usuario SCTP ("usuario SCTP" para abreviar) y un servicio de red de paquetes sin conexión como IP. El servicio básico ofrecido por SCTP es la transferencia confiable de mensajes de usuario entre usuarios SCTP iguales. Realiza este servicio dentro del contexto de una asociación entre dos puntos finales SCTP.

SCTP está orientado a la conexión por naturaleza, pero la asociación SCTP es un concepto más amplio que la conexión TCP. SCTP proporciona los medios para que cada punto extremo SCTP proporcione al otro punto final (durante el inicio de la asociación) una lista de direcciones de transporte (es decir, varias direcciones IP en combinación con un puerto SCTP) a través de las cuales se puede llegar y desde donde se originará Paquetes SCTP. La asociación abarca transferencias sobre todas las posibles combinaciones de fuente / destino que pueden generarse a partir de las listas de cada punto final (Stewart, 2007).

Figura 3

Asociación SCTP.



Nota. La imagen anterior muestra la asociación del protocolo SCTP, dividido en capas de aplicación, transporte y capa de red. Tomado de "Stream Control Transmission Protocol" por R. Stewart, 2007, 4960, p. 6. <https://tools.ietf.org/pdf/rfc4960.pdf>. Obra de dominio público.



### 2.1.9. Estructura de segmento SCTP

Según Juniper Networks (2021), un paquete SCTP consta de las siguientes secciones:

#### 2.1.9.1. Sección de encabezado común

Todos los paquetes SCTP requieren una sección de encabezado común. Esta sección ocupa los primeros 12 bytes del paquete. A continuación se describen los campos en la sección:

- Número de puerto de origen (Source port number): Identifica el puerto de envío.
- Número de puerto de destino (Destination port number): Identifica el puerto receptor. Los hosts utilizan el número de puerto de destino para enrutar el paquete al destino o aplicación correspondiente.
- Etiqueta de verificación (Verification tag): Distingue los paquetes obsoletos de una conexión anterior. Este es un valor aleatorio de 32 bits creado durante la inicialización.
- Suma de verificación (Checksum): Utiliza el algoritmo de verificación de redundancia cíclica (CRC32) para detectar errores que podrían haberse introducido durante la transmisión de datos.

#### 2.1.9.2. Sección de fragmentos de datos

Esta sección ocupa la parte restante del paquete. A continuación son descritos los campos en esta sección:

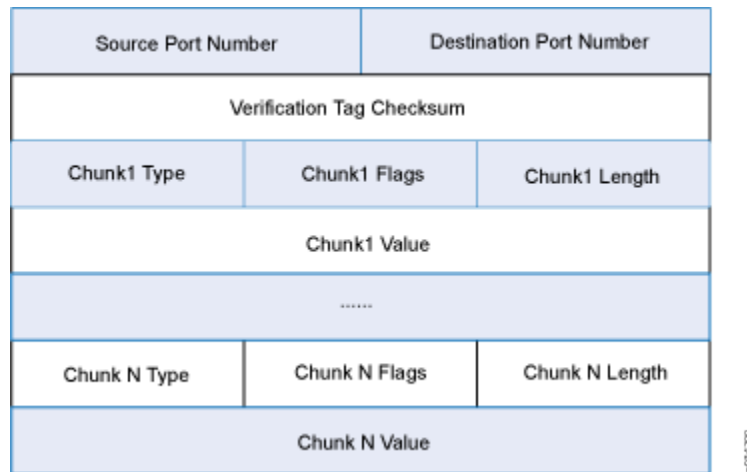
- Tipo de Chunk (Chunk Type): Identifica el contenido del campo de valor del Chunk.
- Banderas de Chunk (Chunk Flags): Consta de 8 flag-bits cuya definición varía según el tipo de Chunk.
- Longitud del Chunk (Chunk Length): Especifica la longitud total del Chunk en bytes.
- Valor de Chunk (Chunk Value): Un campo de datos de propósito general.

De forma gráfica se puede ver la estructura de los paquetes SCTP en la figura mostrada a continuación.



Figura 4

Estructura de segmentos SCTP.



Nota. La figura muestra la estructura de un paquete SCTP. Tomado de “Securing GTP and SCTP Traffic User Guide for Security Devices” por Juniper Networks, 2021. <https://www.juniper.net/documentation/us/en/software/junos/gtp-sctp/topics/topic-map/security-gprs-sctp.html> . Derechos reservados por Juniper Networks.

Es preciso indicar que, el número de chunks en un paquete SCTP está determinado por el tamaño de MTU del camino de transmisión (Palo Alto Networks, 2022).

#### 2.1.10. Paquetes SCTP

Según Oracle (2022), los paquetes SCTP consisten en un encabezado común y uno o más Chunks, cada uno de los cuales tiene un propósito específico.

- DATA chunk: Transporta datos de usuario.
- INIT chunk: Inicia una asociación entre puntos finales SCTP.
- INIT ACK chunk: Reconoce el establecimiento de la asociación.
- SACK chunk: Reconoce los fragmentos de DATOS recibidos e informa al extremo del par sobre las lagunas en las subsecuencias recibidas de los fragmentos DATA.
- HEARTBEAT chunk: Prueba la accesibilidad de un punto final SCTP.
- HEARTBEAT ACK chunk: Reconoce la recepción de un fragmento HEARTBEAT.
- ABORT chunk: Fuerza el cierre inmediato de una asociación.
- SHUTDOWN chunk: Inicia un cierre elegante de una asociación.
- SHUTDOWN ACK chunk: Reconoce la recepción de un fragmento SHUTDOWN.
- ERROR chunk: Informa varias condiciones de error.
- COOKIE ECHO chunk: Utilizado durante el proceso de establecimiento de la asociación.
- COOKIE ACK chunk: Reconoce la recepción de un fragmento COOKIE ECHO
- SHUTDOWN COMPLETE chunk: Completa un cierre de asociación elegante.

### 2.1.11. Establecimiento de la asociación

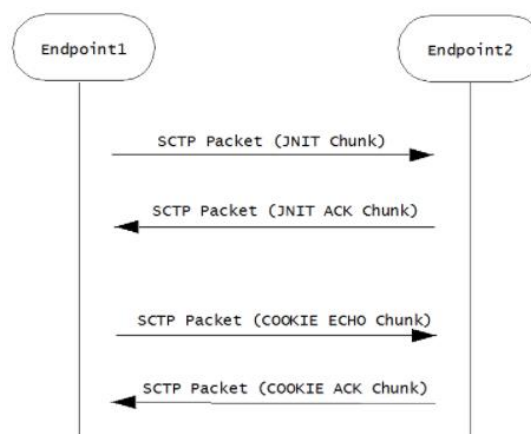
Según IBM (2022), el establecimiento de la asociación bajo el protocolo SCTP, se compone de un Four-Way Handshake (Apretón de Manos de Cuatro Vías) que se lleva a cabo en el siguiente orden:

- El cliente envía una señal INIT al servidor para iniciar una asociación.
- Al recibir la señal INIT, el servidor envía una respuesta INIT-ACK al cliente. Esta señal INIT-ACK contiene una cookie de estado. Esta cookie de estado debe contener un Código de Autenticación de Mensaje (MAC), junto con una marca de tiempo correspondiente a la creación de la cookie, la duración de la cookie de estado y la información necesaria para establecer la asociación. El servidor calcula el MAC en función de una clave secreta que solo él conoce.
- Al recibir esta señal INIT-ACK, el cliente envía una respuesta COOKIE-ECHO, que simplemente refleja la cookie de estado.
- Después de verificar la autenticidad de la cookie de estado utilizando la clave secreta, el servidor asigna los recursos para la asociación, envía una respuesta COOKIE-ACK reconociendo la señal COOKIE-ECHO y mueve la asociación al estado ESTABLISHED.

Este proceso puede ser visualizado en la Figura 6.

Figura 5

*Establecimiento de asociación SCTP.*



Nota. Se representa los paquetes que son enviados para establecer una asociación. Tomado de “Stream Control Transfer Protocol Overview” por Oracle, 2022, <https://docs.oracle.com/en/industries/communications/enterprise-session-border-controller/8.3.0/configuration/stream-control-transfer-protocol-overview.html#GUID-D3EE9CF5-2A18-45BB-9E5B-9BC1D584EE0D>.



#### 2.1.12. Transmisiones durante una asociación

La transmisión de datos debe ocurrir sólo en los estados ESTABLISHED (Establecido), SHUTDOWN-PENDING (Apagado Pendiente), y SHUTDOWN-RECEIVED (Apagado Recibido). La única excepción a esto es que los fragmentos de datos pueden agruparse con un fragmento COOKIE-ECHO de salida cuando están en el estado COOKIE-WAIT.

Los fragmentos de datos deben recibirse sólo de acuerdo con las siguientes reglas: ESTABLISHED, SHUTDOWN-PENDING, y SHUTDOWN-SENT. Un fragmento de datos recibido en estado CLOSED (cerrado) no se vence. Un fragmento de datos recibido en cualquier otro estado debería descartarse.

Este manejo de los fragmentos también es replicado en el manejo de los SACK (Selective Acknowledgment)

Un SACK debe procesarse en un estado de ESTABLISHED, SHUTDOWN-PENDING, y SHUTDOWN-RECEIVED. Un SACK entrante puede procesarse en COOKIE-ECHOED. Un SACK en el estado CLOSED no se vence. Un trozo SACK recibido en cualquier otro estado debe ser descartado.

Un receptor SCTP debe poder recibir un mínimo de 1500 bytes en un paquete SCTP. Esto significa que un punto final SCTP no debe indicar menos de 1500 bytes en su `a_rwnd` (Advertised Receiver Window) inicial enviado en INIT o INIT ACK.

Para la eficiencia de la transmisión, SCTP define mecanismos para la agrupación de mensajes de usuarios pequeños y la fragmentación de mensajes de usuarios grandes.

En esta sección, el término "emisor de datos" se refiere al punto extremo que transmite un fragmento de DATOS y el término "receptor de datos" se refiere al punto final que recibe un fragmento de DATOS. Un receptor de datos transmitirá fragmentos SACK.

Al convertir los mensajes del usuario en fragmentos de datos, un punto final fragmentará los mensajes del usuario más grandes que la MTU de la ruta de asociación actual en múltiples fragmentos de datos. El receptor de datos normalmente reensamblará el mensaje fragmentado de los fragmentos de datos antes de la entrega al usuario.

El remitente puede agrupar múltiples datos y fragmentos de control en un único paquete SCTP para la transmisión, siempre que el tamaño final del paquete no exceda la MTU (Maximum Transmission Unit) de la ruta actual. El receptor desagregará el paquete en los trozos originales. Los trozos de control deben venir antes de los trozos de datos en el paquete.

Los mecanismos de fragmentación y agrupamiento son opcionales para ser implementados por el emisor de datos, pero deben ser implementados por el receptor de datos, es decir, un punto

extremo debe recibir y procesar correctamente los datos agrupados o fragmentados (Stewart, 2007).

### 2.1.13. Liberación de una asociación

Según IBM (2022), SCTP también admite el cierre correcto de una asociación activa a pedido del usuario SCTP. Ocurre mediante la siguiente secuencia de eventos:

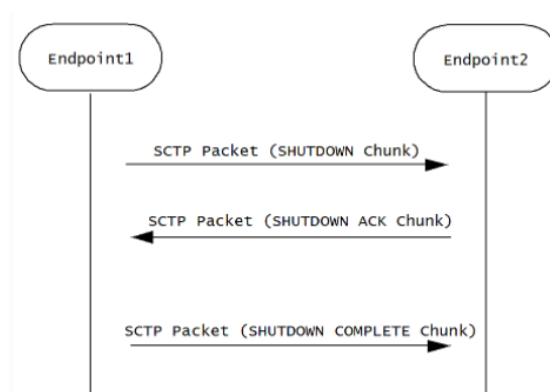
- El cliente envía una señal de SHUTDOWN al servidor, que le dice que el cliente está listo para cerrar la conexión.
- El servidor responde enviando un acuse de recibo SHUTDOWN-ACK.
- Luego, el cliente envía una señal de SHUTDOWN-COMPLETE al servidor.

SCTP también admite el cierre abrupto (señal ABORT) de una asociación activa a solicitud del cliente SCTP o debido a un error en la pila SCTP. Sin embargo, SCTP no admite conexiones semiabiertas.

Ello puede ser observado en la Figura 6 mostrada a continuación.

Figura 6

*Cierre de asociación SCTP.*



Nota. El cierre de la asociación SCTP se desarrolla de la forma mostrada anteriormente. Tomado de “Stream Control Transfer Protocol Overview” por Oracle, 2022, <https://docs.oracle.com/en/industries/communications/enterprise-session-border-controller/8.3.0/configuration/stream-control-transfer-protocol-overview.html#GUID-D3EE9CF5-2A18-45BB-9E5B-9BC1D584EE0D>.

### 2.1.14. Manejo de Retransmisiones

Un punto final SCTP utiliza un temporizador de retransmisión T3-rtx para garantizar la entrega de datos en ausencia de cualquier retroalimentación de su par. La duración de este temporizador se conoce como RTO (tiempo de espera de retransmisión).

Cuando el par de un punto extremo es multi-homed, el punto extremo calculará un RTO separado para cada dirección de transporte de destino diferente de su punto extremo par.

El cálculo y la gestión de RTO en SCTP siguen de cerca cómo TCP gestiona su temporizador de retransmisión. Para calcular el RTO actual, un punto final mantiene dos variables de estado





por dirección de transporte de destino: SRTT (tiempo de ida y vuelta suavizado) y RTTVAR (variación de tiempo de ida y vuelta) (Stewart, 2007).

#### 2.1.15. RTT

Según la compañía de investigación y asesoría en el rubro de Tecnologías de la Información, Gartner (2018), El RTT mide (en milisegundos) la latencia de una red, es decir, el tiempo transcurrido entre el inicio de una solicitud de red y la recepción de una respuesta. La alta latencia tiende a tener un mayor impacto que el ancho de banda en la experiencia del usuario final en aplicaciones interactivas, como la navegación web.

Kurose y Ross (2021) mencionan que: La estimación del tiempo de ida y vuelta entre el emisor y el receptor se consigue de la siguiente manera. El RTT de muestra, denominado RTT o SampleRTT, para un segmento es la cantidad de tiempo entre el momento en que se envía el segmento y cuando se recibe un acuse de recibo para el segmento. En cualquier punto en el tiempo, el SampleRTT se está estimando para solo uno de los segmentos transmitidos, lo que conduce a un nuevo valor de SampleRTT aproximadamente una vez por cada RTT.

Obviamente, los valores de SampleRTT fluctuarán de un segmento a otro debido a la congestión en los enrutadores y a la carga variable en los sistemas finales. Debido a esta fluctuación, cualquier valor de SampleRTT puede ser atípico. Para estimar un RTT típico, es natural tomar algún tipo de promedio de los valores de SampleRTT. Este promedio es llamado SRTT o EstimatedRTT. Al obtener un nuevo SampleRTT, se actualiza EstimatedRTT de acuerdo con la siguiente fórmula:

$$EstimatedRTT = (1 - \alpha) \times EstimatedRTT + \alpha \times SampleRTT$$

La fórmula anterior está escrita en forma de una declaración de lenguaje de programación: el nuevo valor de EstimatedRTT es una combinación ponderada del valor anterior de EstimatedRTT y el nuevo valor de SampleRTT. El valor recomendado de  $\alpha$  es 0.125 (es decir, 1/8), en cuyo caso la fórmula anterior se convierte en:

$$EstimatedRTT = 0.875 \times EstimatedRTT + 0.125 \times SampleRTT$$

#### 2.1.16. RTO

Basado en el valor del RTT se realiza la estimación del temporizador denominado RTO. Este es el que indica el tiempo que se esperara por una respuesta de confirmación de envió de paquetes. Según Kurose y Ross (2021) se precisa que la estimación del valor de este temporizador recurre a la variación del RTT, denominada RTTVAR o DevRTT. El valor inicial del DevRTT es el cálculo de SampleRTT/2, posteriores transmisiones son calculadas mediante la siguiente formula:



$$DevRTT = (1 - \beta) \times DevRTT + \beta \times |SampleRTT - EstimatedRTT|$$

Lo cual indica que si el SampleRTT tiene fluctuaciones pequeñas, la variación también será pequeña; de lo contrario el DevRTT será alto. El valor por defecto de la constante  $\beta = 0.25$ . El valor de inicialización del RTO es igual a 1 segundo. Una vez obtenido el valor del DevRTT es que puede efectuarse la estimación del valor del RTO, a través de la siguiente formula:

$$RTO = EstimatedRTT + 4 \times DevRTT$$

#### 2.1.17. Redes Neuronales Artificiales (RNA)

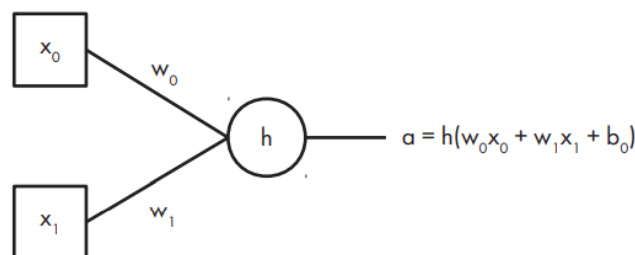
Las RNA son un modelo computacional inspirado en la estructura del cerebro humano, que consisten de una red unidades de procesamiento de información simples. Las unidades de procesamiento son llamadas neuronas, ellas se conectan a otras neuronas, lo cual permite la generación de modelos complejos. Las neuronas son alimentadas por la salida de otra neurona (a excepción de las neuronas de entrada), mediante una conexión que presenta un peso asociado. Dichos pesos afectan la forma en la que se procesara la información (Kelleher, 2019).

