



**UNIVERSIDAD TÉCNICA DE BABAHOYO**

**FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA**

**PROCESO DE TITULACIÓN**

**OCTUBRE 2017 – MARZO 2018**

**EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA**

**PRUEBA PRÁCTICA**

**INGENIERIA EN SISTEMAS**

**PREVIO A LA OBTENCIÓN DEL TITULO DE INGENIERO EN SISTEMAS**

**TEMA:**

**ANÁLISIS DEL RENDIMIENTO DEL SERVIDOR WEB QUE ALOJA EL SISTEMA  
ACADÉMICO INTEGRADO DE LA UNIVERSIDAD TÉCNICA DE BABAHOYO**

**EGRESADO:**

**BAYRON JAMIL GARCIA CAMPAÑA**

**TUTOR:**

**ING. JOFFRE VICENTE LEÓN ACURIO, MSC**

**AÑO 2018**

## I. INTRODUCCION

La Universidad Técnica de Babahoyo es una institución de educación superior la cual cuenta con un Data Center en el que aloja a varios sistemas y servicios con la finalidad de optimizar los procesos que anteriormente se los llevaba de manera manual, haciendo mención al Sistema Académico Integral de la Universidad Técnica de Babahoyo (SAIUTB), el mismo que se encarga de procesar información estudiantil, bibliotecaria, internado, centro de idiomas y titulación.

Desde sus inicios los servidores informáticos estaban destinados al uso exclusivo de empresas de gran tamaño, y a su vez a un pequeño grupo de usuarios por lo tanto no se veían afectados con problemas de procesamiento de peticiones, pero con el avance de las redes de comunicaciones podemos notar que a un mismo servidor se puede acceder desde cualquier parte del planeta. Es así como surgen métodos y técnicas para contrarrestar la deficiencia de procesamiento de peticiones y evitar que los clientes se vean afectados y limitados de acceso a dicha información que estos equipos suelen alojar.

La presente investigación surge de la necesidad de mejorar el procesamiento de las peticiones que recibe el servidor web que aloja al SAIUTB, además de brindar la facilidad de mantenimiento y actualización a nuevas tecnologías que han surgido hasta la actualidad y a su vez eliminar de manera considerable el famoso cuello de botella. Para analizar esta problemática es necesario mencionar sus causas. Una de ellas es que los equipos actuales no abastecen a procesar la cantidad de usuarios que ingresan de manera masiva en ciertas temporadas tales como épocas de matriculación.

Estos problemas a nivel de servidor causan muchos inconvenientes, tanto al personal administrativo del departamento de sistemas que son los encargados del manejo y mantenimiento de dicho servidor, como a los usuarios del mismo (personal docente y estudiantil), teniendo que cortar el procesamiento fluido de datos para realizar tareas de mantenimiento y reinicio del servidor, esto incluso llega a aplazar en muchas ocasiones los tiempos programados en las matriculaciones de los estudiantes.

Dicha investigación hace referencia a la implementación de un nuevo servidor web para mejorar el acceso al SAIUTB. Como principal alternativa tenemos a Nginx (pronunciado en inglés “engine x”), es un programa de código abierto que ha llegado a situarse en el segundo lugar de los servidores web en el mundo de acuerdo a datos estadísticos de Netcraft, compañía dedicada al análisis de servidores web (Netcraft, 2017).

Mediante la implementación de este nuevo servidor web se podría mejorar el procesamiento de aproximadamente diez mil conexiones simultáneas y otras características destacables son: rendimiento, estabilidad, simplicidad de configuración y un bajo consumo de recursos.

## II. DESARROLLO

### **Reseña histórica de la implementación del Data Center en la Universidad Técnica de Babahoyo.**

El 5 de octubre de 1971 es concebida la Universidad Técnica de Babahoyo bajo decreto presidencial en el mandato del Dr. José María Velasco Ibarra, la misma que en su momento se creó con dos facultades: Ciencias de la Educación e Ingeniería Agronómica y Medicina Veterinaria. Actualmente cuenta con cuatro unidades académicas: Facultad de Ciencias Jurídicas, Sociales y de Educación, Facultad de Administración, Finanzas e Informática, Facultad de Ciencias Agropecuarias y Facultad de Ciencias de la Salud.

La Universidad Técnica de Babahoyo para un mejor desempeño y procesamiento de información a lo largo de sus años se ha ido dotando de infraestructura en cuanto al área tecnológica, llegando así a contar con su propio Data Center, el mismo que aloja varios servicios y sistemas entre ellos cabe destacar el SAIUTB el mismo que se encarga de procesar la información que en épocas o temporadas especiales como la de matriculación estudiantil que suelen ser de tamaño considerable.

Este estudio de caso se apega a la línea de investigación desarrollo de sistemas de la información, comunicación y emprendimientos empresariales y tecnológicos, y de acuerdo a la carrera la sub línea de investigación es desarrollo de sistemas informáticos, se utilizó la metodología de investigación analítica y como herramientas se usó entrevistas al encargado de los servidores en el departamento de sistemas de la Universidad Técnica de Babahoyo.

