

Tipo de artículo: Artículo original

# Análisis de infraestructura tecnológica para mejorar los procesos de infoestructura del Datacenter de la Universidad Técnica de Manabí

## *Analysis of technological infrastructure to improvise the infostructure processes of the Technical University of Manabí datacenter*

Geovanny Joao Macias Mera<sup>1\*</sup> , <https://orcid.org/0000-0003-3758-5439>

Cesar Armando Moreira Zambrano<sup>2</sup> , <https://orcid.org/0000-0002-0781-0757>

<sup>1</sup> Estudiante, Facultad de Ciencias Informáticas, Universidad Técnica de Manabí. Ecuador. E-Mail: [gmacias9896@utm.edu.ec](mailto:gmacias9896@utm.edu.ec)

<sup>2</sup> Docente, Facultad de Ciencias Informáticas, Universidad Técnica de Manabí. Ecuador. E-Mail: [armando.moreira@utm.edu.ec](mailto:armando.moreira@utm.edu.ec)

\* Autor para correspondencia: [armando.moreira@utm.edu.ec](mailto:armando.moreira@utm.edu.ec)

### Resumen

El objetivo de esta investigación es realizar un análisis que permita identificar el comportamiento de la infraestructura tecnológica de la Universidad Técnica de Manabí (UTM) a través de mecanismos de simulación mediante un ambiente controlado realizando pruebas de estrés y ataques de denegación de servicio a los sistemas y aplicaciones en producción de la UTM que permita distribuir la carga entre los servidores con un balanceador a nivel web para brindar servicios de alta disponibilidad en los sistemas que se encuentran alojados en los servidores, la investigación se la realizo por medio de la metodología PPDIIO, haciendo mención que en toda administración de los sistemas de información existe un tema que tiene un nivel muy elevado de importancia y es el poder obtener alto rendimiento al utilizar técnicas para lograr distribuir el trabajo entre múltiples ordenadores, por intermedio de la aplicabilidad de sistemas convergentes e hiperconvergentes, por lo cual al término de las pruebas de estrés usando la herramienta JMeter se pudo comprobar que los sistemas brindan una mayor estabilidad al utilizar los correctos algoritmos en el balanceador de carga dentro de sus servidores mejorando el rendimiento, optimizando los tiempos de respuesta, como conclusión el uso de balanceadores de carga en los servidores web ayuda a distribuir el peso de manera equilibrada, aumentando la disponibilidad y mejorando la capacidad de respuesta a los servicios.

**Palabras clave:** Infraestructura, infoestructura, balanceo de carga, alta disponibilidad.

### Abstract

*The objective of this research is to perform an analysis to identify the behavior of the technological infrastructure of the Technical University of Manabí (UTM) through simulation mechanisms by means of a controlled environment performing stress test and denial of service attacks to the systems and applications in production of the UTM that allows to distribute the load between the servers with a web level balancer to provide high availability services in the systems that are hosted on the servers, the research was carried out by means of the PPDIIO methodology, mentioning that in all administration of information systems there is an issue that has a very high level of importance and is the power to obtain high performance by using techniques to achieve distribute the work among multiple computers, through the applicability of converged and hyperconverged systems, therefore, at the end of the stress tests using Jmeter tool, it was possible to verify that the systems provide greater stability by using the correct algorithms in the load balancer within their servers, improving performance and optimizing response times. In conclusion, the use of load balancer in web servers helps to distribute the weight in a balanced way, increasing availability and improving responsiveness to services.*



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

**Keywords:** *Infrastructure, infostructure, load balancing, high availability.*

**Recibido:** 21/02/2023  
**Aceptado:** 24/05/2023  
**En línea:** 01/06/2023

## Introducción

En la actualidad el internet tiene una demanda de internautas muy alta y es por esto que los sistemas deben de estar disponibles para los servicios que se les requiera, para lograrlo estos sistemas deben de contar con alta disponibilidad para obtener el rendimiento esperado, (Cano & Rocha, 2019) menciona que los sistemas con grandes cantidades de usuarios tienen servidores funcionales con sistemas convergentes que cumplen con la función de ayudar al rendimiento del sistema, (Sadhu et al., 2022) menciona que cuando el incremento de usuarios dentro de un sistema sobrepasan la cantidad establecida en la infraestructura, el sistema suele presentar falencias y en la mayoría de ocasiones estos problemas suelen ser que la plataforma en la cual está desplegado el sistema se inhiba.

Existen soluciones de balanceo de carga que permiten distribuir la carga de los procesos de infoestructura dentro de la infraestructura del centro de datos, haciendo pruebas de carga y estrés al balanceador para verificar la estabilidad del sistema, encaminando de esta manera a mejoras mediante simulaciones en las que se realizan pruebas a la calidad de rendimiento de los servidores por medio de aplicaciones beneficiando a los estudiantes de la universidad técnica de Manabí solucionando las posibles causas por el cual al momento de acceder al sistema existen este tipo de problemas.

En la actualidad, gracias a los avances tecnológicos se puede llegar a tener grandes adelantos o progresos de una manera más accesible para lo cual se desplegó un laboratorio controlado mediante herramientas de simulación que permitan establecer cargas de trabajo y peticiones a los sistemas alojados en el centro de datos.

Definiendo una arquitectura de comunicación adecuada partiendo de un previo análisis para distribuir la carga entre los servidores que permitan aplicar algoritmos de balanceadores de carga como: Round Robin, Least Connections, Ip Hash para verificar los tiempos de respuestas ante un tráfico de datos dentro de los servidores. Con la finalidad de mejorar los tiempos de respuesta en las aplicaciones en producción del centro de datos.

Según (Barrera Pérez et al., 2019) menciona que gracias a la virtualización se pueden crear ordenadores, servidores, Datacenter e incluso crear clusters para poder obtener beneficios como tal, utilizando este tipo de herramienta se puede simular un laboratorio para realizar las pruebas de rendimiento dentro de la los servidores web.

