



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

ANÁLISIS COMPARATIVO DE RENDIMIENTO A SERVIDORES WEB DE
DISTRIBUCIÓN LIBRE UTILIZANDO APACHE BENCHMARK

RAMIREZ AZANZA MARIA GABRIELA
INGENIERA DE SISTEMAS

MACHALA
2019



UTMACH

FACULTAD DE INGENIERÍA CIVIL
CARRERA DE INGENIERÍA DE SISTEMAS

ANÁLISIS COMPARATIVO DE RENDIMIENTO A SERVIDORES
WEB DE DISTRIBUCIÓN LIBRE UTILIZANDO APACHE
BENCHMARK

RAMIREZ AZANZA MARIA GABRIELA
INGENIERA DE SISTEMAS

MACHALA
2019



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

EXAMEN COMPLEXIVO

ANÁLISIS COMPARATIVO DE RENDIMIENTO A SERVIDORES WEB DE
DISTRIBUCIÓN LIBRE UTILIZANDO APACHE BENCHMARK

RAMIREZ AZANZA MARIA GABRIELA
INGENIERA DE SISTEMAS

VALAREZO PARDO MILTON RAFAEL

MACHALA, 22 DE AGOSTO DE 2019

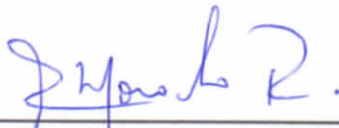
MACHALA
22 de agosto de 2019

Nota de aceptación:

Quienes suscriben, en nuestra condición de evaluadores del trabajo de titulación denominado Análisis comparativo de rendimiento a servidores web de distribución libre utilizando Apache Benchmark, hacemos constar que luego de haber revisado el manuscrito del precitado trabajo, consideramos que reúne las condiciones académicas para continuar con la fase de evaluación correspondiente.



VALAREZO FARDO MILTON RAFAEL
0704518893
TUTOR - ESPECIALISTA 1



MOROCHO ROMAN RODRIGO FERNANDO
0703820464
ESPECIALISTA 2



LOJÁN CUEVA EDISON LUIS
0703249698
ESPECIALISTA 3

Fecha de impresión: jueves 22 de agosto de 2019 - 13:46

Urkund Analysis Result

Analysed Document: RAMIREZ AZANZA MARIA GABRIELA_PT-010419.pdf (D54804886)
Submitted: 8/13/2019 10:25:00 PM
Submitted By: titulacion_sv1@utmachala.edu.ec
Significance: 3 %

Sources included in the report:

https://es.wikipedia.org/wiki/Servidor_web
<https://duplika.com/blog/que-son-los-servidores-web-y-por-que-son-necesarios/>
<https://blog.infranetworking.com/apache-vs-nginx-vs-litespeed/>
<https://fr.slideshare.net/juliojmj/servidor-web-13032139>
https://www.ecured.cu/Servidor_web

Instances where selected sources appear:

6

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

La que suscribe, RAMIREZ AZANZA MARIA GABRIELA, en calidad de autora del siguiente trabajo escrito titulado Análisis comparativo de rendimiento a servidores web de distribución libre utilizando Apache Benchmark, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

La autora declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

La autora como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 22 de agosto de 2019



RAMIREZ AZANZA MARIA GABRIELA
0706578259

DEDICATORIA

Dedico este trabajo a mi madre, la cual ha sido un pilar fundamental en todo este camino recorrido, a mi padre que con su perseverancia me ha enseñado a no decaer y seguir adelante.

A mi hermano Francis por su ejemplo y dedicación, ya que admiro todo lo que ha logrado, y por último a mi hermana Camila pese a su corta edad posee una madurez que me ha ayudado.

Srta. María Gabriela Ramírez Azanza

AGRADECIMIENTO

Agradezco a mis padres por su esfuerzo y apoyo incondicional que me han brindado para poder alcanzar mis objetivos y metas en el transcurso de este camino que no ha sido fácil. A mis maestros que con sus conocimientos me han enseñado todo lo concerniente de la carrera, y sus experiencias han sido de gran ayuda para poder ser una excelente profesional.

Srta. María Gabriela Ramírez Azanza

RESUMEN

ANÁLISIS COMPARATIVO DE RENDIMIENTO A SERVIDORES WEB DE DISTRIBUCIÓN LIBRE UTILIZANDO APACHE BENCHMARK.

Ramírez Azanza María Gabriela,
0706578259

Los servidores web se utilizan actualmente por la gran mayoría de empresas, ya que cualquier persona que tenga acceso al internet puede acceder a una página web, no importando en que alojamiento web se encuentre; y mucho mejor si se puede utilizar libremente. Los sistemas operativos de distribución libre son basados en Linux, incluyen su propio entorno de escritorio y su licencia es libre, es decir, libertad de usar el programa, con cualquier propósito. El rendimiento del servidor web es un problema crítico para los sitios que atienden un gran volumen de solicitudes. Por lo que existen herramientas para medir el rendimiento del servidor que se vaya a utilizar, para verificar si es óptimo. Los servidores web más utilizados son: Apache, Nginx y OpenLiteSpeed, ya que son de código abierto, permitiendo personalizar y ampliar por módulos sin afectar la base de los mismos. El presente trabajo se ha realizado con la finalidad de realizar un análisis comparativo acerca de cuál es el mejor servidor web instalado en un sistema operativo de distribución libre, y con la ayuda de la herramienta Apache Benchmark también de código abierto, permitiendo medir el rendimiento de cada servidor mencionado, ayudando a precisar la capacidad de concurrencia que comprueba condiciones de carrera o bloqueos.

Palabras claves: Análisis Comparativo, Apache Benchmark, Distribución Libre, Rendimiento, Servidores Web.

ABSTRACT

COMPARATIVE ANALYSIS OF PERFORMANCE TO FREE DISTRIBUTION WEB SERVERS USING APACHE BENCHMARK.

Ramírez Azanza María Gabriela,
0706578259

Web servers are currently used by the vast majority of companies, since anyone who has access to the Internet can access a web page, no matter in which web hosting it is; and much better if it can be used freely. Free distribution operating systems are Linux-based, include their own desktop environment, and are licensed free, that is, free to use the program for any purpose. Web server performance is a critical issue for sites that handle a large volume of requests. So there are tools to measure the performance of the server to be used, to verify if it is optimal. The most used web servers are: Apache, Nginx and OpenLiteSpeed, since they are open source, allowing to customize and expand by modules without affecting their base. This work has been carried out with the aim of making a comparative analysis about which is the best web server installed in a free distribution operating system, and with the help of the Apache Benchmark tool also open source, allowing to measure the performance of each server mentioned, helping to specify the concurrency capacity that checks race conditions or blockages.

Keywords: Comparative Analysis, Apache Benchmark, Centos, Free Distribution, Performance, Web Server.

CONTENIDO

DEDICATORIA	3
AGRADECIMIENTO	4
RESUMEN	5
ABSTRACT	6
ÍNDICE DE ILUSTRACIONES	8
ÍNDICE DE TABLAS	9
1. INTRODUCCIÓN	10
1.1. Marco contextual.....	11
1.2. Problema.....	11
1.3. Objetivo general	11
2. DESARROLLO	12
2.1. Marco teórico.....	12
2.1.1. <i>Sistemas Operativos de Distribución Libre</i>	12
2.1.2. <i>Linux</i>	12
2.1.3. <i>CentOS 7</i>	12
2.1.4. <i>Servidores web</i>	13
2.1.5. <i>Protocolo HTTP</i>	13
2.1.6. <i>Apache Http Server</i>	13
2.1.7. <i>Nginx</i>	13
2.1.8. <i>LiteSpeed Web Server</i>	14
2.1.9. <i>Rendimiento Servidor Web (Benchmark)</i>	14
2.1.10. <i>Apache Benchmark</i>	14
2.2. Solución del problema	15
2.2.1. <i>Selección de la herramienta</i>	15
2.2.2. <i>Parámetros de calificación</i>	15
2.2.3. <i>Características a medir en el análisis comparativo</i>	15
2.2.4. <i>Diseño de las pruebas</i>	16
2.3. Resultados.....	17
2.3.1. <i>Evaluación de características propuestas a cada servidor</i>	17
2.3.2. <i>Evaluación de carga y estrés mediante pruebas a cada servidor</i>	17
3. CONCLUSIONES	19
BIBLIOGRAFÍA	20
ANEXOS.....	22

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Resultado tiempo de atención por porcentaje de petición con la página web de prueba	18
Ilustración 2. Instalación del servidor HTTP Apache mediante comando yum	22
Ilustración 3. Inicio del servidor HTTP apache con el comando service.....	22
Ilustración 4. Ingreso a navegador para verificar servidor activo.....	22
Ilustración 5. Configuración en el archivo hosts con el editor de texto nano	23
Ilustración 6. Ruta del archivo para configurar el servidor Apache HTTP	23
Ilustración 7. Configuración del archivo httpd.conf con el editor de texto gedit	23
Ilustración 8. Ruta de configuración de hosts y creación de archivo titulacion.conf....	24
Ilustración 9. Configuración del archivo titulacion.conf con el editor de texto gedit	24
Ilustración 10. Ruta de archivo de servidor HTTP para almacenar páginas web	24
Ilustración 11. Reinicio del servidor HTTP Apache y verificación de página web en host cliente (Windows)	24
Ilustración 12. Instalación del servidor Nginx con el comando yum	25
Ilustración 13. Iniciar, habilitar y verificar estado del servidor Nginx	25
Ilustración 14. Cambio de dirección IP a la máquina del servidor con el comando ifconfig.....	25
Ilustración 15. Verificación de funcionamiento del servidor con dirección IP de la PC	26
Ilustración 16. Ruta de archivo de configuración del servidor Nginx	26
Ilustración 17. Configuración del archivo default.conf con editor de texto gedit	26
Ilustración 18. Ruta de archivo de servidor NGINX para almacenar páginas web	27
Ilustración 19. Reinicio del servidor NGINX y verificación de página web en host cliente (Windows).....	27
Ilustración 20. Instalar repositorio con el comando rpm	28
Ilustración 21. Instalación del servidor OpenLiteSpeed (última versión)	28
Ilustración 22. Instalación de mariadb server con el comando yum	28
Ilustración 23. Instalación de php versión 5.6 y dependencias mediante yum	28
Ilustración 24. Configuración de nombre de usuario y contraseña	28
Ilustración 25. Creación de enlace simbólico para lshp y lshp5	29
Ilustración 26. Iniciar y habilitar mariadb mediante comando systemctl	29
Ilustración 27. Configuración de mariadb con el comando mysql_secure_installation	29
Ilustración 28. Verificación del estado del servidor y visualización de IP de la máquina	29
Ilustración 29. Verificación de funcionamiento de Servidor OpenLiteSpeed.....	30
Ilustración 30. Creación de Carpeta donde se almacenará página	30
Ilustración 31. Inicio en interfaz web (WebAdmin) del servidor OpenLiteSpeed.....	30
Ilustración 32. Creación de Virtual Hosts en OpenLiteSpeed.....	31
Ilustración 33. Creación y configuración de rutas para almacenamiento de página web	31
Ilustración 34. Creación y configuración de archivo Config File del Virtual Host	32
Ilustración 35. Configuración General del Virtual Host creado en OpenLiteSpeed....	32
Ilustración 36. Configuración General de Index Files del Virtual Host.....	33
Ilustración 37. Configuración de archivo log de error para el virtual host	33

Ilustración 38. Configuración de archivo Access log para el virtual host	33
Ilustración 39. Configuración de seguridad para el virtual host	34
Ilustración 40. Creación de Listener para el puerto 80 (HTTP)	34
Ilustración 41. Añadidura del virtual host al listener puerto 80 (HTTP).....	35
Ilustración 42. Reinicio del servidor para efectuar los cambios	35
Ilustración 43. Funcionamiento de página web en host cliente (Windows).....	35
Ilustración 44. Prueba 1. Tiempo de atención por porcentaje de petición a página web	37
Ilustración 45. Prueba 2. Tiempo de atención por porcentaje de petición a página web	37
Ilustración 46. Prueba 3. Tiempo de atención por porcentaje de petición a página web	37
Ilustración 47. Prueba 4. Tiempo de atención por porcentaje de petición a página web	38
Ilustración 48. Prueba 5. Tiempo de atención por porcentaje de petición a página web	38
Ilustración 49. Utilización de la herramienta Apache Benchmark en el servidor OpenLiteSpeed	39

