

MODELO FURPS PARA EL ANÁLISIS DEL RENDIMIENTO DE FRAMEWORKS JSF

FURPS MODEL FOR PERFORMANCE ANALYSIS OF FRAMEWORKS JSF

Alex Fabian Yungan Gualli

Universidad Nacional de Chimborazo.

Riobamba, Ecuador.

E-mail: ayungan.fis@unach.edu.ec ORCID: <https://orcid.org/0000-0002-2712-886X>

Cristian Hugo Morales Alarcón

Universidad Nacional de Chimborazo.

Riobamba, Ecuador.

E-mail: cmorales@unach.edu.ec ORCID: <https://orcid.org/0000-0002-0197-0581>

Jorge Edwin Delgado Altamirano

E-mail: jdelgado@unach.edu.ec ORCID: <https://orcid.org/0000-0002-6782-806X>

Lady Marieliza Espinoza Tinoco

Universidad Nacional de Chimborazo.

Riobamba, Ecuador.

E-mail: lespinoza@unach.edu.ec ORCID: <https://orcid.org/0000-0001-6569-3686>

Recepción: 20/07/2019 **Aceptación:** 22/10/2019 **Publicación:** 30/12/2019

Citación sugerida:

Yungan Gualli, A.F., Morales Alarcón, C.H., Delgado Altamirano, J.E. y Espinoza Tinoco, L.M. (2019). Modelo FURPS para el análisis del rendimiento de frameworks JSF. *3C TIC. Cuadernos de desarrollo aplicados a las TIC*, 8(4), 65-83. doi: <http://doi.org/10.17993/3ctic.2019.84.65-83>

RESUMEN

La presente investigación analiza el rendimiento de los frameworks front end Java Server Faces PrimeFaces y BootsFaces, basados en el modelo de calidad de software FURPS. Se utilizó el tipo de estudio transversal, investigación descriptiva, comparativa y correlacional, donde se establece una muestra de 385 peticiones a través del Protocolo de Transferencia de Hipertexto para cada uno de los grupos de investigación, para lo cual se creó un prototipo con cada uno de los frameworks mencionados. Para compararlos se utilizó el promedio de: consumo de recursos, peticiones de hipertexto realizadas correctamente y el tiempo de respuesta, estos datos se obtienen por medio del programa JMeter. Los resultados permiten determinar el rendimiento, el mejor tiempo de respuesta.

PALABRAS CLAVE

PrimeFaces, BootsFaces, Modelo FURPS, Frameworks, Front-end.

ABSTRACT

The present investigation analyzes the performance of the front end frameworks Java Server Faces, PrimeFaces and BootsFaces, based on FURPS software quality model. It was used the type of cross-sectional study, descriptive, comparative and correlational research, where a sample of 385 through the Hypertext Transfer Protocol requests was established for each of the research groups, for which a prototype was created with each of the mentioned frameworks. To compare them, it was used the average of: consumption of resources, hypertext requests made correctly and the response time, these data are obtained through the JMeter program. The results allow to determine the performance, the best response time.

KEYWORDS

PrimeFaces, BootsFaces, FURPS model, Frameworks, Front-end.

1. INTRODUCCIÓN

En la actualidad los sistemas web son una herramienta de gran importancia para las empresas debido a las ventajas que ofrecen, como es el ahorro de tiempo y dinero, los únicos requisitos son: tener un navegador web y una conexión a internet lo cual facilita el acceso a la información. La plataforma Java según International Business Machines Corporation (IBM) es un entorno para desarrollar y gestionar applets y aplicaciones Java (IBM, 2014), estas aplicaciones están compuestas de tres componentes principales: el lenguaje Java, este lenguaje de programación fue el más utilizado en el año 2018 (TIOBA, 2019), además de los paquetes Java y la máquina virtual Java (Ordax, Barón y Ocana, 2012).

Java Platform Enterprise Edition (Java EE) es una plataforma de programación que permite el desarrollo de aplicaciones web robustas que garantizan la confiabilidad y seguridad (Pech, Gomez, y Lara, 2011), también simplifica el desarrollo de aplicaciones y reduce la necesidad de programación y formación para programadores, al crear componentes modulares normalizados y reutilizables (Oracle, 2014a), está conformado por las aplicaciones cliente y los applets que se ejecutan en el cliente Java Servlet, Java Server Faces (JSF) y Java Server Pages (JSP) (Bergsten, 2004). Una aplicación web puede ser desarrollada utilizando la tecnología Java Server Faces (JSF), la cual permite representar componentes de interfaz de usuario, el manejo de eventos, definir la navegación de páginas, el apoyo a la internalización y accesibilidad (Oracle, 2010), por lo cual esta tecnología simplifica el desarrollo de interfaces de usuarios.