Este tipo de tecnología, como lo menciona (Chingo & Gómez, 2020) permite ejecutar las pruebas de estrés y ataques de denegación de servicio en un ambiente controlado a un menor costo.

En el complejo ambiente de la infraestructura de los centros de datos es necesario realizar análisis para verificar el



funcionamiento correcto de estos, en su investigación (Peña Casanova & Anías Calderón, 2020) menciona que la infraestructura es esencial para la correcta gestión, donde se puedan integrar los procesos correctamente para que pueda soportar los procedimientos que se llevan a cabo dentro de los servidores para un adecuado balanceo, distribuyendo las peticiones realizadas.

Además, (Cando-Segovia & Medina-Chicaiza, 2021) menciona que obteniendo una adecuada infraestructura, la cual cuente con alta disponibilidad, se puede maximizar la escalabilidad y poder satisfacer las necesidades requeridas en los sistemas computacionales, evitando fallas al momento que existan saturaciones mientras estén funcionando en los servidores universitarios, tales como: el sistema de gestión académica, el sistema de planificación y control académico, sistema de inscripciones y demás.

Durante los momentos críticos en los periodos académicos, como es la época de inscripción se crea un alto flujo de solicitudes al sistema y esto al mismo tiempo crea tráfico dentro de los servidores lo que hace que se creen problemas en la estabilidad del sistema, en su investigación (Moreira Zambrano et al., 2019) menciona el tema de la infoestructura en los centros de datos como un tema de gran consideración, pues con los años la tecnología ha avanzado a tal punto que las instituciones educativas dependen cada día más de la tecnología para gestionar sus procesos académicos de una forma más eficiente.

Existen técnicas que ayudan a brindar un mejor servicio en los centros de datos mejorando su alta disponibilidad reduciendo las fallas en el sistema, mejorando el rendimiento, (Gilces & Villamar, 2019) menciona que la usabilidad de balanceadores de carga dentro de los servidores web de los centros de datos mejorarían la alta disponibilidad y la fiabilidad de la carga de trabajo de los servicios, permitiendo a los usuarios un mejor acceso.

Tomando en cuenta esto para los servidores de la UTM, la carga de trabajo podrá ser distribuida entre varios servidores, equilibrando el balanceo, aumentando la seguridad y reduciendo las posibles saturaciones del sistema, como menciona (Mulla et al., 2019), sin embargo (Omran M A et al., 2022) expresa que al aumentar los servidores y distribuyendo la carga entre ellos se podrá añadir mayor capacidad y escalabilidad mejorando el rendimiento de las solicitudes de los usuarios al enviar peticiones al sistema.

Es importante mencionar que los sistemas deben de tener la capacidad de tener una continuidad dentro del servicio garantizando la estabilidad ante posibles fallos en los ordenadores, haciendo referencia a (Perafan et al., 2018) en su investigación, al tener servidores funcionando de manera distribuida usando balanceadores de carga se obtiene alta disponibilidad dentro de los servicios, sin embargo (Tintín-Perdomo et al., 2018) comenta que los servicios de alta disponibilidad obtienen continuidad con los protocolos correctos evitando el surgimiento de problemas al momento de una caída.



Es por esto que, si se implementa dentro del centro de datos de la Universidad Técnica de Manabí balanceadores de carga en los servidores web, la alta disponibilidad proporcionaría una gran tolerancia en cuanto a los fallos presentados durante el proceso de ejecución de los servidores.

Se puede mencionar que para verificar la estabilidad de un sistema ante grandes cargas de solicitudes es necesario realizar pruebas de estrés a los servidores, analizando como responden ante cargas extremas, como lo indica (Flores & Ugalde, 2022) implementar estas pruebas de estrés saturando los servidores, enviando cierta cantidad de solicitudes por segundo a los sistemas principales de la universidad estudiando el estado de los servidores y analizan su comportamiento ante cargas extremas de peticiones. Según (Aguilar et al., 2022) implementar estas pruebas ayudan a verificar la estabilidad del sistema y verificar cómo va a reaccionar ante cargas elevadas de trabajo.

En los centros de datos es necesario contar con el uso de algoritmos de balanceo de carga para crear un equilibrio en sus servidores, distribuyendo correctamente la información, dirigiendo las solicitudes enviadas a los servidores que están disponibles, como lo menciona (Omid et al., 2019).

En su investigación (Vargas Bermúdez & Báez Pérez, 2019) hace alusión a los algoritmos de balanceo carga, estos distribuyen el envío de peticiones dentro del servidor, mejorando el tráfico en el centro de datos, garantizando el servicio continuo concediendo mayor rapidez al sistema al brindar más recursos en el servidor.

Sin embargo (Ibrahim et al., 2021) explica que los algoritmos como Round Robin, Ip Hash y Least Connections son algunos de los algoritmos utilizados para mejorar considerablemente la capacidad y los tiempos de respuesta en el datacenter.

## **Materiales y métodos**

En esta investigación se planteó realizar un estudio analizando la infraestructura tecnológica de la Universidad Técnica de Manabí, realizando pruebas mediante una simulación en máquinas virtuales para posteriormente obtener resultados aplicando procesos de balanceo de carga y alta disponibilidad de las aplicaciones. La metodología empleada para esta investigación fue PPDIOO basada en su técnica para precisar las actividades que se realizaron en cada etapa del desarrollo de ejecución del análisis de la infraestructura, como se presenta en la Figura 1.





**Figura 1.** Metodología PPDIOO.

### Fase de preparación

En esta etapa, se realiza una investigación acerca de las diversas herramientas que existen para poder llevar a cabo la simulación de la infraestructura del Datacenter como se indica en la tabla 1, teniendo en cuenta que se evaluó cuál es el mejor método para cumplir con los objetivos propuestos garantizando el mejor efecto y obteniendo la solución correcta logrando resultados de manera eficaz y eficiente.