ÍNDICE DE TABLAS

Tabla 1. Requisitos de sistema CentOS 7	12
Tabla 2. Parámetros de Apache Benchmark.....	14
Tabla 3. Tipos de pruebas a servidores web.....	15
Tabla 4. Números de peticiones y cantidad de concurrencia para cada prueba	16
Tabla 5. Resultados de acuerdo a la prueba 6 de cada servidor.....	17
Tabla 6. Prueba 6. Resultados de porcentaje de peticiones atendidas de servidores .	18
Tabla 7. Valores de calificación de acuerdo al Proceso Analítico Jerárquico (AHP).....	36
Tabla 8. Calificación a Servidores Web de acuerdo con cada característica.....	36

1. INTRODUCCIÓN

Existe un crecimiento explosivo en tamaño y uso de la World Wide Web, asumiendo como resultado un aumento de la carga en sus redes y servidores, y destaca los protocolos en los que se basa la Web. La mejora del rendimiento de la Web ha sido objeto de muchas investigaciones recientes, que abordan diversos aspectos del problema, como un mejor almacenamiento en caché de la Web, mejoras del Protocolo de transferencia de hipertexto (HTTP), mejores servidores HTTP y servidores proxy, e implementaciones de sistema operativo para servidor web. [1] [2]

Los sistemas operativos de distribución libre, permiten a los usuarios operar sus computadoras sin restricciones de uso, desarrollo y mejoramiento. Existen sistemas operativos para servidores web, mediante un programa que sirve datos en forma de páginas web, la comunicación de estos datos entre cliente y servidor se hace por medio de un protocolo denominado HTTP [3]. Como ejemplo de dichos sistemas se encuentra Centos basado en la distribución Red Hat Enterprise Linux, con licencia GPL, se define como robusto, estable, fácil de instalar y utilizar. Las ventajas de sistemas operativos Linux son: reducción de costos, seguridad, estabilidad, comunidad de programadores y entornos gráficos. [4]

Los servidores web realizan conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier aplicación del lado del cliente, por lo que deben responder solicitudes concurrentes altas. Según [5] el servidor más utilizado hasta el momento es *Servidor HTTP Apache*, es de código abierto, multiplataforma, y licencia Apache 2.0, es uno de los más estables, y su gran ventaja que es de código abierto, por lo que se puede personalizar y ampliar por módulos sin afectar la base del servidor. Su antecesor es *Nginx* también de código abierto, usado como proxy inverso y puede gestionar miles de conexiones simultáneas. Y *OpenLiteSpeed*, es uno de los servidores más rápidos, trabaja en base a eventos, es decir, sirve a todas las conexiones solamente con algunos procesos, ahorrando recursos. [6] [7]

El rendimiento web es uno de los más importantes factores para los desarrolladores, ya que lo primero que notas cuando analizas el rendimiento, es si la carga de la página es lenta. Existen herramientas para medir el rendimiento de servidores web, como, por ejemplo: *Apache Benchmark* es una herramienta de pruebas de carga y evaluación comparativa para el servidor del protocolo HTTP, se instala automáticamente con el servidor web Apache, o se puede instalar por separado. [8] [9] [10]

En el presente trabajo se analizará el rendimiento de servidores web como: Servidor Apache HTTP, Nginx y OpenLiteSpeed, instalados en el sistema operativo Centos 7, y con la ayuda de una página web informativa acerca de los lugares turísticos de El Oro, se realizará una prueba de rendimiento con la herramienta Apache Benchmark, la cual se instala junto con el Servidor Apache, donde se simulará un gran número de peticiones a los servidores web y, al finalizar, reportará un informe con el resultado del test.

La presente documentación se divide en tres secciones, las cuales se especificarán a continuación: Capítulo 1: conlleva la información importante del documento, empezando con la introducción acerca del tema abordado, seguido de un contexto el problema a solucionar y el objetivo general del estudio. Capítulo 2: se presenta la investigación clave para iniciar con el respectivo análisis del tema, de la mano con los procesos de solución y demostrando su resultado para llegar al objetivo. Capítulo 3: finalmente de acuerdo a los objetivos planteados, se da a conocer las conclusiones que se obtuvieron y comunicar el respectivo análisis comparativo.

1.1. Marco contextual

Los servidores web se vuelven cada vez más importantes para las empresas, cuyo negocio principal se realiza en internet. El rendimiento de un servidor web es relevante, ya que, satisface peticiones de otras máquinas de forma remota que trabaja usando el modelo cliente-servidor. Por lo que el rendimiento de dicho servidor debe superar las expectativas del usuario, y sobre todo evaluar continuamente el servidor web a utilizar para posteriormente utilizar esa información para la correcta toma de decisiones.

1.2. Problema

En la actualidad existen muchas empresas que utilizan servidores web, en los últimos años han tenido un crecimiento a nivel mundial, a ritmos del 23% cada año. Este servicio almacena archivos de un sitio y los emite por internet para que las páginas que aloja puedan ser visitadas por los usuarios, por ende, si existe una concurrencia alta de usuarios, dicho servidor debe ser capaz de soportarla.

1.3. Objetivo general

Realizar un análisis comparativo a servidores web instalados en un sistema operativo de distribución libre, mediante la herramienta Apache Benchmark para estar al tanto de cuál es el mejor al momento de usarlo, basado en el tiempo de respuesta y capacidad de concurrencia.

2. DESARROLLO

2.1. Marco teórico

2.1.1. *Sistemas Operativos de Distribución Libre*

Con este tipo de sistemas el usuario tiene completa libertad para ejecutar programas, estudiar su funcionamiento para poder adaptarlo a sus necesidades individuales. Para manipularlo se requiere conocer su código fuente, se puede saber cómo está estructurado, y también copiarlo y distribuirlo libremente. Existen varios sistemas operativos de distribución libre, los más conocidos son: Linux, Ubuntu, Red hat, XUbuntu, etc. [11]

2.1.2. *Linux*

GNU/LINUX, es un sistema operativo como MacOS o Windows, es decir, Linux es el software necesario para que tu ordenador te permita utilizar programas como: editores de texto, navegadores de internet, etc. Puede usarse mediante un interfaz gráfico, pero también mediante línea de comandos. Tiene su origen en Unix, apareció en los años sesenta desarrollado por Dennis Ritchie y Ken Thompson. Entre sus distribuciones más populares están: Debian, Centos, Fedora, Arch Linux, etc. [12]

2.1.3. *CentOS 7*

Es el sistema operativo más popular, superior, ligero, rápido y fiable para un servidor web. Es una plataforma corporativa destinada, esencialmente, a su implementación en empresas y organizaciones de gran tamaño. Aunque la página w3techs.com indica que Ubuntu y Debian se utilizan con mayor frecuencia. CentOS cuenta con las siguientes características: estabilidad, seguridad, ciclos largos de mantenimiento y soporte. [13]

Sin entorno de escritorio	Con entorno de escritorio
Memoria RAM: 64MB (mínimo)	Memoria RAM: 1 GB (mínimo)
Espacio en Disco Duro: 1 GB (mínimo) – 2 GB (recomendado)	Espacio en Disco Duro: 20 GB (mínimo) – 40 GB (recomendado)
Procesador: Intel x86-64 (64 bit)	Procesador: Intel x86-64 (64 bit)

Tabla 1. Requisitos de sistema CentOS 7

Fuente: Elaboración propia

2.1.4. Servidores web

Es un programa informático que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del Modelo OSI. [14]

2.1.5. Protocolo HTTP

El Protocolo de transferencia de hipertexto (Hypertext Transfer Protocol), es el protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. Está soportado sobre los servicios comunes de los entornos UNIX. La función del servidor es escuchar en un puerto de comunicaciones TCP, por defecto el puerto 80, y esperar las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libres de errores. [15]

2.1.6. Apache Http Server

Es un software de servidor web gratuito, de código abierto y multiplataforma con el cual se ejecutan el 46% de los sitios web de todo el mundo. el nombre oficial es Apache HTTP Server, es mantenido y desarrollado por Apache Software Foundation. Les permite a los propietarios de sitios web servir contenido en la web, es uno de los servidores más antiguos y confiables, con la primera versión lanzada hace más de 20 años, en 1995. [16]

2.1.7. Nginx

Es un servidor web de software libre y de código abierto, bajo la licencia BSD Simplificada. Desde su éxito inicial como servidor web, ahora también es usado como proxy inverso, cache de HTTP, y balanceador de carga, es ligero y de alto rendimiento. Pronunciado como “engine-ex”, y es utilizado por compañías de alto perfil como: GitLab, Microsoft, IBM, Google, Adobe, Cisco, Facebook, Twitter, Apple, Intel, entre otros. [17]

2.1.8. LiteSpeed Web Server

LiteSpeed Web Server (LSWS), es un servidor web con un rendimiento muy superior a sus competidores, sobre todo cuando lo comparamos con Apache, ya que con Nginx la diferencia de rendimiento es menor. Hay que destacar que el desarrollo de LiteSpeed es de los más activos, ya que ha sido el primer servidor web en soportar HTTP/3 en rama estable. La ventaja de LiteSpeed es que lee e interpreta los archivos .htaccess, por lo que no es necesario adaptarnos a un cambio de sintaxis para hacer ciertos rewrites como en el caso de Nginx y sus archivos de configuración. [18]

2.1.9. Rendimiento Servidor Web (Benchmark)

Es una técnica usada para medir el rendimiento de un sistema o componente del mismo. Más formalmente puede entenderse que un benchmark es el resultado de la ejecución de un programa informático o un conjunto de programas en una máquina, con el objetivo de estimar el rendimiento de un elemento concreto, y poder comparar los resultados. [19]

2.1.10. Apache Benchmark

Es un programa que permite medir el rendimiento de un servidor web, ofrece valores indicativos de la cantidad de peticiones por segundo que puede procesar el servidor, su principal ventaja es su extrema sencillez, mediante el comando ab con un par de parámetros se obtiene información muy útil acerca del servidor a analizar. Se instala automáticamente con el servidor web Apache, o también se la puede instalar por separado, es parametrizable y gratuita. [20] [21]

Parámetros de Apache Benchmark
A (auth-username:password): suministra las credenciales de autenticación basic al servidor.
c (concurrency): cantidad de conexiones concurrentes.
e (csv-file): escribe un archivo de valores separados por comas.
g (gnuplot-file): escribe todos los valores medidos en un archivo gnuplot.
H (custom-header): anexa encabezados adicionales a la solicitud.
k (keepalive): realiza múltiples solicitudes dentro de una sesión HTTP.
n (requests): número total de solicitudes para ejecutar.

Tabla 2. Parámetros de Apache Benchmark

Fuente: Elaboración propia

2.2. Solución del problema

2.2.1. Selección de la herramienta

La herramienta seleccionada es Apache Benchmark, ya que permite comparar cada servidor de protocolo de transferencia de hipertexto (HTTP), mediante pruebas de carga y estrés. Los servidores web seleccionados para las respectivas pruebas son Servidor HTTP Apache, Servidor Nginx y Servidor OpenLiteSpeed. La diferencia de cada tipo de prueba mencionada anteriormente se mostrará en la tabla 3.

Tipos de pruebas a servidores web	
Pruebas de carga (load testing)	Evalúan cuánto tiempo se tarda el servidor para atender peticiones.
Pruebas de estrés (stress testing)	Evalúan las respuestas del sistema a periodos de mayor volumen de actividad.

Tabla 3. Tipos de pruebas a servidores web

Fuente: Elaboración propia

2.2.2. Parámetros de calificación

Adicional a los tipos de prueba a cada servidor, se tomaron en cuenta 10 características para la evaluación. Las características entre factores y funcionalidades están relacionadas para realizar pruebas de carga, las cuales se les asigna un peso subjetivo (Anexo 4), pero tomando en consideración la recomendación del Proceso Jerárquico Analítico (AHP) [22], es una técnica estructurada para tratar con decisiones complejas.

2.2.3. Características a medir en el análisis comparativo

A continuación, se describen las 10 características a evaluar. El número entre llaves es la intensidad de importancia que se asigna en cada caso.

1. **Soporte técnico personalizado {2}**: brindan soporte técnico especializado y personalizado para resolver problemas.
2. **Hardening de Cabeceras HTTP {3}**: protege ante ataques y filtra información vital.
3. **Soporte WAF {6}**: Firewall para aplicaciones (WAF), ofreciendo protecciones contra ataques XSS, inyección SQL, etc.