## **Servidores web.**

Un servidor web es una aplicación o programa informático el cual sirve para almacenar archivos de una página web y presentarlos mediante internet para de esta forma poder ser visualizados por los usuarios o clientes. Generalmente, un servidor web es una computadora de características muy avanzadas y su información en ella almacenada es transmitida mediante la red de redes llamada Internet. La forma en que el usuario accede a ella es a través de un navegador web el cual se comunica con el servidor y el mismo le arroja un resultado visual en pantalla de acuerdo a lo solicitado (Reig, 2010).

Actualmente toda empresa o en su mayoría tanto de carácter pública como privada podemos constatar que cuentan con sistemas o servicios que son accesibles a través de la Internet los mismos que están alojados en uno o varios servidores web, el o los mismos que pueden estar ubicados localmente o en la nube. De acuerdo a la demanda y necesidad actualmente se cuenta con una amplia gama de servidores web, entre ellos podemos enlistar los más conocidos y de uso a nivel mundial los mismos que cuentan con diferentes características y usos (SW Hosting, 2014).

- Apache
- Nginx
- Internet Information Services (IIS)
- Tomcat
- Sun Java System Application Server
- Google Web Server
- Lighttpd
- Thttpd (Forgiarini, 2017).

A continuación, una breve descripción de los servidores web que se analizaron para el desarrollo de este estudio de caso.

**Apache:** Es un servidor web HTTP de código abierto, desarrollado por Apache Software Foundation por sus siglas en inglés ASF bajo la autoría de Robert McCool en el año de 1995, su última versión estable es la 2.4.27 liberada el 11 de julio de 2017, el mismo que es de carácter multiplataforma: Unix (BSD, GNU/Linux), Microsoft Windows y Macintosh. Entre sus características principales tenemos: (Grupoapache, 2011).

- Adaptable a diferentes entornos.
- Sistema de notificación de errores HTTP.
- Gestión avanzada de indexación y alias de directorios (Grupoapache, 2011).

**Nginx:** Este servidor web es ligero y de alto rendimiento desarrollado por NGINX, Inc., su autor fue Igor Sysoev, lanzado por primera vez en el año 2004, la última versión estable 1.12.2 liberada el 17 de octubre de 2017, su funcionamiento es multiplataforma y bajo licencia BSD simplificada. Como características principales tenemos las siguientes: (Esaú, 2016).

- Servidor de archivos estáticos, índices y auto indexado.
- Proxy inverso con opciones de caché.
- Balanceo de carga.
- Tolerancia a fallos.
- Soporte de HTTP y HTTP2 sobre SSL.
- Soporte para autenticación.
- Compatible con IPv6.

- Habilitado para soportar más de 10.000 conexiones simultáneas (Esaú, 2016).

Antes de continuar profundizando en la investigación cabe mencionar que como objetivo de este estudio de caso se tiene como prioridad efectuar un análisis del servidor web actual y en base a la investigación bibliográfica realizar una propuesta de posible solución.

En cuanto al servidor físico que aloja al servidor web, tenemos que cuenta con las siguientes características empezando por 12 Gb de memoria RAM, doble procesador Intel Xeon X5650 estos trabajan a 2.67Ghz, 2 discos duros de 500Gb HDD los mismos que trabajan en modo espejo (es decir que, si uno falla, el otro se levanta automáticamente) además están distribuidos en un array 1+0 y a su vez trabajan a 7200 rpm, por otro lado, cuenta con una tarjeta de red con 4 puertos gigabit ethernet.

La parte lógica del servidor, cuenta con un sistema de distribución GNU/Linux el cual es CentOS 7 el mismo que cuenta con 10 años de soporte, por parte de la comunidad, esta distribución está instalada con el mínimo de paquetería para uso exclusivo del servidor, por otro lado, cuentan con un Firewall llamado IPTables el cual viene por defecto.

El Data Center de la Universidad Técnica de Babahoyo tiene a Apache HTTPD 2.4 como su servidor web, en el cual se encuentra alojado el SAIUTB el mismo que fue desarrollado en el conocido lenguaje de programación PHP 5.6. Cabe destacar que Apache es un servidor web que actualmente se sitúa en los primeros lugares en cuanto a su uso a nivel global, no obstante, los inconvenientes que se han suscitado en el transcurso de los años nos

dejan en claro que esta solución no es la óptima para el tipo de recursos físicos que con los que está equipado el Data Center Universitario actualmente (Axarnet, 2017).

Apache es un servidor web HTTP de código abierto, y gracias a su licencia de código libre ayudo a que haya crecido considerablemente a lo largo de los años. En primera instancia fue pensado para usarse en sistemas operativos Linux/Unix, posteriormente fue adaptado para trabajar en otros sistemas operativos que incluían a Mac y Windows. La ventaja de este servidor web es la funcionalidad de añadir módulos para ampliar sus capacidades (Fumás, 2014).