**Tabla 1:** Herramientas para la trabajar durante la simulación.

Número	Sistemas	Herramientas
4	Sistemas operativos	Ubuntu 22.04.1 LTS
1	Sistema de Virtualización	VMware Workstation Tech Preview 2018 Pro
1	Sistema de base de datos	Mysql 8.0.32
3	Servidor Web	Nginx 1.18.0
1	Pruebas de cargas y estrés	JMeter 5.5

### Fase de planeación

Se puede observar en la tabla 2. Se logró identificar los requerimientos que se necesitan para llegar al proceso de simulación, analizar los sistemas operativos que se utilizarán, la intercomunicación entre máquinas, la cantidad de ordenadores virtuales que se implementarán dentro del diseño para poder realizar las pruebas correspondientes donde puedan soportar las cargas enviadas.

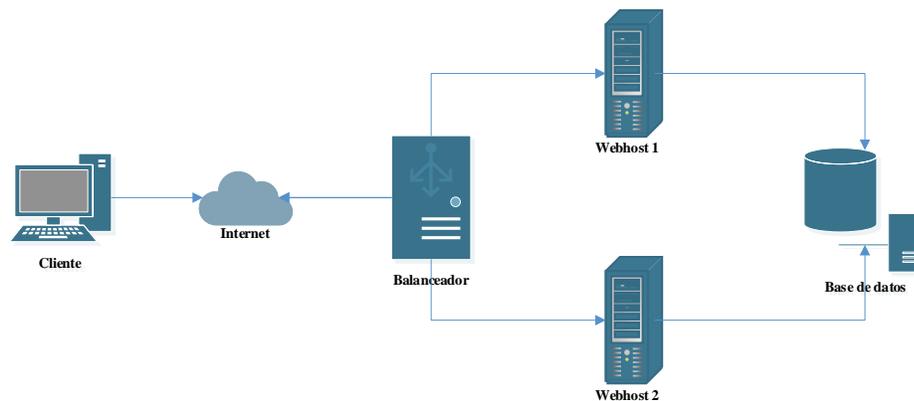


**Tabla 2:** Componentes de infraestructura.

Componentes para la simulación
1. Infraestructura que soporte errores o fallas
2. Plataformas para virtualizar las máquinas que se requieren
3. Sistemas operativos funcionales dentro de las máquinas virtuales
4. Implementación de servicios dentro de los sistemas
5. Pruebas para verificar el comportamiento del sistema

### Fase de diseño

En esta fase se procedió con el diseño de la topología correspondiente para el correcto funcionamiento de un sistema de balanceo de carga, como se indica en la figura 2, en el cual se detalla el diagrama del balanceador propuesto.



**Figura 2:** Topología propuesta implementando balanceadores de carga

En la figura 3 se muestra el diseño topológico que utiliza actualmente la Universidad Técnica de Manabí en su red informática. En dicha configuración se muestra como fluye el tráfico de datos por la red, la cual tiene la capacidad de soportar la conexión de un gran número de usuarios, específicamente 35.000 estudiantes de pregrado y posgrado, a los cuales se les da acceso a las diversas plataformas con las que cuenta la institución.





Web Host 2	Ubuntu, Apache2
Host Base de Datos	Ubuntu, MySQL-Server

### Fase de operación

En esta fase se evalúa el desempeño de la infraestructura mediante el ingreso de datos para comprobar mediante pruebas de estrés y ataques de denegación de servicio la estabilidad del sistema, mostrando y dando resultado sobre infraestructura que soportara dichos datos, en la figura 4 se puede observar la interfaz del aplicativo web donde se ingresa la información que se almacena en la base de datos del balanceador.



Figura 4: Aplicativo para el ingreso de datos.

### Fase de optimización

Durante la última fase se verificó la estabilidad de los servidores al implementar balanceadores de carga mediante pruebas de estrés y ataques de denegación de servicio, comprobando la funcionalidad de los sistemas al enviar determinadas cantidades de peticiones, saturando los servicios y comprobando su estabilidad cuando se llevan los servidores a su límite máximo.

## Resultados y discusión

Al realizar las pruebas de estrés con la herramienta JMeter para analizar el rendimiento de los servidores, simulando un envío masivo de solicitudes hacia el sistema para medir el comportamiento de los servicios ante estas cargas, ya implementada la herramienta, se definen las variables como la dirección IP del sitio y el puerto por el cual se va a trabajar dentro del plan de prueba como se observa en la tabla 5.



**Tabla 5:** Plan de prueba.

Nombre	Valor
Sitio	10.10.10.156
Puerto	80, 443

Dentro del Jmeter, avanzando con la configuración se crea una acción de prueba para poder dar inicio a las cargas de estrés que se le harán al servidor para ver cómo responde, se ingresan los datos para poder conectar el servidor web como se detalla en la tabla 6.

**Tabla 6:** Ingreso de datos al HTTP Request.

<b>Protocolo:</b>	http	<b>Nombre del servidor o IP:</b>	10.10.10.156	<b>Puerto:</b>	80
<b>Petición HTTP:</b>	GET	<b>Ruta:</b>	Matrículas		
<b>Protocolo:</b>	https	<b>Nombre del servidor o IP:</b>	10.10.10.156	<b>Puerto:</b>	443
<b>Petición HTTPS:</b>	GET	<b>Ruta:</b>	Matrículas		

Como se indica en la tabla 7, se determinará la cantidad de usuarios y la frecuencia en periodos de subida de ejecución del proceso en la prueba de estrés mediante la configuración del grupo de hilos. Este varía en el tiempo establecido dependiendo de la carga que se le aplique al balanceador, la cual puede visualizar sus tiempos de ejecución y ver cómo van cambiando los gráficos a medida que avanza.

Conociendo que la Universidad Técnica de Manabí cuenta con 35000 estudiantes, se emplea una prueba de estrés enviando peticiones, aumentando las cantidades en cada testeo para analizar su comportamiento, llegando hasta los 100.000 hilos para verificar como responde el sistema ante estos envíos a los servidores.

**Tabla 7:** Colocación de Hilos para la ejecución de tareas.

Nombre:	Matrículas
Número de Hilos	5000
Periodo de Subida (en segundos):	1

Como se detalla en la tabla 8, durante el tiempo de espera dentro del árbol de resultados se puede observar datos de respuesta, como en qué dirección IP está trabajando el balanceador en ese instante o también el resultado del muestreador.