4. **Idiomas soportados {6}**: siendo multilingües, les da la capacidad de ser fácilmente adaptables sin importar la cultura en la cual se utilicen.
5. **Archivo .htaccess {7}**: soporte para interpretar archivos .htaccess junto con sus instrucciones.
6. **Multiplataforma {8}**: característica que describe la portabilidad del servidor.
7. **Interfaz Web {5}**: configuración en una interfaz amigable, permite a los usuarios configurar de forma fácil, eficaz y cómoda.
8. **Visualización en tiempo real {9}**: permite visualizar los resultados de las pruebas y las gráficas de datos mientras la prueba se ejecuta.
9. **Proxy HTTP {9}**: examina el tráfico web para identificar contenido sospechoso, que puede ser virus o contenido con formato incorrecto.
10. **AntiDDOS {9}**: es el ataque de denegación de servicio distribuido, llevando a cabo un gran flujo de información desde varios puntos de conexión hacia un mismo punto de destino.

2.2.4. Diseño de las pruebas

En esta etapa se diseñaron 6 pruebas que miden el número máximo de peticiones y el número máximo de conexiones simultáneas que soporta el servidor web.

Las pruebas se diseñaron usando un documento HTML estático, acerca de los lugares turísticos de la provincia de El Oro. Dichas pruebas sirvieron para probar el comportamiento del servidor con distintos niveles de concurrencia y de peticiones, experimentando su respuesta a distintos niveles de carga. La tabla 4 resume cantidad de pruebas realizadas.

N° Prueba	N° Peticiones	Concurrencia
1	500	100
2	1000	100
3	1000	150
4	10000	100
5	10000	150
6	50000	200

Tabla 4. Números de peticiones y cantidad de concurrencia para cada prueba

Fuente: Elaboración propia

2.3. Resultados

2.3.1. Evaluación de características propuestas a cada servidor

En anexo 5 se realiza la evaluación de las 10 características indicadas anteriormente, obteniendo la suma del puntaje total de cada servidor y así estar al tanto que servidor posee la mayor puntuación.

La calificación obtenida por cada servidor web, está referenciada con licencia open source, ya que la respectiva instalación se la realiza en el sistema operativo Centos 7 de distribución libre.

Dando como resultado que, el servidor OpenLiteSpeed obtuvo la mayor puntuación de 56/64, ya que posee soporte para el archivo .htaccess, también con la ayuda de su interfaz web brinda mejora en la configuración, y posee visualización en tiempo real. Seguido por el servidor Nginx con una calificación de 37/64, ya que no tiene soporte para el archivo .htaccess y finalmente el servidor Apache obtuvo una calificación de 33/64, llegando a obtener la calificación más baja.

2.3.2. Evaluación de carga y estrés mediante pruebas a cada servidor

Una vez finalizadas las pruebas a cada servidor web, se analizaron los tiempos de respuesta de acuerdo con el número de solicitudes de usuarios y la cantidad de conexiones concurrentes, con la ayuda de la herramienta Apache Benchmark. La prueba 6 se usa como base para establecer el comportamiento de los servidores respecto al tiempo requerido para completar dicha prueba, en la tabla 5 se define el tiempo necesario para las pruebas (seg), solicitudes por segundo (seg), tiempo por solicitud (ms) y tiempo de conexión (ms).

Resultados	N° Prueba	Servidores Web		
		Apache HTTP	NGINX	OpenLiteSpeed
Tiempo necesario para las pruebas	6	11.742 seg	6.435 seg	6.584 seg
Solicitudes por segundo (promedio por petición)	6	4258.31 seg	7770.44 seg	7593.93 seg
Tiempo por solicitud (promedio grupo de peticiones)	6	46.967 ms	25.739 ms	26.337 sec
Tiempo por solicitud (promedio una petición)	6	0.235 ms	0.129 ms	0.132 ms
Tiempo de conexión (total) (min – max)	6	7 – 3742 ms	12 – 1364 ms	13 – 243 ms

Tabla 5. Resultados de acuerdo a la prueba 6 de cada servidor

Fuente: Elaboración propia

En la tabla 6 se observa el rendimiento de los servidores web, de acuerdo a los valores de porcentaje de peticiones atendidas, dichos valores son el tiempo en milisegundos que se necesita para atender un cierto porcentaje de peticiones.

Porcentaje de peticiones atendidas	Servidores Web		
	Apache HTTP	NGINX	OpenLiteSpeed
50%	39 ms	17 ms	21 ms
66%	46 ms	27 ms	29 ms
75%	49 ms	28 ms	29 ms
80%	51 ms	29 ms	30 ms
90%	59 ms	42 ms	44 ms
95%	70 ms	54 ms	52 ms
98%	82 ms	65 ms	61 ms
99%	111 ms	79 ms	65 ms
100%	3742 ms	1364 ms	243 ms

Tabla 6. Prueba 6. Resultados de porcentaje de peticiones atendidas de servidores

Fuente: Elaboración propia

En la ilustración 1 se contrasta la información de acuerdo a los resultados de la tabla 6, dicha información recalca el tiempo de atención por porcentaje de peticiones atendidas de cada servidor, mediante la página web de prueba acerca de los lugares turísticos de la provincia de El Oro. El servidor más rápido en atender las peticiones es OpenLiteSpeed, seguido del servidor NGINX y por último el servidor Apache HTTP. Las pruebas 1 al 5 se presentan en el anexo 6, con la ayuda de gráficos se visualizan los respectivos resultados agrupados. En anexos 7 se presentan capturas del código de la prueba 6 del servidor más rápido, utilizando la herramienta Apache Benchmark.

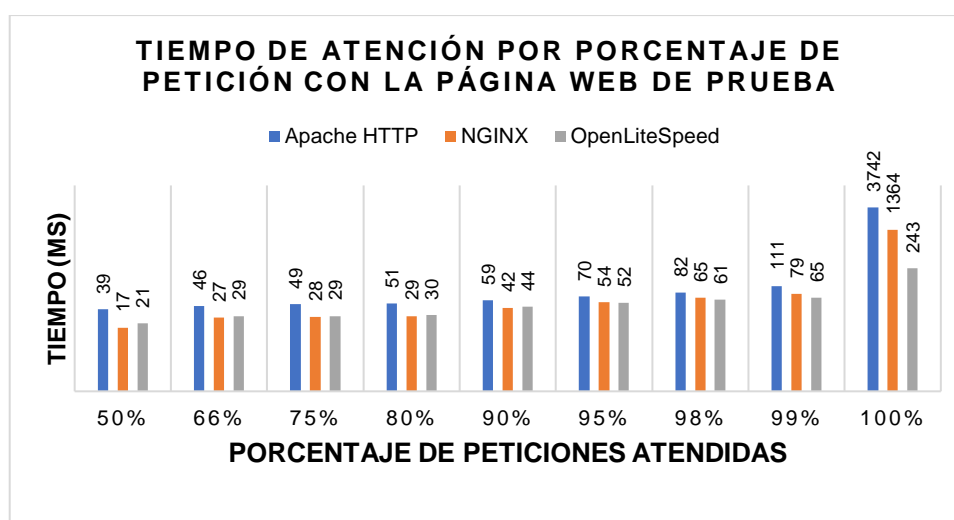


Ilustración 1. Resultado tiempo de atención por porcentaje de petición con la página web de prueba

Fuente: Elaboración propia

3. CONCLUSIONES

- Se logró la instalación y configuración de tres servidores web en el sistema operativo Centos 7, el cual es de distribución libre. El servidor Apache HTTP fue fácil de instalar ya que dispone de mucha documentación para la respectiva guía. Nginx posee una configuración basada en Apache HTTP, por lo que se presentaron inconvenientes. En el caso de OpenLiteSpeed se realizaron varios pasos para la instalación, y no existe mucha documentación, por ende, fue un poco más tedioso.
- Se evaluó a los servidores web con la ayuda del Proceso Analítico Jerárquico (AHP), seleccionando 10 características, acordes al rendimiento, factores y funcionalidades relacionadas con las pruebas de carga de los servidores, asignando un peso subjetivo a cada característica. La suma total de las características fue de 64 puntos, obteniendo como resultado OpenLiteSpeed con 56 puntos, Nginx con 37 puntos y Apache HTTP con 33 puntos.
- Con la herramienta Apache Benchmark se llegó a la conclusión que el servidor OpenLiteSpeed es uno de los más rápidos en responder a peticiones de usuarios y cantidad de usuarios concurrentes. Obteniendo un tiempo de conexión en milisegundos (ms) como mínimo de 13 ms y como máximo de 243 ms, de acuerdo a la prueba número 6, la cual se caracteriza por proponer la mayor cantidad de usuarios y conexiones concurrentes.
- De acuerdo a los resultados obtenidos mediante el benchmark realizado a cada uno de los servidores web se llegó a la conclusión que OpenLiteSpeed es uno de los mejores al momento de alojar una página web, ya que su entorno se basa en procesos no en eventos, ahorrando recursos, logrando tener un impacto en el rendimiento del equipo, así como en la velocidad de despacho de la web. Posee su propio modo de ataques DDOS, haciéndolo mucho más seguro a la hora de mitigar un ataque masivo, también cuenta con soporte para interpretar archivos .htaccess ya que es una de las opciones más buscadas por los programadores. Y posee una interfaz web amigable donde se puede configurar todo con mayor comodidad y eficacia.

BIBLIOGRAFÍA

- [1] A. Albacete Carreño, M. P. Antolínez Merchán y M. C. Cañamero Alvarado, «La crónica de sucesos. Evolución del género en el world wide web,» *Ámbitos. Revista Internacional de Comunicación*, nº 26, 2014.
- [2] M. Kumar y Meenu, «Analysis of visitor's behavior from web log using web log expert tool,» *International conference of Electronics, Communication and Aerospace Technology (ICECA)*, vol. 2, pp. 296-301, 2017.
- [3] «An Experimental View on Fairness between HTTP/1.1 and HTTP/2,» *International Conference on Information Networking (ICOIN)*, pp. 399-401, 2019.
- [4] Z. Zhang, S. Qin, X. Wang y D. Zhan, «Implementation of embedded Linux based on PC/104 platform,» *IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 1073-1077, 2015.
- [5] M. Vujović, M. Savić, D. Stefanović y I. Pap, «USAGE OF NGINX and websocket in IoT,» *Telecommunications Forum Telfor (TELFOR)*, pp. 289-292, 2015.
- [6] L. Zhang y Q. Zhu, «The Overtime Waiting Model for Web Server Performance Evaluation,» *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 229-232, 2014.
- [7] C. Gómez Montoya, C. Candela Uribe y L. Sepúlveda Rodríguez, «Seguridad en la configuración del Servidor Web Apache,» *Revista Inge CuC*, vol. 9, nº 2, pp. 31-38, 2013.
- [8] K. Hong, Y. Kim, H. Choi y J. Park, «SDN-Assisted Slow HTTP DDoS Attack Defense Method,» *IEEE Communications Letters*, vol. 22, nº 4, pp. 688-961, 2018.
- [9] Q. Fan y Q. Wang, «Performance Comparison of Web Servers with Different Architectures: A Case Study Using High Concurrency Workload,» *Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, pp. 37-42, 2015.
- [10] B. Brennan, «Business Benchmark,» *Modern English teacher*, vol. 23, nº 2, pp. 73-74, 2014.
- [11] J. Jorba Esteve y R. Suppi Boldrito, *Software Libre: Administración avanzada de GNU/Linux*, Barcelona: Eureka Media, SL, 2004.
- [12] D.-A. Leblanc, *Administración de sistemas Linux*, Madrid: ANAYA Multimedia, 2001.
- [13] D. Morril, *Configuración de sistemas Linux*, Madrid: ANAYA Multimedia, 2002.