Java Server Faces (JSF) ha creado varios frameworks que contienen un conjunto de componentes que permiten realizar interfaces gráficas, modernas y atractivas, que se integran fácilmente con herramientas como Omnifaces, permitiendo el desarrollo de aplicaciones web rápidamente y son adecuados para desarrolladores que carecen de habilidades para la programación de front end (Farrell, 2018).

PrimeFaces es un framework para programación de interfaz de usuario que se puede utilizar para desarrollar rápidamente aplicaciones sofisticadas para una organización o empresa (Oracle, 2014a),

y otro de los frameworks JSF con un propósito similar es BootsFaces, el cual se considera como un marco de trabajo potente y ligero basado en Bootstrap 3 y que utiliza además, la tecnología jQuery, permitiendo el desarrollo front-end para aplicaciones empresariales de forma rápida y fácil (Urzo y Massera, 2019), se caracteriza generalmente porque ofrece una gran cantidad de componentes ricos o widgets que ayuda a los desarrolladores a obtener un sitio bien diseñado, ligero y con poco esfuerzo; aprovecha el sistema de red de Bootstrap y le permite combinar con características JSF templating, las páginas se adaptan automáticamente en todos los dispositivos, también, permite personalizar la apariencia de los componentes basados en Bootstrap, este framework ocupa espacio hasta 10 veces menos en comparación con otros marcos de trabajo JSF, además, cada componente se carga sólo con los recursos necesarios es decir con las Hojas de estilo en cascada, Cascading Style Sheets (CSS) y JavaScript necesarios. Uno de los puntos claves de éxito que tiene una aplicación web es la usabilidad y el rendimiento debido a que esto puede afectar no solamente a la experiencia del usuario al visitar el sitio web (Guaman, Palacios, y Contenido, 2019), sino también puede afectar de manera positiva o negativa su rendimiento (Solvetic, 2015), por tal razón es de suma importancia analizar cuál de los dos frameworks ofrece un mejor rendimiento, con la finalidad de ayudar a los desarrolladores a escoger herramientas adecuadas al momento de implementar una aplicación de estas características.

Uno de los puntos claves de éxito que tiene una aplicación web es la usabilidad y el rendimiento debido a que esto puede afectar no solamente a la experiencia del usuario al visitar el sitio web (Guaman, Palacios, y Contenido, 2019).

El rendimiento web es la medición del tiempo que transcurre desde cuando se ingresa a un sitio o a una aplicación web determinada y esta se muestra en su totalidad, en esta petición a través del Protocolo de Transferencia de Hipertexto (HTTP) pueden verse afectados los recursos que usa el servidor de aplicaciones (Solvetic, 2015), modelo basado en Funcionalidad, Usabilidad, Confiabilidad, Prestación y Soporte (FURPS) es un modelo desarrollado por Hewlett-Packard en el año 1987 en el que se desarrollan

un conjunto de factores de calidad de software, bajo el acrónimo de FURPS (Constanzo, 2014), en cuanto al rendimiento este modelo de calidad de software determina que las métricas que permiten determinar cuál tecnología proporciona mejores características son: tiempo de respuesta, consumo de recursos y eficacia.

Tabla 1. Trabajos relacionados.

Tema	Descripción y hallazgos
Análisis del rendimiento de librerías de componentes Java Server Faces en el desarrollo de aplicaciones web	Realiza una comparación de PrimeFaces y RichFaces en la cual se establece las siguientes dimensiones: tiempo promedio de respuesta de página, y tiempo promedio de respuesta en Ajax, con el objetivo de analizar cuál de estos ofrece un mejor rendimiento, esta investigación se realiza sobre una aplicación en N capas y sobre un servidor web Tomcat y sobre una población de 384 peticiones por cada grupo de estudio (Guaman <i>et al.</i> , 2018).
Análisis comparativo de frameworks JSF 2.0: IceFaces, PrimeFaces y RichFaces; para la implementación en el desarrollo del sistema de gestión de proyectos ambientales de la empresa Kaymanta	Realiza una evaluación técnica de los frameworks JSF IceFaces, PrimeFaces y RichFaces para lo cual se basa en el Modelo de Construcción de Calidad Individual (IQMC) y las características de calidad propuestas en la norma ISO 25000, las cuales permiten establecer las métricas para medir la calidad de software, una de estas métricas es el rendimiento (Ortega y Rodríguez, 2014).
Análisis comparativo de frameworks para el desarrollo de aplicaciones web en java.	Realiza un análisis comparativo de los frameworks Spring, Struts, JSF y Angular JS para lo cual se basa en la ISO 25000 y IQMC, el cual da como resultado que JSF ha demostrado ser un framework poderoso en la capa de presentación por cual nos aconseja el uso de esta tecnología solo en la capa de presentación (Sánchez, Tuesta y Cabrera, 2015).
Investigation on performance testing and evaluation of PReWebN: a java technique for implementing web application	Nos menciona que la prueba de rendimiento de la aplicación web es esencial desde la perspectiva de los usuarios y de los desarrolladores, para lo cual crean un prototipo de aplicación web de investigación basada en la técnica Java para estudiar el rendimiento y evaluar la técnica utilizada para desarrollar la aplicación web, para hacer esta prueba de rendimiento hace pruebas de carga y estrés en los prototipos utilizando Mercury LoadRunner para estudiar el rendimiento, la estabilidad, la escalabilidad, la confiabilidad, la eficiencia y la rentabilidad de la técnica. el rendimiento depende de métricas tales como hits / s, tiempo de respuesta, rendimiento, errores / s y resumen de transacciones el estudio revela que la aplicación web desarrollada con la técnica Java es más estable, confiable, escalable y rentable que su otra contraparte, como la tecnología Microsoft .NET (Kalita, Khanikar, y Bezboruah, 2011).