#### 2.1.18. Neurona genérica

Las RNA son universalmente funciones de aproximación. Ellas usan una estructura de grafo para representar una serie de pasos computacionales que procesan entradas, generando un valor de salida, típicamente interpretado como una probabilidad. Las RNA son construidas en capas, compuestas por nodos, llamados neuronas, que son conectados a través de vértices.

Figura 7

Neurona genérica



Nota. Tomado de Practical Deep Learning: A Python Based Introduction por R. Kneusel, 2021, (edición n° 1, p. 171).

Derechos reservados 2020 por No Starch Press, Inc.

Considerando que el flujo es de derecha a izquierda, es posible verse que la neurona “h” recibe entradas “ $x_0, x_1$ ”, mediante conexiones y sus pesos “ $w_0, w_1$ ”, para generar únicamente una salida “a”.

La neurona es la que implementa una función matemática llamada función de activación, la misma que calcula la salida del nodo, en base a las entradas y el bias. El bias es utilizado para ajustar el rango de entrada de modo que sea adecuado para la función de activación (Kneusel, 2021).



### 2.1.19. Topología de las RNA

La topología explica la forma en que los nodos están conectados, también es llamada arquitectura de la RNA. Las RNA están comúnmente construidas por capas, lo que simplifica su computación y facilita su entrenamiento.

Una red Feedforward tiene una sola capa de entrada, una o más capas ocultas, y una capa de salida. Las capas ocultas están conformadas por nodos que aceptan como entradas, las salidas de los nodos de capas anteriores. Las conexiones entre capas son típicamente, fully connected (conectadas completamente).

El número de capas ocultas y el número de neuronas de cada una definen la arquitectura de la red. Es probado que una sola capa oculta con la cantidad de neuronas necesarias, es suficiente para aprender el comportamiento de cualquier función. Sin embargo, esto no significa que sea recomendado.

Seleccionar una arquitectura de la red apropiada, tiene un gran impacto sobre el aprendizaje del modelo, más debido a que no se tienen directrices estrictas de cómo hacerlo. Sin embargo, algunas recomendaciones que son de utilidad se presentan a continuación:

- Si la entrada implica relaciones espaciales, como imágenes, se deben usar Redes Neuronales Convolucionales.
- Usar la cantidad de capas ocultas necesarias, lo recomendable es no más de tres.
- La cantidad de neuronas de la capa de entrada tiene que ser la misma que la cantidad de entradas a la red.

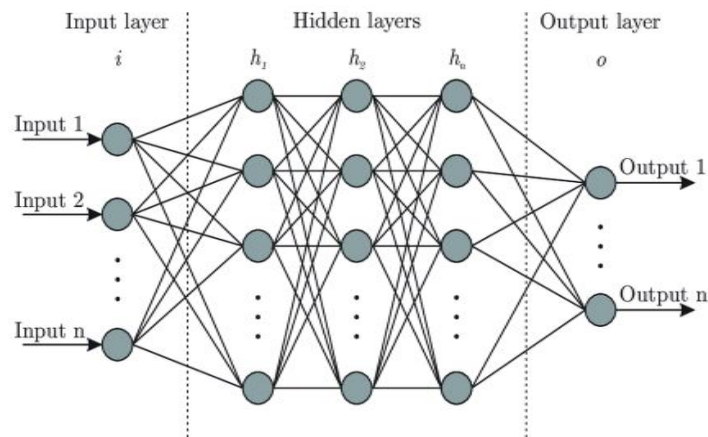
La última capa de la red es la capa de salida. En el caso de que la RNA este modelando un valor continuo, conocido como regresión, entonces el único nodo presente en dicha capa no usa ninguna función de activación más que la función identidad (Kneusel, 2021).

### 2.1.20. Perceptron Multicapa (Multilayer Perceptron)

Este tipo de arquitectura, conocida por sus siglas MLP, consiste de una combinación de unidades lógicas o neuronas organizadas en una o más capas. En esta RNA, el valor de salida de una neurona perteneciente a una capa, es el valor de entrada de cada una de las neuronas de la capa subsecuente. En MLP, todas las neuronas de capas continuas se encuentran conectadas. Usualmente todas las neuronas de una capa usan la misma función de activación, pero no es una regla. Cada capa puede tener diferente número de neuronas (Burkov, 2019).

Figura 8

Perceptron Multicapa



Nota. Tomado de Activation Functions in Neural Networks por S. Sharma y S. Sharma, 2020, (edición n° 1, p. 310)

<https://ijeast.com/papers/310-316,Tesma412,IJEAST.pdf> . Derechos reservados 2020 por IJEAST.

Las MLP consisten de nodos que se utilizan para la clasificación y predicción de datos, una vez proporcionados los datos de entrada a la red (Sharma & Sharma, 2020).

#### 2.1.21. Funciones de activación

La función de activación toma un solo valor de entrada. Recibe la suma de las entradas que fueron multiplicadas por sus respectivos pesos, juntamente con el bias. En particular, permite cálculos no lineales, de modo que el modelo pueda aprender funciones complejas.

La función de activación es alimentada por las entradas o también por la salida de neuronas que se encuentran en la capa anterior, haciendo uso de las conexiones que conjuntamente a sus pesos determinan el impacto que tiene cada entrada sobre la neurona. Al mismo tiempo es alimentada por el bias.

La forma en la que trabaja la neurona es recepcionando las entradas que son multiplicadas por los pesos de sus conexiones, para ser sumadas y dadas a la función de activación, produciéndose así la salida de la neurona.

Las RNA requieren de funciones de activación no lineales, de otra forma, ellas estarían solo disponibles para aprender mapeos lineales, y no serían útiles (Kneusel, 2021).

##### 2.1.21.1. Función Lineal

Esta función tiene su salida directamente proporcional a su entrada y su grafico es una línea recta. Esta puede ser definida por la función:

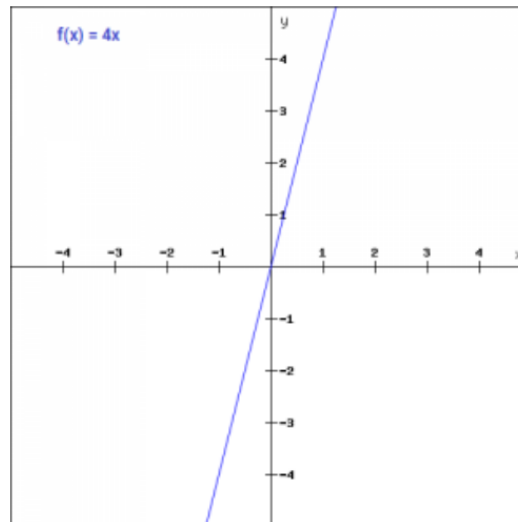
$$f(x) = ax$$



El valor de la constante  $a$  puede ser cualquier valor definido por el investigador. La siguiente figura muestra gráficamente, la función descrita:

Figura 9

*Función de activación Lineal*



Nota. Tomado de Activation Functions in Neural Networks por S. Sharma y S. Sharma, 2020, (edición n° 1, p. 310) <https://ijeast.com/papers/310-316,Tesma412,IJEAST.pdf> . Derechos reservados 2020 por IJEAST.

El gradiente no es cero, sino un valor constante que es independiente del valor de entrada  $x$ , lo que implica que los pesos y los sesgos se actualizarán durante el paso de retropropagación aunque el factor de actualización será el mismo.

No hay mucho beneficio de usar la función lineal porque la red neuronal no mejoraría el error debido al mismo valor de gradiente para cada iteración. Además, la red no podrá identificar patrones complejos de los datos (Sharma & Sharma, 2020).

#### 2.1.21.2. Función sigmoidea

Es la función de activación más utilizada ya que es una función no lineal. La función sigmoidea transforma los valores en el rango de 0 a 1. Se puede definir como:

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

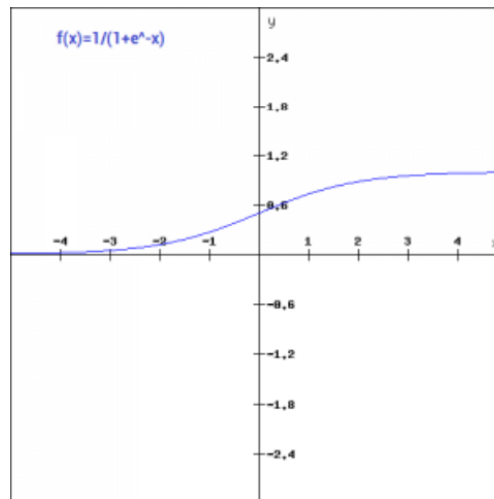
La función sigmoidea es continuamente diferenciable por la forma de S suave. Además, la función sigmoidea no es simétrica con respecto a cero lo que significa que los signos de todos los valores de salida de las neuronas serán los mismos (Sharma & Sharma, 2020).



La figura siguiente describe el comportamiento de esta función:

Figura 10

*Función de activación Sigmoidea*



Nota. Tomado de Activation Functions in Neural Networks por S. Sharma y S. Sharma, 2020, (edición n° 1, p. 310) <https://ijeast.com/papers/310-316,Tesma412,IJEAST.pdf> . Derechos reservados 2020 por IJEAST.

### 2.1.21.3. Función tangente hiperbólica

La función tangente hiperbólica o Tanh es similar a la función sigmoidea pero es simétrica alrededor del origen. Esto da como resultado diferentes señales de salidas de capas anteriores que se alimentarán como entrada a la siguiente capa. Se puede definir como:

$$f(x) = \left( \frac{2}{1 + e^{-2x}} \right) - 1$$

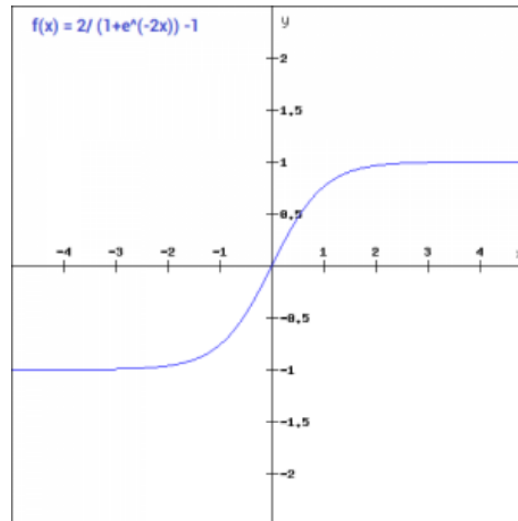
Dicha función es continua y diferenciable, los valores se encuentra en el rango de -1 a 1. En comparación con la función sigmoidea, el gradiente de la función Tanh es más pronunciado (Sharma & Sharma, 2020).



A continuación se muestra la función Tanh, gráficamente:

Figura 11

*Función de activación Tangente Hiperbólica*



Nota. Tomado de Activation Functions in Neural Networks por S. Sharma y S. Sharma, 2020, (edición n° 1, p. 310)  
<https://ijeast.com/papers/310-316,Tesma412,IJEAST.pdf> . Derechos reservados 2020 por IJEAST.

#### 2.1.21.4. Función ReLU

ReLU significa unidad lineal rectificadora y es una función de activación no lineal que al igual que la función sigmoidea, se usa ampliamente en redes neuronales. La ventaja de usar la función ReLU es que no todas las neuronas se activan al mismo tiempo. Esto implica que una neurona se desactivará sólo cuando la salida de la transformación lineal es cero. Se puede definir matemáticamente como:

$$f(x) = x, x \geq 0 \\ = 0, x < 0$$

En esta función no se activan todas las neuronas al mismo tiempo, se activa un cierto número de neuronas a la vez.

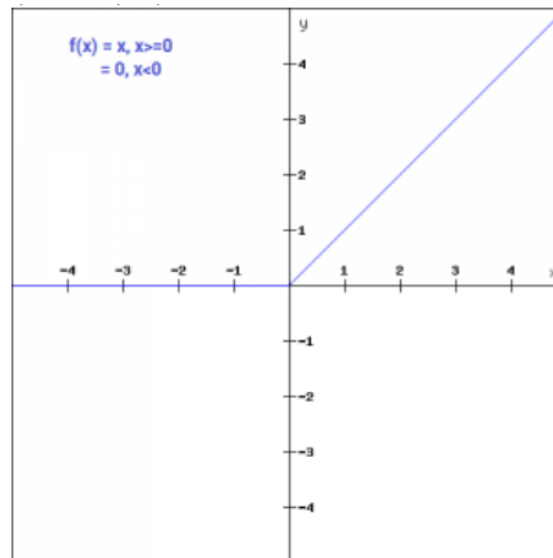
En algunos casos, el valor de gradiente es cero, por lo que los pesos y sesgos no se actualizan durante el paso de retropropagación en el entrenamiento de la red neuronal (Sharma & Sharma, 2020).



La figura relacionada a esta función se muestra a continuación:

Figura 12

*Función de activación ReLU*



Nota. Tomado de Activation Functions in Neural Networks por S. Sharma y S. Sharma, 2020, (edición n° 1, p. 310) <https://ijeast.com/papers/310-316,Tesma412,IJEAST.pdf> . Derechos reservados 2020 por IJEAST.

### 2.1.22. Algoritmo Gradiente Descendiente

Es un algoritmo que permite inicializar y modificar los pesos, y los bias de la RNA, con el objetivo de reducir el error sobre el conjunto de datos de entrenamiento. Se hace uso del valor promedio de la función de pérdida (loss function), para medir el error, el cual indica que tan errónea es la red. Debido a que se desea un modelo que cometa pocos errores, se necesita encontrar el valor de los parámetros que lleven a un menor valor en la función de pérdida, búsqueda de mínimos. Es posible conocerse si la red es correcta o errónea en base al valor esperado correspondiente a cada grupo de entrada.

El gradiente descendiente (gradient descent), hace uso de gradientes, que son medidas de inclinación. Cuanto mayor sea el gradiente, más pronunciada esta la función en ese punto. Para usar el descenso gradiente en búsqueda del valor más pequeño de la función de pérdida, necesitamos poder encontrar el gradiente, ello es posible a través del algoritmo backpropagation (propagación hacia atrás) (Kneusel, 2021).

### 2.1.23. Algoritmo de Retropropagación (Backpropagation)

Este es un algoritmo fundamental para las RNA, porque es el que les permite aprender. Este proporciona las gradientes necesarias para iniciar en la salida de la red, y moverse hacia atrás a través de la red, hacia las entradas; al mismo tiempo propaga el error desde la función de pérdida a cada parámetro de la red. En el camino, calcula el valor del gradiente para cada peso y el bias.





Tanto el algoritmo de descenso gradiente como el de backpropagation, trabajan juntos para actualizar los pesos y bias, y recorrer todas las neuronas, de adelante hacia atrás, de tal forma que la próxima vez que pasemos las muestras de entrenamiento a través de la red, el valor de la función de pérdida se reduzca. Siendo esta la finalidad del entrenamiento, esperando que resulte en una red que haya aprendido las características generales de los datos (Kneusel, 2021).

#### 2.1.24. Proceso de aprendizaje

El entrenamiento es el proceso donde se modifican los pesos y bias de una red neuronal, con la finalidad de generar valores de salida deseados. El primer paso para entrenar una red neuronal es seleccionar valores iniciales para los pesos y sesgos. Luego usamos el descenso de gradiente para modificarlos, según el valor de la tasa de aprendizaje (Learning Rate, indica en cuanto variara el descenso gradiente), el Momentum (previene cambios rápidos en el descenso gradiente) y el gradiente, de tal forma que reduzcamos el error sobre el conjunto de entrenamiento. Conjuntamente, el algoritmo de Retropropagacion, recorre la red de derecha a izquierda, propagando el error de la función de pérdida y permitiendo al algoritmo de Descenso Gradiente efectuar cambios de los pesos y bias de todas las neuronas.

Se usa el valor promedio de la función de pérdida para medir el error, que indica qué tan mal está la red actualmente. Se sabe si la red es correcta o incorrecta porque se tiene la salida esperada para cada muestra de entrada en el conjunto de entrenamiento.

Durante el entrenamiento, se puede usar la precisión, o alguna otra métrica, en el conjunto de validación para decidir cuándo detener el entrenamiento. Si se entrena durante mucho tiempo, se da el sobreajuste, donde el error de entrenamiento disminuye a medida que el modelo aprende más y más, pero deja de aprender cosas generales (Kneusel, 2021).

#### 2.1.25. Predicción mediante RNA

La predicción de series temporales se basa en el modelado de valores futuros de cualquier parámetro específico en función de sus datos actuales. Los modelos basados en series de tiempo son pronósticos de lluvia promedio, tasas de tráfico de Internet, pronóstico de tendencias en negocios, pronóstico del tiempo, propagación de enfermedades contagiosas, etc.

Las RNA tienen cada vez más éxito en la predicción de datos de series temporales a través de modelos basados en regresión. Comienzan con un comportamiento inicial aleatorio y pueden entrenarse gradualmente de forma supervisada o no supervisada, según la aplicación y el escenario.