- [14] D. Denaro, «Servidor Web: con shield wi-fi,» *Nueva Electrónica*, nº 325, pp. 35-41, 2015.
- [15] I. Stanivuk, V. Bjelić, T. Samardžić y Đ. Simić, «Expanding lua interface to support HTTP/HTTPS protocol,» *International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, pp. 407-410, 2017.
- [16] V. Piantadosi, S. Scalabrino y R. Oliveto, «Fixing of Security Vulnerabilities in Open Source Projects: A Case Study of Apache HTTP Server and Apache Tomcat,» *IEEE Conference on Software Testing, Validation and Verification (ICST)*, pp. 68-78, 2019.
- [17] M. Data, M. Luthfi y W. Yahya, «Optimizing single low-end LAMP server using NGINX reverse proxy caching,» *International Conference on Sustainable Information Engineering and Technology (SIET)*, pp. 21-23, 2017.
- [18] Quest Software Inc., User Guide: LiteSpeed, California: Quest Software Inc., 2017.
- [19] P. Castillo Valdiviezo, J. Bernier Villamor, A. Mora García, P. García Sánchez y M. I. García Arenas, «Servidores web de altas prestaciones,» *Enseñanza y aprendizaje de ingeniería de computadores: Revista de Experiencias Docentes en Ingeniería de Computadores*, nº 3, pp. 133-142, 2013.
- [20] R. Sen, A. Farris y P. Guerra, «Benchmarking Apache Accumulo BigData Distributed Table Store Using Its Continuous Test Suite,» *IEEE International Congress on Big Data*, pp. 334-341, 2013.
- [21] J. A. Hernández, M. E. Pérez y M. E. García, «Metodología para realizar pruebas de carga a un servidor Web sobre un sistema empotrado. Un caso de estudio,» *Pistas Educativas*, nº 108, pp. 1450-1465, 2014.
- [22] C. Zapata y C. Cardona, «Comparación de las características de algunas herramientas de software para pruebas de carga,» *Revista Avances en Sistemas e Informática*, vol. 8, nº 2, pp. 143-154, 2011.

ANEXOS

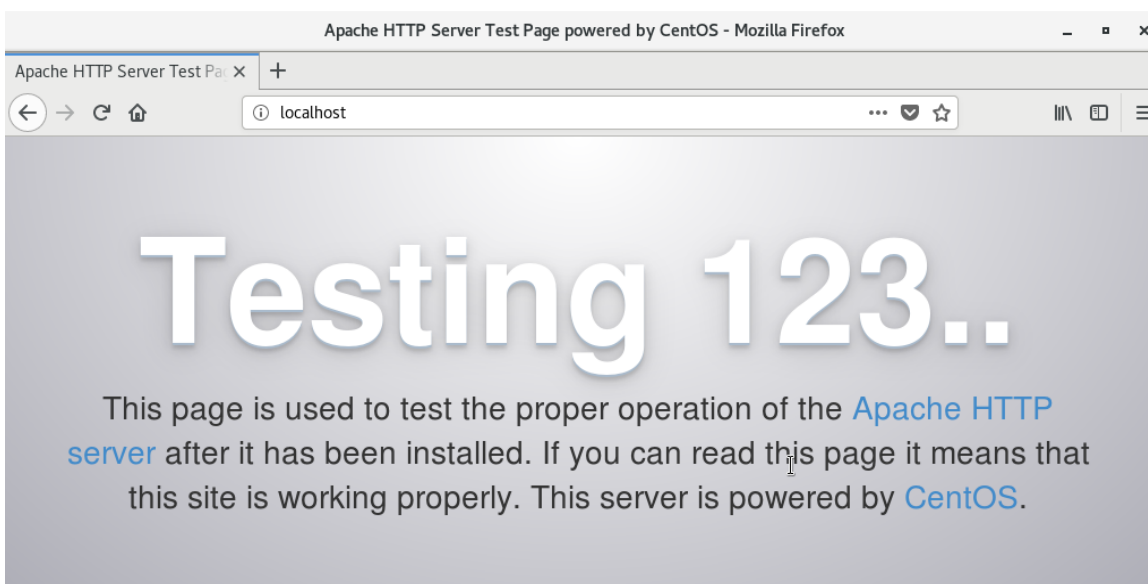
ANEXO 1. INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR WEB HTTP APACHE

```
[root@localhost gabyramirez]# yum install httpd
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.uce.edu.ec
 * epel: mirror.cedia.org.ec
 * extras: mirror.uce.edu.ec
 * updates: mirror.uce.edu.ec
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete httpd.x86_64 0:2.4.6-89.el7.centos debe ser instalado
--> Procesando dependencias: httpd-tools = 2.4.6-89.el7.centos para el paquete:
httpd-2.4.6-89.el7.centos.x86_64
--> Procesando dependencias: libaprutil-1.so.0()(64bit) para el paquete: httpd-2
.4.6-89.el7.centos.x86_64
--> Procesando dependencias: libapr-1.so.0()(64bit) para el paquete: httpd-2.4.6
-89.el7.centos.x86_64
```

Ilustración 2. Instalación del servidor HTTP Apache mediante comando yum

```
[root@localhost gabyramirez]# service httpd start
Redirecting to /bin/systemctl start httpd.service
```

Ilustración 3. Inicio del servidor HTTP apache con el comando service



Just visiting?

The website you just visited is either experiencing problems or is undergoing routine maintenance.

Are you the Administrator?

You should add your website content to the directory `/var/www/html/`.

To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d`.

Ilustración 4. Ingreso a navegador para verificar servidor activo

```
[root@localhost sysconfig]# nano /etc/hosts
```



```
GNU nano 2.3.1 Fichero: /etc/hosts
192.168.0.1 www.elorotavel.com elorotavel.com
127.0.0.1 www.elorotavel.com localhost localhost.localdomain localhost4 localh
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
```

Ilustración 5. Configuración en el archivo hosts con el editor de texto nano

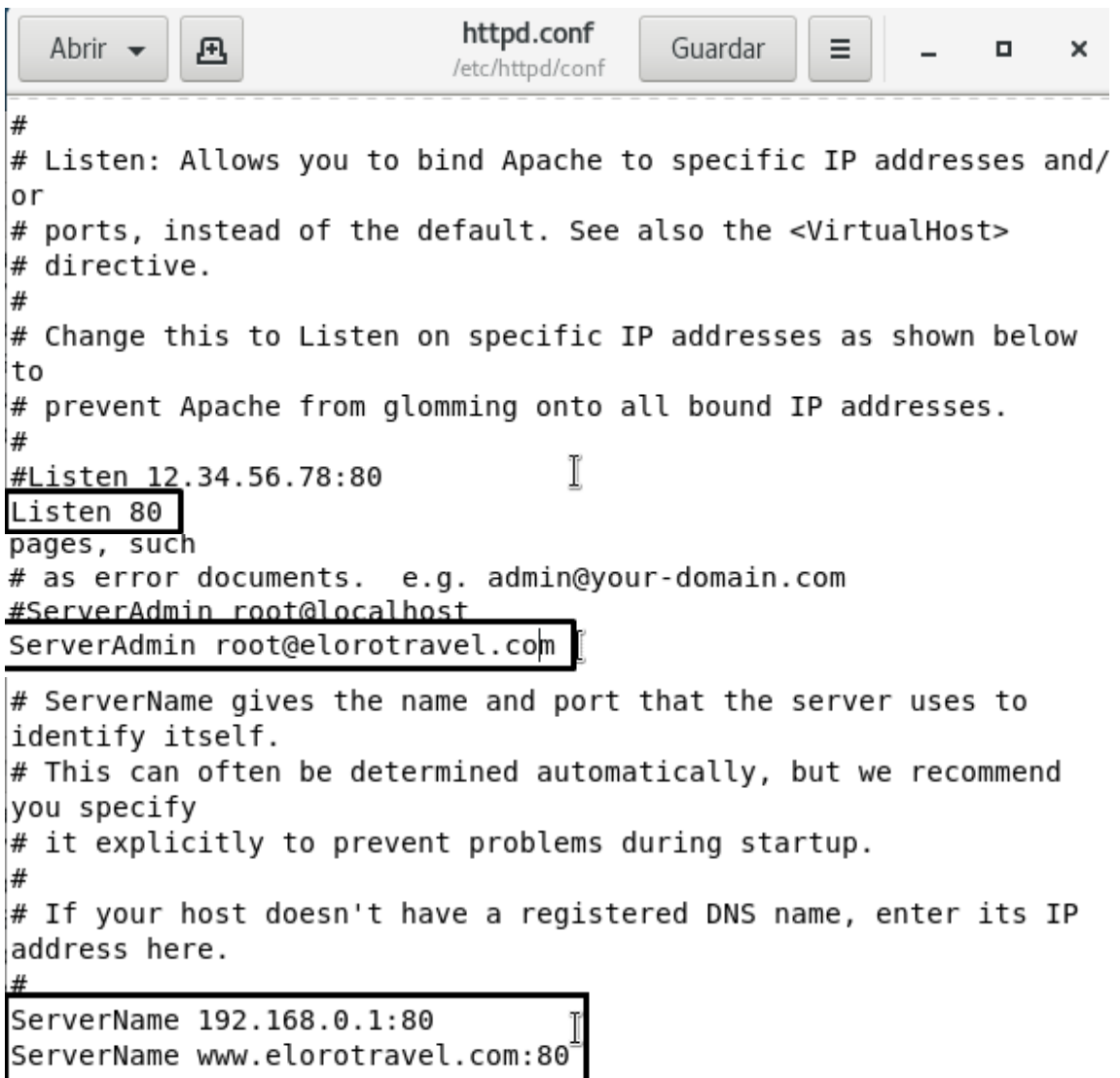
```
[root@localhost gabyramirez]# cd /etc/httpd/conf
```

```
[root@localhost conf]# ls
```

```
httpd.conf magic
```

```
[root@localhost conf]# gedit httpd.conf
```

Ilustración 6. Ruta del archivo para configurar el servidor Apache HTTP




```
Abrir ▾ httpd.conf Guardar ≡ - □ ×
/etc/httpd/conf
#
# Listen: Allows you to bind Apache to specific IP addresses and/
or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below
to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80
pages, such
# as error documents. e.g. admin@your-domain.com
#ServerAdmin root@localhost
ServerAdmin root@elorotavel.com
#
# ServerName gives the name and port that the server uses to
identify itself.
# This can often be determined automatically, but we recommend
you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP
address here.
#
ServerName 192.168.0.1:80
ServerName www.elorotavel.com:80
```

Ilustración 7. Configuración del archivo httpd.conf con el editor de texto gedit

```
[root@localhost httpd]# ls
conf  conf.d  conf.modules.d  logs  modules  run
[root@localhost httpd]# cd conf.d
[root@localhost conf.d]# ls
autoindex.conf  php.conf  README  userdir.conf  welcome.conf
[root@localhost conf.d]# touch titulacion.conf
```

Ilustración 8. Ruta de configuración de hosts y creación de archivo titulacion.conf

```
[[root@localhost conf.d]# gedit titulacion.conf
```



```
<VirtualHost *:80>
    ServerAdmin root@elorotravel.com
    DocumentRoot /var/www/html/elorotravel
    ServerName www.elorotravel.com
    ServerAlias elorotravel.com
</VirtualHost>
```

Ilustración 9. Configuración del archivo titulacion.conf con el editor de texto gedit

```
[root@localhost html]# cd /var/www/html
[root@localhost html]# ls
elorotravel  index.html
[root@localhost html]# cd elorotravel/
[root@localhost elorotravel]# ls
correo.php  gracias.html  index.html  package.json  scss
css         gulpfile.js  js         package-lock.json  vendor
debug.log  img         LICENSE   README.md
```

Ilustración 10. Ruta de archivo de servidor HTTP para almacenar páginas web

```
[root@localhost conf.d]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```



Ilustración 11. Reinicio del servidor HTTP Apache y verificación de página web en host cliente (Windows)

ANEXO 2. INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR WEB NGINX

```
[root@localhost ~]# yum install nginx
```

```
Dependencias resueltas
```

```
=====
```

Package	Arquitectura	Versión	Repositorio	Tamaño
=====				
Instalando:				
nginx	x86_64	1:1.12.2-3.el7	epel	531 k
Instalando para las dependencias:				
nginx-all-modules	noarch	1:1.12.2-3.el7	epel	16 k
nginx-filesystem	noarch	1:1.12.2-3.el7	epel	17 k
nginx-mod-http-geoip	x86_64	1:1.12.2-3.el7	epel	23 k
nginx-mod-http-image-filter	x86_64	1:1.12.2-3.el7	epel	27 k
nginx-mod-http-perl	x86_64	1:1.12.2-3.el7	epel	36 k
nginx-mod-http-xslt-filter	x86_64	1:1.12.2-3.el7	epel	26 k
nginx-mod-mail	x86_64	1:1.12.2-3.el7	epel	54 k
nginx-mod-stream	x86_64	1:1.12.2-3.el7	epel	76 k

```
=====
```

Resumen de la transacción

```
=====
```

Instalar 1 Paquete (+8 Paquetes dependientes)

Ilustración 12. Instalación del servidor Nginx con el comando yum