Estos estudios fueron de utilidad para el desarrollo de la presente investigación, sin embargo, cabe mencionar que ninguno de los estudios mencionados realiza el análisis del rendimiento del framework

JSF BootsFaces y tampoco utilizan el Modelo FURPS como base para realizar este análisis. Por lo cual esta investigación tuvo como objetivo realizar un análisis del rendimiento de los frameworks JSF PrimeFaces y BootsFaces (Yungan y Morales, 2019) basándose en el modelo FURPS, para lo cual se utilizó el software JMeter, herramienta que se puede utilizar para probar aplicaciones que utilizan servidores HTTP o FTP (Nevedrov, 2006) y se caracteriza principalmente por permitir pruebas de carga, integración de plugin, análisis de datos y reportes; por tener un núcleo altamente extensible para realizar las pruebas y sobre todo se adapta a las métricas de rendimiento del modelo FURPS. En base a lo mencionado en la investigación se estableció los siguientes objetivos específicos: determinar los parámetros de análisis de rendimiento de frameworks JSF basado en el modelo FURPS y analizar los resultados del factor rendimiento de los frameworks PrimeFaces y BootsFaces basado en el modelo FURPS.

Esta investigación tuvo como objetivo realizar un análisis del rendimiento de los frameworks JSF PrimeFaces y BootsFaces (Yungan y Morales, 2019) basándose en el modelo FURPS.

2. METODOLOGÍA

Para el análisis del rendimiento de cada uno de los frameworks JSF PrimeFaces y BootsFaces se estableció la siguiente hipótesis de investigación: La evaluación de los frameworks JSF PrimeFaces y BootsFaces mejora el rendimiento de los sistemas web basados en java EE. Para lo cual se planteó las siguientes hipótesis estadísticas:

- H_0 : No existe una diferencia significativa del rendimiento de los frameworks JSF PrimeFaces y BootsFaces.
- H_1 : Existe una diferencia significativa del rendimiento de los frameworks JSF PrimeFaces y BootsFaces.

2.1. DISEÑO Y TIPO DE ESTUDIO

El diseño de la investigación fue experimental debido a que se comparan dos grupos de estudio que no fueron seleccionados aleatoriamente (Manterola y Otzan, 2015), por lo cual se aplicó el tipo de estudio transversal, debido a que este tipo de estudio permite analizar un evento en un momento dado, por lo cual supera las limitaciones de tiempo (Cabrera, Bethencourt, González, y Álvarez, 2014), según el nivel de conocimiento se aplicó la investigación descriptiva debido a que esta investigación permite realizar la recopilación de los datos con la finalidad de organizar y tabular información (Abreu, 2012), para posteriormente mostrarla mediante gráficos y tablas, según el método a utilizar se aplicó la investigación comparativa debido a que se realiza la comparación entre dos grupos (Abreu, 2012).