**Tabla 8:** Muestra de resultados.

Vista de resultados de la muestra	
Nombre de Hilo:	matrículas 1-299



Comienzo de muestra:	2023-02-09 23:33:04 COT
Tiempo de carga:	46833
Connect Time:	4783
Latencia:	46833
Tamaño en bytes:	3531
Bytes enviados:	258
Headers size in bytes:	225
Body size in bytes:	3306
Conteo de muestra:	1
Conteo de Error:	0
Tipo de Dato ("text" "bin" ""):	Text
Código de respuesta:	200
Mensaje de respuesta:	OK
ContentType:	text/html; charset=UTF-8
DataEncoding:	UTF-8

Como se observa en la tabla 9, una vez que se ejecuta la prueba, en la sección de los gráficos de los resultados se puede notar que el servicio está soportando un promedio de 3,152.478 peticiones por minuto.

**Tabla 9:** Datos del gráfico de resultados.

Número de muestras	Desviación	Última Muestra	Rendimiento
5000	9616	93061	3.152.478/min

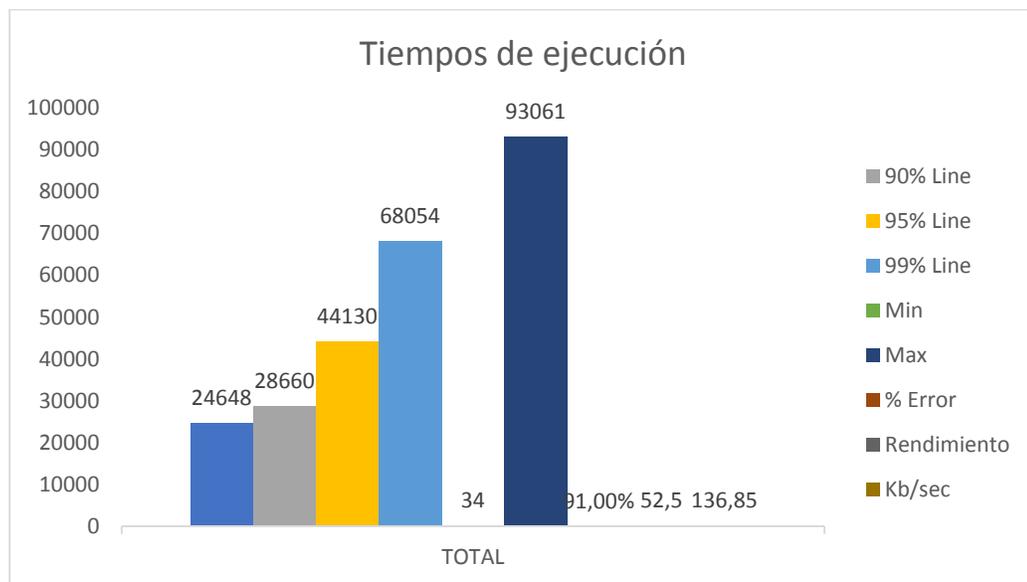
Se puede mencionar que dentro de la tabla existen apartados los cuales expresan el número de muestras ejecutadas con las que se trabajó, seguido de la media que calcula el valor en milisegundos, los tiempos en línea como son 90%, 95% y 99% que representan el tiempo superado para llegar al servidor, mientras que el Min es el tiempo estimado más corto que se tarda la petición en llegar al servidor, de igual forma Max es el tiempo estimado más extenso que se tarda en llegar al servidor, error% que es el porcentaje de muestras con errores, rendimiento que muestra cómo recibe el servidor por segundo las peticiones y los Kb/sec que muestran con Kb que envían y reciben por segundo el cliente y el servidor. Como se puede observar en la tabla 10, al aplicar la prueba al balanceador como tal utilizando el algoritmo para la administración de recursos llamado Round Robin, el cual es el más utilizado dentro de los algoritmos de balanceo de carga porque envía las conexiones hacia los servidores al azar, se obtuvieron los datos estadísticos una vez ejecutada la prueba de carga y estrés dando los datos mostrados en la siguiente tabla.



**Tabla 10:** Datos estadísticos de los tiempos con algoritmo Round Robin.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Max	% Error	Rendimiento	Kb/sec	Sent Kb/sec
TOTAL	4901	24648	21614	28660	44130	68054	34	93061	91.00%	52.5/sec	136.85	1.31

Se puede observar en la figura 5 como se muestran los datos estadísticos de la tabla 10 ya graficados mediante una estructura más ordenada donde se examina a detalle los datos resultantes donde indica que el margen de error es del 91% mientras que las líneas de porcentaje muestran los tiempos en que fueron entregadas las peticiones y que los valores siguen por encima del tiempo promedio, es decir, han cumplido o superado el tiempo de llegada al servidor.