```
[root@localhost conf.d]# service nginx start
Redirecting to /bin/systemctl start nginx.service
[root@localhost conf.d]# systemctl enable nginx
[root@localhost conf.d]# service nginx status
Redirecting to /bin/systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since jue 2019-07-04 10:20:51 -05; 10min ago
 Main PID: 6123 (nginx)
   CGroup: /system.slice/nginx.service
           └─6123 nginx: master process /usr/sbin/nginx
             └─6124 nginx: worker process
```

Ilustración 13. Iniciar, habilitar y verificar estado del servidor Nginx

```
[root@localhost conf.d]# ifconfig enp0s3 192.168.0.2 netmask 255.255.255.0 up
[root@localhost conf.d]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.2 netmask 255.255.255.0 broadcast 192.168.0.255
    ether 08:00:27:df:72:7a txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11 bytes 1749 (1.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Ilustración 14. Cambio de dirección IP a la máquina del servidor con el comando ifconfig

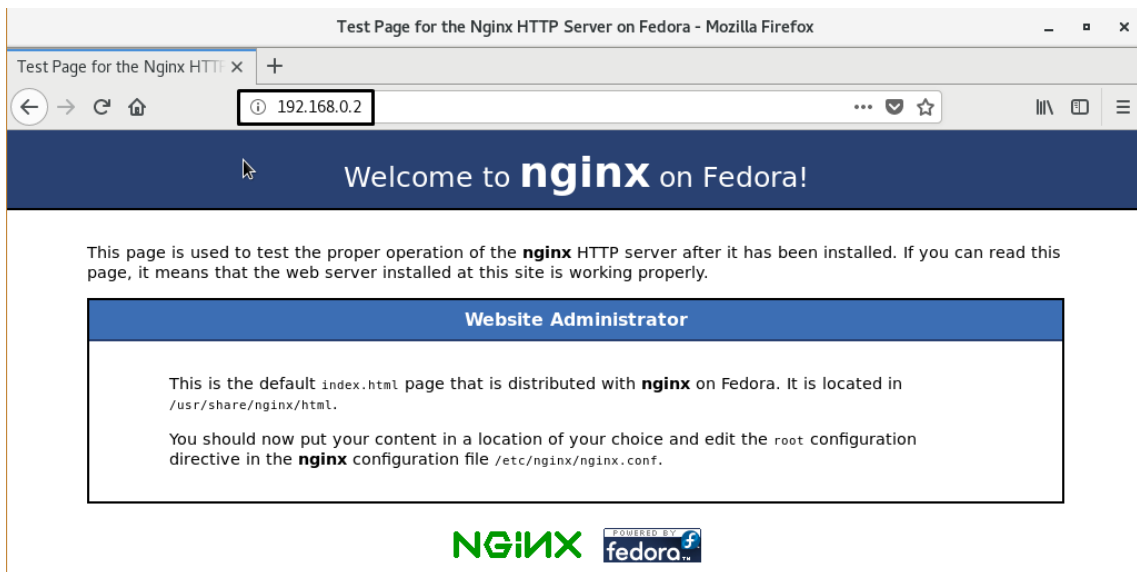


Ilustración 15. Verificación de funcionamiento del servidor con dirección IP de la PC

```
[root@localhost nginx]# cd /etc/nginx/conf.d
[root@localhost conf.d]# ls
[root@localhost conf.d]# touch default.conf
[root@localhost conf.d]# ls
default.conf
```

Ilustración 16. Ruta de archivo de configuración del servidor Nginx

```
[root@localhost conf.d]# gedit default.conf
```

```
#default server

server {
    # listen      80;
    listen       192.168.0.1:80 default_server;
    server_name  www.elorotravel.com;

    include /etc/nginx/default.d/*.conf;

    location / {
        root     /usr/share/nginx/html/elorotravel;
        index   index.html index.htm;
    }

    error_page 404          /404.html;
    location = /404.html {
        root /usr/share/nginx/html;
    }
}
```

Ilustración 17. Configuración del archivo default.conf con editor de texto gedit

```
[root@localhost elorotravel]# cd /usr/share/nginx/html/
[root@localhost html]# ls
404.html  elorotravel  nginx-logo.png  prueba.html
50x.html  index.html   poweredby.png
[root@localhost html]# cd elorotravel/
[root@localhost elorotravel]# ls
correo.php  gracias.html  index.html  package.json  scss
css         gulpfile.js  js          package-lock.json  vendor
debug.log  img          LICENSE    README.md
```

Ilustración 18. Ruta de archivo de servidor NGINX para almacenar páginas web

```
[root@localhost titulacion]# service nginx restart
Redirecting to /bin/systemctl restart nginx.service
```

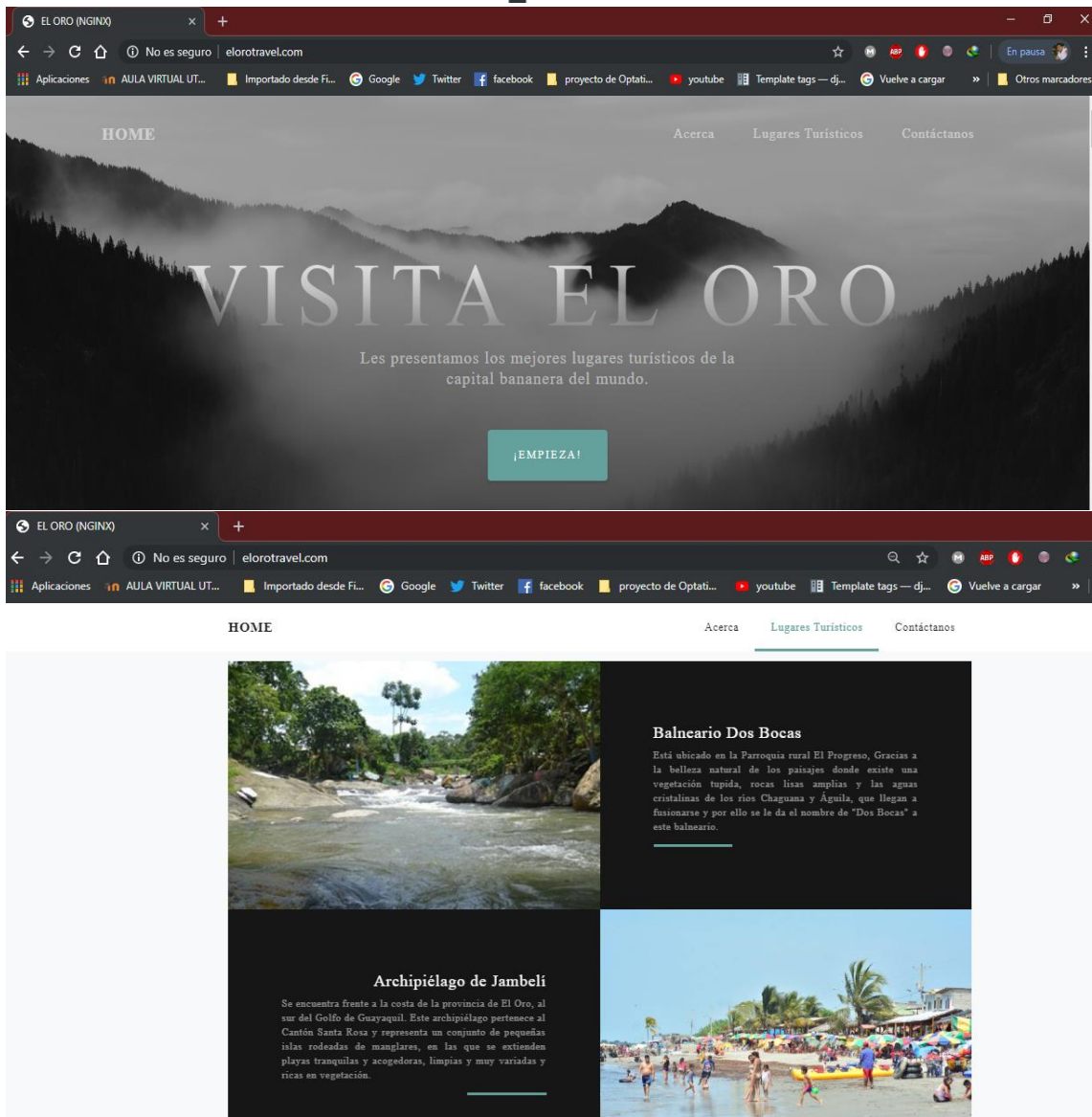


Ilustración 19. Reinicio del servidor NGINX y verificación de página web en host cliente (Windows)

ANEXO 3. INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR WEB OPENLITESPEED

```
[root@localhost ~]# rpm -ivh http://rpms.litespeedtech.com/centos/litespeed-repo-1.1-1.el7.noarch.rpm
Recuperando http://rpms.litespeedtech.com/centos/litespeed-repo-1.1-1.el7.noarch.rpm
Preparando... ##### [100%]
Actualizando / instalando...
  1:litespeed-repo-1.1-1.el7.centos ##### [100%]
[root@localhost ~]# █
```

Ilustración 20. Instalar repositorio con el comando rpm

```
[root@localhost ~]# yum -y install openlitespeed
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
epel/x86_64/metalink | 3.4 kB 00:00
* base: mirror.esepoch.edu.ec
* epel: mirror.uta.edu.ec
* extras: mirror.esepoch.edu.ec
* updates: mirror.esepoch.edu.ec
```

Ilustración 21. Instalación del servidor OpenLiteSpeed (última versión)

```
[root@localhost ~]# yum -y install mariadb mariadb-server
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirror.esepoch.edu.ec
* epel: mirror.uta.edu.ec
* extras: mirror.esepoch.edu.ec
* updates: mirror.esepoch.edu.ec
```

Ilustración 22. Instalación de mariadb server con el comando yum

```
[root@localhost ~]# yum install lsphp56-* --skip-broken
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirror.esepoch.edu.ec
* epel: mirror.uta.edu.ec
* extras: mirror.esepoch.edu.ec
* updates: mirror.esepoch.edu.ec
```

Ilustración 23. Instalación de php versión 5.6 y dependencias mediante yum

```
[root@localhost ~]# /usr/local/lsws/admin/misc/admpass.sh

Please specify the user name of administrator.
This is the user name required to login the administration Web interface.

User name [admin]: admin

Please specify the administrator's password.
This is the password required to login the administration Web interface.

Password:
Retype password:
Administrator's username/password is updated successfully!
```

Ilustración 24. Configuración de nombre de usuario y contraseña


```
[root@localhost ~]# ln -sf /usr/local/lsws/lsp56/bin/lsp56 /usr/local/lsws/fcgi-bin/lsp56
[root@localhost ~]# █
```

Ilustración 25. Creación de enlace simbólico para lsp56 y lsp56

```
[root@localhost ~]# systemctl start mariadb
[root@localhost ~]# systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service
to /usr/lib/systemd/system/mariadb.service.
```

Ilustración 26. Iniciar y habilitar mariadb mediante comando systemctl

```
[root@localhost ~]# mysql_secure_installation
```

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current password for the root user. If you've just installed MariaDB, and you haven't set the root password yet, the password will be blank, so you should just press enter here.