Cabe mencionar de manera generalizada los problemas que se pueden suscitar al usar el servidor web Apache, los formatos de configuración no están estandarizados por lo tanto no existe documentación oficial para todo tipo de error que pueda surgir. La falta de integración también es una de sus contras y para culminar, no cuenta con una buena administración, esto debido a que carece de personal que brinde soporte técnico o una oficina central a quien acudir en caso de ser necesario (BlueHosting, 2016).

El servidor web que aloja al SAIUTB es el que se encarga de procesar todas las peticiones tanto del personal docente como estudiantil, y procesar toda esa información no siempre se realiza adecuadamente y en ocasiones se generan ralentizaciones y colapsos del servidor.

Profundizando en los problemas causados por el colapso del servidor web podemos mencionar que el impacto va más allá que solo el aplazamiento de matrículas o el reinicio del



servidor web para que siga en su buen funcionamiento, sino que además de eso la institución pierde prestigio por la pobre administración del servidor que es tan indispensable para los usuarios.

La cantidad máxima de conexiones en simultaneo siempre serán las mismas sin importar que características físicas tengan los servidores, a menos que se tomen las medidas adecuadas de configuración para optimizar el rendimiento, esto quiere decir que se deberá enfocar mucho en saber configurar el software del servidor (Drassill, 2015).

Para saber la cantidad máxima de conexiones en simultaneo se depende mucho de la RAM del servidor, es decir, a mayor memoria RAM, mayor el número de conexiones, si bien no existe un valor fijo (es decir X clientes por cada X RAM) (Drassill, 2015).

A continuación, se realizará diferentes cálculos para conocer la cantidad máxima de conexiones en simultaneo del servidor que aloja el SAIUTB, no todas las conexiones consumen la misma RAM, con lo que se realizará una proyección estimada. Todo ello se puede obtener con el siguiente comando:

```
ps -y1C apache2 --sort:rss | awk '{SUM += $8; I += 1} END {print SUM/I/1024}'
```

El resultado que se obtiene con este comando es representado en megabytes y puede variar ya que depende mucho de las conexiones activas y el tipo de contenido que se esté consumiendo, en el caso del servidor de la UTB ha sido 11834 número que lo representamos redondeado en 11 MB de RAM, consumidos de media por conexión.

A continuación, debemos saber qué cantidad de memoria RAM es consumida por el resto de procesos, ejecutando el siguiente comando:

```
ps -N -ylC apache2 --sort:rss | awk '{SUM += $8} END {print SUM/1024}'
```

El resultado obtenido también se representará en megabytes, en el caso del servidor de la UTB proporcionó como resultado un consumo de 800 MB, con esta información podremos hacer el cálculo general de la cantidad de conexiones en simultaneo que podremos tener, dicho calculo lo podremos obtener mediante la siguiente operación:

**$(RAMTOTAL - RAM\_RESTOPROCESOS) / RAM\_POR\_CONEXIÓN$**

En el caso del servidor de la UTB

$$(12288 - 800) / 10 = 1148.8$$

Realizado este cálculo matemático se obtendrá una cifra utópica de conexiones en simultaneo de 1149 esta cifra tendrá que ser reducida en aproximadamente un 10% ya que esa es la cantidad máxima obtenida con el uso del 100% de memoria RAM, lo que generaría un cuelgue de todos los procesos. Entonces:

$$1149 - 10\% = 1034.1$$

El resultado aproximado óptimo de conexiones simultaneas al servidor que aloja el sistema SAIUTB sería de 1034, antes que dicho sistema empiece a ralentizar o colgarse.

## **Problemas en el rendimiento del servidor web que aloja el sistema académico integrado de la Universidad Técnica de Babahoyo.**

En la Universidad Técnica de Babahoyo existe un servidor en el cual se encuentra alojado el Sistema Académico Integrado, dicho sistema es el encargado de realizar los siguientes procesos: registro de ficha estudiantil, matriculación en línea, consulta de horarios académicos, consulta de notas académicas, revisión de historial de notas académicas, historial estudiantil y documentaciones. Por lo cual dicho servidor está expuesto a peticiones tanto del personal docente como estudiantil, los cuales están en un rango de 7500 a 8500 personas, dependiendo del manejo de cupos para los estudiantes.

El servidor que aloja el sistema como ya mencionamos anteriormente es el que se encarga de procesar todas las peticiones tanto del personal docente como estudiantil, estas peticiones y procesos suelen incrementarse de manera muy considerable en periodos de matriculación debido al tiempo limitado en los cronogramas de todas sus facultades, por lo que dicho servidor tiende a disminuir su rendimiento e incluso en muchas ocasiones a saturarse y colapsar por completo por la formación de famoso cuello de botella.

Las constantes caídas del servidor originan muchos inconvenientes, tanto al personal administrativo del Departamento de Sistemas que son los encargados del manejo y mantenimiento de dicho servidor, como a los usuarios del mismo (personal docente y estudiantil), teniendo que cortar el procesamiento fluido de datos para realizar tareas de mantenimiento y reinicio del servidor, esto incluso llega a aplazar en muchas ocasiones los tiempos programados para matriculaciones de los estudiantes.