2.2. POBLACIÓN Y MUESTRA

Se estableció como población el número de peticiones HTTP realizadas al Sistema Web, por lo cual se considera que la población es infinita. Para calcular la muestra se aplicó la siguiente ecuación (donde: n es la cantidad de muestra; Z es el nivel de confianza; p es la probabilidad de éxito; q es la probabilidad de fracaso y es el error muestral), una vez aplicada el resultado de la muestra es de 385 peticiones HTTP.

$$n = \frac{Z^2 * p * q}{e^2}$$

2.3. VARIABLES DE ESTUDIO

Las variables independientes de la investigación son: los frameworks JSF PrimeFaces y BootsFaces, el primero es un marco de interfaz de usuario que se puede utilizar para desarrollar rápidamente aplicaciones sofisticadas para la empresa o para los sitios web estándar (Oracle, 2014b), el segundo es un marco JSF potente y ligero basado en Bootstrap 3 y la interfaz de usuario jQuery que le permite desarrollar aplicaciones empresariales de front-end rápido y fácil (Urzo y Massera, 2019), las variables independientes mencionadas anteriormente contienen cinco indicadores: cantidad de líneas de código,

cantidad de componentes, tamaño de archivo, tiempo en el mercado y número de actualizaciones (Gizas, Christodoulou, y Papatheodorou, 2012; Murai y Klyuev, 2016). La variable dependiente es el rendimiento que tiene estos frameworks en el prototipo creado para medición, los indicadores de esta variable se basan en el modelo FURPS: Media de peticiones HTTP realizadas correctamente, promedio de tiempo de respuesta, promedio de uso de memoria RAM, promedio de uso de disco duro y promedio de uso del CPU.

2.4. INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Una aplicación o sitio web está formada por archivos con código de lenguaje de marcas como por ejemplo HTML el cual es el más extendido en la actualidad, y otros archivos como imágenes, documentos o scripts. Para acceder es necesario conocer la dirección IP o dirección DNS del sitio. El cliente en su navegador a través del protocolo HTTP realiza peticiones al servidor web, el mismo que luego de procesarlas las responde, de esta forma llegamos a observar y a utilizar las páginas y aplicaciones web en la internet.

En este contexto con la finalidad de medir el factor de rendimiento del modelo FURPS es necesario realizar peticiones HTTP hacia a los prototipos PrimeFaces y BootsFaces para esto se utilizó el software JMeter, una petición se emite utilizando métodos GET, POST, PUT y DELETE del protocolo HTTP (Barrueco y Subirats, 2014).

2.5. PROCEDIMIENTOS Y ANÁLISIS

Se realizó un análisis previo con los indicadores de las variables independientes debido a que el rendimiento puede depender o determinarse a partir de este análisis, en el segundo paso se creó un prototipo con cada uno de los frameworks JSF este prototipo contiene las funcionalidades básicas de un sistema informático, página de: autenticación, registro de datos, eliminación de la información,

modificación de la información, y de visualización de la información, en el tercer paso se realizó un análisis del rendimiento sobre los prototipos creados utilizando JMeter, el cual permitió realizar 385 peticiones HTTP, con 35 usuarios por segundo, debido a que estos son todos los usuarios involucrados en utilizar el sistema, cabe recalcar que los prototipos mencionados se subieron a un servidor con las siguientes características: sistema operativo CentOS 7.5, Memoria RAM de 4G, Disco Duro SSD de 50G, procesador 2.4GHZ y servidor web Glassfish 5, Glassfish es la implementación de referencia de código abierto de Java EE, también es un proyecto de código abierto bajo la licencia de GPL, este servidor web permite el despliegue de aplicaciones web Java EE (Glassfish, 2019), en el cuarto paso, una vez recogida y almacenada toda la información se procedió a tabular, analizar y representar ilustraciones estadísticas, en el quinto paso, se realizó la prueba de normalidad y de igualdad de varianza, de los dos grupos, la cual al no ser satisfactoria para la aplicación de un T-Student, se hizo uso de la prueba estadística de U de Mann Whitney la cual es una prueba no paramétrica aplicada a dos muestras independientes, en este caso a 385 peticiones HTTP con un nivel de confianza de 95 % y el 5% de error, en el quinto paso se realiza la verificación de la hipótesis planteada y en el sexto paso se emite conclusiones y recomendaciones en base al estudio realizado.

3. RESULTADOS

3.1. ANÁLISIS COMPARATIVO DE FRAMEWORK JSF PRIMEFACES Y BOOTSFACES

Se ha realizado una investigación previa en la cual se comparan los frameworks JSF PrimeFaces en la versión 6.2 y BootsFaces en la versión 1.4 en base a los siguientes parámetros: cantidad de líneas de código, esta información se obtuvo con el programa Cloc versión 1.80, el indicador cantidad de componentes se obtuvo por medio de la página oficial de cada uno de los frameworks JSF, el indicador tamaño de archivo se obtuvo por medio del explorador de archivos de Windows 10, el indicador tiempo

en el mercado se obtuvo por medio de la fecha de lanzamiento de la primera versión hasta la enero del 2019, el indicador número de actualizaciones se obtuvo por medio de las versiones publicadas en el repositorio de cada uno de los frameworks JSF. En la tabla 2, se realiza una comparación de los dos frameworks JSF en base a los parámetros mencionados.

