Evaluación de un Método de Monitorización de Calidad de Servicios Cloud: Una Replicación Interna[1]

Priscila Cedillo[1, 2], Emilio Insfran[1], Silvia Abrahão[1]

[1] DSIC – Universitat Politècnica de València
Camino de Vera s/n, 46022 Valencia, España
{icedillo, einsfran, sabrahao}@dsic.upv.es
[2] Universidad de Cuenca
Av. 12 de Abril y Av. Loja s/n, 0101168, Cuenca, Ecuador
priscila.cedillo@ucuenca.edu.ec

Resumen.
Contexto: El modelo de negocio que ofrece la computación en la nube tiene un gran número de ventajas tanto para proveedores como para consumidores. Sin embargo, es imprescindible controlar la calidad de los servicios provistos, lo que se puede alcanzar a través de soluciones de monitorización. Sin embargo, se ha prestado poca atención a las percepciones de los usuarios que las utilizan. En un trabajo previo, hemos realizado un cuasi-experimento para evaluar las percepciones de un grupo de estudiantes en el uso de un método de monitorización (Cloud MoS@RT) de calidad de servicios cloud en tiempo de ejecución.
Objetivo: Proporcionar mayor evidencia sobre la facilidad de uso percibida, utilidad percibida e intención de uso de profesionales utilizando Cloud MoS@RT.
Método: Hemos ejecutado una replicación interna del cuasi-experimento base con un grupo de profesionales. La tarea experimental consistió en utilizar Cloud MoS@RT para configurar la monitorización de la calidad de un servicio en la plataforma Microsoft Azure. Los participantes también rellenaron un cuestionario que nos ha permitido evaluar su percepción sobre la utilidad del método.
Resultados: Los resultados indican que los participantes han percibido el método como fácil de usar y útil, y han manifestado su intención de uso futuro.
Conclusiones: Los resultados están alineados con el cuasi-experimento base y confirman que Cloud MoS@RT puede ser utilizado de manera efectiva tanto por estudiantes como profesionales sin la necesidad de un extensivo entrenamiento y conocimiento de la plataforma cloud.

Palabras Clave: Cloud Computing, Software as a Service, Monitorización, Calidad de Servicios, Cuasi-Experimento, Replicación.

1 Introducción

Como una tecnología joven, la computación en la nube y sus herramientas de monitorización de servicios aún sufren la falta de consenso sobre un criterio de evaluación apropiado para determinar su grado de adecuación, calidad, eficiencia o utilidad de

[1] Este trabajo ha sido financiado por el proyecto Value@Cloud (TIN2013-46300-R)

cara a los usuarios. Es deseable tener herramientas de monitorización que puedan ser evaluadas y comparadas de forma que los usuarios y clientes tengan evidencia contrastada para juzgar su calidad y tomar decisiones informadas sobre su adopción. En los últimos años, se han realizado algunos estudios empíricos con el objetivo de evaluar los métodos y las herramientas de monitorización existentes. Estos trabajos reportan experiencias del uso de soluciones de monitorización aunque la mayoría se centran en aspectos muy dependientes de la plataforma y de bajo nivel de abstracción como la eficiencia, latencia o rendimiento [12][14][15].

En trabajos anteriores, hemos propuesto una infraestructura y un método de monitorización de la calidad de servicios de software en la nube mediante modelos en tiempo de ejecución [6][7][9]. El método consta de tres actividades principales: *configuración de la monitorización*, *ejecución de la monitorización*, *análisis de resultados*. El uso de los modelos en tiempo de ejecución nos permite tener una infraestructura de monitorización extensible y dinámica ya que toda la información de la configuración y de los requisitos de monitorización están disponibles como modelos durante la ejecución de la monitorización, pudiendo además ser actualizados, lo que causa la adaptación automática de la infraestructura de monitorización sin necesidad de parar el sistema. Esta adaptación en tiempo de ejecución de la infraestructura de monitorización es una característica fundamental para un entorno tan dinámico como es el cloud. También hemos presentado en un trabajo previo una primera evaluación de la actividad de *configuración de la monitorización* a través de un cuasi-experimento [5] con un grupo de 58 estudiantes del grado en Ingeniería Informática de la Universitat Politècnica de València. Los resultados evidenciaron que los participantes han encontrado el método fácil de usar, útil, y que tendrían intención de usarlo en el futuro. Sin embargo, como todos los participantes eran estudiantes de grado sin experiencia profesional, era necesario contrastar estos resultados con participantes que tengan experiencia profesional y también en otros contextos.