Las capas en la red ayudan a extraer las características de las entradas. Se supone que este procedimiento de entrenamiento es similar al de los humanos, sin embargo, la lógica



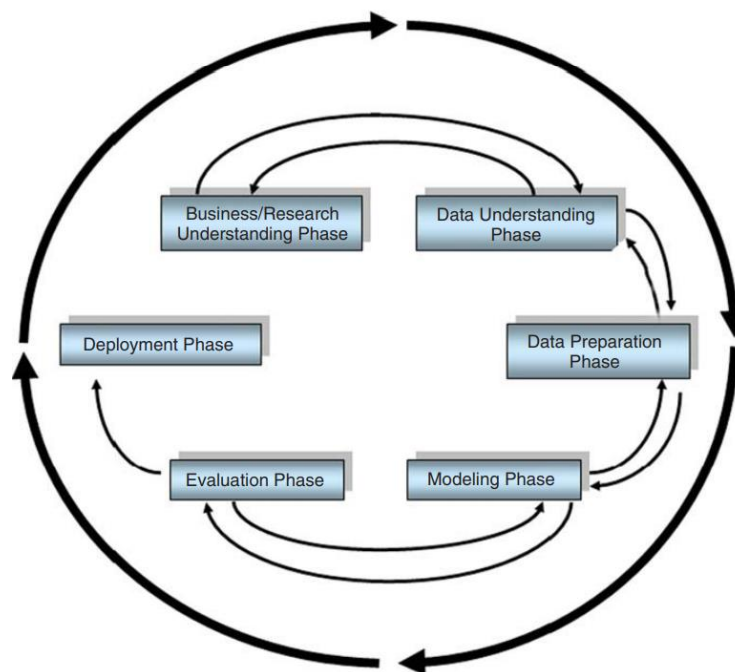
determinada y la información dentro de la red sigue siendo una caja negra que es una limitación importante de la red neuronal (Nannapaneni, 2018).

#### 2.1.26. Metodología CRISP-DM

CRISP – DM provee un proceso estandarizado para adaptar los proyectos de minería de datos en una estrategia de resolución de problemas de negocios o investigaciones. Las fases que contiene son adaptadas a proyectos donde se trabaje con datos y generación de modelos. Es debido a ello que el uso de esta metodología es ampliamente utilizado para trabajos relacionados a ciencia de datos y aprendizaje de máquinas. Según esta metodología, todos los proyectos de minería de datos y ciencia de datos tienen un ciclo de vida, que consta de 6 fases. Cada una de las fases implicadas depende de la fase previa, en caso se tenga que realizar un ajuste en una fase previa es válido retornar a dicha fase. De forma gráfica se puede ver esta metodología a continuación.

Figura 13

*Fases de la metodología CRISP-DM*



Nota. Data science using Python and R por Larose D. y Larose C, 2019, (edición n° 1) <https://es.ok.lat/book/5411346/3cf7ed> . Derechos reservados 2019 por John Wiley & Sons.

A continuación, se revisarán las fases de esta metodología, acorde a Larose D. y Larose C, (2019). Además de ello, en el capítulo de desarrollo del proyecto, se verá la investigación alineada a cada una de las fases de la metodología.



#### 2.1.26.1. Entendimiento el Negocio

Enunciar los objetivos del proyecto y requerimientos en términos del negocio o la investigación. Para después proceder a trasladar tanto objetivos como restricciones a la formulación de la definición del problema. Finalmente, se requiere preparar la estrategia para alcanzar esos objetivos.

#### 2.1.26.2. Entendimiento de los Datos

La primera parte de esta fase inicia con la recolección o captura de datos, posteriormente estos datos tendrán que ser analizados con la finalidad de conocerlos y entenderlos, lo cual servirá para evaluar si los datos de los que disponemos son los requeridos para la ejecución de la investigación.

Esta fase también incluye la selección de un grupo de variables, únicamente el necesario para el desarrollo del proyecto, este grupo reducido será el universo de variables. Dicho subconjunto de variables, es extraído del grupo de datos obtenido durante la captura.

#### 2.1.26.3. Preparación de los Datos

La labor intensa que aquí se desarrolla, cubre todos los aspectos de la preparación final del grupo de datos, el cual será usado para los procedimientos siguientes. Esta etapa parte de un grupo de datos sucios, que deben ser tratados para conseguir un conjunto de datos de calidad. La acción indicada, evitará un impacto negativo en los procesos posteriores, pues la construcción de un modelo sobre datos sucios no permitirá predicciones correctas. Al igual que en el entendimiento de datos, durante esta fase también se realiza una selección, sin embargo, la forma en que se da es más especializada, pues las variables son escogidas mediante un análisis que impactará sobre el modelo.

Es aquí donde se desarrollan las transformaciones de algunas variables, si es que es necesario. El escalado de los datos es otra de las actividades que están contempladas en esta etapa.

#### 2.1.26.4. Modelamiento

El modelamiento comienza con la determinación de la técnica de modelado apropiada para el objetivo del proyecto. Pertenece aquí la calibración del modelo para optimizar los resultados, esto significa la configuración de los parámetros del modelo.

Se cuenta con una amplia variedad de técnicas que serán aplicadas al proyecto de acuerdo a la percepción del investigador.



Desde aquí es posible regresar hasta la fase de preparación de datos, en búsqueda de satisfacer el requerimiento de la técnica de modelado escogida.

#### 2.1.26.5. Evaluación

En la presente etapa de la metodología se desarrolla la evaluación de uno o más modelos. Estos modelos tienen que ser evaluados para conocer su calidad y efectividad, antes de desplegarlos para su uso. Esta fase también implica determinar si el modelo alcanza los objetivos planteados durante el entendimiento del negocio.

Se concluye esta evaluación, con la decisión de hacer uso de los resultados de todo el proceso seguido hasta llegar a este punto.

#### 2.1.26.6. Despliegue

La generación del modelo no significa la culminación del proyecto. Contempla el desarrollo basado en el modelo, lo que significa la materialización del modelo generado. Esta fase es la última que se sigue para la ejecución del proyecto.

### 2.2. Antecedentes de la investigación

#### 2.2.1. Antecedentes a Nivel Internacional

- **“Round Trip Time Prediction using Recurrent Neural Networks with Minimal Gated Unit”**

Según Dong, Du & Yan (2019), en su investigación Round Trip Time Prediction Using Recurrent Neural Networks With Minimal Gated Unit, señalaron que:

Motivados por mejorar el QoS que exigen muchas aplicaciones de red, se propone realizar la predicción del valor del RTT haciendo uso de las RNR (Redes Neuronales Recurrentes). Los resultados de esta investigación apuntan a una mejora en cuanto a mayor precisión del valor del RTT.

El aporte de este antecedente se encuentra en la forma en la que se hace la medición del RTT que implementa RNR, en contraste con otros métodos de estimación, toma indicadores importantes como MAE, RMSE y  $R^2$ ; que también serán empleados en la presente investigación.

- **“Round-Trip Time Anomaly Detection”**

Según Brahneborg, Afzal, Causevic, Sundmark & Bjorkman (2018), en el artículo publicado en la conferencia Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering, se muestra lo siguiente:



Esta investigación realiza un estudio a profundidad sobre el comportamiento del RTT durante transmisiones de datos. Bajo diferentes indicadores estadísticos, se analiza el valor del RTT a lo largo del tiempo, debido a que este es bastante variable. Así también, la investigación centra esfuerzos en estudiar los valores anómalos del RTT, lo que implica el estudio de outliers y valores producto de fluctuaciones y variaciones en la red.

El aporte que se puede considerar para el presente trabajo lo obtenemos abstrayendo que, el comportamiento del RTT es bastante variable, por lo que se requiere del desarrollo de algoritmos centrados en su estimación. De tal modo que sean capaces de seguir y adaptarse a su variabilidad.

#### 2.2.2. Antecedentes a Nivel Nacional

- **“Análisis de los factores que influye con el rendimiento de la red de datos de la corte superior de justicia de La Libertad basados en los parámetros de QoS – 2017”**

Acorde a la investigación de Palomino Suarez (2017), se estudió los parámetros del QoS y el impacto de los mismos sobre el rendimiento de la red, ambos fueron incluidos en las variables de estudio. En dicha investigación se llega a la conclusión de que la mayor afectación sobre la red, después de haber sido analizados los parámetros, es determinada por el Jitter.

Este estudio proporciona para la presente investigación un soporte en cuanto al uso de los parámetros QoS, como características que impactan en la red. De esta forma, es que los parámetros analizados para servir de entradas en la RNA adquieren mayor sustento.

- **“Redes neuronales artificiales para la optimización del cálculo de tiempo de ida y vuelta del protocolo de control de transmisión sobre Linux”**

Según Tejada Chávez (2015), en su tesis para obtener el título de ingeniero de Sistemas en la Universidad Andina del Cusco realizó una investigación que alcanzó el desarrollo de un modelo predictivo para el cálculo óptimo del RTT durante una conexión en el protocolo de transporte TCP, todo ello implementando las Redes Neuronales Artificiales.

Posteriormente lleva a cabo una comparación del modelo propuesto, con el modelo predictivo estadístico, basado en el modelo predictivo de Van Jacobson.

Sus resultados finales, concluyen que, el modelo predictivo propuesto realiza un consumo de tiempo inferior al que toma el modelo predictivo habitual, y muestra que la estimación de tiempo basado en las RNA presenta un comportamiento más eficiente.



El trabajo realizado brinda algunos puntos de referencia para la determinación de la función de activación y para la determinación de los indicadores de variables, que pueden utilizarse en la presente investigación.



## CAPÍTULO III: METODOLOGÍA

### 3.1. Tipo de investigación

El desarrollo de la investigación es realizada a través de la metodología CRISP-DM, esta metodología provee de un proceso estructurado, que es ampliamente utilizado para proyectos relacionados a Inteligencia Artificial.

Es debido a lo mencionado que, las fases que componen esta metodología se adaptan a la presente investigación y permiten alcanzar los objetivos planteados.

Desde un enfoque general, el proceso que se sigue implica el tratamiento de los datos generados a partir de la simulación, la definición de las variables, la generación del modelo de RNA y finalmente la evaluación de los resultados y la comparativa entre las predicciones del modelo estándar y el modelo propuesto.

### 3.2. Diseño de la investigación

La presente investigación recurre a las fases de CRISP-DM para su ejecución.

La primera fase, Entendimiento del Negocio, es sustancial para identificar y delimitar el problema en el que se enfoca esta investigación. El cual está relacionado a una estimación inadecuada del valor del RTT en el protocolo SCTP, conjuntamente se precisa el planteamiento de un modelo de estimación alternativo basado en RNA.

Consecuentemente, la fase de Entendimiento de los Datos es la que contempla aspectos relacionados a la obtención de los datos con los que se construirá el modelo, esto implica la identificación de las 4 variables que serán analizadas (Length, Jitter, Packet Loss y Throughput). Es aquí donde también se ven los detalles del entorno de simulación, como la arquitectura lógica y física de la red, de la misma forma, las características de los videos transmitidos.

Culminadas las fases anteriormente expuestas, se puede iniciar la Preparación de los datos. Esta etapa requiere de un tiempo prolongado durante su ejecución, y se aboca al tratamiento de los datos, ello consiste en la limpieza y la normalización. El conjunto de datos capturados presenta valores vacíos y en algunos casos valores extremadamente dispersos, por ello es que mediante lenguajes como R, se realiza la limpieza. En el mismo sentido, es requerido estandarizar los datos, pues los valores de cada variable se encuentran en diferentes escalas de valores, lo que produciría gran dispersión entre los datos.

La etapa de Modelamiento presenta alta dependencia de la fase anteriormente tratada, porque es en función a los datos, que el análisis se llevará a cabo. Esto implica trabajar en la selección



de predictores o variables que determinarán el modelo, mediante estrategias de selección, criterios de selección y análisis de correlación. Esta etapa también profundiza en el diseño de la RNA y contempla la construcción de esta misma. El modelo resultante ataca el problema planteado para la investigación.

La fase de Evaluación describe la comparativa entre el modelo estándar y el modelo propuesto, así también, el análisis de las predicciones del modelo propuesto en función a sus variables e indicadores.

Finalmente, en la etapa de Despliegue se incorpora el código fuente de la implementación del modelo propuesto basado en RNA.

### 3.3. Instrumentos

Los instrumentos que intervinieron en la investigación son mencionados en esta sección obviando detalles, debido a que los mismos serán tratados de forma más estructurada conjuntamente a cada fase de la metodología CRISP – DM, por lo tanto, en este apartado son tocados desde una perspectiva genérica.

La ejecución de la investigación requirió de instrumentos y equipos tecnológicos como:

- Ordenadores
- Cables de red
- Tarjeta de red externa

De la misma manera, fue necesaria la utilización de programas y librerías, como:

- Wireshark
- Multiple Back Propagation (MBP)
- FFMPEG
- Librerías R, etc.

### 3.4. Recolección y Análisis de Datos

La recolección de los datos fue llevada a cabo haciendo uso de herramientas de análisis de paquetes de redes de datos. Bajo los escenarios de prueba con diferentes condiciones tanto en la información transmitida, como en el entorno de red, se realizó la recopilación de los datos.

Cada una de las tramas enviadas por el emisor y receptor son capturadas por el programa Wireshark, en él se recopilan todos los detalles de cada una de las tramas, generando así una cantidad inmensa de información. Partiendo de la información recolectada se lleva a cabo el





análisis de la misma, a través de un estudio de los datos y de técnicas estadísticas que señalen las características del modelo que satisfaga los requerimientos de la investigación.

Se partió de un entendimiento de la información, basado en la revisión bibliográfica concerniente al protocolo de transporte SCTP, para posteriormente decantar en un estudio estadístico de las variables y su efecto sobre el objetivo de la investigación.

En la sección concerniente al desarrollo del proyecto se ampliará a detalle el procedimiento explicado en párrafos anteriores.



## CAPÍTULO IV: DESARROLLO E IMPLEMENTACIÓN

El desarrollo del proyecto estuvo sujeto a las directrices de la metodología ampliamente utilizada para investigaciones relacionadas a la ciencia de datos y machine learning, CRISP - DM.

Bajo el ciclo de vida de esta metodología, se definió de forma clara las fases que se debían atravesar para conseguir un análisis de datos adecuado, lo que también permitió la selección de las variables a emplearse como entradas de la RNA.

Alineado a cada una de las fases mencionadas, a continuación se detalla el proceso seguido en la investigación.

### 4.1. Entendimiento del Negocio

SCTP, es un protocolo de nivel de transporte orientado a la conexión que, a diferencia del protocolo de comunicación TCP y su concepto de conexión, introduce una nueva forma de comunicación entre sistemas denominada asociación. Este protocolo está especificado en los RFC's (Request For Comments, Peticiones de Comentarios) 3286 y 4960.

SCTP permite multiplexar varios flujos de nivel de aplicación diferentes a través de una asociación SCTP (Multi-Streaming). Además de que permite que los datos se transfieran a través de dos rutas de salida cuando un host está conectado a dos o más redes (Multi-Homing) (Kurose & Ross , 2021, pág. 192).

Sin embargo, a pesar de que este protocolo implementa características particulares, también recurre a características e implementa algoritmos de la misma forma como lo hacen los protocolos TCP y UDP. Una de estas características es el cálculo del RTT (Round Trip Time) y el RTO (Retransmission Timeout), a través del algoritmo de Jacobson. Donde el RTT es el tiempo que transcurre desde que un segmento es enviado, hasta que este mismo segmento sea confirmado por una confirmación de recibo. Basado en el valor asignado al RTT es que el RTO puede ser calculado. El RTO o temporizador, es el mecanismo de retransmisión para la recuperación de la conexión frente a segmentos perdidos. Alrededor del RTT surge el problema de cuánto tiempo debería de tener asignado (Kurose & Ross , 2021, pág. 265).

Según Stewart (2007), el manejo del RTO en SCTP se hace de la misma forma como TCP gestiona su temporizador de retransmisión. El algoritmo de Jacobson se encarga de estimar el valor del RTT y su variación, para posteriormente, en base a este valor sea calculado el valor del RTO (Yasuda & Yoshida, 2018, pág. 1). Este algoritmo presenta limitaciones para realizar la estimación de estos valores con precisión. Sin embargo, el RTO requiere de ser estimado con



precisión, pues este valor tiene un impacto negativo en la congestión, lo que consecuentemente exige en la activación de mecanismos de control de congestión, para solucionar este nuevo problema (Peterson & Davie, 2021).

La asignación del RTO está directamente relacionada al RTT, pues se realiza a través de un cálculo donde se tiene como variables únicamente al valor del RTT y la variación del RTT, esta estimación se encuentra detallada en capítulos anteriores. Asignar al RTT un tiempo más elevado del necesario implica que el emisor de la transmisión esperará un periodo de tiempo amplio para retransmitir a pesar de que esto sea necesario. Al contrario de ello, si el RTT es demasiado corto conllevará a retransmisiones innecesarias originadas al no tenerse una confirmación de recibo, es esto lo que supone el incremento de la congestión del canal (Yasuda & Yoshida, 2018, pág. 1).

A partir de ello se afirma que, el RTT puede generar problemas en las redes, como congestión, lo que acarrea demoras y cambios en el enrutamiento. Por lo que, una estimación del RTT más aproximada al RTT Real es fundamental para evitar este tipo de problemas (Chen, y otros, 2020).