Tabla 2. Comparación de los framework JSF PrimeFaces y BootsFaces. El contenido de esta tabla permite determinar que el Framework BootsFaces posee una menor: cantidad de líneas de código, cantidad de componentes, tamaño de archivo, tiempo en el mercado y número de actualizaciones.

Indicador	Framework JSF	
	BootsFaces	PrimeFaces
Cantidad de líneas de código	89.295	257.843
Cantidad de componentes	79	162
Tamaño de archivo	2,130 MB	6,070 MB
Tiempo en el mercado	31/10/2014	23/2/2009
	4 años	9 años
Número de actualizaciones	31	58

3.2. ANÁLISIS DE LOS INDICADORES: DIMENSIÓN EFICACIA

En la dimensión eficacia se analiza el indicador medio de peticiones HTTP, donde se establece que una petición se realiza correctamente cuando el código de estado de respuesta es superior a 200 (Mozilla, 2017), en la Tabla 3 se puede observar los resultados del indicador mencionado anteriormente.

Tabla 3. Dimensión eficacia, todas las peticiones HTTP realizadas se realizaron correctamente tanto en el framework JSF PrimeFaces y BootsFaces.

Dimensión	Indicador	Framework JSF	
		PrimeFaces	BootsFaces
Eficacia	Media de peticiones HTTP realizadas correctamente	385	385
	Promedio (%)	100%	100%

3.3. ANÁLISIS DE LOS INDICADORES: DIMENSIÓN TIEMPO DE RESPUESTA

Esta dimensión se compone del indicador promedio de tiempo de respuesta, que es el valor que tarda un servidor en responder una petición HTTP (Marchionni, 2011). El tiempo de respuesta del framework JSF BootsFaces fue de 367,904ms y del framework JSF PrimeFaces fue de 277,353ms, la diferencia entre ellos fue de 90,551ms, por lo cual se puede determinar que el framework JSF PrimeFaces ofrece un mejor tiempo de respuesta.

3.4. ANÁLISIS DE LOS INDICADORES: DIMENSIÓN CONSUMO DE RECURSOS

Luego de realizar el análisis de cada uno de los indicadores que conforman esta dimensión se procedió a consolidar los datos, los mismos se detallan en la tabla 4. en esta se puede observar que en el indicador porcentaje promedio de uso de CPU y el porcentaje promedio de Memoria RAM y Disco Duro.

Tabla 4. Dimensión consumo de recursos, se detalla los datos consolidados de cada uno de los indicadores.

Dimensión	Indicador	Framework JSF	
		BootsFaces	PrimeFaces
Consumo de Recursos	Promedio de uso del CPU %	99,133	98,546
	Promedio de uso de Memoria RAM %	19,250	25,340
	Promedio de uso de Disco Duro %	0,026	0,052

Posterior al análisis de los indicadores de acuerdo a las dimensiones, se procedió a consolidar los datos, para lo cual se calcula los promedios de cada uno de las dimensiones, estos datos se detallan en la figura 1 y se puede observar que en la primera dimensión el framework JSF PrimeFaces tiene menor tiempo de respuesta y la diferencia con el framework JSF BootsFaces es de 90,551ms por lo que se puede determinar que el framework JSF PrimeFaces es mejor que su competidor, en la dimensión eficacia se puede determinar que los dos frameworks JSF BootsFaces y PrimeFaces tiene un porcentaje del 100% así que cumple con este indicador, en la dimensión consumo de recursos se puede determinar que el

framework JSF BootsFaces consume un 39,47% y el framework JSF PrimeFaces consume un 41,31% y la diferencia de los dos es de 1,843% por lo cual se determina que el framework JSF BootsFaces consume menos recursos.

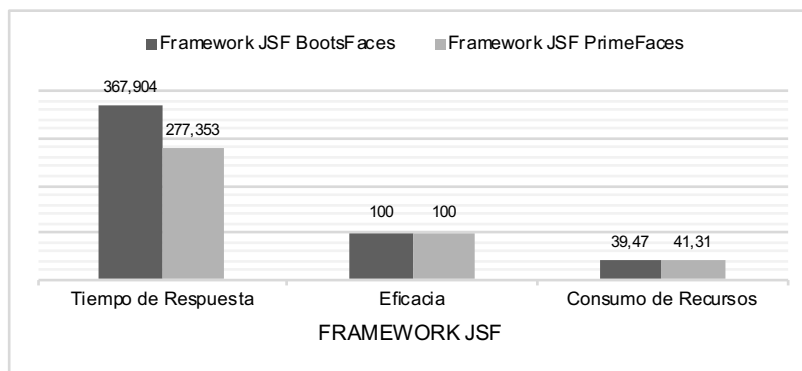


Figura 1. Datos consolidados, se detalla el promedio de cada una de las dimensiones.