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

```
Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!
```

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

```
Reload privilege tables now? [Y/n] y
... Success!
```

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

Ilustración 27. Configuración de mariadb con el comando mysql_secure_installation

```
[root@localhost ~]# service lsws status
litespeed is running with PID 5395.
[root@localhost ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::9bfe:6f84:947a:ladd prefixlen 64 scopeid 0x20<link>
ether 08:00:27:df:72:7a txqueuelen 1000 (Ethernet)
RX packets 107630 bytes 97178649 (92.6 MiB)
```

Ilustración 28. Verificación del estado del servidor y visualización de IP de la máquina

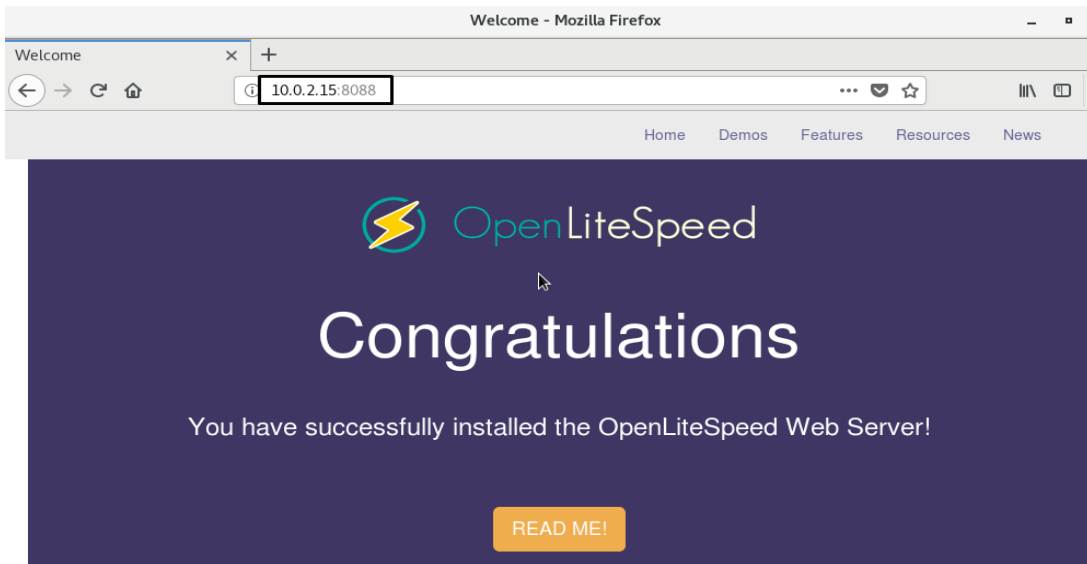


Ilustración 29. Verificación de funcionamiento de Servidor OpenLiteSpeed

```
[root@localhost lsws]# cd /usr/local/lsws
[root@localhost lsws]# ls
add-ons  backup  cgid  Example  GPL.txt  lsphp56  phpbuild  VERSION
admin    bin     conf  fcgi-bin  lib      modules  share
autoupdate  cachedata  docs  gdata    logs     php      tmp
[root@localhost lsws]# mkdir elorotravel
[root@localhost lsws]# ls
add-ons  backup  cgid  elorotravel  gdata  logs  php  tmp
admin    bin     conf  Example     GPL.txt  lsphp56  phpbuild  VERSION
autoupdate  cachedata  docs  fcgi-bin    lib      modules  share
```

Ilustración 30. Creación de Carpeta donde se almacenará página

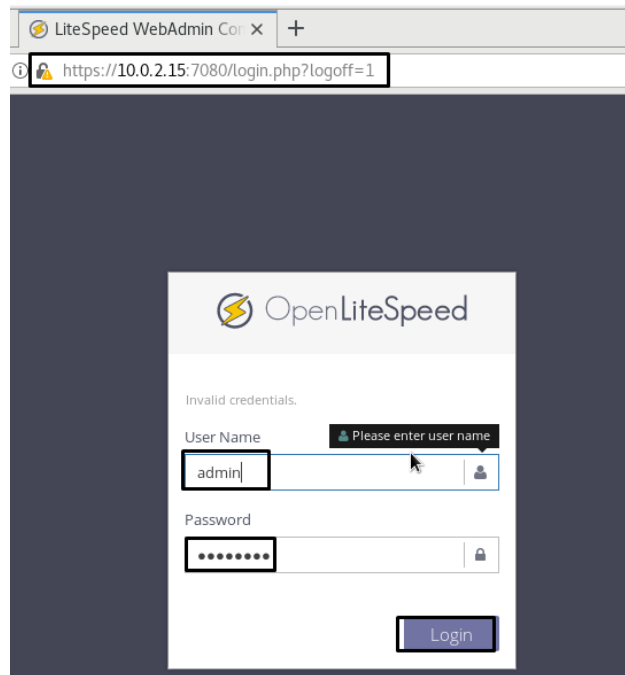


Ilustración 31. Inicio en interfaz web (WebAdmin) del servidor OpenLiteSpeed

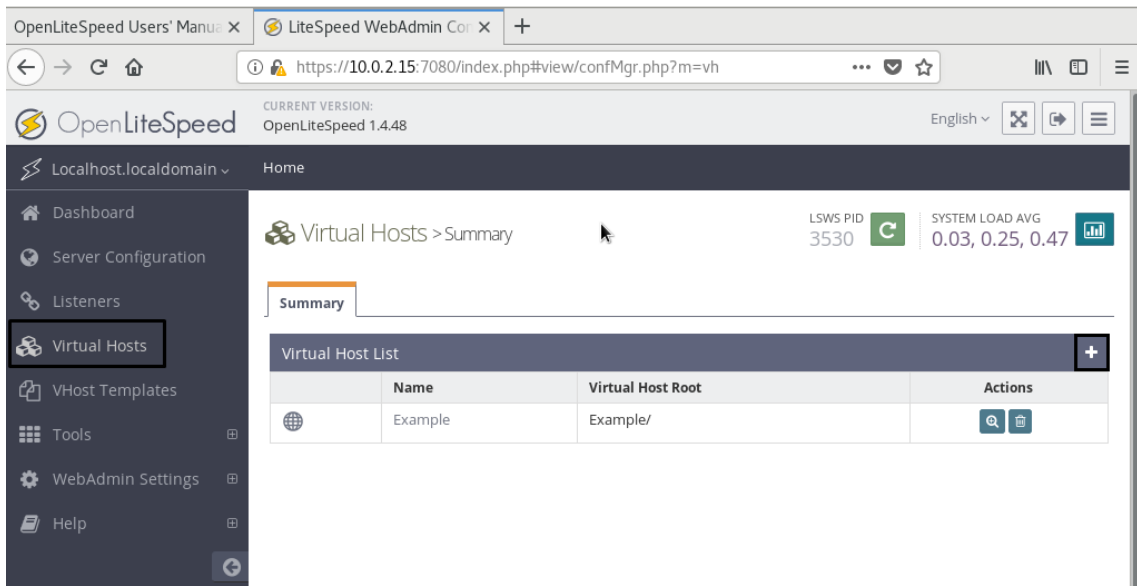


Ilustración 32. Creación de Virtual Hosts en OpenLiteSpeed

```
[root@localhost lsws]# mkdir elorotavel/{conf,html,logs}
[root@localhost lsws]# ls elorotavel/
conf html logs
[root@localhost lsws]# chown lsadm:lsadm elorotavel/conf
```

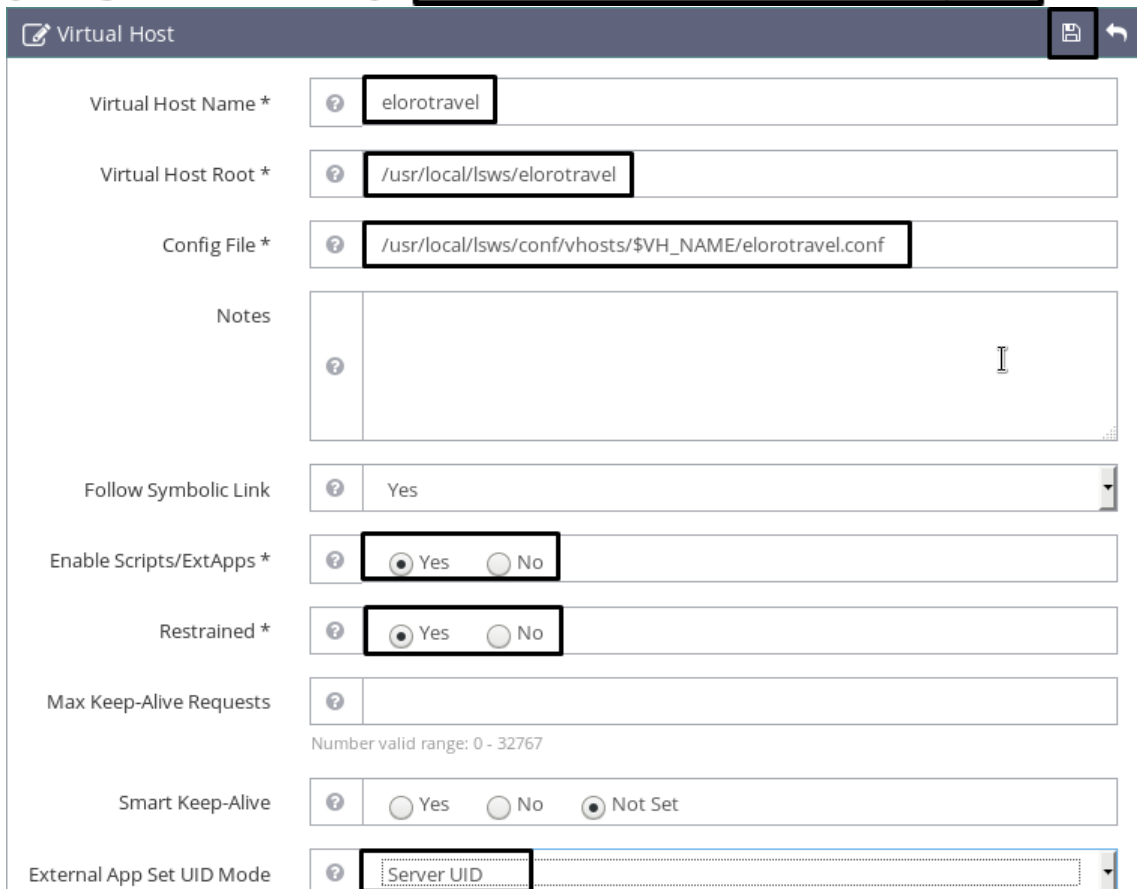


Ilustración 33. Creación y configuración de rutas para almacenamiento de página web