**Figura 5:** Gráficos estadísticos de los tiempos con algoritmo Round Robin.

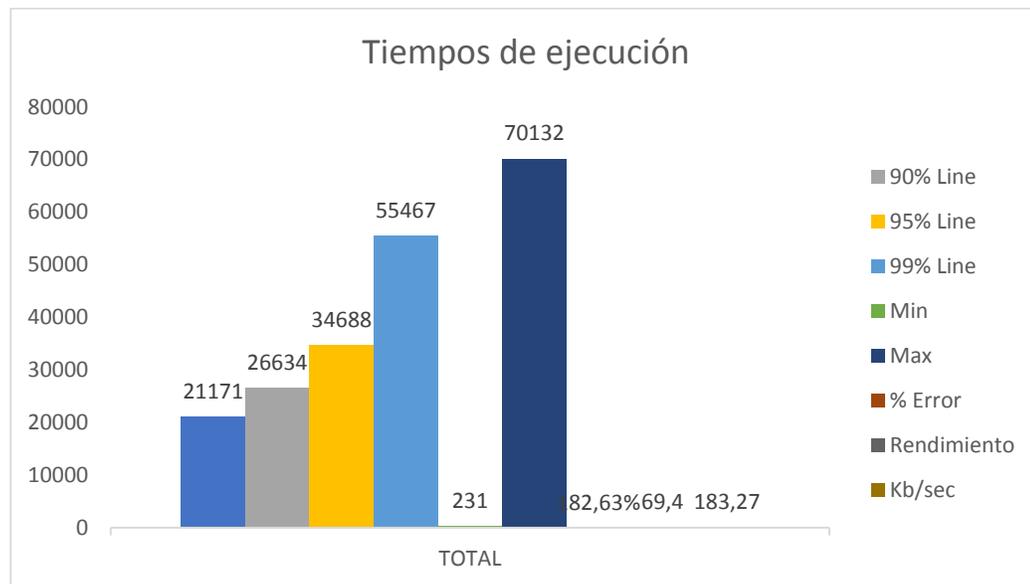
Al realizar las pruebas en el algoritmo Least Connections el cual envía las conexiones al servidor con menos uso para lograr que él envió de datos sea equitativo para todos los servidores que estén funcionales, muestra los resultados de las peticiones que están en línea, como se pueden observar en la tabla 11.



**Tabla 11:** Datos estadísticos de los tiempos con algoritmo Least Connections.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Max	% Error	Rendimiento	Kb/sec	Sent Kb/sec
TOTAL	4927	21171	21423	26634	34688	55467	231	70132	82.63%	69.4/sec	183.27	3.04

Dentro de la figura 6, se puede observar mediante el gráfico estadístico los datos que fueron mostrados previamente en la tabla 11 al proceder con la prueba de estrés al balanceador de carga, donde se indica que el margen de error es del 82,63% mostrando las líneas de porcentaje por encima del tiempo promedio, cumpliendo con los tiempos establecidos.



**Figura 6:** Gráficos estadísticos de los tiempos con algoritmo Least Connections.

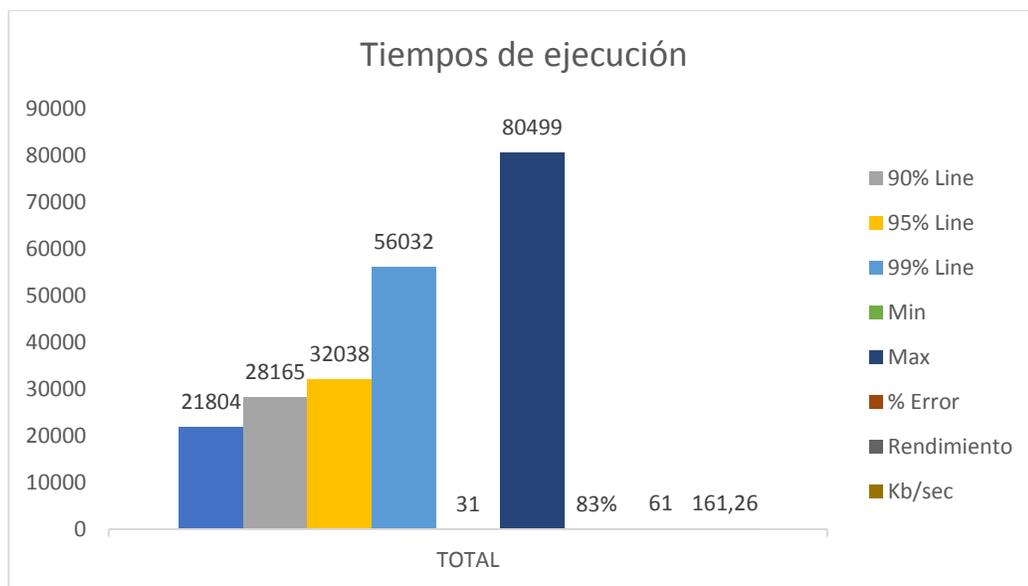
En la tabla 12, se puede observar los datos estadísticos recopilados de las pruebas de estrés en el balanceador de carga, en el cual se utilizó el algoritmo Ip Hash el cual trabaja utilizando un servidor para atender las peticiones enviadas, donde se obtuvo un resultado que demuestra el estado en línea de las peticiones.



**Tabla 12:** Datos estadísticos de los tiempos con algoritmo Ip Hash.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Max	% Error	Rendimiento	Kb/sec	Sent Kb/sec
TOTAL	4977	21804	21450	28165	32038	56032	31	80499	83%	61/sec	161.26	2.67

En la figura 7, se indica de manera gráfica los datos estadísticos de la tabla 12 los cuales fueron reunidos después de saturar los servidores mediante pruebas de estrés para verificar el comportamiento de estos, mostrando que las líneas de porcentaje apuntan los tiempos en que fueron entregadas las peticiones, dando a conocer que los valores de las líneas de porcentaje siguen por encima de la media establecida, mostrando la funcionalidad del servidor ante las pruebas de saturación.



**Figura 7:** Gráficos estadísticos de los tiempos con algoritmo Ip Hash.

Una vez finalizada las pruebas de estrés al balanceador se pudieron obtener datos necesariamente útiles, los cuales ayudaron a comprobar que el algoritmo que mejor prestación da al aplicar estas pruebas mediante la herramienta JMeter es el algoritmo de balanceo de carga Round Robin.

Por otra parte, se realizaron pruebas de estrés mediante ataques de denegación de servicio a los servidores que alojan las páginas principales de la Universidad técnica de Manabí, como se indica en la tabla 13, las cuales respondieron



acorde a la cantidad de peticiones enviadas al ser saturadas mediante pruebas realizadas con la herramienta JMeter de manera similar al sobrecargar los servicios para determinar su comportamiento en condiciones extremas.