3.5. COMPROBACIÓN DE LA HIPÓTESIS

Se realizó una prueba de normalidad de Kolmogorov-Smirnov al indicador promedio de tiempo de respuesta el cual da como resultado que en el prototipo de BootsFaces la Sig. es de 0,2 y en el prototipo de PrimeFaces es de 0,18, debido a que los valores de Sig. son mayores a 0,05 se determina que si cumple con la prueba de normalidad. Sin embargo, la prueba de igualdad de varianzas realizadas al indicador promedio de tiempo de respuesta indica que la Sig. fue de 0,00 el cual no es mayor a 0,05 por ende se determina que no cumple con la igualdad de varianzas.

La dimensión consumo de recursos está compuesta por los siguientes indicadores: promedio de uso de CPU, promedio de uso de la memoria RAM y promedio de uso de disco duro, a estos se realiza la prueba de normalidad de Kolmogorov-Smirnov, en los tres casos se obtiene un nivel de Sig. de 0,0.

En este contexto y al no cumplir con los supuestos de normalidad y de igualdad de varianza de los indicadores mencionados, se aplica la prueba estadística de U Mann Whitney, con un nivel de confianza del 95%, un error del 5% y con una muestra de 385 por cada uno de los grupos los cuales son los frameworks JSF PrimeFaces y BootsFaces se determina que: existe una diferencia significativa del rendimiento de los frameworks JSF PrimeFaces y BootsFaces, por lo cual se procede a aceptar la Hipótesis H_1 y se rechaza la hipótesis H_0 o nula. Por lo cual en cuanto a la hipótesis de la investigación se determina que: La evaluación de los frameworks JSF PrimeFaces y BootsFaces mejora el rendimiento de los sistemas web basados en Java EE, la prueba estadística realizada se detalla en la Tabla 5.

Tabla 5. Comprobación de la hipótesis, la comprobación de la hipótesis se realiza con la prueba estadística U de Mann Whitney debido a que los datos no cumplen con la prueba de normalidad y no cumplen con la igual da varianzas de Levene.

	Estadísticos de prueba				
	Tiempo de respuesta	Uso de Memoria Ram %	Uso del CPU %	Uso de Disco Duro%	Peticiones HTTP realizadas correctamente
U de Mann-Whitney	16 354	0	32 276,5	34 808,5	74 112,5
W de Wilcoxon	90 659	74 305	106 581,5	109 113,5	148 417,5
Z	-18,717	-24,016	-13,836	-12,739	0
Sig. asintótica (bilateral)	0	0	0	0	1
a. Variable de agrupación: FRAMEWORK JSF					

4. DISCUSIÓN Y CONCLUSIONES

4.1. DISCUSIÓN

La calidad de software se implementa para que una tecnología de información se encuentre en lo posible libre de fallas y a su vez proporcione aspectos funcionales coherentes para su aplicación en el desarrollo o industria. Esta investigación midió la diferencia entre el rendimiento de los Frameworks JSF BootsFaces y Primefaces, el factor de rendimiento del modelo FURPS ha sido utilizado satisfactoriamente para medir

la diferencia entre estas dos tecnologías, la utilización de sus métricas así también de su procedimiento a conllevado a determinar los siguientes resultados:

Se acepta la hipótesis del investigador o H_1 que establece que existe una diferencia significativa del rendimiento de los frameworks, debido a que el consumo de recursos del framework JSF BootsFaces es de 39,47% y el de PrimeFaces es de 41,31%, en cuanto a la eficacia se determinó el número de peticiones HTTP que se realizaron correctamente hacia a los prototipos realizados con los frameworks proporcionó como resultado que en los dos casos no existe una diferencia entre los dos promedios el cual fue del 100%, en cuanto al tiempo de respuesta se determina que el framework JSF PrimeFaces tiene un mejor tiempo de respuesta el cual es 367,904ms, mientras que el framework JSF BootsFaces tiene un tiempo de respuesta de 277,353ms. Estos resultados se asemejan a los resultados de los autores (Guamán, Palacios, y Contenido, 2019, Escobar, Rodríguez, Coral, y Hinojosa, 2014), donde se menciona que el tiempo de renderización de los componentes es menos de 1s en el framework JSF PrimeFaces y esto concuerda con el indicador promedio de tiempo de respuesta de la presente investigación, el cual en el framework JSF PrimeFaces fue de 277,353ms lo que es equivalente a 0,277s, por lo cual se puede constatar que a pesar de que en la investigación de (Escobar *et al.*, 2014) se realizó sobre el framework JSF PrimeFaces en su versión 5 y en la presente investigación sobre la versión 6.2, el tiempo de respuesta sigue siendo menor a 1s, también los autores (Escobar *et al.*, 2014) mencionan que el framework JSF PrimeFaces tiene actualizaciones frecuentes, y en esta investigación se determinó que desde su lanzamiento el framework JSF PrimeFaces hasta la fecha de realización de esta investigación ha tenido un total de 58 actualizaciones por lo cual se puede deducir que tiene actualizaciones frecuentes.