Además de lo expuesto anteriormente, se pone gran énfasis en la estimación del valor del RTT, pues también es perceptible en el rendimiento de las aplicaciones de red, más aún en las aplicaciones de red con estrictos requisitos de QoS (Dong, Du, & Yan, 2019).

Una de esas aplicaciones es el video almacenado o por streaming, pues según Kurose y Ross (2021) exige periodos de demora muy cortos. Otra aplicación puede ser vista en las teleoperaciones, donde el RTT juega un rol importante debido a que, la demora en el control de señales sumado a las variaciones drásticas en los tiempos de demora, deterioran la capacidad de respuesta e impactan su estabilidad (Yasuda & Yoshida, 2018, pág. 1). Lo mencionado también resalta la necesidad de obtener un valor del RTT adecuado, de modo se pueda calcular un RTO preciso, lo que evita que se generen problemas de congestión (Peterson & Davie, 2021). Por otro lado, la variabilidad del valor del RTT provocada por el entorno de red no confiable, recae en paquetes fuera de orden, retransmitidos, duplicados, caídas en las conexiones o congestión de los canales; exige el empleo de métodos de estimación que se adapten de mejor manera al valor real del RTT. Uno de estos métodos recurre al uso de RNA (Redes Neuronales Artificiales), debido a que los modelos basados en ella presentan buenos resultados en la predicción de datos de series temporales a través de modelos basados en regresión. Los que comienzan con un comportamiento inicial aleatorio y pueden entrenarse gradualmente (Nannapaneni, 2018).



De ello se desprende que, aún es posible modificar este protocolo para mejorar el cálculo del tiempo estimado. En este contexto, se propuso modificar la función del algoritmo actual donde se realiza la predicción del RTT, buscando una alternativa a la estimación del tiempo de espera, asignada a las transmisiones a través de una asociación SCTP, haciendo uso de las RNA. Propuesta construida en un entorno de pruebas que contempló la transmisión de flujos multimedia, para replicar un entorno con estrictos requisitos de QoS.

Para alcanzar el objetivo planteado en la investigación, se establecieron las actividades en base a las fases de la metodología CRISP-DM. El siguiente cuadro muestra estas actividades.



#### 4.1.1. Cronograma de trabajo

Tabla 3

*Cronograma de trabajo.*

Nombre de la tarea(Fases Metodología CRISP-DM)	Duración (días)	Inicio	Finalización
Entendimiento del Problema	9 días	20/09/2021	28/09/2021
Entendimiento de Datos	5 días	29/09/2021	03/10/2021
• Captura de Datos	1 días	29/09/2021	29/09/2021
• Análisis de los Datos	4 días	30/09/2021	03/10/2021
Preparación de Datos	9 días	04/10/2021	12/10/2021
• Limpieza de Datos	4 días	04/10/2021	07/10/2021
• Normalización de Datos	3 días	08/10/2021	10/10/2021
• Selección de Predictores	2 días	11/10/2021	12/10/2021
Modelado	15 días	13/10/2021	27/10/2021
• Definición del Modelo	5 días	13/10/2021	17/10/2021
• Desarrollo de la propuesta	10 días	18/10/2021	27/10/2021
Evaluación del Modelo	5 días	28/10/2021	01/11/2021
Despliegue	4 días	02/11/2021	05/11/2021

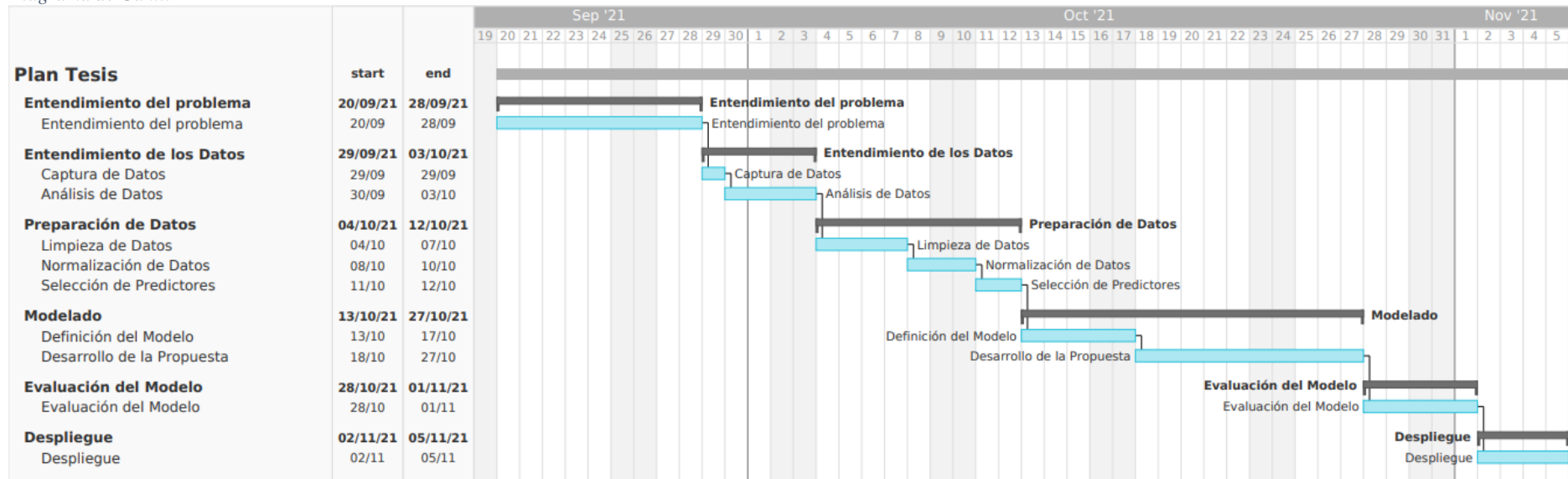
Nota. Actividades para llevar a cabo la investigación junto con el tiempo que tomarán y el intervalo de tiempo en el que se desarrollaran. Fuente de elaboración propia.



### 4.1.2. Diagrama de Gantt

Figura 14

Diagrama de Gantt.



Nota. Actividades desarrolladas en la presente investigación. Fuente elaboración propia.



## 4.2. Entendimiento de los Datos

### 4.2.1. Captura de Datos

El conjunto de datos necesario para el desarrollo del proyecto es obtenido a partir del entorno de red simulado y los flujos multimedia transmitidos. El intercambio de paquetes producidos durante este proceso es capturado por el analizador de paquetes Wireshark. Esta herramienta brinda muchas funcionalidades y muestra una amplia variedad de datos que permiten conocer a profundidad las condiciones bajo las cuales se llevó a cabo el flujo de datos, así como información referida a los paquetes.

La captura de paquetes brindó gran cantidad de información, esto exigió tener identificadas las variables que serían de importancia para la investigación y entender la significancia de cada una de ellas. La cantidad de registros obtenidos de la captura es de 189,141 paquetes, lo suficiente para que posteriormente a la fase de preparación, tengamos un grupo de datos significativo para el entrenamiento y testeo de la RNA. Todos ellos fueron obtenidos en un periodo de 14 minutos 41 segundos de transmisión de flujos, dividido en un video con duración de 09 minutos 56 segundos y otro de 04 minutos y 45 segundos.

Consecuentemente, fue necesario explorar el gran grupo de datos de tal forma que sea posible hallar la información con la que se trabajaría en la presente investigación.

### 4.2.2. Entorno de simulación

El entorno de simulación fue una réplica de entornos reales de red. Por lo cual se trabajó con un emulador de red WAN llamado WANem, que fue configurado con parámetros que describen el comportamiento real del tráfico de la red, los valores con los que fueron configurados dichos parámetros son tomados de una investigación llevada a cabo, que indica los valores correspondientes a las condiciones que caracterizan normalmente una red.

Para tener alta exigencia de recursos, se planteó la transferencia de flujo multimedia entre dos ordenadores, por lo que, se transmitió y reprodujo videos con diferentes características, debido a que los entornos reales son el medio de intercambio de una amplia variedad de archivos.

A continuación se detallan las condiciones de la simulación.

#### 4.2.2.1. Arquitectura lógica de la red

Se trabajó con máquinas virtuales, una de ellas contenía el emulador de red WAN, WANem versión 2.3. Este programa funciona como un entorno de red al cual se vinculan los computadores que se conectarán. Para que la simulación sea válida es necesario emular las



condiciones de un entorno real, el programa permite modificar ciertos parámetros como el Jitter, la congestión, la latencia, entre otros. De ellos únicamente se configuraron los valores de los parámetros que según la investigación de Carrasco, E. , Gamarra, E. y Palomino, E. (2017), caracterizan un entorno real de red:

- RTT: 82 ms
- Retardo: 144 ms
- Fluctuación: 6

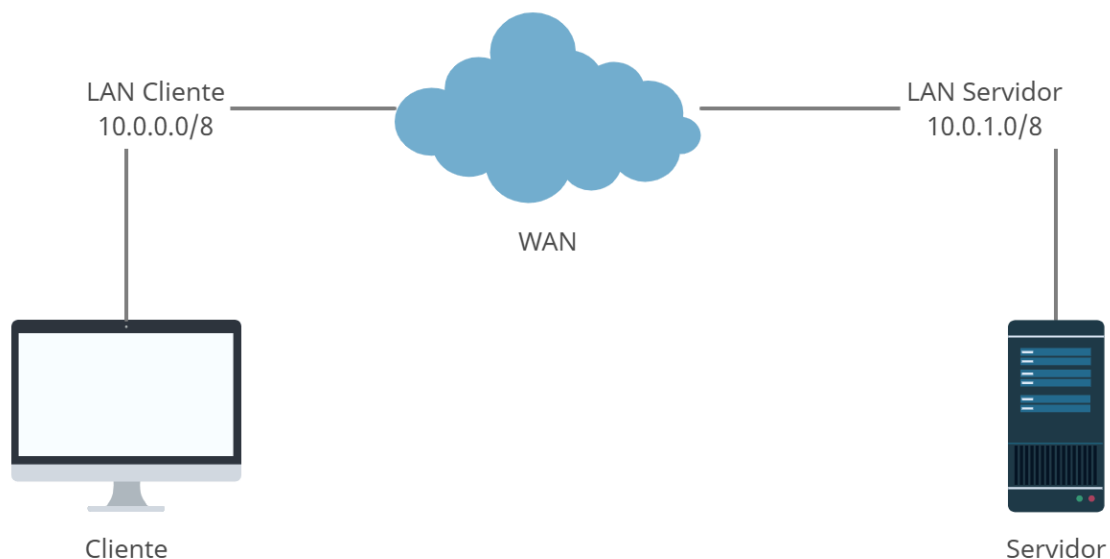
Por otro lado, se emplearon dos máquinas virtuales como computadoras, cliente y servidor, en la conexión establecida para la transmisión. Ambos nodos utilizaron Debian como sistema operativo, las características del Sistema Operativo son:

- Arquitectura: 64 bits
- Versión: 10.5.0 (Buster, estable)

La siguiente imagen presenta la arquitectura lógica del entorno de simulación.

Figura 15

*Arquitectura lógica del entorno de simulación*



Nota. Arquitectura lógica bajo la cual se desarrolló la simulación. Fuente elaboración propia.

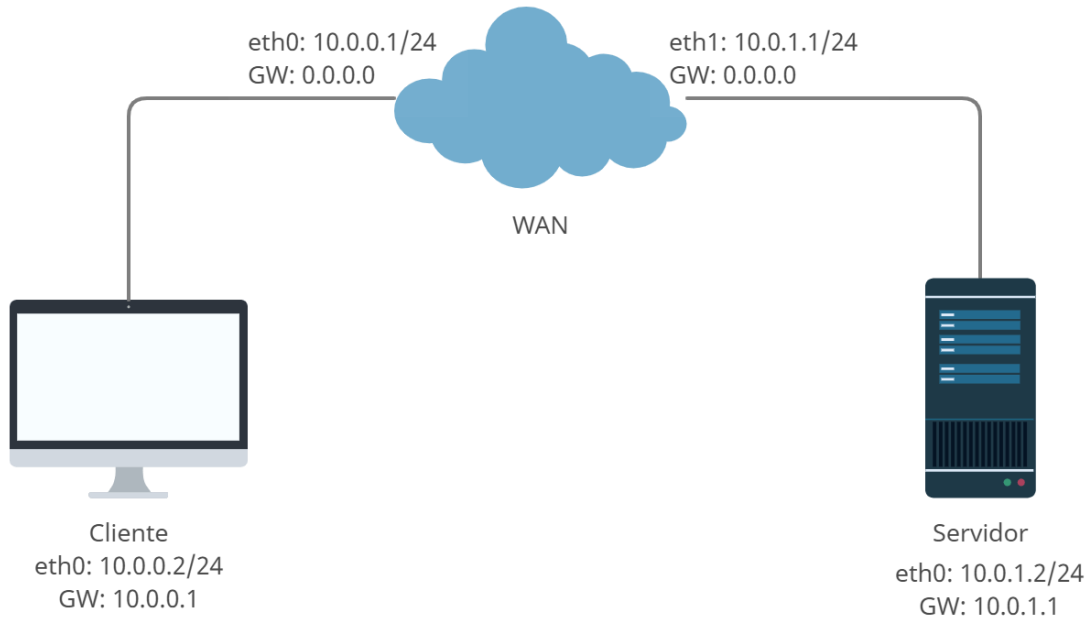




Viendo esta arquitectura a mayor profundidad se tiene:

Figura 16

Arquitectura lógica y configuraciones



Nota. Arquitectura lógica de la simulación y la configuración de sus puertos de red. Fuente elaboración propia.

#### 4.2.2.2. Configuración de las máquinas

- Cliente:
  - IP address: 10.0.0.2
  - Netmask: 255.255.255.0
  - Gateway: 10.0.0.1
- Servidor:
  - IP address: 10.0.1.2
  - Netmask: 255.255.255.0
  - Gateway: 10.0.1.1
- Red WAN:
  - Eth0:
    - IP address: 10.0.0.1
    - Netmask: 255.255.255.0
    - Gateway: 0.0.0.0
  - Eth1:



- IP address: 10.0.1.1
- Netmask: 255.255.255.0
- Gateway: 0.0.0.0

#### 4.2.2.3. Programas y librerías empleadas

De la misma forma, fueron requeridos en los nodos cliente y servidor, algunos programas y librerías adicionales que intervinieron a lo largo del proceso de desarrollo.

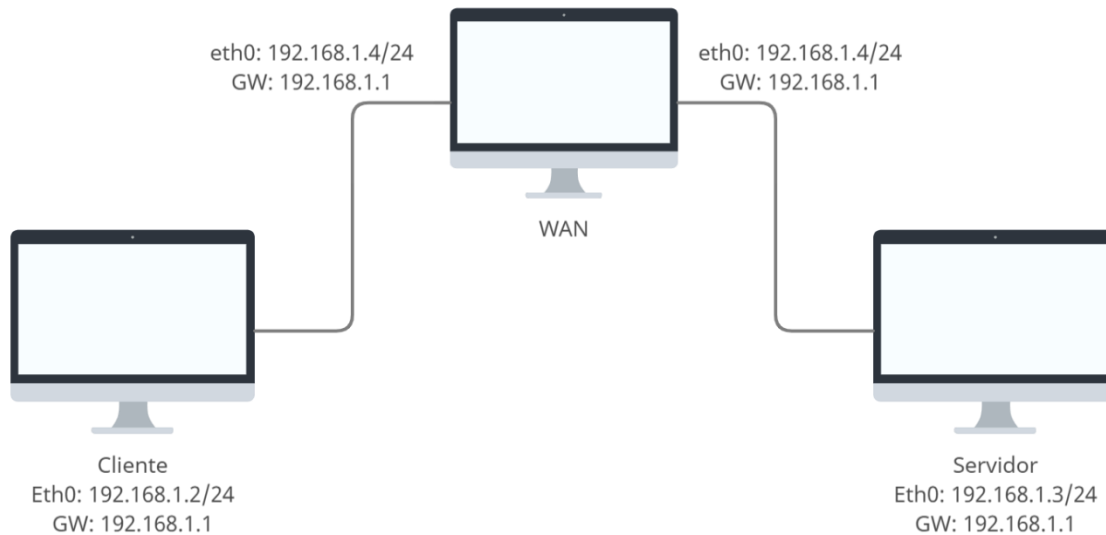
- **lksctp-tools**: Provee de una librería a nivel de espacio de usuario para el uso de SCTP. Soporta la ejecución del programa FFmpeg sobre el protocolo SCTP.
- **libsctp-dev**: Este paquete complementa la funcionalidad de lksctp-tools. Soporta la ejecución del programa FFmpeg sobre el protocolo SCTP.
- **FFmpeg**: Framework multimedia capaz de codificar, decodificar, transmitir, filtrar y reproducir archivos multimedia. Se utilizó esta plataforma para la transmisión de los videos. Donde el nodo cliente se encargó de reproducir los paquetes que recepcionaba, y donde el nodo servidor enviaba el video.
- **Wireshark**: Es un programa analizador de protocolos de red, que te permite ver la red a un nivel más bajo y con mayor profundidad. Empleado para realizar la captura de los paquetes transmitidos, así como, examinar las condiciones y parámetros de la red durante la simulación.

#### 4.2.2.4. Arquitectura física de la red

Los equipos que intervinieron en la simulación, tuvieron Windows como sistema operativo nativo. El mismo que alojó las máquinas virtuales mediante el programa de virtualización, Virtual Box.