**Tabla 13:** Direcciones de las páginas vulneradas.

Nombre del servidor o IP	Descripción
app.utm.edu.ec	Sistema de gestión académica de la de la UTM
preinscripcion.utm.edu.ec	Sistema de inscripciones de la UTM
spca.utm.edu.ec	Sistema de planificación y control académico de la UTM
utm.edu.ec	Página principal de la UTM

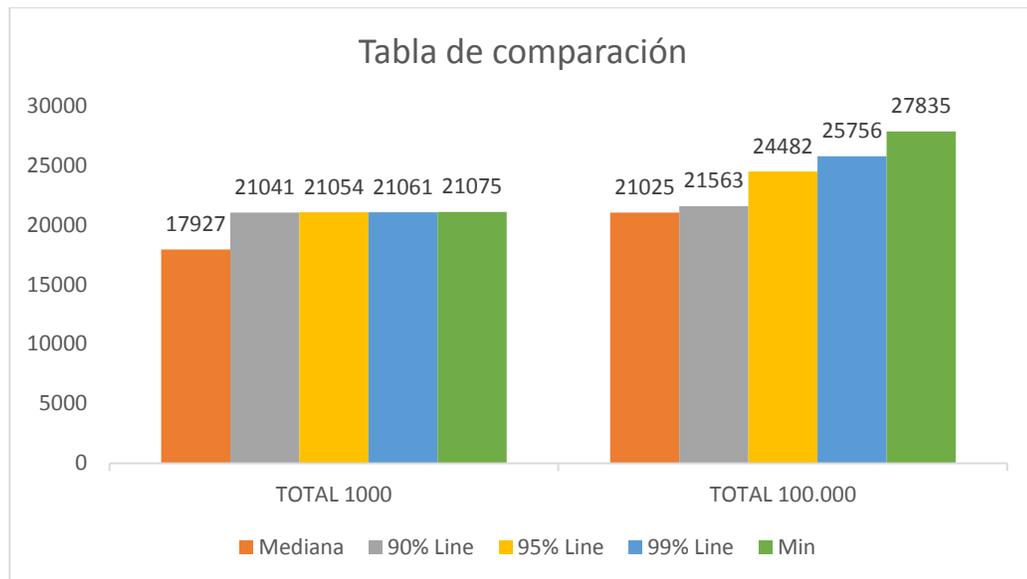
Una vez efectuada las pruebas de estrés, enviando varias peticiones con diferentes números de hilos incrementados en cada ejecución como se observa en la tabla 14, finalmente haciendo una última prueba enviando 100.000 hilos se pudo comprobar que a medida que se iban aumentando las peticiones por segundo los servicios web iban ralentizando sus procesos hasta llegar al fallo, debido a esto, el sistema suele presentar problemas al saturar los servidores, pues estos no cuentan con las medidas para soportar dichas cantidades.

**Tabla 14:** Comparación de datos obtenidos de prueba de estrés.

Etiqueta	Media	Mediana	90% Line	95% Line	99% Line	Min	Max	% Error	Rendimiento	Kb/sec	Sent Kb/sec
<b>TOTAL 1000</b>	17927	21041	21054	21061	21075	190	21108	81%	45	184,29	1,09
<b>TOTAL 100.000</b>	21025	21563	24482	25756	27835	30	39090	91,71%	281,1	952,94	2,96

Al llevar a cabo las pruebas de estrés con los determinados números de hilos en cada petición se pudo notar una respuesta diferente como se observa en la figura 8, al enviar 1000 y 5000 números de hilos, los sistemas continuaron trabajando de la misma manera, mientras que al enviar 25000 números de hilos en sus peticiones los sistemas comienzan a ralentizarse, al ejecutar las 50000 peticiones los sistemas ralentizaron aún más mostrando pequeñas fallas en su servicio, mientras que al enviar las 100.000 peticiones se observó una caída notable dentro de los servicios de los aplicativos de la UTM, mostrando datos estadísticos como se observan en la figura 8 donde los muestreadores superaron los tiempos para llegar al servidor.





**Figura 8:** Gráficos estadísticos de comparación de pruebas.

Estas pruebas indican que, si se implementara un balanceador de carga dentro de los servicios web, se podría distribuir de la manera más óptima las cargas de trabajo dentro del sistema, repartiendo los envíos de datos que llegan a los servidores, equilibrando las solicitudes enviadas con la finalidad de maximizar el rendimiento dentro de los procesos del centro de datos logrando que los servicios funcionarán brindando alta disponibilidad y otorgando la continuidad en los servicios.

Otra manera de observar el balanceo de carga mediante un software libre el cual es comúnmente utilizado para dar mantenimiento y poder administrar un cluster llamado openMosixView que, además, permite visualizar y administrar el rendimiento de los nodos del balanceador de carga, como se puede visualizar en la figura 9.

Al inicializar los servicios el balanceador está equilibrado distribuyendo la carga dentro del cluster de alto rendimiento al tener funcionando sus 6 nodos, potenciando el nodo maestro con los recursos de los nodos esclavos, como se observa en la figura 9, distribuyendo el trabajo dentro de los equipos que están balanceando el sistema concediendo alta disponibilidad al servidor actualizando automáticamente cada 5 segundos distribuyendo la carga de manera correcta.



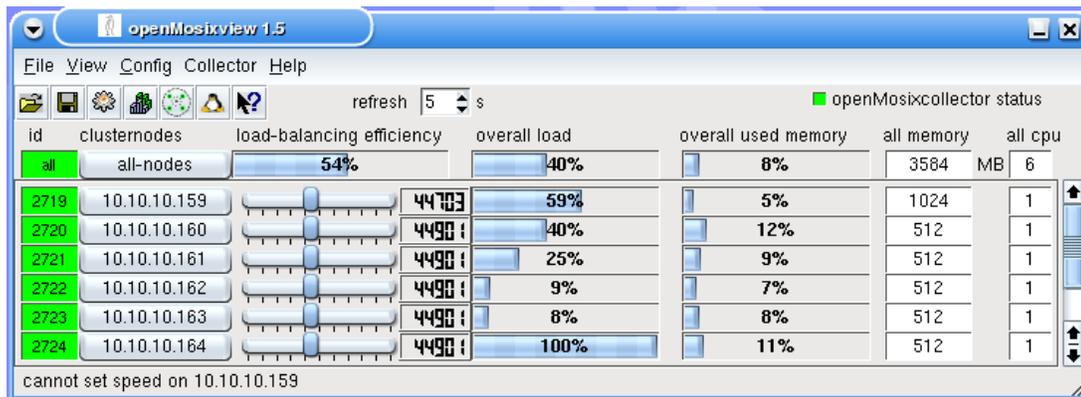


Figura 9: Balanceo de carga en openMosixView.