En este trabajo, presentamos una replicación interna del estudio inicial con un grupo de 14 profesionales que han asistido a un máster profesional en Ingeniería del Software de la Universidad Nacional de Asunción (Paraguay). El objetivo de este cuasi-experimento, es por tanto, verificar los resultados obtenidos en el cuasi-experimento base pero con participantes que tienen experiencia profesional y en un contexto distinto. Aunque son necesarias más replicaciones con participantes con experiencia en monitorización de servicios cloud, los resultados obtenidos son positivos y están alineados con los resultados obtenidos en el cuasi-experimento base. Los resultados confirman que Cloud MoS@RT puede ser utilizado de manera efectiva tanto por estudiantes como por profesionales noveles sin la necesidad de un entrenamiento intensivo ni con conocimientos muy profundos en las plataformas cloud.

El artículo está organizado de la siguiente manera: en la Sección 2 se discuten otros estudios empíricos realizados en el ámbito de los métodos y herramientas de monitorización de servicios. En la Sección 3 se presenta una breve descripción del método de monitorización a ser evaluado. En la Sección 4 se describe el cuasi-experimento base. En la Sección 5 se describe la replicación interna del cuasi-experimento. En la Sección 6 se discuten las amenazas a la validez, y finalmente, en la Sección 7 se presentan las conclusiones y los trabajos futuros.

2 Trabajos Relacionados

En los últimos años, se han realizado algunos estudios empíricos para evaluar la monitorización de servicios. El método MoDe4SLA propuesto por Bodenstaff [3] permite el manejo y monitorización de dependencias durante la composición de servicios Web. El método tiene en cuenta si la composición se realiza correctamente y el efecto de la composición en el rendimiento. La evaluación se realizó con un cuasi-experimento con 34 participantes expertos en el desarrollo y manejo de servicios, de los cuales 11 pertenecían a la industria, 9 trabajaban en la industria y en la universidad, y 23 pertenecían únicamente a la universidad. Los autores validaron la utilidad de su método pidiendo a los participantes que realizaran tres composiciones de diferente complejidad a través de simulaciones utilizando MoDe4SLA. Sin embargo, el objetivo del experimento no estaba dirigido solo a monitorizar la calidad de servicios, sino a obtener una buena composición de los mismos y su efecto en el rendimiento.

En el proyecto SLA@SOI [16] se ha propuesto un marco de trabajo para el manejo de los servicios en base a los *Service Level Agreements (*SLAs). En este contexto, se han presentado algunas evaluaciones para ilustrar la aplicabilidad de su solución, que incluye un marco de trabajo para la monitorización de servicios denominado EVEREST, en dominios gubernamentales [1]. Sin embargo, estas evaluaciones presentan pruebas de concepto y sus resultados están centrados en la solución completa sin centrarse en la evaluación del enfoque de monitorización.

En el trabajo de Emeakaroha *et al*. [11] se presentó una arquitectura para la monitorización de servicios llamada CASViD, que tiene como objetivo la detección de violaciones de los SLAs para aplicaciones desplegadas en la nube, su solución fue evaluada haciendo uso de una prueba de concepto. Los autores evaluaron dos aspectos: (i) la habilidad de la arquitectura para monitorizar aplicaciones en tiempo de ejecución para así detectar las violaciones del SLA, y (ii) su capacidad de automáticamente determinar el intervalo de medición apropiado para una monitorización eficiente. Sin embargo, la evaluación se ha centrado en aspectos de rendimiento y no incluye las percepciones de los usuarios al utilizar esta solución por lo que no se tiene evidencia sobre cómo los usuarios percibieron el uso de esta solución de monitorización.

En el trabajo de Meng *et al*. [12] se ejecutaron experimentos por medio de simulaciones de un ambiente cloud con un sistema del mundo real y trazas de red. Los resultados muestran que su enfoque conseguía bajar significativamente los costos, tener una alta escalabilidad, y un mejor rendimiento multitenencia que otros. Sin embargo, su evaluación no incluye ambientes reales ni usuarios para proveer retro-alimentación y contribuir hacia mejorar el enfoque en base a las percepciones de los usuarios.