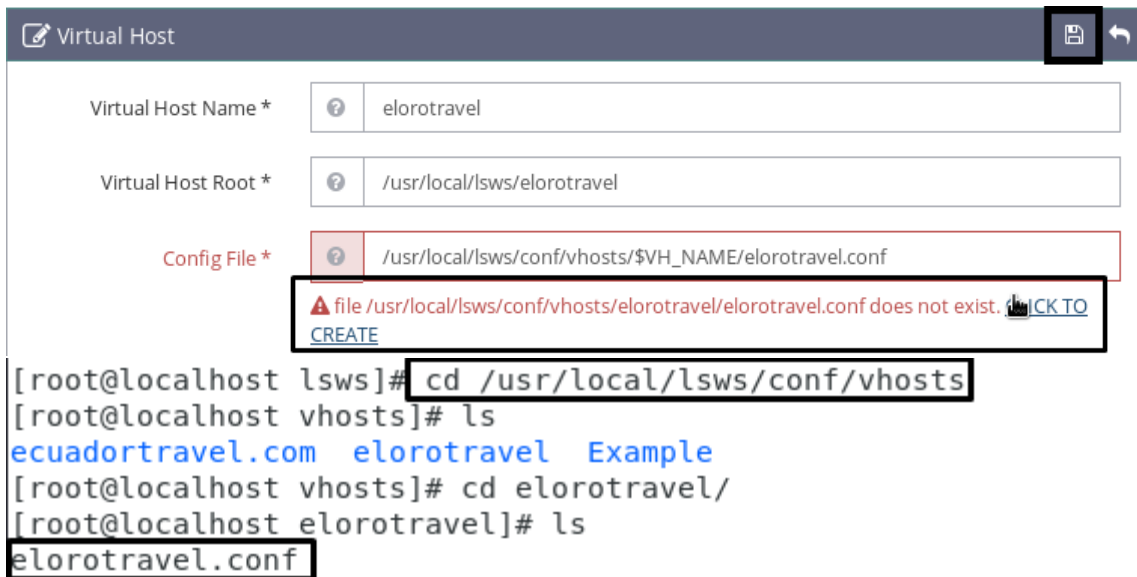


Ilustración 34. Creación y configuración de archivo Config File del Virtual Host

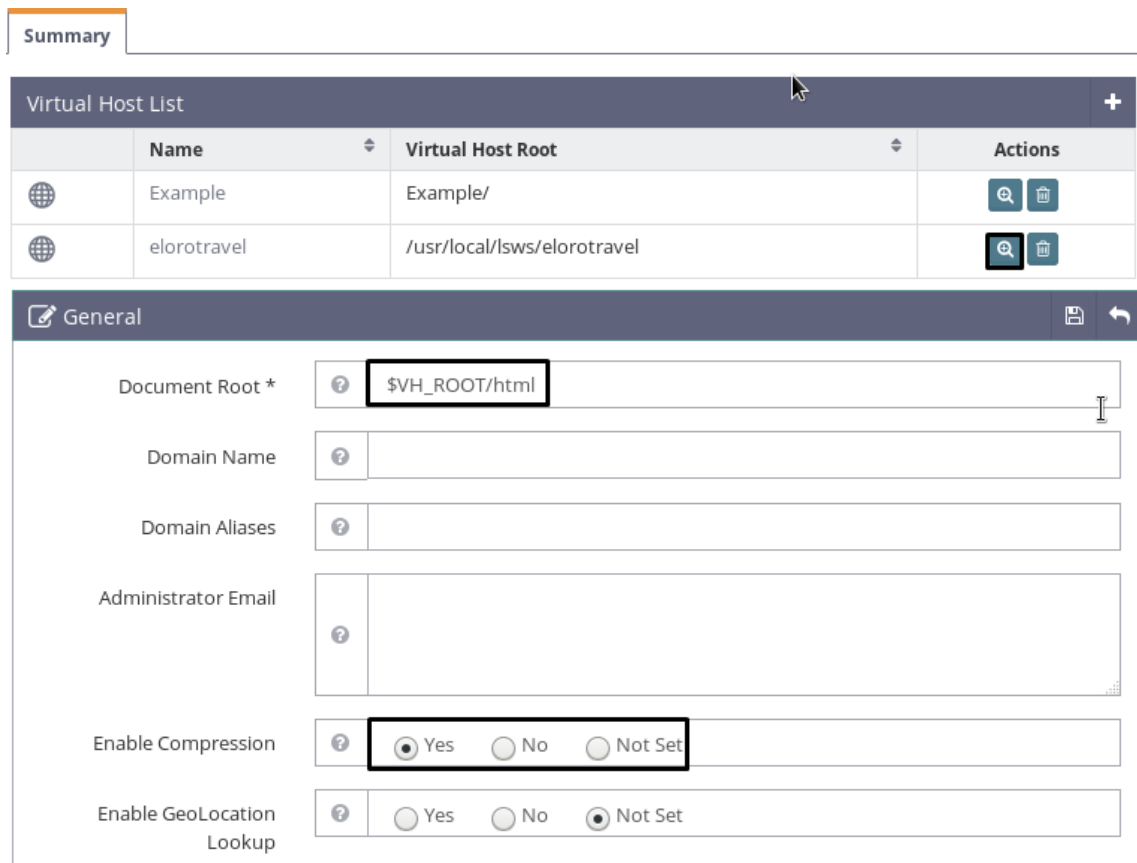


Ilustración 35. Configuración General del Virtual Host creado en OpenLiteSpeed

Index Files

Use Server Index Files *

Index Files

Auto Index Yes No Not Set

Auto Index URI

Ilustración 36. Configuración General de Index Files del Virtual Host

Virtual Host Log

Use Server's Log * Yes No

File Name

Log Level

Rolling Size (bytes)

Ilustración 37. Configuración de archivo log de error para el virtual host

Access Log

Log Control *

File Name

Piped Logger

Log Format

Log Headers Referrer UserAgent Host None

Rolling Size (bytes)

Keep Days
Number valid range >= 0

Bytes log

Compress Archive Yes No Not Set

Ilustración 38. Configuración de archivo Access log para el virtual host

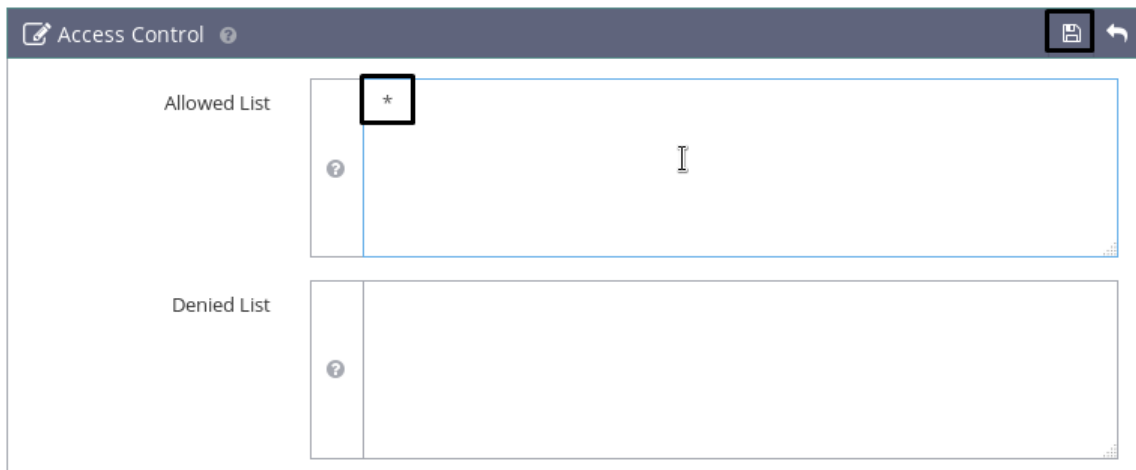


Ilustración 39. Configuración de seguridad para el virtual host

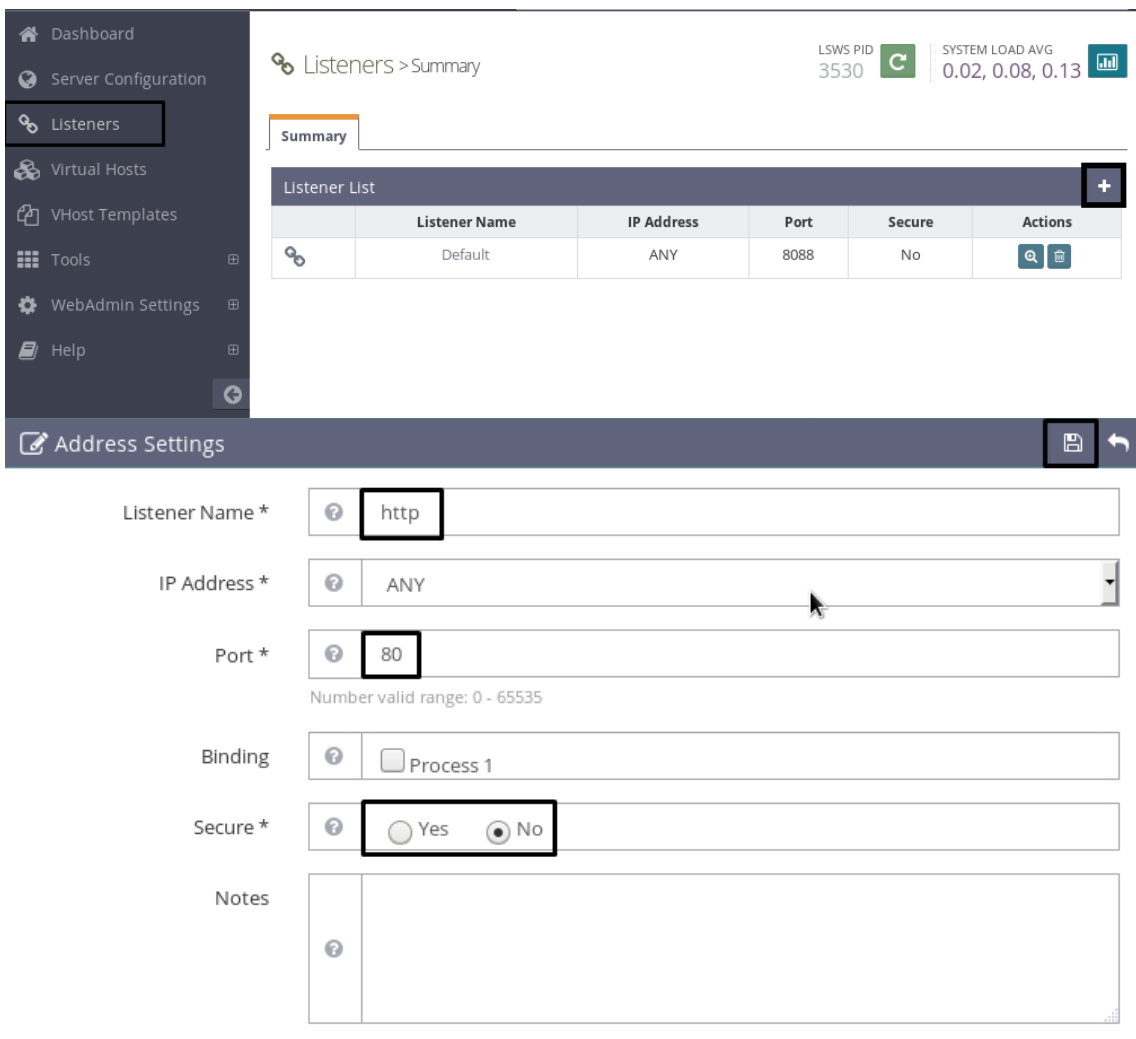


Ilustración 40. Creación de Listener para el puerto 80 (HTTP)



Summary

Listener List						+
	Listener Name	IP Address	Port	Secure	Actions	
	Default	ANY	8088	No		
	ssl	ANY	443	Yes		
	http	ANY	80	No		

Virtual Host Mappings

Virtual Host *

Domains *

Ilustración 41. Añadida del virtual host al listener puerto 80 (HTTP)

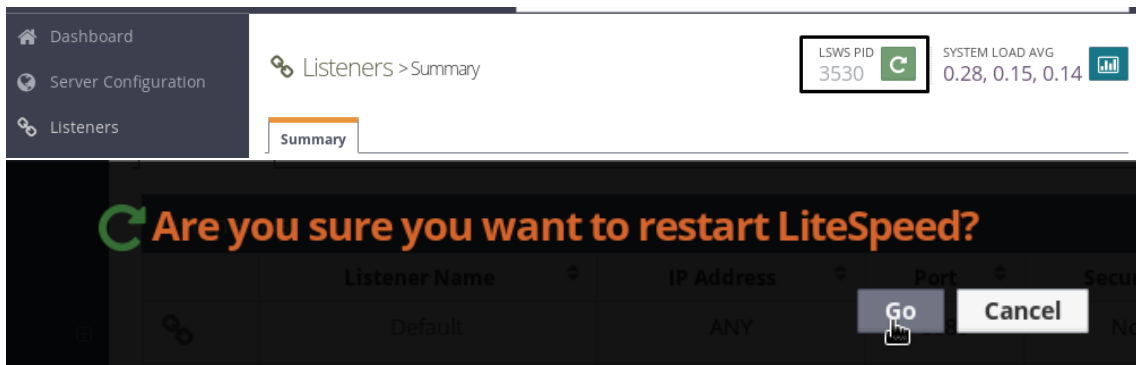


Ilustración 42. Reinicio del servidor para efectuar los cambios

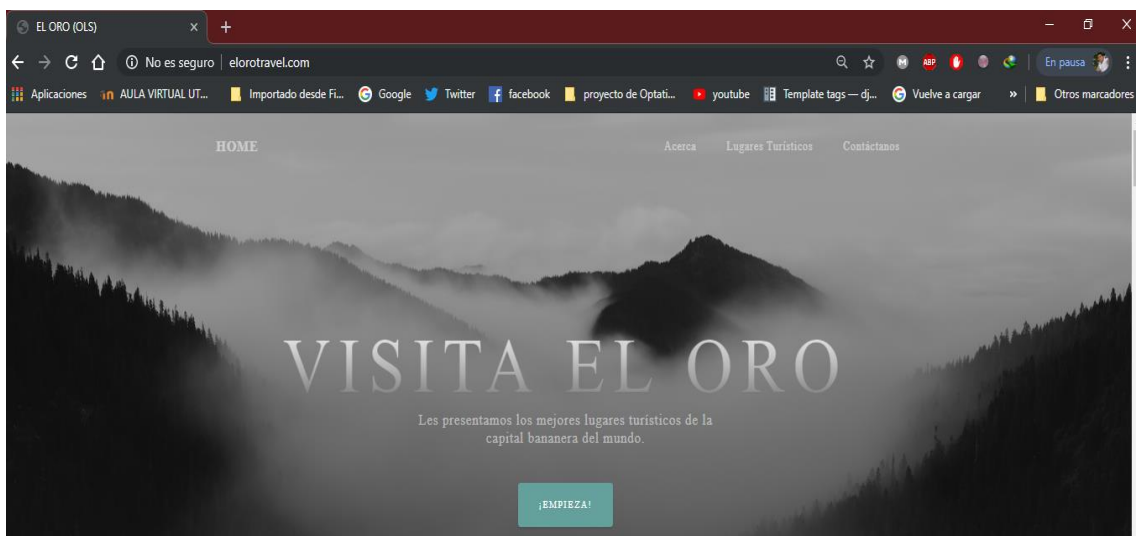


Ilustración 43. Funcionamiento de página web en host cliente (Windows)

ANEXO 4. VALORES DE CALIFICACIÓN PROCESO ANALÍTICO JERÁRQUICO (AHP)

Proceso Analítico Jerárquico (AHP)		
Escala numérica	Escala verbal	Explicación
1	Igual Importancia	Dos elementos contribuyen igualmente al criterio.
3	Importancia moderada	El juicio y la experiencia favorecen a un elemento frente al otro.
5	Importancia fuerte	El juicio y la experiencia favorecen fuertemente a un elemento frente al otro
7	Importancia muy fuerte	Un elemento domina fuertemente. Probado en práctica.
8	Importancia absoluta	Un elemento domina al otro con el mayor orden de magnitud posible.