## Discusión

En esta investigación se efectuaron pruebas para evaluar el rendimiento de los servidores distribuyendo la carga de una manera eficiente bajo condiciones extremas, mostrando datos que respaldan los resultados de (Ali et al., 2022). Donde se observa que destacan que la funcionalidad de los servidores mejora en cuanto a la disponibilidad del sistema ayudando a que los servidores den mejores resultados, dando una mayor optimización en cuanto a las respuestas brindadas por los balanceadores logrando alta disponibilidad dentro de los servicios.

El estudio realizado respalda los hallazgos de (Elveny et al., 2020) quienes indican que implementar balanceadores de carga ayudan a soportar las sobrecargas que ocurren durante las fases en las que existe una alta demanda de peticiones, otorgando mayores beneficios para sus usuarios en cuanto al tiempo de respuesta.

Además, al utilizar estas herramientas se puede aportar superioridad en el rendimiento de los sistemas distribuyendo las peticiones de los usuarios respaldando la investigación de (Ma & Chi, 2022). La cual menciona que los balanceadores ayudan al rendimiento de los servidores porque estos distribuyen la carga para poder lograr un balance, destinando los procesos entre todos los servidores que conectados al balanceador logran un equilibrio, manteniendo el flujo de datos.

Al utilizar las pruebas de estrés verificando el rendimiento con el que trabajan de los servidores se puede comprobar el estado de estos, gracias a las herramientas que existen hoy en día, como lo indica en su trabajo de (Putra, 2022) donde menciona que para estudiar el estado de los servidores y verificar su comportamiento se aplican pruebas de estrés para saturar los servicios y observar su comportamiento, una herramienta bastante utilizada es Apache JMeter la cual trabaja enviando ataques de denegación de servicios a los servidores, sin embargo, (Lebedev & Mihaelis, 2021) indica que la manera en la que trabaja JMeter es produciendo una sobrecarga de datos simulando el ingreso de usuarios dentro de la aplicación enviando solicitudes al servidor saturándolo mediante pruebas de estrés para poder evidenciar el comportamiento del servidor al tener grandes cantidades de solicitudes sobrecargando los sistemas.



## Conclusiones

La implementación de balanceadores de carga mejora significativamente los procesos de despliegue y producción en los centros de datos debido a que logran distribuir las cargas de trabajo entre los servidores disponibles, evitando que los servicios se saturen ante la demanda de solicitudes en el sistema obteniendo un mejor rendimiento y una alta disponibilidad en los sistemas de la universidad reduciendo los tiempos de espera por inactividad mejorando la escalabilidad del sistema.

Los servicios de alta disponibilidad permiten que los servidores se equilibren y ayuden a proteger la información ante cualquier tipo de fallo dentro del sistema, debido a que los servidores que funcionan correctamente utilizando algoritmos de balanceo de carga como Round Robin, Least Connections o IpHash suelen reducir el tráfico de datos del servidor, pues estos brindan un alto rendimiento y una alta disponibilidad.

Los balanceadores de carga se pueden definir como herramientas escalables que ayudan a mejorar la infoestructura dentro de la infraestructura, logrando que los sistemas y servicios tengan una mayor estabilidad dentro de los servidores, estos distribuyen el tráfico de la red entre los nodos disponibles para así aumentar la disponibilidad de los sistemas, los cuales para verificar su funcionalidad en muchas ocasiones son sometidos a pruebas de estrés para determinar la estabilidad ante cargas excesivas en condiciones extremas.

## Agradecimientos

Quiero expresar mi más profundo agradecimiento, primeramente a Dios por haberme permitido llegar hasta este momento y darme la voluntad y fortaleza para seguir luchando por alcanzar mis objetivos, a la universidad por brindarme excelentes años y aumentar mi conocimiento en la materia, también a mis docentes que se han esforzado día a día por compartir sus vivencias y conocimientos para fortalecerme profesionalmente, a mi familia y amigos por ser fortaleza y apoyo en los momentos difíciles que se han presentado a lo largo de mi carrera profesional.

## Conflictos de intereses

En la investigación realizada no se tiene conflicto de intereses, por lo tanto, se permite compartir y utilizar el contenido del presente artículo.

## Contribución de los autores

1. Conceptualización: Geovanny Joao Macias Mera, Cesar Armando Moreira Zambrano.



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

2. Curación de datos: Geovanny Joao Macias Mera.
3. Análisis formal: Geovanny Joao Macias Mera.
4. Adquisición de fondos: No aplica
5. Investigación: Geovanny Joao Macias Mera.
6. Metodología: Geovanny Joao Macias Mera.
7. Administración del proyecto: Geovanny Joao Macias Mera, Cesar Armando Moreira Zambrano.
8. Recursos: Cesar Armando Moreira Zambrano.
9. Software: Geovanny Joao Macias Mera, Cesar Armando Moreira Zambrano.
10. Supervisión: Geovanny Joao Macias Mera, Cesar Armando Moreira Zambrano.
11. Validación: Cesar Armando Moreira Zambrano.
12. Visualización: Geovanny Joao Macias Mera, Cesar Armando Moreira Zambrano.
13. Redacción – borrador original: Geovanny Joao Macias Mera, Cesar Armando Moreira Zambrano.
14. Redacción – revisión y edición: Geovanny Joao Macias Mera, Cesar Armando Moreira Zambrano.

## Financiamiento

La investigación fue respaldada financieramente por recursos propios.