Finalmente, analizando de forma global estos trabajos, se han identificado limitaciones en cuanto a las evaluaciones empíricas realizadas, tales como (1) la poca cantidad de estudios que tienen en cuenta la experiencia percibida de los usuarios al hacer uso de la solución de monitorización; (2) la carencia de estudios que analicen la interacción de los usuarios con la solución de monitorización para la definición de los requisitos a ser monitorizados (la mayoría se centraba en la efectividad o en el rendimiento), y (3) la falta de réplicas que ayuden a confirmar o refutar los resultados inicialmente obtenidos en diferentes contextos.

3 Monitorización de Calidad de Servicios con Cloud MoS@RT

En trabajos anteriores, hemos propuesto un método de monitorización de calidad de servicios cloud mediante el uso de modelos en tiempo de ejecución (Cloud MoS@RT) [9][6]. El método permite la evaluación de la calidad del servicio y la detección de incumplimientos en acuerdos de nivel de servicios (SLA). Este método está soportado por una infraestructura de monitorización [7] que incluye un *configurador* que permite al usuario configurar los atributos de calidad a ser monitorizados y un *middleware* [8] que recoge la información de los servicios cloud en tiempo de ejecución con el objetivo de determinar su calidad en base al cumplimiento de los requisitos no-funcionales incluidos en el SLA.

El método Cloud MoS@RT consta de tres actividades principales, descritas en la Fig. 1: a) la *Configuración de la Monitorización* que recibe los requisitos no-funcionales a ser monitorizados (del SLA y otros requisitos adicionales de interés) y mediante el uso de un Modelo de Calidad SaaS proporciona guías a los usuarios para que éstos seleccionen los atributos de calidad, métricas y otros parámetros de configuración, que serán incluidos en un modelo de calidad en tiempo de ejecución, b) *Monitorización* que utiliza el modelo de calidad en tiempo de ejecución generado, recolecta los datos de los servicios y opera con ellos, de acuerdo a la configuración establecida y, c) el *Análisis de Resultados* que analiza los datos provistos por la monitorización y evalúa la calidad del servicio y el cumplimiento del SLA.

Esta sección proporciona el contexto para la replicación mediante la descripción de la información clave del experimento base. Seguimos las recomendaciones descritas en [4]. El experimento base [5] describe un cuasi-experimento realizado con el objetivo de evaluar la utilidad de Cloud MoS@RT, teniendo en cuenta la percepción del usuario al momento de realizar la configuración de la monitorización de los servicios (primera actividad del método). La evaluación se centra en la actividad de configuración debido a que es la única que involucra la participación del usuario. Las demás actividades (monitorización y análisis de resultados) se realizan de forma automática por el middleware de monitorización una vez que el modelo en tiempo de ejecución haya sido generado.

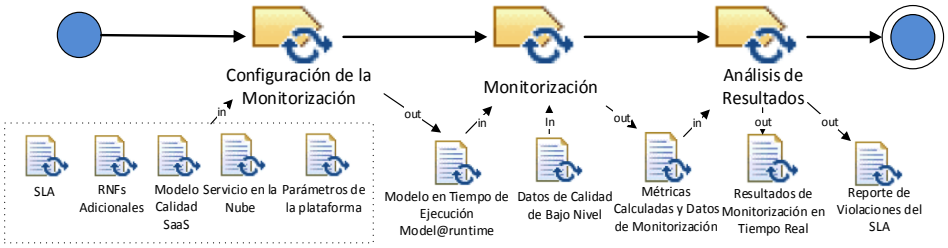

Fig. 1. Método Cloud MoS@RT

4 Cuasi-experimento Base

El estudio empírico base es un cuasi-experimento. Los cuasi-experimentos se realizan cuando los participantes no pueden asignarse aleatoriamente a una condición experimental o, alternativamente, existe la falta de un grupo de control. Este es nuestro caso, debido a la falta de un grupo de control, ya que no existe actualmente un método o infraestructura de monitorización para servicios cloud que pueda ser considerada estándar o ampliamente aceptada en la industria. El cuasi-experimento ha sido diseñado siguiendo los pasos del proceso experimental en Ingeniería del Software propuesto por Wohlin *et al.* [17].