Tomando en cuenta que la demanda de peticiones al servidor web de la UTB suele incrementarse por lo general en épocas de matrícula estudiantil, se puede argumentar que al contar con un único servidor el procesamiento de tal demanda se ve limitada a aproximadamente 1034 usuarios como ya se mencionó anteriormente, además, un solo usuario puede realizar múltiples peticiones en simultáneo dependiendo de los requerimientos.

La redundancia de peticiones hacia el servidor por parte de los usuarios se incrementa al notar que una página web carga parcialmente o no carga, debido a que el servidor no cuenta con un espacio de cacheo o configuración adecuada que permita al usuario visualizar un recargo de la página, desde un servicio de cache.

Por la problemática redactada anteriormente surge la necesidad de realizar un estudio, de cuáles serían las soluciones en potencia a todos los problemas, para el flujo adecuado de procesamiento de datos en el servidor web.

### **Soluciones Potenciales.**

Debido a los problemas antes mencionados, el servidor que guarda el SAIUTB requiere mejoras para solucionarlo, dentro de lo posible se puede mencionar tres potenciales soluciones que desde el punto de vista analítico pueden ser las siguientes:

- Cambiar de Apache Server a Nginx.
- Balancear la carga del servidor web.
- Y establecer un servicio de cache para una mayor velocidad de carga.

**A continuación, se sustenta con detalles técnicos porque se sugieren estas soluciones.**

Nginx es un servidor proxy HTTP y reverso, un servidor proxy de correo y un servidor proxy genérico TCP/UDP. Desde hace mucho tiempo atrás, este servidor se viene usando en muchos sitios rusos de alta demanda, tales como: Yandex, Mail.Ru, VK y Rambler. Según datos estadísticos de Netcraft, Nginx sirvió o proceso en 30.17% de los sitios web con mayor concurrencia en diciembre de 2017 (Nginx, 2017).

Apache es el servidor web más usado en el mundo con un 42.32% de cuota de uso en servidores a nivel mundial, mientras que Nginx paso a posicionarse en segundo lugar con una cuota de 22.23% de uso a nivel mundial, según Netcraft a septiembre del 2017 (Netcraft, 2017).

Pero que Apache sea el más utilizado a nivel mundial no quiere decir que sea el mejor del mundo, tanto Apache como Nginx son servidores de código abierto y juntos son los responsables de servir más de la mitad de todo el tráfico de internet del mundo (Ellingwood, 2015).

Si bien Apache y Nginx comparten muchas cualidades no deberían considerarse completamente intercambiables, a continuación, se enlista las razones del porque Nginx y no Apache Server.

- Es más ligero.
- Es multiplataforma y fácil de instalar.
- Se puede usar en paralelo con Apache.
- Se puede usar como cache.

- Funciona como balanceador de carga.
- Soporte profesional y comunitario.
- Compatible con las aplicaciones web más populares (Wajse, 2015).

La diferencia más significativa e importante, es que Nginx es más ligero, ya que reduce significativamente el consumo de memoria RAM y realiza un mayor procesamiento de datos por hilos en CPU, por la forma real en que manejan las conexiones y el tráfico, puesto que Nginx se basa en una arquitectura orientada a eventos. Apache se basa en una arquitectura basada en procesos (Andres, 2014).

Por lo tanto, Nginx desde su creación ha visto un alto crecimiento debido a que utiliza menos recursos que otros servidores web y su fácil escalabilidad con bajos recursos de hardware. Nginx se destaca por entregar contenido estático con rapidez y su desarrollo cuenta con un diseño para pasar solicitudes dinámicas a otro software que sea apropiado para su presentación (Ellingwood, 2015).

Concurrentemente Nginx es elegido por los administradores de TI por su eficacia de uso de recursos y capacidad de responder ante altas cargas de datos. Los defensores utilizan Nginx en el servidor web principal por su enfoque, además de su funcionalidad proxy (Ellingwood, 2015).

Nginx tuvo su lanzamiento posterior a Apache, con una amplia experiencia acerca de problemas de concurrencia que afectarían a los sitios a escala. Valiéndose de esta experiencia,

Nginx se creó desde cero para hacer uso de un algoritmo de manejo de conexión asíncrono, no bloqueante, impulsado por eventos (Ellingwood, 2015).

Nginx crea procesos de trabajo, estos procesos de manera individual pueden procesar miles de conexiones, efectuando un mecanismo de bucle rápido que permanentemente sirve para verificar y manejar eventos de los procesos de trabajo que se lleven a cabo. Al momento de desacoplar el trabajo real de las conexiones cada trabajador se centra en una conexión únicamente cuando se desencadena un nuevo evento (Ellingwood, 2015).

Todas y cada una de las conexiones administradas por el trabajador se ubica dentro del bucle de evento junto a otras conexiones ya existentes. Estos eventos son procesados de manera asíncrona mientras se encuentren dentro del mismo ciclo, permitiendo que el trabajo sea gestionado de manera no bloqueante. Una vez finalizada la conexión, esta pasa a eliminar el bucle (Ellingwood, 2015).