Figura 17

Arquitectura física de la red de simulación



Nota. Arquitectura física de la simulación y la configuración de los equipos. Fuente elaboración propia.

#### 4.2.2.5. Características de los ordenadores

- Ordenador Cliente:
  - Sistema operativo: Windows 10
  - Marca: Lenovo
  - Modelo: G400s
  - Procesador: Intel Core i3-3110M 2.40GHz
  - Memoria RAM: 4 GB DDR3
  - Interfaz de red: LAN Gigabit 10/100/1000 MB/S
- Ordenador Servidor:
  - Sistema operativo: Windows 7
  - Marca: HP
  - Modelo: Pavilion
  - Procesador: Intel Core i3-3110M 2.40GHz
  - Memoria RAM: 4 GB DDR3
  - Interfaz de red: LAN Gigabit 10/100/1000 MB/S



- Ordenador WAN:
  - Sistema operativo: Windows 7
  - Marca: Advance
  - Modelo: AN-6431
  - Procesador: Intel Core Inside 2.0GHz
  - Memoria RAM: 2 GB DDR3
  - Interfaz de Red Interna: LAN Gigabit 10/100/1000 MB/S
  - Interfaz de Red Externa: LAN Gigabit 10/100/1000 MB/S

La tarjeta de red externa fue adicionada al equipo, para lograr la conexión con el equipo cliente y servidor.

#### 4.2.2.6. Características de la Conexión

- Conexión: Cableada
- Tipo: UTP (Unshielded Twisted Pair)
- Categoría: Cat 5e
- Calibre: 24 AWG

#### 4.2.3. Escenarios de prueba

Se trabajó con dos escenarios de prueba, diferenciados únicamente por los archivos transmitidos. Los escenarios muestran características muy diferentes, entendiendo que los entornos reales pueden transmitir información de diversas características.

##### 4.2.3.1. Escenario 1

- Video 1: Cortometraje animado “Piper”
  - Resolución: 640 x 360 px
  - Formato: MPEG – 4
  - Tamaño: 17 MB
  - Duración: 04:45 min
  - Velocidad de Fotogramas: 29.87 FPS

##### 4.2.3.2. Escenario 2

- Video 2: Cortometraje animado “Big Buck Bunny”
  - Resolución: 1920 x 1080 px
  - Formato: MPEG – 4
  - Tamaño: 243 MB



- Duración: 09:56 min
- Velocidad de Fotogramas: 24 FPS

#### 4.2.3.3. Descripción de Variables

En la investigación se tomó como universo de variables para el análisis, a los parámetros de la Calidad de Servicio (QoS, Quality of Service), acorde a Sunny Valley Networks (2022) estos son Latencia, Jitter, Throughput y Packet Loss Rate. La determinación de este universo de variables responde a la necesidad de generar una propuesta que incluya parámetros que caractericen el entorno de red, en búsqueda de un mayor ajuste en la predicción del RTT, en lugar de tomar el valor del RTT de anteriores transmisiones de datos.

El universo de variables con el que se trabaja es de gran importancia para la investigación y es uno de los mayores aportes de la misma, debido a que este conjunto de variables es muy diferente al que en investigaciones relacionadas se han empleado.

Los parámetros QoS son cuantificables, además de que, son aplicados para determinar la disponibilidad del servicio. Los parámetros básicos son:

- Retardo o Latencia (Delay):  
Retardo de tránsito de extremo a extremo, es el tiempo transmitido para que un paquete pase desde el emisor al receptor, a través de la red.
- Variación (Jitter):  
La variación del retardo de tránsito de extremo a extremo. En las redes de conmutación de paquetes, Jitter define la distorsión de los tiempos de llegada entre paquetes en comparación con los tiempos entre paquetes de transmisión.
- Rendimiento (Throughput):  
Medida de la velocidad a la que los datos pueden ser enviados a través de la red.
- Tasa de Pérdida de Paquetes (Packet Loss Rate):  
Normalmente, cuando un camino a un dispositivo está sobrecargado y no puede aceptar datos de entrada en un momento dado se produce la pérdida de paquetes temporal.

Las variables que fueron utilizadas para la investigación fueron las que proveían información concerniente a los parámetros de QoS con excepción de la variable de Latencia. Se optó por descartar esta variable debido a que la Latencia explica el mismo comportamiento del RTT, pero únicamente para la trayectoria del envío de una trama. Por lo indicado, es que esta variable no permitiría alcanzar una nueva propuesta, diferente a la que implementa el algoritmo estándar, que recurre al valor del RTT en tiempos anteriores, para la estimación del nuevo RTT. Además



de esta determinación, se incluyó la variable tamaño de la trama en bytes (a ella se le llamara Longitud a lo largo de la investigación), esto responde a la observación de una posible relación entre la variable mencionada y el RTT.

La naturaleza de cada parámetro dio paso a manipular el formato de los campos de manera adecuada. La naturaleza de la información con la que se trabajó es de tipo cuantitativa en su totalidad, pues se tiene como unidad de medida el tiempo o cantidades.

#### 4.3. Preparación de los Datos:

Considerando que esta etapa implica un tiempo prolongado y atención, fue necesario seguir al detalle cada actividad como la limpieza de datos y dar formato según el tipo de dato.

##### 4.3.1. Limpieza de Datos

La limpieza de datos es una actividad de suma trascendencia, debido a que se debe asegurar que la información que será procesada es consistente, caso contrario el modelo generado será erróneo y a consecuencia de ello, las predicciones también.

Este procedimiento implica actividades como la eliminación de valores nulos o vacíos, en caso contrario estos valores distorsionan los procedimientos consecuentes a este paso. Persiguiendo el mismo propósito de mantener la consistencia de los datos, se aplicó también la eliminación de valores duplicados e inconsistentes.

La uniformización de los datos es otra de las actividades aquí desarrolladas. Uniformizar los datos implica modificar los datos que comprendan valores demasiado elevados o bajos, pues ellos distorsionan el análisis estadístico que se aplica.

Las acciones descritas anteriormente en esta actividad nos aseguran en gran medida la consistencia de los datos que se emplearán para los modelos que se propondrán.

##### 4.3.2. Normalización de los datos

La normalización o escalamiento de los datos es la actividad que consiste en la compresión o extensión de los valores a un rango de valores definido.

El escalamiento empleado para el conjunto de datos obtenido fue el Escalamiento Estándar, que implica restar la media de cada observación y luego dividir por la desviación estándar. Este escalado hace que todos los datos preserven una distribución similar, para lo cual transforma los datos originales en otros cuya distribución estadística tiene una media 0 y una desviación estándar igual a 1.



Se eligió este tipo de escalado frente al Escalado de Mínimos y Máximos, debido a que este último es ideal para grupos de datos donde los valores de todas las variables se encuentran en rangos de valores muy similares. Lo cual no se aplicaba para los valores del conjunto de datos del proyecto, en vista que se dispone de valores decimales muy pequeños en algunos campos, y en otros se tienen valores enteros, lo que los hace muy diferentes en términos de escala.

Por otro lado, se tuvo la opción de trabajar con el escalado robusto, que trabaja con la mediana y los rangos intercuartiles, esto no permite que los valores de las variables se vean afectados por valores atípicos. Sin embargo, durante la etapa de limpieza de datos se manipularon los datos que eran anormales y perturbaban la distribución en la variable. Por lo que el uso de un escalado estándar es suficiente.

El lenguaje de programación con el que se hizo el escalamiento de los datos es el R, mediante la función “scale()” que realiza un escalamiento estándar, perteneciente a la librería “scales”. Una vez normalizados los datos podemos comenzar las tareas de análisis de variables que serán parte del modelo, a esta etapa se le conoce como la selección de variables o selección de predictores.

#### 4.4. Modelamiento

##### 4.4.1. Selección de predictores

La selección de predictores es una tarea que consiste en agregar y/o quitar predictores iterativamente en el modelo predictivo, para que basado en el conjunto de datos se encuentre el subconjunto de variables que resulte en el modelo de mejor desempeño, es decir, un modelo que reduce el error de predicción.

En la investigación se disponía de un grupo de 4 variables (Tamaño del paquete, Jitter, Packet Loss y Throughput), las cuales podían conformar el modelo. Sin embargo, es necesario definir cuáles de las variables disponibles tienen impacto sobre la variable que se busca predecir. Fue de esta forma que, se recurrió a la selección de predictores. Es necesario precisar que se cuenta con tres estrategias para esta selección. Selección hacia Adelante (Forward selection), la que empieza sin predictores en el modelo pero iterativamente añade a los que contribuyen más. La estrategia parecida es la de Selección hacia Atrás (Backward selection) que al contrario de la estrategia explicada, empieza con todos los predictores en el modelo, pero que iterativamente va eliminando los predictores que menos contribuyen. La última estrategia es la llamada Selección Escalonada (Stepwise selection o sequential replacement), que combina las dos estrategias antes mencionadas. El modelo inicia sin predictores y secuencialmente los añade.



Después de añadir una nueva variable, elimina otras variables que no impactan en el desempeño del modelo.

En R se invoca a la librería “leaps” que provee de la función “regsubsets()”. El parámetro con el que se configuró la función fue el “seqrep” en referencia al método de selección escalonada.

```
#selección de predictores
```

```
pred <- regsubsets(RTT~., data = ds, nvmax = 5, method="seqrep")
```

```
summary(pred)
```

La función mencionada retorna varios modelos combinando el grupo de variables disponibles, acorde al nivel de error de los modelos. Los modelos generados los encontramos en la imagen a continuación.

Figura 18

*Modelos generados acorde a la cantidad de variables*

```
Selection Algorithm: 'sequential replacement'
      Length Jitter.DeltaTimeDisplayed PacketLoss.PerSecond Throughput.Bytes
1 ( 1 ) " " "*" " " " "
2 ( 1 ) " " "*" "*" " "
3 ( 1 ) " " "*" "*" "*"
4 ( 1 ) "*" "*" "*" "*" "*"

```

Nota. Modelos más adecuados de acuerdo a la cantidad de variables. Fuente elaboración propia.

La imagen anterior muestra 5 modelos generados, donde:

- Modelo 1: Contiene únicamente la variable Jitter.
- Modelo 2: Incluye las variables Jitter y Packet Loss.
- Modelo 3: Emplea las variables Jitter, Packet Loss y Throughput.
- Modelo 4: Contiene todas las variables, Length, Jitter, Packet Loss y Throughput.

De todos los modelos que se tienen a disposición es necesario determinar cuál de ellos será el ideal para la investigación mediante los criterios de selección de variables.

#### 4.4.2. Criterios de Selección de Variables

Para identificar cuál de los modelos obtenidos es el más adecuado para la investigación, se requiere ver los niveles de error y desempeño según los criterios de selección de variables, como el  $R^2$ , BIC,  $R^2$  Ajustado.





Al grupo de modelos generados se les aplica estos criterios de selección.

```
summary(pred)$rsq
```

```
summary(pred)$bic
```

```
summary(pred)$adjr2
```

Lo que arroja los siguientes resultados:

Figura 19

*Criterios de selección para los posibles modelos*

```
> rsq
[1] 0.005070567 0.009135325 0.010758928 0.011576778
> bic
[1] -1375.001 -2489.972 -2929.100 -3144.364
> adjr2
[1] 0.005066955 0.009128129 0.010748153 0.011562422
```

Nota. Arquitectura física de la simulación y la configuración de los equipos. Fuente elaboración propia.

Como en la imagen anterior se muestra, cada modelo obtuvo un valor diferente respecto a un indicador particular. Es necesario realizar un análisis de cada uno de ellos con la finalidad de explicar la selección de variables para un modelo más adecuado para la investigación.

- Coeficiente de determinación R2 (RSQ):

Esta medida estadística indica que tan cerca se encuentran los datos respecto a la línea de regresión ajustada. En otras palabras, explica la variación existente entre los datos observados y los valores de predicción de un modelo lineal. El valor de este coeficiente oscila entre 0 y 1, a mayor aproximación a 0, menor el ajuste del modelo a la variable que se intenta explicar. Si al contrario de ello se presenta una aproximación más cercana a 1, entonces se dice que el modelo está más ajustado.

Según los resultados obtenidos, el modelo que muestra un mejor ajuste de entre todos los simulados, es el cuarto modelo con 0.012. Esto se debe a que la cantidad de variables que incluye este último modelo son todas las disponibles, lo cual quiere decir que siendo mayor la cantidad de variables, se genera un modelo más ajustado. Sin embargo, esto no es óptimo en vista de que, tan solo se usa todas las variables disponibles, mas no las que tienen un impacto relevante. A eso se suma que la diferencia de este indicador para cada modelo, presenta una diferencia insignificante.



- Coeficiente de determinación ajustado  $R^2$  (ADJR2):

Este coeficiente explica la intensidad de las variables independientes respecto a la variable dependiente. En términos matemáticos, se entiende como el porcentaje de la variación de la variable dependiente, explicado a partir de las variables independientes.

El análisis de este criterio expone que el cuarto modelo con un valor de 0.012, es el más alto, por lo tanto, dicho modelo es el que mejores condiciones ofrece para la predicción. Pero, al igual que el  $R^2$  simplemente se toman todas las variables, aunque la inherencia que tengan sea mínima, por lo que este modelo puede tener mayor precisión en la predicción cayendo en un sobreajuste, y a la vez ser un modelo ineficiente, que emplea todas las variables a pesar de su irrelevancia.

Por otra parte, en un análisis global respecto a este indicador para cada modelo, podemos observar que la variación de los valores para cada uno de los modelos es insignificante para determinar el modelo más adecuado. Entonces podemos afirmar que todos los modelos presentan un nivel de error similar.

- Criterio de información Bayesiana (BIC):

Cuando se ajustan los modelos, es posible aumentar la probabilidad de obtener un mejor modelo, añadiendo parámetros, pero hacerlo puede llevar a un sobreajuste. En este sentido, también, se encuentra que el incluir todas las variables disponibles para generar un modelo recae en modelos ineficientes. Por lo que el BIC resuelve este problema incluyendo una penalidad por el número de variables en el modelo.

El menor valor que adquiere un modelo bajo este criterio se traduce en el mejor modelo, siendo este el modelo 1 con -1375.001, que incluye únicamente el Jitter. Observando los otros valores podemos ver que los modelos 2 y 3 obtuvieron -2489.972 y -2929.100 respectivamente, cercanos al del mejor modelo. A diferencia del modelo 4 con -3144.364, que varía demasiado respecto al mejor modelo según el indicador BIC. Es así, que los modelos más adecuados bajo este criterio son el 1, 2 y 3.

Considerando que ambos indicadores analizados devolvieron resultados diferentes y poco precisos es necesario aplicar una técnica estadística adicional, con el objetivo de determinar el mejor modelo. Es de esta forma que se emplea el estudio de correlación de variables. Este último análisis devuelve el impacto de cada predictor sobre el RTT.



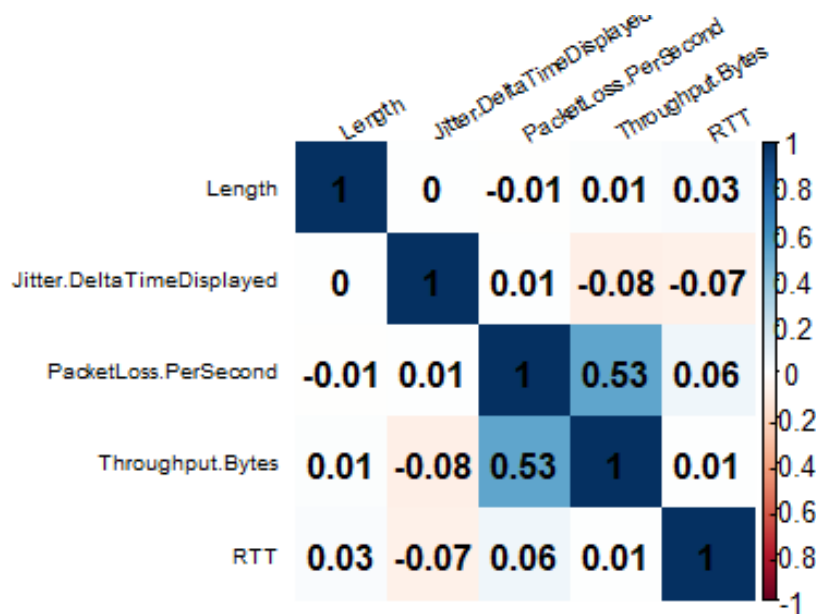
- Análisis de Correlación de Variables

La correlación fue aplicada bajo el método de Pearson debido a que este método permite evaluar la relación lineal entre variables continuas. En otras palabras el cambio de una variable está vinculado con cambios sobre la otra variable de forma proporcional. A diferencia del método de Spearman que trabaja con variables jerárquicas.

Con esta configuración se obtuvo los siguientes valores de correlación:

Figura 20

Correlación de variables en función del RTT



Nota. Diagrama de correlación de variables respecto al RTT. Fuente elaboración propia.

Donde la relación de las variables sobre el RTT se da de la siguiente forma. La variable que impacta mínimamente en el cálculo del RTT es el tamaño del paquete en Bytes, por lo cual asumimos que esta variable no tiene efecto ni deben ser considerada para el modelo.

La variable que presenta mayor correlación con el RTT es el Jitter con un valor de -0.07 por lo cual vemos que esta variable si debe ser parte del modelo.