4.1 Preguntas de investigación e hipótesis

El objetivo principal del cuasi-experimento base (UPV1) consiste en evaluar la fase de configuración del método Cloud MoS@RT del punto de vista de sus usuarios. De acuerdo al paradigma *Goal-Question Metric (GQM)* [2] el objetivo del experimento se puede definir de la siguiente manera:

Tabla 1. Objetivo del experimento base

Analizar	la fase de configuración de Cloud MoS@RT
Con el propósito de	Evaluarla
Con respecto a	la facilidad de uso percibida, utilidad percibida e intención de uso
Desde el punto de vista de	evaluadores noveles que utilizan el método de monitorización para evaluar servicios cloud en plataformas específicas y del investigador interesado en los resultados de evaluación
En el contexto de	un grupo de estudiantes en Ingeniería del Software de la Universitat Politècnica de València

En el ámbito de la Ingeniería del Software es importante tener en cuenta el rol de las personas en el proceso de aceptación y adopción de una solución tecnológica. Esto ha sido estudiado en el campo de las Ciencias Sociales a través del desarrollo de modelos teóricos que explican los factores que afectan la aceptación de tecnologías, metodologías y métodos. En este campo, los modelos actuales de aceptación de tecnología tienen sus raíces en varios modelos teóricos que incorporan constructores para medir las reacciones psicológicas del usuario y factores organizacionales de una manera sistemática. Entre estos modelos están el *Technology Acceptance Model (TAM)* propuesto por Davis [10], un modelo teórico ampliamente usado desde la perspectiva del uso general de un sistema, y el *Method Evaluation Model (MEM)* propuesto por Moody [13] que provee mecanismos para evaluar el rendimiento actual, la aceptación y la posible adopción de un método de sistemas de información en la práctica. El experimento base ha utilizado el TAM para plantear las preguntas de investigación a ser estudiadas:

- RQ1: ¿Es Cloud MoS@RT percibido como fácil de usar y útil?
- RQ2: ¿Existe una intención de uso de Cloud MoS@RT en el futuro?

Estas preguntas de investigación pueden ser evaluadas a través de la prueba de varias hipótesis. En particular, la primera pregunta de investigación puede ser estudiada mediante las siguientes hipótesis:

- $H1_0$: Cloud MoS@RT es percibido como difícil de usar, $H1_0=\neg H1_1$.
- $H2_0$: Cloud MoS@RT no es percibido como un método útil, $H2_0=\neg H2_1$

Por otra parte, la segunda pregunta de investigación puede ser estudiada a través de la formulación de la siguiente hipótesis:

- $H3_0$: No existe intención de utilizar Cloud MoS@RT en el futuro $H3_0=\neg H3_1$.

4.2 Selección del Contexto

El contexto está determinado por el método de monitorización, la selección del servicio cloud a ser evaluado (objeto experimental) y la selección de los participantes.

El método de monitorización a ser evaluado es Cloud MoS@RT. Como se ha mencionado anteriormente, nos centraremos en la actividad de configuración de la monitorización, la cual genera el Modelo de Calidad en Tiempo de Ejecución. Esta actividad es relevante debido a la necesidad de la interacción del usuario con la infraestructura de monitorización. Los participantes ejecutarán el rol del Configurador de la Monitorización (ver Fig. 2) el cual incluye las siguientes tareas: (i) selección de los atributos de calidad, (ii) selección de métricas independientes de la plataforma, (iii) mapeo de métricas a los contadores de la plataforma cloud, y (iv) generación del modelo de calidad en tiempo de ejecución.