La modalidad de procesar las conexiones hace que Nginx escale de manera increíble incluso con bajos recursos de hardware. Este servidor cuenta con un solo subproceso y los procesos no se crean para operar una nueva conexión, el uso del CPU y la memoria se mantienen consistentes en él tiempo, incluso en momentos de gran demanda (Ellingwood, 2015).

Además del cambio que representaría pasar de Apache a Nginx Server, sería viable adquirir otro servidor físico con las mismas o mejores características que el actual, al realizar

esa adquisición se podría balancear la carga que tiene uno solo, pudiendo así distribuir las peticiones entre ambos servidores, optimizando el uso de los recursos, maximizando el número de peticiones atendidas por unidad de tiempo, minimizar la latencia y brindar la latencia a fallos (Pico, 2016).

La escalabilidad para procesar grandes cantidades de datos en un servidor web que ha llegado a su límite, es más conveniente agregar servidores mediante escalabilidad horizontal, ya que escalar verticalmente, adquiriendo servidores con mejores características a los servidores actuales significaría una alta inversión (Pico, 2016).

Una de las ventajas más destacables que tiene la implementación de Nginx es que también permite configurarlo como balanceador de carga con una configuración muy sencilla y de manera nativa.

Como ultima recomendación se menciona en este documento realizar el cacheo del sitio para acelerar la web y mejorar la experiencia de usuario. Por si fuese poco Nginx también nos permite configurarlo como un servicio de cacheo de proxy inverso, el cual es un tipo de servidor situado entre el cliente y el servidor web el cual recupera los recursos de estos. Estos recursos son devueltos al usuario como si el propio servidor web los hubiese enviado, básicamente actúa como intermediario que puede desempeñar distintos roles (Vega, 2017).

Si se utiliza Nginx como servidor de caché, realiza peticiones al servidor web para saber si este cuenta con los recursos de una solicitud, generando un MISS y guardando en la memoria



dicho recurso. Por lo tanto, la próxima petición a ese recurso es entregada por el mismo proxy de forma eficaz originando un HIT, lo cual impide que se realice una petición al servidor web en un TTL determinado. Pudiendo montar un servidor centralizado de contenido estático, desde el cual se genere todo el contenido que se llegase a necesitar (Vega, 2017).

### III. CONCLUSION

Luego de analizar los problemas que afectaban al SAIUTB y al servidor web de la Universidad Técnica de Babahoyo, se empezó con el análisis de las características que tiene el servidor y de qué forma está estructurado a nivel de software, además, se utilizó una fórmula de medición para obtener un dato estadístico aproximado de peticiones o usuarios que puede soportar el servidor.

Llegando a la conclusión que el servidor que se utiliza actualmente, no cuenta con los suficientes recursos físicos necesarios para el manejo de todas las peticiones que generan los usuarios que acceden al SAIUTB y que, además, el actual servidor web utilizado (Apache Server) no es el óptimo para administrar las peticiones de servidores de alta demanda con bajos recursos.

Concluyendo, para solucionar dichos problemas se recomienda principalmente el cambio de servidor web de Apache a Nginx, el mismo que se ha mencionado a lo largo de este documento contando con muchas ventajas para el manejo de peticiones y contenido, a su vez se recomienda la adquisición de un nuevo servidor físico de iguales o mejores características que el actual, el cual permitiría realizar un balanceo de carga y distribuir la demanda de peticiones entre ambos servidores y por último realizar un servicio de cacheo para aligerar la administración de peticiones en el servidor web, ya que tanto el balanceo como el servicio de caché lo podemos hacer con Nginx sin la utilización de software adicional o de terceros.