## Referencias

- Aguilar, R., García, N., & Díaz, J. (2022). A Systematic Mapping Study in the area of Software Testing. *International Journal of Combinatorial Optimization Problems & Informatics.*, 13(4), 47 - 57.
- Ali, A., Maghawry, H. A., & Badr, N. (2022). Performance testing as a service using cloud computing environment: A survey. *Journal of software (Malden, MA)*, 34(12). <https://doi.org/https://doi.org/10.1002/smr.2492>
- Barrera Pérez, M. Á., Serrato Losada, N. Y., Rojas Sánchez, E., & Mancilla Gaona, G. (2019). State of the art in software defined networking (SDN). *Visión Electrónica*, 13(1), 178-194. <https://doi.org/https://doi.org/10.14483/22484728.14424>
- Cando-Segovia, M. R., & Medina-Chicaiza, P. (2021). Prevención en ciberseguridad: enfocada a los procesos de infraestructura tecnológica. *3C TIC Cuadernos de desarrollo aplicados a las TIC*, 10(1), 17- 41. <https://doi.org/https://doi.org/10.17993/3ctic.2021.101.17-41>



- Cano, J., & Rocha, A. (2019). Ciberseguridad y ciberdefensa. Retos y perspectivas en un mundo digital. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*(32), vii-vii. <https://doi.org/http://dx.doi.org/10.17013/risti.32.0>
- Chingo, R., & Gómez, O. (2020). Tecnología de contenedores y su aplicación en el aprendizaje de ciberseguridad: una revisión sistemática de literatura. *RECIBE, Revista electrónica de Computación, Informática, Biomédica y Electrónica*, 9(2), C1-20. <https://doi.org/https://doi.org/10.32870/recibe.v9i2.186>
- Elveny, M., Winata, A., Siregar, B., & Syah, R. (2020). Load balancers in a container technology system using docker swarms on a single board computer cluster. *Elementary Education Online*, 19(4), 744-751. <https://doi.org/doi:10.17051/ilkonline.2020.04.178>
- Flores, M. E., & Ugalde, F. R. (2022). Métricas de potencia y consumo en equipo computacional portátil bajo estrés mediante pruebas de desempeño. *Ingeniería: Revista de la Universidad de Costa Rica*, 32(2), 201-204. <https://doi.org/DOI 10.15517/ri.v32iNE2.50666>
- Gilces, C. E., & Villamar, R. P. (2019). Aplicación de Balanceo De Carga Dinámico Para Servidores, Basada En Redes Definidas Por Software. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, 3(32), 67-82. <https://doi.org/10.17013/risti.n32.67-82>
- Ibrahim, I. M., Ameen, S. Y., Yasin, H. M., Omar, N., Kak, S. F., Rashid, Z. N., Salih, A. A., Salim, N. O., & Ahmed, D. M. (2021). Web server performance improvement using dynamic load balancing techniques: A review. *Asian Journal of Research in Computer Science*, 10(1), 47-62. <https://doi.org/10.9734/ajrcos/2021/v10i130234>
- Lebedev, V. S., & Mihaelis, V. V. (2021). Overview of software tools for the organization of load testing of information systems. *Information Technology and Mathematical Modeling in the Management of Complex Systems*, 10(2), 40-45. [https://doi.org/10.26731/2658-3704.2021.2\(10\).40-45](https://doi.org/10.26731/2658-3704.2021.2(10).40-45)
- Ma, C., & Chi, Y. (2022). Evaluation test and improvement of load balancing algorithms of nginx. *IEEE access: practical innovations, open solutions*, 10, 14311-14324. <https://doi.org/doi: 10.1109/ACCESS.2022.3146422>
- Moreira Zambrano, C. A., Riveros Villareal, V. S., & Zambrano Romero, W. D. (2019). Infoestructura tecnológica y sus elementos estructurales para la educación superior. *Mundo Recursivo*, 2(1), 38 - 55.
- Mulla, M. M., Raikar, M. M., Meghana, M. K., Shetti, N. S., & Madhu, R. K. (2019). Load balancing for software-defined networks. *Springer*, 235-244. [https://doi.org/https://doi.org/10.1007/978-981-13-5802-9\\_22](https://doi.org/https://doi.org/10.1007/978-981-13-5802-9_22)
- Omid, J., Subhi, Z., & Zebari, R. (2019). A state of art survey for web server performance measurement And load balancing mechanisms. *International Journal of Scientific & Technology Research*, 8(12), 535-543.



- Omran M A, A., Z Zainal, A., N A, Z., & Z Abal, A. (2022). Software defined network based load balancing for network performance evaluation. *International journal of advanced computer science and applications: IJACSA*, 13(4), 117 - 124. <https://doi.org/10.14569/ijacsa.2022.0130414>
- Peña Casanova, M., & Anías Calderón, C. (2020). Modelo para la gestión de infraestructuras de tecnologías de la información. *TecnoLógicas*, 23(48), 31-53. <https://doi.org/https://doi.org/10.22430/22565337.1449>
- Perafan, H., Guia, N., Rey, D., & Duarte, D. (2018). Diseño de un cluster de alta disponibilidad para un entorno educativo virtual universitario. *Revista INGENIERÍA UC*, 25(1), 108-116. <https://www.redalyc.org/articulo.oa?id=70757668014>
- Putra, E. L. (2022). A literature review on load and performance testing methods for website optimization. *Proceedings of the Informatics Conference*, 8(16), 28-28.
- Sadhu, P. K., Yanambaka, V. P., & Abdelgawad, A. (2022). Internet of Things: Security and Solutions Survey. *Sensors (Basel, Switzerland)*, 22(19), 7433. <https://doi.org/https://doi.org/10.3390/s22197433>
- Tintín-Perdomo, V. P., Caiza-Caizabuan, J. R., & Caicedo-Altamirano, F. S. (2018). Arquitectura de redes de información. Principios y conceptos. *Dominio de las Ciencias*, 4(2), 103-122. <https://doi.org/10.23857/dc.v4i2.780>
- Vargas Bermúdez, F. A., & Báez Pérez, C. I. (2019). Estrategias de planificación para datos y procesos en computación Grid: estado del arte. *Ingenierías USBmed*, 10(1), 40-52. <https://doi.org/https://doi.org/10.21500/20275846.3805>