Analizando las dos variables restantes, Packet Loss(Per Second) y Throughput(Bytes) con valores 0.06 y 0.01 respectivamente. Entre las dos variables, la más adecuada para complementar el modelo sería el Packet Loss, sin embargo, conjuncionando las tres estrategias empleadas para la selección de predictores donde el  $R^2$  ajustado nos indica que los modelos en estudio son similares por lo tanto cualquiera puede ser escogido. El criterio BIC nos sugiere trabajar con los modelos 1, 2 y 3. La correlación de variables nos da mayores luces de la



afectación de cada predictor sobre la variable dependiente, y nos muestra en qué medida impactan sobre esta. Descomponiendo las opciones disponibles podemos ver que el modelo cuatro, propuesto por el ADJR2 no sería adecuado, debido a que dos de las variables no tienen impacto real sobre el RTT. Referente al modelo 1 se sugiere que se tome como valor único el Jitter, y por el lado del modelo 2 se recomienda tomar el valor del Jitter y Packet Loss. En este punto, el investigador decide acogerse al modelo 2, debido a que se consideró más conveniente la manipulación y procesamiento de más de una variable.

Es preciso mencionar que la investigación también habría sido válidamente realizada con solo el Jitter, como sugiere el modelo 1 según el criterio BIC.

#### 4.4.3. Estrategia de Inteligencia Artificial

El Machine Learning proporciona dos estrategias para poder desarrollar el modelo, clasificación y regresión. La presente investigación requiere de la predicción de series de tiempo, por lo tanto, es necesario emplear técnicas como las máquinas de vector de soporte, árboles de decisión, redes neuronales, entre otras.

La investigación recurrió a las redes neuronales artificiales debido a que en investigaciones relacionadas, precedentes, se halla que mejores predicciones del valor del RTT es arrojada por esta técnica. Las investigaciones relativas a la actual, nos brindan mayores luces de cuál de las técnicas proporcionó mayor aproximación. Según algunas investigaciones previas, los modelos que mejores resultados arrojan son los que incluyen técnicas de Machine Learning como, Perceptrón Multicapa, Redes Neuronales recurrentes, etc.

Con la finalidad de incorporar la red neuronal desarrollada dentro de un Kernel, se vio conveniente trabajar únicamente con 2 variables procesadas por una red neuronal perceptrón multicapa. Debido a que esta tecnología no implicara la disposición de muchos recursos computacionales para su procesamiento, esto añadido al hecho de que se está utilizando solo dos variables.

La red neuronal ha sido construida utilizando herramientas como el programa MBP (Multiple Back Propagation). Esta herramienta necesitó de algunas configuraciones. A ella se le pasaron los dos grupos de datos necesarios para el desarrollo de la RNA, los datos de entrenamiento y los datos de prueba. Durante 17 horas y 25 segundos se mantuvo el entrenamiento de la RNA. A lo largo de ese periodo de tiempo, se ejecutaron 219383 épocas, alcanzando un nivel de error de 0.056 (Root Mean Square Error, RMSE). Consecuentemente, la finalización del entrenamiento y pruebas de la RNA, fue realizado manualmente.



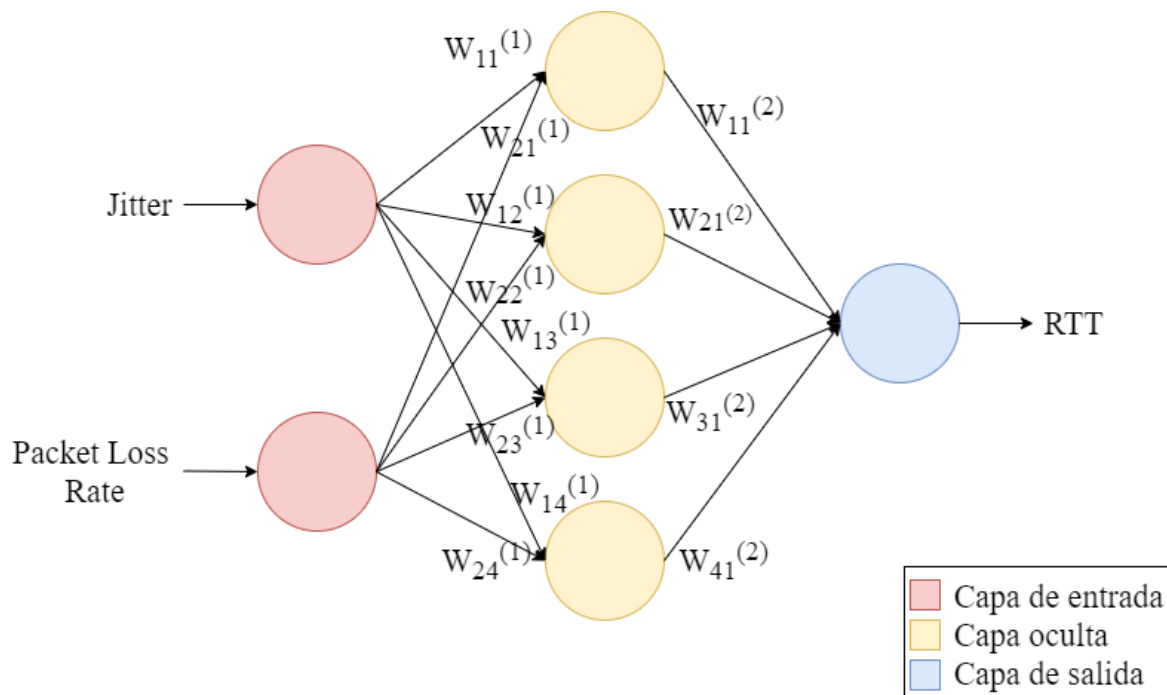
En vista de que este nivel de error no mostraba variaciones durante mucho tiempo, se optó por finalizar el procedimiento. Hecho que es válido y depende de la observación del investigador.

#### 4.4.4. Diseño de la red neuronal

Se propone una RNA con una topología que consta de una capa de entrada, una capa oculta y una capa de salida. Además de que está configurada con una Tasa de Aprendizaje de 0.25 y Momentum de 0.8 . A continuación se muestra el diseño de la topología de la RNA propuesta.

Figura 21

Topología de la RNA de estimación del RTT



Nota. Topología de la RNA empleada en la investigación. Fuente elaboración propia.

A partir de un análisis se determinó que las variables adecuadas para definir el valor del RTT son el “Jitter” y el “Packet Loss Rate”, es por esta razón que ambas variables son las entradas de la RNA.

La capa de entrada contiene 2 neuronas, cada una recibe como entradas las variables indicadas anteriormente, estas no sufren de ninguna variación debido a que la función de activación contenida en ambas neuronas es Lineal. Las entradas recibidas pasan con sus mismos valores a la capa oculta, con conexiones entre todas las neuronas de las 2 capas.

La capa oculta de la topología es la que inicia el procesamiento de las entradas, pues realiza la multiplicación de las entradas por los pesos sinápticos de su conexión para finalmente aplicarle



la función Sigmoide como función de activación. En esta capa se tienen 4 neuronas todas con la misma función de activación.

Como última capa de la topología se encuentra la capa de salida, ella consolida las salidas de las neuronas pertenecientes a la capa oculta. Consta de una sola neurona que tiene las mismas características descritas para las neuronas de la capa anterior. El valor de salida obtenido de esta última capa corresponde al valor del nuevo RTT.

#### 4.4.5. Resultado de la RNA

Concluidas las fases de entrenamiento de la RNA y pruebas de la misma, es posible acceder a los pesos sinápticos. Ellos cumplen un papel importante, pues determinan la intensidad de las conexiones entre las neuronas que conforman la topología de la RNA. El cuadro visible a continuación, presenta la correspondencia entre las conexiones sinápticas y los pesos.

- Capa 1:

Tabla 4

*Pesos sinápticos entre capa de entrada y capa oculta*

		Capa de Entrada		
		Bias	Neurona 01	Neurona 02
Capa Oculta	Neurona 01	$b_1^{(1)} = -121.492$	$W_{11}^{(1)} = -111.229$	$W_{21}^{(1)} = -10.0569$
	Neurona 02	$b_2^{(1)} = 37.0291$	$W_{12}^{(1)} = 45.2544$	$W_{22}^{(1)} = 5.75404$
	Neurona 03	$b_3^{(1)} = -22.6322$	$W_{13}^{(1)} = -19.4119$	$W_{23}^{(1)} = 3.15083$
	Neurona 04	$b_4^{(1)} = 77.6689$	$W_{14}^{(1)} = 88.6816$	$W_{24}^{(1)} = -8.78477$

Nota. Pesos sinápticos de las neuronas de las capas de entrada y oculta. Fuente elaboración propia.



- Capa 2:

Tabla 5

*Pesos sinápticos entre la capa oculta y la capa de salida*

		Capa de Salida
		Neurona 01
<b>Capa Oculta</b>	<b>Bias</b>	$b_1^{(2)} = -2.90933$
	<b>Neurona 01</b>	$W_{11}^{(2)} = -1.11033$
	<b>Neurona 02</b>	$W_{21}^{(2)} = -1.1057$
	<b>Neurona 03</b>	$W_{31}^{(2)} = 2.46844$
	<b>Neurona 04</b>	$W_{41}^{(2)} = -1.15397$

Nota. Pesos sinápticos de las neuronas de las capas ocultas y de salida. Fuente elaboración propia.

Hechas todas las configuraciones de los parámetros determinados manualmente (Tasa de Aprendizaje, Momentum y Topología), y los parámetros generados a partir del proceso de entrenamiento (Pesos Sinápticos). Fue alcanzando un RMSE de 0.556, valor que indica un margen de error aceptable y a la vez descarta un sobreajuste del modelo.

Con el proceso de entrenamiento concluido con éxito, es posible utilizar la opción de generación de código automática, en lenguaje C, que brinda la aplicación. Para la presente investigación se hizo uso de dicha funcionalidad, porque esta acción no afecta de ninguna forma los objetivos planteados, más al contrario es una herramienta que facilita alcanzarlos de forma más rápida. Es preciso mencionar que las RNA son un medio para lograr los objetivos de la investigación.

#### 4.5. Evaluación

Buscando conocer el comportamiento del método propuesto, es llevada a cabo esta fase de la metodología. El método propuesto en la investigación fue sometido a un análisis, posteriormente a la comparación con el valor del RTT esperado y el método estándar que implementa el protocolo SCTP para la estimación de su RTT. Tanto el análisis como la comparación de los métodos se ejecutaron en función a las variables dependientes e independientes determinadas.

##### 4.5.1. Análisis comparativo de los métodos en el cálculo del RTT

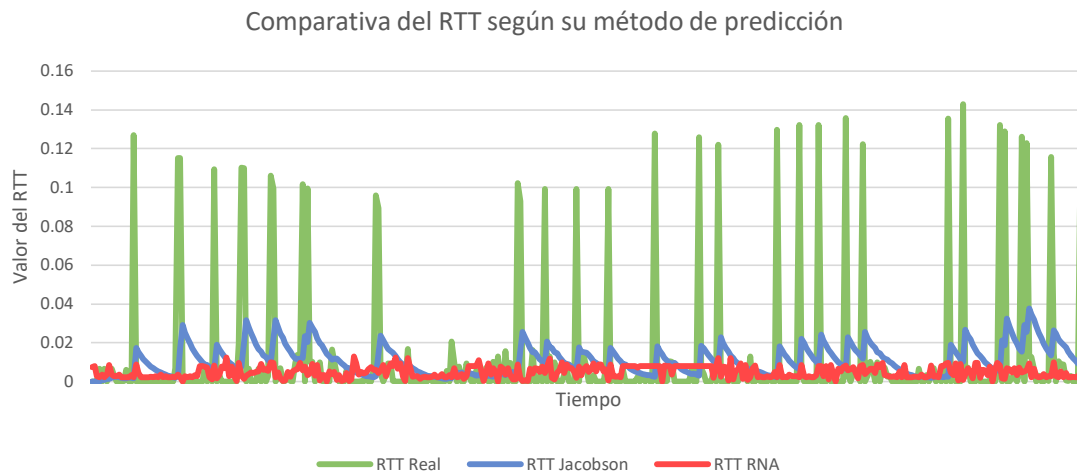
El análisis que exhibirá de mejor manera la asignación del RTT bajo los diferentes métodos es mediante una comparativa de los resultados esperados, los obtenidos mediante los métodos de Jacobson y el que está basado en RNA.



Para la gráfica mostrada a continuación se tiene como ejes el tiempo, que representa la transmisión de los paquetes a lo largo de esta medida. Por otra parte, se tiene que el eje vertical acapara los valores asignados al RTT.

Figura 22

*Análisis comparativo de los valores del RTT*



Nota. Comparativa del valor del RTT bajo los diferentes modelos. Fuente elaboración propia.

La gráfica generada hace posible ver el comportamiento de los resultados arrojados por los métodos de estimación y el valor esperado.

La línea verde es el valor real del RTT, el que fue capturado según se detalla en fases anteriores de la metodología usada. Estos valores son a los que se espera que los métodos de estimación estén ajustados, una mayor aproximación significa una mejor estimación por parte del método.

Referente al método estándar, línea de color azul, es posible observar que hay una estimación que sigue de forma parcial el comportamiento esperado. La forma en la que el método estima el valor del RTT, se puede resaltar cuando se encuentra ante valores del mismo que cambien radicalmente, como en el caso de picos elevados del RTT real. Posteriormente a encontrar estos valores elevados, se presenta un descenso gradual de los valores estimados por el método estándar. Lo que se debe a que, el valor del RTT en instantes de tiempo anterior, alimenta a la variable de entrada que emplea este método para generar su cálculo. Por lo tanto, es posible encontrar que el método estándar se comporta mejor durante variaciones inesperadas en la red.

Las estimaciones obtenidas como resultado del método que se basa en RNA, por su parte, expresan un seguimiento más aproximado para el caso de valores del RTT real que se sitúan en la parte inferior del gráfico, línea roja. Esto quiere decir que, el valor estimado por la RNA tiene





mayor aproximación al valor real, cuando se trata de flujos continuos de transmisión. En el caso de variaciones drásticas, propias de entornos de red inestables, los valores generados por el método propuesto, presentan dificultades para acercarse al valor deseado. Todo ello se debe probablemente, a que durante la fase de limpieza de datos, y con la finalidad de contar con datos de calidad, se procede con la eliminación de valores que alteren de forma drástica el conjunto de datos que se analizará.

Sintetizando, los resultados obtenidos partiendo del método estándar de estimación, son adecuados para valores del RTT real que sean picos elevados, traducido como, entornos de red inestables y variables. Al otro lado, el método de estimación basado en RNA, se ajusta de mejor manera a flujos constantes de datos, debido a que para el caso de entornos de red variable, los valores estimados no muestran mucha proximidad al valor deseado.

El análisis de estos resultados se ve a detalle en el siguiente capítulo.

#### 4.6. Despliegue

Siguiendo el modelo desarrollado en etapas anteriores se procedió a generar el código fuente de la RNA en lenguaje C. A continuación se muestra lo mencionado:

```
#include <math.h>
double Code(double * inputs, double * outputs) {

    double mainWeights[] = {-121.492356957314939, -111.229363729106979, -
    10.056862650769254, 37.029091064501074, 42.254369114148645,
    5.754043123268831, -22.632179789420046, -19.411883132428109,
    3.150828056890509, 77.668941635635633, 88.681567283581089, -
    8.784768278542858, -2.909329016234212, -1.110334894038119, -
    1.010572289155784, 2.468440270987181, -1.153969006970546};

    double * mw = mainWeights;
    double hiddenLayer1outputs[4];
    int c;

    hiddenLayer1outputs[0] = *mw++;

    for(c = 0; c < 2; c++) hiddenLayer1outputs[0] += *mw++ * inputs[c];

    hiddenLayer1outputs[0] = 1.0 / (1.0 + exp(-hiddenLayer1outputs[0]));
    hiddenLayer1outputs[1] = *mw++;

    for(c = 0; c < 2; c++) hiddenLayer1outputs[1] += *mw++ * inputs[c];

    hiddenLayer1outputs[1] = 1.0 / (1.0 + exp(-hiddenLayer1outputs[1]));
    hiddenLayer1outputs[2] = *mw++;

    for(c = 0; c < 2; c++) hiddenLayer1outputs[2] += *mw++ * inputs[c];

    hiddenLayer1outputs[2] = 1.0 / (1.0 + exp(-hiddenLayer1outputs[2]));
    hiddenLayer1outputs[3] = *mw++;
```



```
for(c = 0; c < 2; c++) hiddenLayer1outputs[3] += *mw++ * inputs[c];  
  
hiddenLayer1outputs[3] = 1.0 / (1.0 + exp(-hiddenLayer1outputs[3]));  
outputs[0] = *mw++;  
  
for(c = 0; c < 4; c++) outputs[0] += *mw++ * hiddenLayer1outputs[c];  
  
outputs[0] = 1.0 / (1.0 + exp(-outputs[0]));  
return outputs[0];  
}
```



## CAPÍTULO V: RESULTADOS

La cantidad de registros con los que se trabajó es de 189,141 paquetes. Cada uno de estos paquetes empleados registra información referente a la comunicación desplegada mediante SCTP.

Partiendo de este grupo de datos, se analizó las variables que tienen mayor impacto sobre la variable que se busca predecir, el RTT.

Uno de los principales aportes que se pueden resaltar de la presente investigación es que, se recurrió a variables relacionadas al QoS de la red, de modo que basado en las características del entorno de red pueda calcularse el RTT.