## BIBLIOGRAFIA

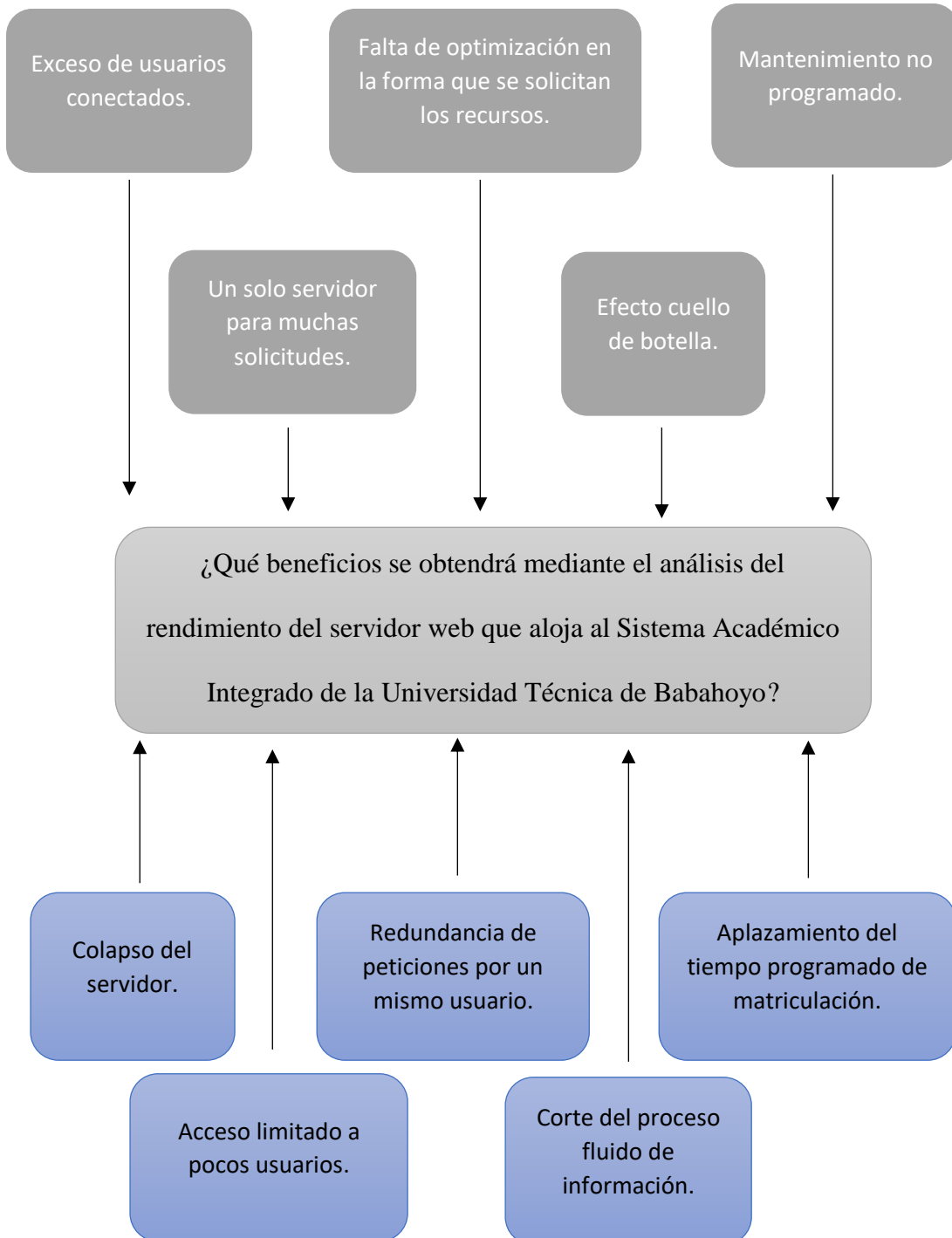
- Accerto. (2014). *Estrategias de reclutamiento 2.0*. Barcelona: Grupo Planeta Spain. Obtenido de [https://books.google.com.ec/books?id=6xIaAwAAQBAJ&printsec=frontcover&source=gbg\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.ec/books?id=6xIaAwAAQBAJ&printsec=frontcover&source=gbg_summary_r&cad=0#v=onepage&q&f=false)
- Accerto. (2014). *Medición de la eficacia de un sitio web*. Barcelona: Grupo Planeta Spain. Obtenido de <https://books.google.com.ec/books?id=iBYaAwAAQBAJ&pg=PA3&dq=analisis+del+trafico+web&hl=es-419&sa=X&ved=0ahUKEWj537TAuOfXAhXGLSYKHU02CpsQ6AEINTAD#v=onepage&q&f=false>
- Aguilar, L. (2016). *Big Data, Análisis de grandes volúmenes de datos en organizaciones*. Mexico: Alfaomega Grupo Editor. Obtenido de [https://books.google.com.ec/books?id=1GywDAAAQBAJ&printsec=frontcover&dq=herramientas+de+analisis+de+rendimiento+de+servidores&hl=es-419&sa=X&ved=0ahUKEWiiy4y7mefXAhXFYyYKHfs\\_CV8Q6AEIKjAB#v=onepage&q&f=false](https://books.google.com.ec/books?id=1GywDAAAQBAJ&printsec=frontcover&dq=herramientas+de+analisis+de+rendimiento+de+servidores&hl=es-419&sa=X&ved=0ahUKEWiiy4y7mefXAhXFYyYKHfs_CV8Q6AEIKjAB#v=onepage&q&f=false)
- Andres. (26 de Marzo de 2014). *Apache vs Nginx: ¿cuál es mejor?* Obtenido de <https://guiadev.com/apache-vs-nginx-cual-es-mejor/>
- Axarnet. (7 de Noviembre de 2017). *Comparativa de los servidores web más utilizados*. Obtenido de <https://www.axarnet.es/blog/comparativa-de-los-servidores-web-mas-utilizados/>
- Banquet, P., & Bobillier, S. (2015). *Linux: administración del sistema y explotación de los servicios de red* (Tercera ed.). Barcelona: Ediciones ENI. Obtenido de [https://books.google.com.ec/books?id=MT0r8Od\\_-j8C&printsec=frontcover&source=gbg\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.ec/books?id=MT0r8Od_-j8C&printsec=frontcover&source=gbg_summary_r&cad=0#v=onepage&q&f=false)
- BlueHosting. (27 de Septiembre de 2016). *Guía de solución de problemas comunes de Apache*. Obtenido de <https://docs.bluehosting.cl/troubleshooting/servidores/guia-de-solucion-de-problemas-comunes-de-apache.html>
- Calomel Org. (1 de Enero de 2017). *tHttpd Server Tutorial*. Obtenido de <https://calomel.org/thttpd.html>
- Castro, V., Armenta, X., & Quiroz, M. (2014). *Fundamentos de las tecnologías de Información: viviendo en una sociedad tecnológica*. Monterrey: Digital del Tecnológico de Monterrey. Obtenido de <https://books.google.com.ec/books?id=vUJuDQAAQBAJ&printsec=frontcover&dq=protocolo+para+interconectar+programas&hl=es-419&sa=X&ved=0ahUKEWjD-eDujfHXAhWxSN8KHa7sAekQ6AEINzAD#v=onepage&q&f=false>