existe una diferencia significativa del rendimiento de los frameworks, debido a que el consumo de recursos del framework JSF BootsFaces es de 39,47% y el de PrimeFaces es de 41,31%.

Pero en lo que no concuerda con los autores mencionados anteriormente es en que menciona que el framework JSF PrimeFaces en lo funcional obtuvo una puntuación de 36 sobre 47, debido a que en la presente investigación en la dimensión eficacia se determinó que el framework JSF PrimeFaces fue del 100%, pero cabe recalcar que esta diferencia puede ser causada por las actualizaciones de este framework.

4.2. CONCLUSIONES

Se observa que el consumo de recursos del framework JSF BootsFaces es de 39,47% y el consumo de recursos del framework JSF PrimeFaces es de 41,31%, por tal razón se determina que framework JSF BootsFaces ofrece un mejor rendimiento debido a que consume un 1,843% menos que el framework JSF PrimeFaces.

En cuanto a la eficacia se determinó el número de peticiones HTTP que se realizaron correctamente a los prototipos desarrollados con los frameworks JSF PrimeFaces y BootsFaces, en el resultado se pudo observar que en los dos casos no existe diferencia entre los dos promedios el cual fue del 100%.

En cuanto al tiempo de respuesta se determina que el framework JSF BootsFaces tiene un tiempo de respuesta de 367,904ms, mientras que el framework JSF PrimeFaces es de 277,353ms, en base a estos datos existe una diferencia de 90,551ms, por lo cual se puede determinar que el framework JSF PrimeFaces ofrece un menor tiempo de respuesta.

El factor de rendimiento del modelo FURPS proporciona una base firme metodológica para evaluar las métricas o características de los frameworks JSF BootsFaces y PrimeFaces. De acuerdo a los resultados obtenidos se puede concluir que si un desarrollador se encuentra interesado en el rendimiento de su aplicación web se incline hacia la utilización de PrimeFaces, debido a que existe una diferencia más amplia en la métrica de tiempo de respuesta que en el consumo de recursos y eficacia, características en la cuales las dos tecnologías son prácticamente iguales.

Esta investigación se basó en medir las características de rendimiento del modelo FURPS, en trabajos futuros se comparará a los dos frameworks BootsFaces y PrimeFaces con los demás factores que incluye el mismo los cuales son Funcionalidad, Usabilidad, Confiabilidad y Soporte.

REFERENCIAS BIBLIOGRÁFICAS

Abreu, J. (2012). Hypothesis, Method & Research Design. *International Journal of Good Conscience*, 190-193.

Apache. (2018). Apache JMeter™. Recuperado de <https://jmeter.apache.org/>

Barrueco, J.M., y Subirats, I. (2014). *OAI-PMH: Protocolo para la transmisión de contenidos en Internet*. Recuperado de <http://eprints.rclis.org/4093/1/cardedeu.pdf>

Bergsten H. (2004). *JavaServer Faces: Building Web-based User Interfaces* (3ª ed). California: O'Reilly Media.

Cabrera, L., Bethencourt, J.T., González, M., y Álvarez, P. (2014). Un estudio transversal retrospectivo sobre prolongación y abandono de estudios universitarios. *RELIEVE*, 12(1), 105-127. Recuperado de https://www.uv.es/RELIEVE/v12n1/RELIEVEv12n1_1.pdf

Constanzo, M. A. (2014). Comparación de modelos de calidad, factores y métricas en el ámbito de la ingeniería de software. *Dialnet*, 6, 8-9. Recuperado de <https://dialnet.unirioja.es/descarga/articulo/5123569.pdf>

Escobar, C., Rodríguez S., Coral, H., y Hinojosa, C. (2014). Análisis comparativo de frameworks JSF 2.0: IceFaces, PrimeFaces y RichFaces; para la implementación en el desarrollo del sistema de gestión de proyectos ambientales de la empresa Kaymanta. *ESPE*, 1, 5-6. Recuperado de <https://repositorio.espe.edu.ec/bitstream/21000/8162/1/AC-SI-ESPE-047672.pdf>