Tabla 7. Valores de calificación de acuerdo al Proceso Analítico Jerárquico (AHP)

Fuente: Elaboración propia

ANEXO 5. CALIFICACIÓN A SERVIDORES WEB DE ACUERDO CON CADA CARACTERÍSTICA

N°	Características	Servidores		
		HTTP Apache	Nginx	OpenLiteSpeed
1	Soporte técnico personalizado {2}	No {0}	Sí {2}	Sí {2}
2	Hardening de Cabeceras HTTP {3}	Sí {3}	Sí {3}	Sí {3}
3	Soporte WAF {6}	Sí {6}	Sí {6}	Sí {6}
4	Idiomas soportados {6}	Inglés {0}	Inglés {0}	Inglés, Chino Mandarín {6}
5	Archivo .htaccess {7}	Soportado {7}	No Soportado {0}	Soportado {7}
6	Multiplataforma {8}	Sí {8}	Sí {8}	No {0}
7	Interfaz Web {5}	No {0}	No {0}	Sí {5}
8	Visualización en tiempo real {9}	No {0}	No {0}	Sí {9}
9	Proxy HTTP {9}	Sí {9}	Sí {9}	Sí {9}
10	AntiDDOS {9}	No {0}	Sí {9}	Sí {9}
	TOTAL {64}	{33}	{37}	{56}

Tabla 8. Calificación a Servidores Web de acuerdo con cada característica

Fuente: Elaboración propia

ANEXO 6. PRUEBAS DEL 1 – 5 DE CADASERVIDOR WEB INSTALADO

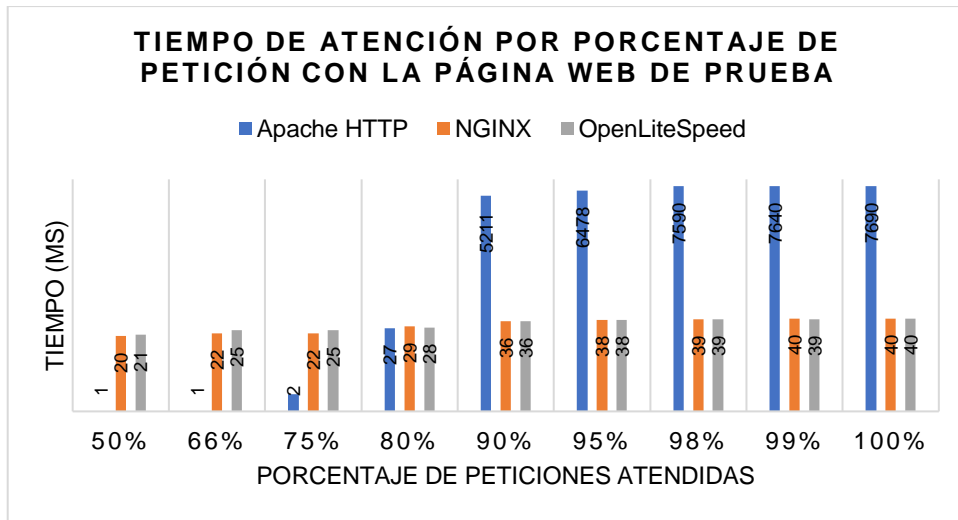


Ilustración 44. Prueba 1. Tiempo de atención por porcentaje de petición a página web

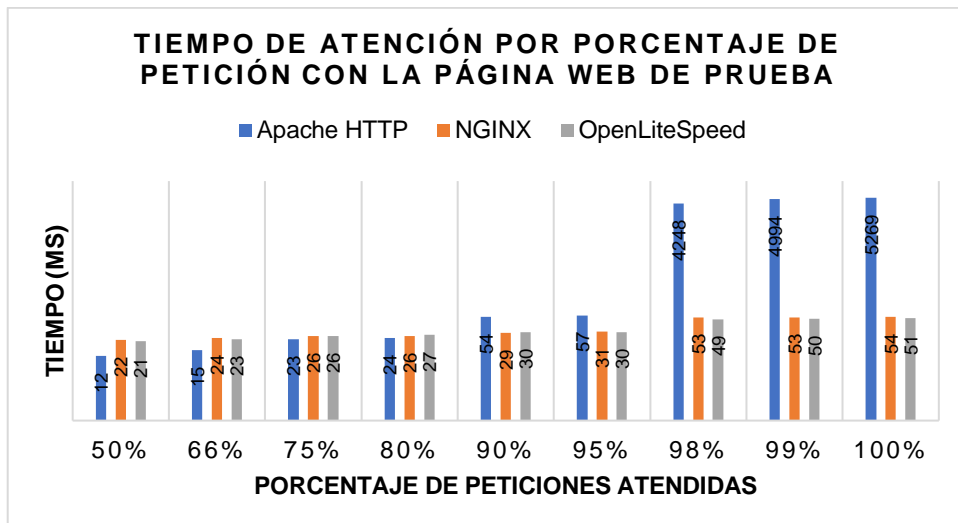


Ilustración 45. Prueba 2. Tiempo de atención por porcentaje de petición a página web

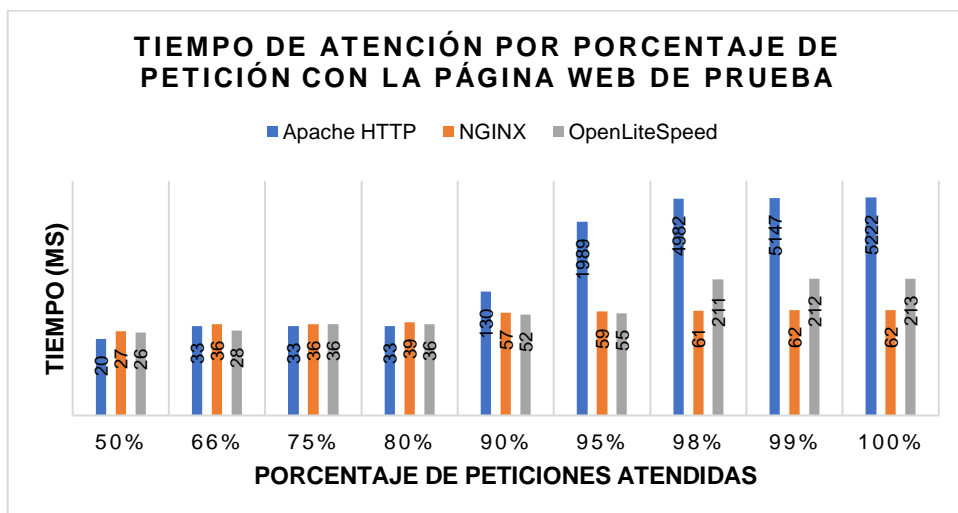


Ilustración 46. Prueba 3. Tiempo de atención por porcentaje de petición a página web

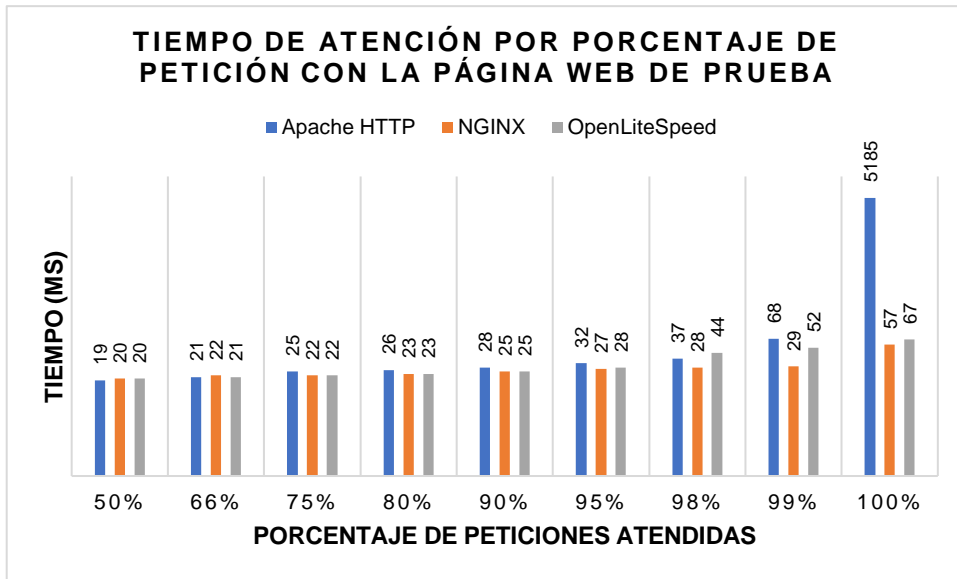


Ilustración 47. Prueba 4. Tiempo de atención por porcentaje de petición a página web

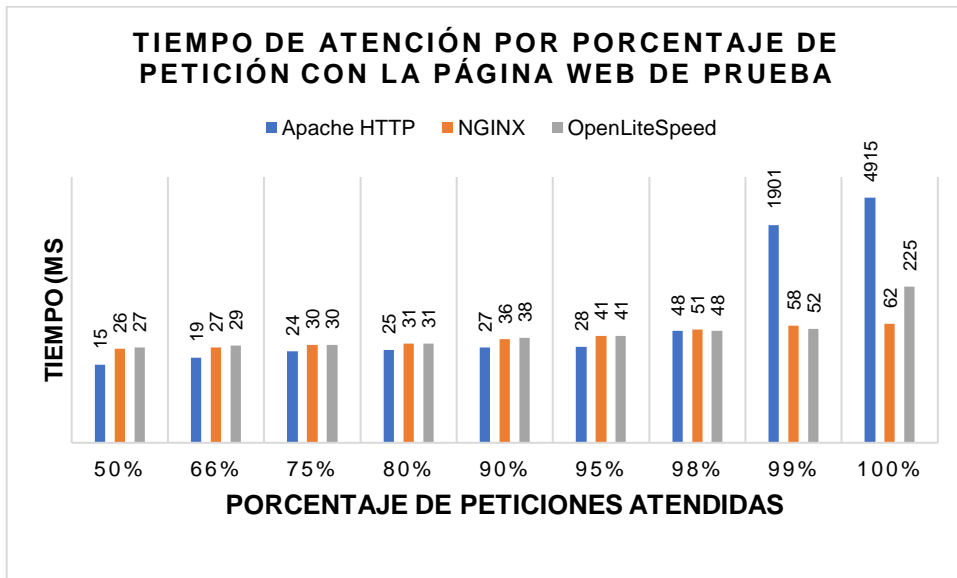


Ilustración 48. Prueba 5. Tiempo de atención por porcentaje de petición a página web

ANEXO 7. PRUEBA DE CARGA CON LA HERRAMIENTA APACHE BENCHMARK

```
[root@localhost ols]# ab -k -n 50000 -c 200 -g ols6.csv http://elorotravel.com/index.html
This is ApacheBench, Version 2.3 <$Revision: 1430300 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net
/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking elorotravel.com (be patient)
Completed 5000 requests
Completed 10000 requests
Server Software:      LiteSpeed
Server Hostname:     elorotravel.com
Server Port:         80

Document Path:       /index.html
Document Length:     21864 bytes

Concurrency Level:   200
Time taken for tests: 6.584 seconds
Complete requests:   50000
Failed requests:     0
Write errors:        0
Keep-Alive requests: 50000
Total transferred:   1105600000 bytes
HTML transferred:    1093200000 bytes
Requests per second: 7593.93 [#/sec] (mean)
Time per request:    26.337 [ms] (mean)
Time per request:    0.132 [ms] (mean, across all concurrent requests)
Transfer rate:       163981.48 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median  max
Connect:      0    0  2.2      0    50
Processing:   13   26 13.0     21   227
Waiting:      7    23 13.4     18   225
Total:       13   26 13.5     21   243

Percentage of the requests served within a certain time (ms)
 50%    21
 66%    29
 75%    29
 80%    30
 90%    44
 95%    52
 98%    61
 99%    65
100%   243 (longest request)
```

Ilustración 49. Utilización de la herramienta Apache Benchmark en el servidor OpenLiteSpeed