- Christensson, P. (11 de Diciembre de 2013). *Definición de IIS*. Obtenido de <https://techterms.com/definition/iis>
- Clifton, M. (2015). *Web Servers Succinctly*. Morrisville: Syncfusion.
- Drassill. (21 de Julio de 2015). *Cómo aumentar las conexiones simultáneas en Apache*. Obtenido de <https://blog.desdelinux.net/como-aumentar-conexiones-simultaneas-apache/>
- Durán, A. (2015). *Acceso a datos en aplicaciones web del entorno servidor. IFCD0210*. Málaga: IC Editorial.
- Durango, A. (2015). *Diseño de Software* (Segunda ed.). Vigo: IT Campus Academy. Obtenido de <https://books.google.com.ec/books?id=1ShpCwAAQBAJ&printsec=frontcover&dq=tolerancia+a+fallos&hl=es-419&sa=X&ved=0ahUKEwjnmrHc9vDXAhUhm-AKHcPYCikQ6AEIQDAF#v=onepage&q&f=false>
- Ellingwood, J. (28 de Enero de 2015). *Apache vs Nginx: Practical Considerations*. Obtenido de <https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations>
- Esaú, A. (23 de Febrero de 2016). *Open Webinars*. Obtenido de <https://openwebinars.net/blog/nginx-vs-apache/>
- Forgiarini, L. (2 de Enero de 2017). *Luis Forgiarini Blog*. Obtenido de <https://luisforgiariniblog.com/ejemplos-servidores-web-mas-utilizados/>
- Fumás, E. (11 de Junio de 2014). *Apache HTTP Server: ¿Qué es, cómo funciona y para qué sirve?* Obtenido de <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>
- Grupoapache. (20 de Junio de 2011). *APACHE SOFTWARE FOUNDATION*. Obtenido de <https://apachefoundation.wikispaces.com/Apache+HTTP+Server>
- Heurtel, O. (2015). *PHP 5.6: desarrollar un sitio web dinámico e interactivo*. Barcelona: Ediciones ENI. Obtenido de <https://books.google.com.ec/books?id=O1JyeUYGwF8C&printsec=frontcover&dq=php+5.6&hl=es-419&sa=X&ved=0ahUKEwjcpNbXlufXAhWJWCYKHQYqB1UQ6AEIJTAA#v=onepage&q&f=false>
- Limoncelli, T., Hogan, C., & Chalup, S. (2016). *The Practice of System and Network Administration* (Tercera ed.). Reading: Addison-Wesley Professional.
- Lockhart, J. (2015). *Modern PHP: New Features and Good Practices*. Sebastopol: O'Reilly Media, Inc. Obtenido de <https://books.google.com.ec/books?id=rnSpBgAAQBAJ&pg=PT308&dq=PHP-FPM&hl=es-419&sa=X&ved=0ahUKEwj27pXs-vDXAhUhc98KHRI7AasQ6AEIODAC#v=onepage&q&f=false>