- Farrell, M.** (2018). *Would You Use JSF for Your Next Project?* Recuperado de <https://dzone.com/articles/would-you-use-jsf-for-your-next-project>
- Gizas, A., Christodoulou, S., y Papatheodorou, T.** (2012). Comparative Evaluation of Javascript Frameworks. *The 21st International Conference on World Wide Web* (pp. 513-514). Francia. Lyon: ACM.
- Glassfish.** (2019). *The Open Source Java EE Reference Implementation*. Recuperado de <https://javaee.github.io/glassfish/>
- Guamán, J., Palacios, D., y Contenido, S.** (2018). Análisis del rendimiento de librerías de componentes Java Server Faces en el desarrollo de aplicaciones web. *NOVASINERGIA*, 1(2), 54-59. Recuperado de <http://novasinergia.unach.edu.ec/index.php/novasinergia/article/view/54/34>
- IBM.** (2014). *Plataforma Java*. Recuperado de https://www.ibm.com/support/knowledgecenter/es/ssw_ibm_i_71/rzaha/platform.htm
- Kalita, M., Khanikar, S., y Bezboruah, T.** (2011). Investigation on performance testing and evaluation of PReWebN: a java technique for implementing web application. *IET Software*, 5(5), 434-444. doi: <https://doi.org/10.1049/iet-sen.2011.0030>
- Manterola, C., y Otzen, T.** (2015). Estudios Experimentales 2ª Parte: Estudios Cuasi-Experimentales. *International Journal of Morphology*, 33(1), 382-387. Recuperado de <https://scielo.conicyt.cl/pdf/ijmorphol/v33n1/art60.pdf>
- Marchionni, E. A.** (s.f.). *Administrador de servidores*. Buenos Aires: Claretiana.
- MDN web docs Mozilla.** (2017). *Códigos de estado de respuesta HTTP* Recuperado de <https://developer.mozilla.org/es/docs/Web/HTTP/Status>

- Murai, K., y Klyuev, V.** (2016). Comparison of Bootstrap and W3. CSS. *2nd International Conference on Applications in Information Technology* (pp. 163-171). Recuperado de http://web-ext.u-aizu.ac.jp/labs/is-se/conference_proceedings/ica-it-16/ica-it-16-paper-54.pdf
- Nevedrov, D.** (2006). *Using JMeter to Performance Test Web Services*. Recuperado de <https://loadstorm.com/files/Using-JMeter-to-Performance-Test-Web-Services.pdf>
- Oracle.** (2010). *Oracle technology network*. Recuperado de <https://www.oracle.com/technetwork/java/javace/overview-140548.html>
- Oracle.** (2014a) *Java recursos de ayuda*. Recuperado de <https://www.java.com/es/download/faq/techinfo.xml>
- Oracle.** (2014b). *Technology network articles java*. Recuperado de <http://www.oracle.com/technetwork/articles/java/java-primefaces-2191907.html>
- Ordax, J. M., Barón, S.D., y Ocaña, P.A.** (2012). *Programación web en Java*. España: Ministerio de Educación, Cultura y Deporte.
- Pech, F., Gomez M. A., y Lara S.U.** (2011). *Desarrollo de Aplicaciones web con JPA, EJB, JSF y PrimeFaces*, México: Instituto Tecnológico Superior de los Ríos.
- Sánchez, C., Tuesta, V., y Cabrera, I.** (2015). Análisis comparativo de frameworks para el desarrollo de aplicaciones web en java. *Ingeniería: Ciencia, Tecnología e Innovación*, 1(2), 60-72. Recuperado de <http://revistas.uss.edu.pe/index.php/ING/article/view/101/100>
- Solvetic.** (2015). *Solvetic tutoriales*. Recuperado de <https://www.solvetic.com/tutoriales/article/1502-c%C3%B3mo-medir-el-rendimiento-de-una-aplicaci%C3%B3n-web/>
- TIOBE.** (2019). *The Software Quality Company*. Recuperado de <https://www.tiobe.com/tiobe-index/>

- Urzo, D., y Massera, R.** (2019). *BootsFaces the next-gen JSF Framework based on Bootstrap*. Recuperado de <https://www.bootsfaces.net/>
- Yungan, A., y Morales, C.** (2019). *Análisis del rendimiento de framework JSF para el desarrollo del sistema web con enfoque al flujo turístico del cantón Riobamba*. [Tesis pregrado]. Riobamba: UNACH.