Fue necesario el análisis de las variables y de los datos, para poder determinar qué variables afectan la estimación del valor del RTT. A consecuencia de este estudio surgieron las variables Jitter y Packet Loss Rate como variables de entrada al modelo para la generación del RTT. Este es un enfoque diferente al de investigaciones similares que persiguen el mismo objetivo del presente proyecto, donde se hacía uso del valor del RTT en instantes de tiempo anteriores.

Es de esta manera que, esta investigación da un aporte a las investigaciones en este campo de estudios.

Posterior a la implementación del modelo, se realizó la comparativa de la estimación que realiza el modelo estándar y el modelo basado en RNA.

Es en base a estos resultados obtenidos, es llevado a cabo el siguiente análisis.

### 5.1. Análisis de métodos en función al RTT

Tabla 6

*Análisis en función al R<sup>2</sup>*

Método	R <sup>2</sup>
Método Estándar	0.068
Método basado en RNA	0.330

Nota. Análisis del RTT estimado por los métodos en función al criterio R<sup>2</sup>. Fuente elaboración propia.

El R<sup>2</sup> es un indicador orientado a determinar el nivel de variación de un valor estimado, respecto al valor real de RTT.



Para el análisis de los resultados de este proyecto se hizo uso del mencionado indicador, ello es de importancia para conocer los niveles de variación con el que los métodos de estimación devuelven los resultados.

El rango de valores que acepta este indicador, va desde 0 a 1, donde una mayor aproximación al 0 significa alta variabilidad de los resultados en contraste con el RTT estimado, caso contrario, la variación de la estimación es mínima. En este sentido, la simulación sometida a las condiciones de red antes expuestas y a flujo de datos multimedia, bajo el método estándar indica un  $R^2$  de 0.068, lo que es un resultado que expresa alta variabilidad respecto al comportamiento con el que el RTT real se mueve. El método basado en RNA muestra una ligera superioridad respecto al indicador que se está analizando. El método propuesto alcanzó un  $R^2$  de 0.330 lo que a comparación del método estándar es mayor. Esta diferencia existente al medir bajo el indicador antes mencionado, quiere decir que, el método basado en RNA tiene mayor aproximación al valor del RTT real, cuando es comparado con el RTT generado por el método estándar.

Por otro lado, el análisis del RTT también se hizo mediante los indicadores MAE y RMSE. Ellos señalan el nivel de error con el que las estimaciones del RTT se están generando.

Tabla 7

*Análisis en función al MAE y RMSE*

	MAE	RMSE
<b>Método Estándar</b>	13.196	15.938
<b>Método RNA</b>	5.233	6.165

Nota. Análisis del RTT estimado por los métodos en función al criterio MAE y RMSE. Fuente elaboración propia.

Analizando los resultados obtenidos, está expuesto que, el método estándar en su indicador MAE es de 13.196 ms, por su parte, el mismo indicador para el método basado en RNA alcanza el 5.233 ms. Este análisis puede interpretarse como, una variación ligeramente mayor en la estimación del RTT del método estándar, con respecto al valor real. Lo que es, un mayor error en la predicción del método estándar respecto al método que recurre a las RNA.

Un escenario similar ocurre cuando se mira hacia el indicador RMSE, donde el valor para el método estándar es de 15.938 ms, superior al presentado por el método basado en RNA que es de 6.165 ms. El grado de error existente indica que, el RTT propuesto que se basa en RNA es más próximo al valor real esperado, comparado con el RTT estimado del método estándar.



5.2. Análisis de métodos en función a la Predicción

El indicador que dejará visualizar el impacto de los dos métodos sobre la variable definida como predicción, es la correlación de variables. Con este tenemos expresado numéricamente la relación existente entre el valor esperado y cada uno de los métodos de estimación del RTT.

El cuadro siguiente representa la correlación en función al RTT esperado, para los dos métodos:

Tabla 8

Correlación de valores RTT estimados

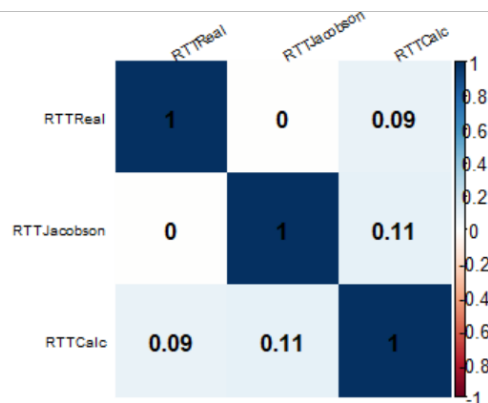
	RTT esperado
<b>Método Estándar</b>	0
<b>Método RNA</b>	0.09

Nota. Análisis de la predicción de los métodos en función al RTT esperado. Fuente elaboración propia.

A continuación se muestra también el gráfico referente a la correlación de variables:

Figura 23

Correlación de variables bajo predicciones de los modelos



Nota. Diagrama de correlación de variables respecto a las predicciones del RTT. Fuente elaboración propia.

El análisis de correlación, que estudia la relación entre las variables sometidas a evaluación, por lo tanto, nos permite medir la predicción que alcanzan las estimaciones que realizan los métodos, respecto a la estimación esperada.

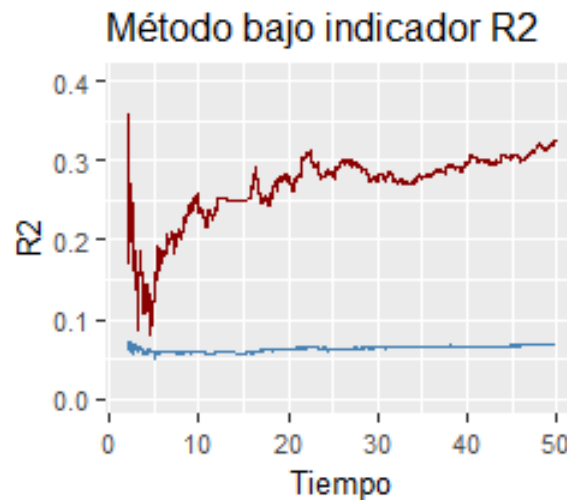
La correlación de los dos métodos, respecto al RTT esperado, muestra que la relación existente con el método basado en RNA presenta un valor de 0.09. En contraste, la relación con el método estándar es de 0. Lo que significa una mayor aproximación en la predicción generada por el método de estimación basado en RNA.

### 5.3. Evolución del $R^2$ por el tiempo

La variación de los valores del RTT estimado y el RTT real, permiten comprender en qué medida se aproximan las estimaciones al valor deseado, y como se mueven en cada instante de tiempo.

Figura 24

*Evolución del  $R^2$  durante la transmisión*



Nota. Gráfico de línea del indicador  $R^2$  en el tiempo de transmisión. Fuente elaboración propia.

Acorde a los análisis realizados, el indicador  $R^2$  del método estándar es de 0.068, lo que señala más variabilidad respecto al valor deseado. Por su parte, el método basado en RNA tiene 0.330 correspondiente al mismo indicador, ello hace referencia a una mayor cercanía de los valores estimados, respecto a los valores deseados.

Con la finalidad de visualizar la evolución de la variabilidad de los métodos, se presenta la figura que desglosa este indicador a lo largo del tiempo. En la figura presentada, se comparan ambos métodos, donde la línea roja pertenece al método basado en RNA, por el otro lado, el método estándar es representado por la línea azul. Para los dos casos, el  $R^2$  a lo largo del tiempo, fluctúa alrededor del valor final del indicador, respectivamente. En el caso de la evolución del  $R^2$  del método basado en RNA, se aprecia un incremento progresivo, en la parte inicial de la figura, que rápidamente se estabiliza alrededor de 0.330, su valor final. Correspondiente al método estándar, se puede ver que desde un inicio, la evolución del  $R^2$ , se mantiene bastante cercana a 0.068, su valor final.

Partiendo de esta figura se muestra que, a lo largo del tiempo, la mayor variabilidad respecto al valor deseado, es presentada por parte del método estándar, pues el comportamiento de su  $R^2$



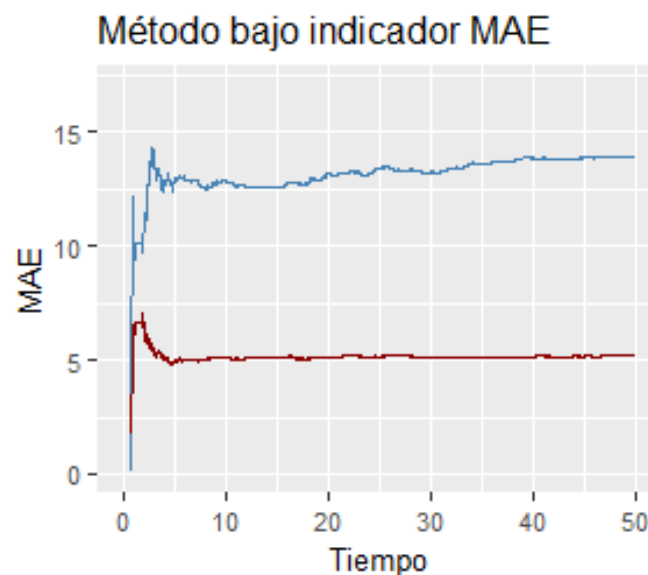
siempre se mantiene más aproximado a 0, en comparación con el indicador del método basado en RNA.

#### 5.4. Evolución del MAE en el tiempo

La forma en la que los niveles de error determinados por el indicador MAE, fueron moviéndose en el tiempo, durante la transmisión de flujos multimedia, es expresada en la siguiente figura. En ella se amplía el panorama de los niveles de error MAE, para los dos métodos en análisis.

Figura 25

*Evolución del MAE durante la transmisión*



Nota. Gráfico de línea del indicador MAE en el tiempo de transmisión. Fuente elaboración propia.

Según el análisis realizado, el valor del MAE para el método estándar es superior al del método basado en RNA, con 13.196 ms y 5.233 ms respectivamente. Ello se refleja en la figura mostrada, y se aprecia que, con la línea roja, los niveles de error MAE del método basado en RNA, en un inicio llegan hasta casi 6.000 ms. Sin embargo, pasado unos segundos desciende y se estabiliza alrededor de su valor final, 5.233 ms. Por su parte, el valor MAE del método estándar, línea azul, también obedece a esta misma fluctuación, y finalmente se sitúa alrededor de su valor final, 13.196 ms.

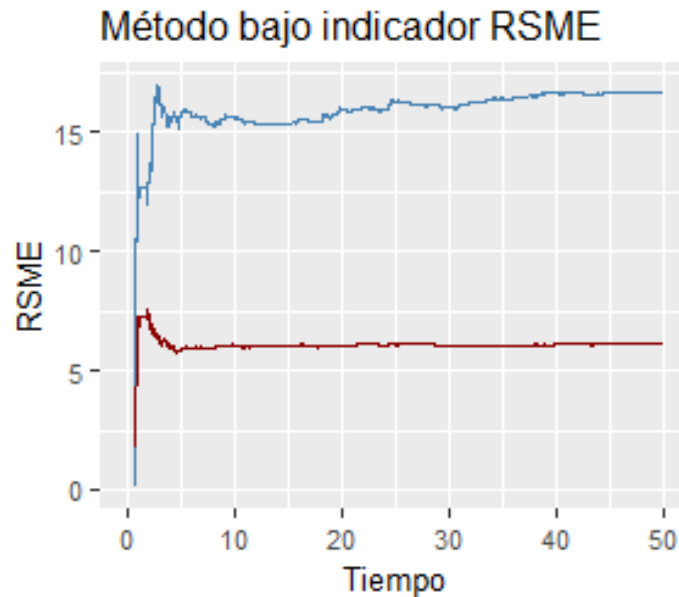
La evolución de los métodos, se puede interpretar de la siguiente forma. A través de toda la transmisión del protocolo, el error de la predicción del método basado en RNA es inferior a la que presenta el método estándar, siendo la diferencia de error entre ambos de 7.963 ms. El nivel de error bajo el indicador MAE es mayormente constante para los dos métodos, no sufren de variaciones drásticas respecto a sus valores finales.

### 5.5. Evolución del RMSE en el tiempo

El comportamiento de los dos métodos, alineados a la desviación representada por el indicador RMSE, es visto a detalle mediante la siguiente figura.

Figura 26

*Evolución del RMSE durante la transmisión*



Nota. Gráfico de línea del indicador RMSE en el tiempo de transmisión. Fuente elaboración propia.

La figura mostrada, indica la medida en la que ambos métodos de estimación se desvían del valor real del RTT. Esta desviación en la estimación es de ayuda para conocer la proporción en la que las estimaciones de ambos métodos, son erróneas respecto al valor real del RTT.

Los segundos iniciales de la transmisión implican alta variabilidad en el valor del RMSE, para ambos métodos, pues el método basado en RNA, línea roja, alcanza un valor de hasta 6.000 ms, y en el caso del método estándar, se llega hasta casi 17.000 ms. Comportamiento que a lo largo del tiempo se estabiliza, respecto al valor final del indicador. Siendo el RMSE del método estándar igual a 15.938 ms, y el correspondiente al método basado en RNA igual a 6.165 ms.

La mayor desviación durante toda la transmisión pertenece al método estándar, lo que significa una estimación del valor del RTT menos aproximada al valor del RTT real, comparada con la estimación que hace el método basado en RNA. La diferencia existente entre ambos métodos de estimación respecto a este indicador es de 9.773 ms, lo cual indica una mejora en la estimación que proporciona el método basado en RNA.





## 5.6. Análisis de Resultados

Partiendo de los diferentes análisis hechos sobre las variables estudiadas a través de sus indicadores, se pueden resaltar los siguientes puntos.

La predicción que realiza el método basado en RNA proporciona resultados más aproximados a los esperados, comparados con la predicción del método estándar. Acorde a la correlación de variables aplicada a los métodos de estimación y la estimación esperada, se presenta una ligera superioridad, en cuanto a la predicción, por parte del método propuesto.

De forma complementaria se encuentra también la evaluación bajo el indicador  $R^2$ , donde se halla menor variabilidad respecto al valor esperado, por parte del método propuesto; esto significa una mayor aproximación del RTT estimado, respecto el RTT real.

Las evaluaciones llevadas a cabo bajo el indicador MAE exponen que la estimación que genera el método basado en RNA, supone un nivel de error inferior cuando se le compara con el método estándar. Lo que en otras palabras significa una estimación menos errónea por parte del método propuesto.

De la misma forma, los niveles de error que se avizoran a través del indicador RMSE, al igual que el indicador mencionado en el párrafo anterior, son inferiores por parte del método basado en RNA, a comparación del nivel de error del método estándar.

Culminada la investigación es posible ver que ambos métodos presentan diferencia respecto a su predicción y su estimación del RTT. Así mismo, es observable que los resultados conseguidos con el método basado en RNA superan ligeramente la predicción y las estimaciones del método estándar.

De esta forma podemos afirmar que la hipótesis de la investigación se concretó. Aunque los resultados del método basado en RNA, sean ligeramente superiores a los correspondientes al método estándar, se considera la investigación como un aporte para este campo de estudios.

En conclusión, se halla que la investigación desarrollada juntamente al método basado en RNA, proporciona una mejor estimación del valor del RTT, que el método estándar.



## CAPÍTULO VI: DISCUSIÓN

La medición de los resultados obtenidos en la investigación fue realizada utilizando indicadores como MAE, RMSE y  $R^2$ . Esta acción hizo posible la evaluación comparativa de la presente investigación junto con la investigación desarrollada por Dong, Du & Yan (2019). A continuación se presenta un cuadro comparativo de los resultados de esta investigación y otras investigaciones.

Especificando únicamente con los resultados propios de la investigación mencionada, se hace la siguiente comparativa.

Tabla 9

*Análisis comparativo con otra investigación*

	MAE	RMSE	$R^2$
Método de estimación RNA para SCTP	5.233	6.165	0.330
Minimal Gated Unit	1.143	1.534	0.997

Nota. Comparativa de resultados con otra investigación. Fuente elaboración propia.

Los resultados obtenidos en las diferentes investigaciones apuntan que la investigación que ha proporcionado mejores resultados para la estimación del valor del RTT es la que emplea el Minimal Gated Unit. Con este método se alcanza una estimación del RTT más aproximado al valor real del RTT, tal como lo expresan los indicadores con los cuales el método fue medido. El MAE de 1.143 ms, RMSE de 1.534 ms y  $R^2$  de 0.997. A diferencia de la presente investigación, donde los indicadores obtuvieron MAE de 5.233 ms, RMSE de 6.165 ms y  $R^2$  de 0.330.



Considerando adicionalmente otros métodos de estimación que fueron propuestos con el mismo propósito, se puede dar una mirada comparativa de los resultados de esta investigación, frente a otras investigaciones con diferentes modelos construidos.

Tabla 10

*Análisis comparativo con otra investigación*

	MAE	RMSE	R <sup>2</sup>
Método Estándar	13.196	15.938	0.068
Método de estimación RNA para SCTP	5.233	6.165	0.330
RNN (Redes Neuronales Recurrentes)	2.257	2.953	0.892
CRTTP (Predictor Cauchy RTT)	2.187	2.768	0.903
LSTM (Larga Memoria a Corto Plazo)	1.927	2.538	0.927
GRU (Unidad Recurrente Cerrada)	1.685	2.162	0.958
MGU (Minimal Gated Unit)	1.143	1.534	0.997

Nota. Comparativa de resultados con otra investigación. Fuente elaboración propia.