- Medina, J. (2014). *Pruebas de Rendimiento TIC* (Segunda ed.). Murcia: Lulu.com. Obtenido de [https://books.google.com.ec/books?id=kTvbBgAAQBAJ&pg=PA27&dq=pruebas+de+stress+web&hl=es-419&sa=X&ved=0ahUKEwj40KX\\_4-bXAhVBLyYKHW08CnUQ6AEIJTAA#v=onepage&q&f=false](https://books.google.com.ec/books?id=kTvbBgAAQBAJ&pg=PA27&dq=pruebas+de+stress+web&hl=es-419&sa=X&ved=0ahUKEwj40KX_4-bXAhVBLyYKHW08CnUQ6AEIJTAA#v=onepage&q&f=false)
- Netcraft. (20 de Julio de 2017). *July 2017 Web Server Survey*. Obtenido de <https://news.netcraft.com/archives/2017/07/20/july-2017-web-server-survey.html>
- Netcraft. (26 de Octubre de 2017). *Web Server Survey*. Obtenido de <https://news.netcraft.com/archives/2017/10/26/october-2017-web-server-survey-13.html>
- Nginx. (28 de Diciembre de 2017). *Nginx*. Obtenido de <https://nginx.org/en/>
- Pico, D. (9 de Julio de 2016). *Configurar Nginx como balanceador de carga*. Obtenido de <https://picodotdev.github.io/blog-bitix/2016/07/configurar-nginx-como-balanceador-de-carga/>
- Reig, D. (20 de Septiembre de 2010). *Duplika*. Obtenido de <https://duplika.com/blog/que-son-los-servidores-web-y-por-que-son-necesarios/>
- Soni, R. (2016). *Nginx: From Beginner to Pro*. Kolkata: Apress. Obtenido de [https://books.google.com.ec/books?id=uQvpDAAAQBAJ&printsec=frontcover&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.ec/books?id=uQvpDAAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- SW Hosting. (20 de Noviembre de 2014). *Blog de SW Hosting*. Obtenido de <https://www.swhosting.com/blog/las-ventajas-de-tener-un-servidor-propio/>
- Talledo, J. (2015). *Implantación de aplicaciones web en entorno internet, intranet y extranet*. Madrid: Ediciones Paraninfo, S.A. Obtenido de <https://books.google.com.ec/books?id=RtESCgAAQBAJ&pg=PA69&dq=concepto+de+servidor+web&hl=es-419&sa=X&ved=0ahUKEwiFvNXqofjXAhXJ6yYKHdcgBjoQ6AEIJjAA#v=onepage&q&f=false>
- Thompson, C. (2015). *Vagrant Virtual Development Environment Cookbook*. Birmingham: Packt Publishing Ltd. Obtenido de [https://books.google.com.ec/books?id=UzDWBgAAQBAJ&printsec=frontcover&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.ec/books?id=UzDWBgAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- Vega, A. (14 de Junio de 2017). *Acelera tu web con Nginx como caché de proxy inverso*. Obtenido de <https://medium.com/@alonus91/accelera-tu-web-con-nginx-como-cach%C3%A9-de-proxy-inverso-fd25722a0dc2>
- Wajse, D. (18 de Agosto de 2015). *Combate entre servidores web: NGINX vs Apache*. Obtenido de <http://latamdigital.softtek.co/combate-entre-servidores-web-nginx-vs-apache>

## ANEXOS

### ARBOL DE PROBLEMAS



**ENTREVISTA REALIZADA POR BAYRON JAMIL GARCIA CAMPAÑA AL ING. SERGIO ABRIL CAMPUZANO ANALISTA DE PROYECTOS Y SOLUCIONES TECNOLÓGICAS.**

**1) ¿Con que tipo de servidor funcionan actualmente los sistemas web?**

Actualmente la se cuenta con el servidor web Apache.

**2) ¿Qué tecnología usa el servidor para el alojamiento de los servidores web?**

Open Source – Código Abierto (Servidor Web Apache y para el lenguaje de servidor PHP 5.6).

**3) ¿Por qué se decidió usar ese tipo de tecnología para los sistemas web?**

1. Los aplicativos recomiendan esta tecnología.
2. La tecnología Open Source tiene soporte de la comunidad en lugar de una empresa, lo cual ayuda a corregir posibles errores rápidamente.
3. Es estable y hace uso óptimo de los recursos de hardware además brinda facilidad para migrar a otras tecnologías Open Source.

**4) ¿Cómo se mide el rendimiento del Sistema Académico Integrado de la Universidad Técnica de Babahoyo?**

El rendimiento es monitorizado a nivel de uso de base de datos a través de herramientas proporcionadas por el administrador de base de datos pgAdmin v4, también se monitoriza en el sistema lo cual nos proporciona el uso de los procesadores, memoria RAM y red.

A nivel del servidor web se monitorizan las peticiones que entran al servidor, el tiempo de respuesta y el uso de red, también se ejecutan pruebas de DoS y de QoS. Una herramienta usada generalmente es Apache JMeter.

**5) ¿En caso de requerimiento de estadísticas del sistema SAIUTB con que programa se realiza las analíticas?**

Se usa Google Analytics.

**6) ¿Con que tipo de seguridad cuenta el servidor web que aloja al sistema SAIUTB?**

Cuenta con firewall a nivel de sistema, además está aplicado un firewall a nivel de servidor web el cual detecta patrones de ataque como SQL injection y Cross-site scripting y varios ataques más.

**7) ¿En caso de fallo del servidor web, con que planes de contingencia cuentan?**

Se cuenta con varias instalaciones de servidores web replicadas en las cuales se activaría el servicio con facilidad.

**8) ¿Que considera Usted que mejoraría el rendimiento del SAIUTB?**

A nivel de hardware adquirir equipos actualizados mucho mejores a los que contamos, a nivel de procesador, discos duros, memoria RAM y tarjetas de red. A nivel de software la implementación de otro tipo de servidor web que optimice los recursos y que los tiempos de respuesta sean menores al servidor Apache y a su vez cuente con un nivel más alto de seguridad.

---

**Ing. Sergio Abril Campuzano**

Analista de Proyectos y Soluciones Tecnológicas

Departamento Soporte Sistemas – UTB