Relacionando bajo los mismos indicadores, los resultados de la investigación realizada, junto con estudios desarrollados utilizando otros métodos de estimación. Se destaca que los indicadores relacionados a los niveles de error (MAE y RMSE), son inferiores en otras investigaciones, donde el mínimo MAE es de 1.143 ms que pertenece al método MGU, que a diferencia del método propuesto que es de 5.233 ms. Así también, sujeto al indicador RMSE, se encuentra con el nivel más bajo, al método MGU con un valor de 1.534 ms, el nivel alcanzado por esta investigación, en referencia al mismo indicador es de 6.165 ms.

Con ello se puede observar que los niveles de error alcanzados por otras investigaciones, son menores al que se alcanzó con el método propuesto, sin embargo, la presente investigación mantiene un menor nivel de error respecto al método estándar.

De la misma forma, en lo que refiere al indicador R<sup>2</sup> se presenta mayor cercanía al valor real del RTT, por parte del método MGU, en esta investigación se alcanza el valor de 0.997. Referente al mismo indicador, en la investigación propuesta se tiene un R<sup>2</sup> de 0.330. Esto se traduce en una superioridad de estimación por parte del método MGU respecto al propuesto. De la misma forma que comparando bajo los otros indicadores, las estimaciones que realiza el método propuesto, son mejores a las que hace el método estándar.

Fundamentado en estos resultados sometidos a comparación es posible afirmar que, aún puede trabajarse más en las estimaciones de la RNA perteneciente a esta investigación.



## GLOSARIO

- **Correlación de variables:**  
Es un tipo de asociación entre dos variables numéricas, específicamente evalúa la tendencia (creciente o decreciente) en los datos.
- **Jitter (Variación):**  
La variación del retardo de tránsito de extremo a extremo. En las redes de conmutación de paquetes, Jitter define la distorsión de los tiempos de llegada entre paquetes en comparación con los tiempos entre paquetes de transmisión.
- **MAE (Error Medio Absoluto o Mean Absolute Error):**  
Proporciona una magnitud promedio del error.
- **Momentum:**  
Acelera la convergencia de los pesos de las neuronas. Esta constante hace estable la actualización de los pesos porque toma en cuenta el gradiente de cada iteración.
- **$R^2$  (Coeficiente de Determinación o Multiple Coefficient of Determination):**  
Representa la variación de la data obtenida.
- **RMSE(Error Cuadrático Medio o Root Mean Squared Error):**  
Representa la desviación del modelo, le da más peso a la imprecisión de la estimación.
- **Tasa de Aprendizaje:**  
Controla el tamaño del cambio de los pesos en cada iteración.
- **Tasa de Pérdida de Paquetes (Packet Loss Rate):**  
Normalmente, cuando un camino a un dispositivo está sobrecargado y no puede aceptar datos de entrada en un momento dado se produce la pérdida de paquetes temporal.



## CONCLUSIONES

Acorde al objetivo general, el trabajo desarrollado propone un método de estimación del valor del RTT del protocolo SCTP, alternativo al método estándar para la transmisión de flujos multimedia; para tal propósito se recurre a las RNA. Esta propuesta requirió de plantear variables de estimación, desarrollo del modelo de estimación, la construcción del modelo y la evaluación de las predicciones.

A partir de toda esta investigación se puede determinar como uno de los principales aportes, y respondiendo al primer objetivo específico, la definición de las variables para determinar la estimación del valor del RTT. El uso del Jitter y Packet Loss Rate como variables de entrada para la RNA, dan al presente estudio gran relevancia, porque en todas las investigaciones similares se tomó únicamente el valor del RTT en instantes de tiempo diferentes.

Empleando estas dos variables es que se plantea el modelo basado en RNA, que permitirá la estimación del RTT, como se propone en el segundo objetivo específico. Esta nueva forma de estimación del RTT significa un aporte para los trabajos de investigación en el ámbito de las ciencias de la computación, así como también, en las telecomunicaciones.

Sumado a ello, correspondiente al tercer objetivo específico, se encuentra que la construcción de la RNA implica una gran atención, pues a pesar de que algunos autores sugieren ciertas formas de realizar el diseño de la red neuronal, esto no es completamente certero. Por lo que, el diseño de la RNA requiere probarse cada vez que se modifican los parámetros o la topología, para conocer su comportamiento. Esta etapa de prueba y error de la RNA significa un prolongado periodo de tiempo.

Haciendo referencia al cuarto objetivo específico, la evaluación de las predicciones llevadas a cabo por los dos métodos, demuestra que las Redes Neuronales Artificiales mejoran la estimación del RTT, en contraste con la estimación del método estándar, lo que es de mucha utilidad para protocolos de transporte con altas exigencias de red, como el protocolo SCTP. Este análisis de resultados mediante los indicadores empleados, permite conocer con detalle cómo es el comportamiento de los métodos de estimación desde diferentes puntos de vista. Por esta razón, se puede percibir que los indicadores son los que definen si el modelo es el adecuado para la solución del problema de investigación.



## RECOMENDACIONES

1. El modelo propuesto está diseñado para ser implementado a nivel del Kernel del sistema operativo, teniendo en cuenta de que el método estándar de predicción del RTT implementa su algoritmo a este nivel. Por lo que, se recomienda que en trabajos futuros se lleve el algoritmo propuesto a nivel del sistema operativo.
2. Por otro lado, se propone el desarrollo de métodos de estimación del valor del RTT, empleando Redes Neuronales Recurrentes, pues se vio a través de otros estudios, que ellas pueden también proporcionar mejoras en esta tarea. Por esta razón, se sugiere tomar las variables determinadas en esta investigación, para determinar el RTT, y usarlas conjuntamente con las Redes Neuronales Recurrentes.



## REFERENCIAS BIBLIOGRÁFICAS

- Brahneborg, D., Afzal, W., Čaušević, A., Sundmark, D., & Björkman, M. (2018). Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering. *Round-Trip Time Anomaly Detection* (pp. 107-114). Berlin: ACM.
- Burkov, A. (2019). *The hundred page Machine Learning Book*. Retrieved from <https://books.google.com.pe/books?id=ZF3KwQEACAAJ>
- Carrasco, E., Gamarra, E., & Palomino, E. (2017). *Evaluación del desempeño de las implementaciones del Protocolo de Control de Transmisión TCP en la red de Comunicaciones de la Universidad Nacional de San Antonio Abad del Cusco*. Cusco.
- Chen, X., Kim, H., Aman, J. M., Chang, W., Lee, M., & Rexford, J. (2020). Workshop on Secure Programmable Network Infrastructure (SPIN'20). *Measuring TCP Round-Trip Time in the Data Plane* (p. 7). New York: ACM. Retrieved from Measuring TCP Round-Trip Time in the Data Plane.
- Dong, A., Du, Z., & Yan, Z. (2019, Abril). Round Trip Time Prediction Using Recurrent Neural Networks With Minimal Gated Unit. *IEEE Communications Letters*, 23(4), 584-587. doi:10.1109/LCOMM.2019.2899603
- Gartner. (2018). *Gartner IT Glossary*. Retrieved from RTT (round-trip time): <https://www.gartner.com/it-glossary/rtt-round-trip-time>
- Hernández Sampieri, R., & Mendoza Torres, C. P. (2018). *Metodología de la investigación. Las rutas cuantitativa, cualitativa y mixta*. Ciudad de México, México: Mc Graw Hill Education.
- IBM. (2021). *Conceptos de red y comunicación*. Retrieved from IBM Knowledge Center: [https://www.ibm.com/support/knowledgecenter/es/ssw\\_aix\\_72/com.ibm.aix.networkcomm/network\\_intro.htm](https://www.ibm.com/support/knowledgecenter/es/ssw_aix_72/com.ibm.aix.networkcomm/network_intro.htm)
- IBM. (2022). *SCTP association startup and shutdown*. Retrieved from <https://www.ibm.com/docs/en/aix/7.2?topic=protocol-sctp-association-startup-shutdown>
- IBM. (2022). *Stream Control Transmission Protocol*. Retrieved from <https://www.ibm.com/docs/en/aix/7.2?topic=protocol-stream-control-transmission>



- Juniper Networks. (2021). *TechLibrary*. Retrieved from Securing GTP and SCTP Traffic User Guide for Security Devices: <https://www.juniper.net/documentation/us/en/software/junos/gtp-sctp/topics/topic-map/security-gprs-sctp.html>
- Kelleher, J. (2019). *Deep Learning*. Massachusetts: The MIT Press.
- Kneusel, R. (2021). *Practical Deep Learning: A Python Based Introduction*. San Francisco: No Starch Press, Inc.
- Kurose, J. F., & Ross, K. W. (2021). *Computer Networking a Top-Down Approach*. Retrieved from <https://www.pearson.com/us/higher-education/program/Kurose-Pearson-e-Text-Computer-Networking-Access-Card-8th-Edition/PGM2877610.html>
- Larose, D., & Larose, C. (2019). *Data science using Python and R*. Connecticut, Estados Unidos de America: John Wiley & Sons. Retrieved from <https://es.book.lat/book/5411346/3cf7ed>
- Nannapaneni, R. (2018). Prediction Using Artificial Neural Networks. *2018 Dell EMC Proven Professional Knowledge Sharing*, pp. 1-18.
- Oracle. (2018). *SCTP (Stream Control Transmission Protocol)*. Retrieved from [https://docs.oracle.com/cd/E93309\\_01/docs.466/SIGTRAN/GUID-77BF741F-3990-458D-946C-7B79B0DB3CC6.htm](https://docs.oracle.com/cd/E93309_01/docs.466/SIGTRAN/GUID-77BF741F-3990-458D-946C-7B79B0DB3CC6.htm)
- Oracle. (2022). *Stream Control Transfer Protocol Overview*. Retrieved from ACLI Configuration Guide: <https://docs.oracle.com/en/industries/communications/enterprise-session-border-controller/8.3.0/configuration/stream-control-transfer-protocol-overview.html#GUID-8D95AA21-4530-4E02-90FB-032E3856CC80>
- Palo Alto Networks. (2022). *SCTP Packets and Chunks*. Retrieved from Document:Mobile Network Infrastructure Getting Started: <https://docs.paloaltonetworks.com/service-providers/8-1/mobile-network-infrastructure-getting-started/sctp/sctp-introduction/sctp-packets-and-chunks>
- Peterson, L. L., & Davie, B. S. (2021). *Computer Networks A System Approach* (Sexta ed.). Cambridge: Morgan Kaufman.





- Sharma, S., & Sharma, S. (2020). Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology* (pp. 310-316). Jaipur: IJEAST.
- Stewart, R. (2007). *Stream Control Transmission Protocol*. Retrieved from Request for Comments 4960: <https://tools.ietf.org/pdf/rfc4960.pdf>
- Sunny Valley Networks. (2022). *What is QoS?* Retrieved from Network Basis: <https://www.sunnyvalley.io/docs/network-basics/what-is-qos>
- Tanenbaum, A. S., & Wetherall, D. J. (2021). *Computer Networks* (6 ed.). New Jersey, Estados Unidos de America: Pearson Education. Retrieved from <https://www.studocu.com/en-us/document/california-university-of-pennsylvania/computer-programming-i/computer-networks-6th-edition-by-andrew-s-tanenbaum/23608956>
- Tejada Chávez, F. (2015). Redes neuronales artificiales para la optimización del cálculo de tiempo de ida y vuelta del protocolo de control de transmisión sobre Linux. Cusco, Perú.
- Yasuda, S., & Yoshida, H. (2018). Prediction of round trip delay for wireless networks by a two-state model. *IEEE Wireless Communications and Networking Conference (WCNC)*, 1-6. doi:10.1109/WCNC.2018.8377039

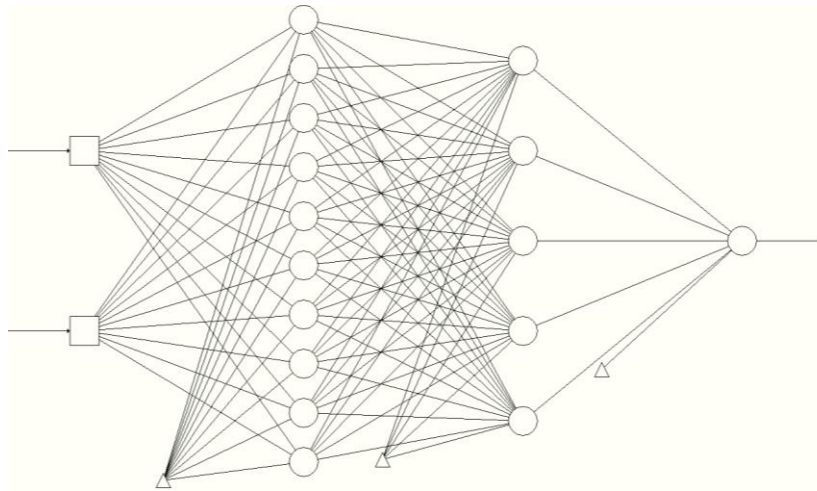


## ANEXOS

Topologías Diseñadas (Entrenamiento prueba - error):

Figura 27

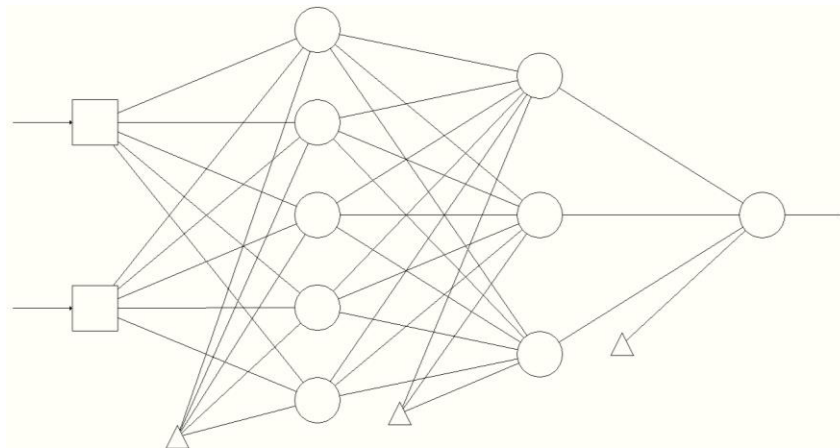
Topología 1



Nota. 1 capa de entrada con 2 neuronas, 1 capa oculta con 10 neuronas, 1 capa oculta con 4 neuronas, 1 capa de salida con 1 neurona. Fuente elaboración propia.

Figura 28

Topología 2

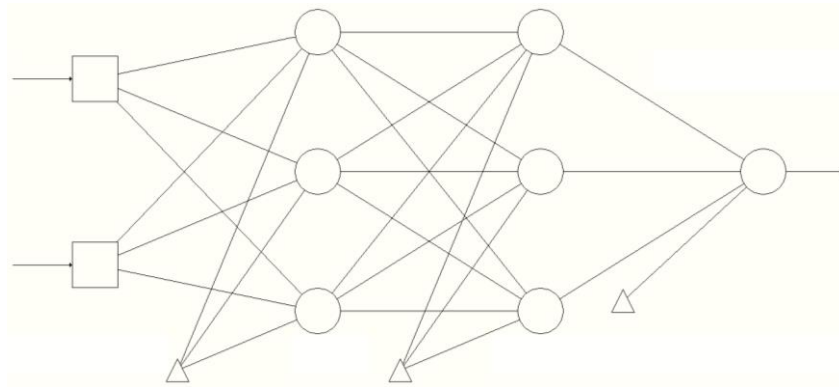


Nota. 1 capa de entrada con 2 neuronas, 1 capa oculta con 5 neuronas, 1 capa oculta con 3 neuronas, 1 capa de salida con 1 neurona. Fuente elaboración propia.



**Figura 29**

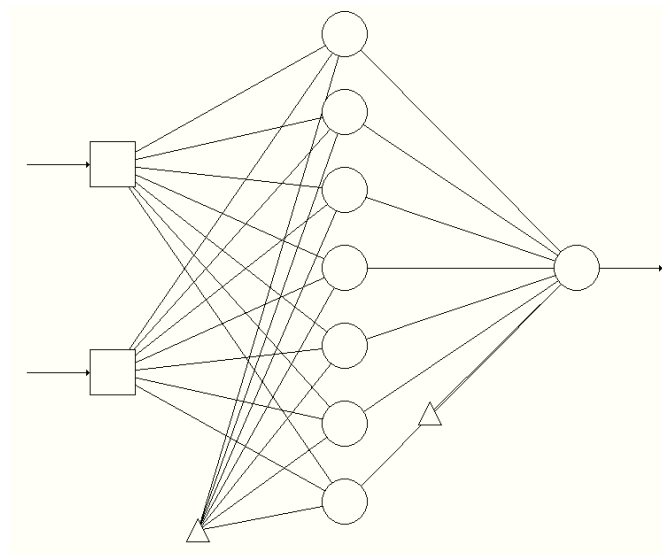
*Topología 3*



Nota. 1 capa de entrada con 2 neuronas, 2 capas ocultas cada una con 3 neuronas, 1 capa de salida con 1 neurona. Fuente elaboración propia.

**Figura 30**

*Topología 4*



Nota. 1 capa de entrada con 2 neuronas, 1 capa oculta con 7 neuronas, 1 capa de salida con 1 neurona. Fuente elaboración propia.