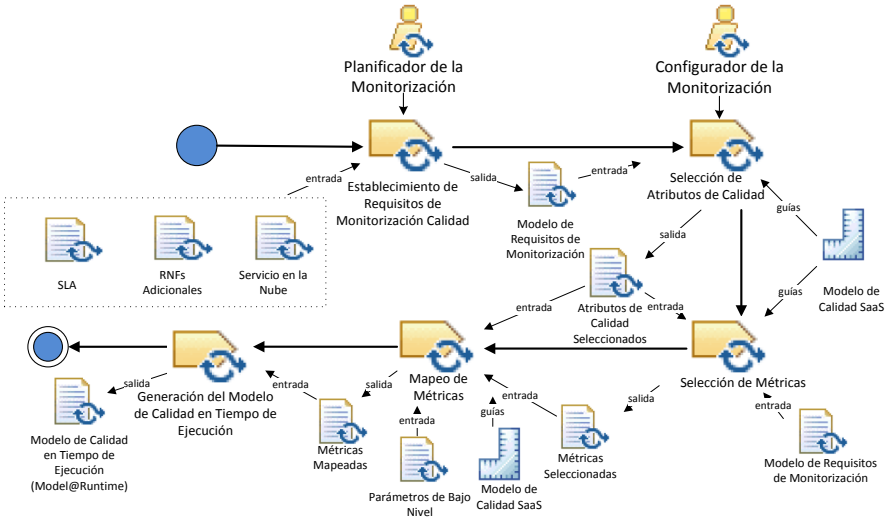

Fig. 2. Configuración de la Monitorización en Cloud MoS@RT

Los participantes recibieron un documento con el *Modelo de Requisitos de Monitorización*, el cual incluye los requisitos no funcionales (RNF) a ser monitorizados. Además, los participantes utilizaron un *Modelo de Calidad SaaS* que describe las

características, sub-características, atributos y métricas de calidad de los servicios, con la finalidad de guiarles en la selección de los atributos y métricas a ser incluidos en el modelo en tiempo de ejecución de una manera sistemática. Una vez realizada la configuración de la monitorización, el *Modelo de Calidad en Tiempo de Ejecución* es generado y usado para monitorizar la calidad de los servicios.

El **servicio cloud** a ser evaluado pertenece a un sitio de subastas en línea. Los altos niveles de calidad que sus servicios demandan (p. e. disponibilidad, fiabilidad, baja latencia) proporcionan un interesante problema a ser resuelto. La configuración de la monitorización ha consistido en la configuración de tres requisitos no funcionales. El Modelo de Requisitos de Monitorización realizado por el *Planificador de la Monitorización* especifica los requisitos y las métricas a ser consideradas por los participantes. Por ejemplo, el servicio debería tener un máximo de 10 operaciones defectuosas por millón, lo cual representa una fiabilidad de 99.999% y puede ser medida utilizando la siguiente función de medición:

$$DPM = \frac{Operaciones\ Intentadas - Operaciones\ exitosas}{Operaciones\ intentadas} * 10^6$$

El problema está basado en un *Software as a Service (SaaS)* real que ha sido desplegado en la plataforma Microsoft Azure. Finalmente, hemos usado muestreo por conveniencia para seleccionar los participantes del experimento. Los **participantes** fueron 58 estudiantes del Grado en Ingeniería Informática (rama de Ingeniería del Software) de la Universitat Politècnica de València, matriculados en la asignatura de Calidad de Software. Se ha tomado una muestra consistente en dos grupos de la asignatura (mañana y tarde) de 37 y 21 participantes respectivamente. Los estudiantes tienen conocimientos previos sobre calidad de productos, incluido modelos de calidad, métricas y métodos de evaluación. Las tareas experimentales se organizaron como parte obligatoria del curso.

4.3 Variables

La variable independiente es el método que se evalúa con un tratamiento (Cloud MoS@RT). La Tabla 2 muestra las variables dependientes de interés basadas en la percepción de los usuarios, de acuerdo al modelo *Technology Acceptance Model* (TAM) [10], las cuales fueron usadas para evaluar Cloud MoS@RT en la práctica.

Tabla 2. Variables dependientes

Variable	Descripción
Facilidad de Uso Percibida (PEOU)	El grado en el cual los participantes creen que será fácil aprender y usar Cloud MoS@RT.
Utilidad Percibida (PU)	El grado en el cual los participantes creen que usando Cloud MoS@RT se incrementará su rendimiento.
Intención de Uso (ITU)	El grado en que los participantes piensan usar el método en el futuro en caso de necesitar monitorizar la calidad de servicios en la nube. Esta variable representa un juicio de la eficacia del método y puede ser utilizada para predecir la posible aceptación del método en práctica (mediante la definición de modelos de regresión).

Estas variables fueron medidas utilizando un cuestionario on-line con una escala de Likert de 1 a 5, con el formato de preguntas opuestas. El cuestionario está compuesto de 14 preguntas cerradas (5 ítems para medir PEOU, 6 ítems para medir PU y 3 ítems para medir ITU). Varios ítems pertenecientes al mismo grupo de variables fueron aleatorizados para prevenir errores de respuesta sistemática. Para asegurar el balance de los ítems, aproximadamente la mitad de las preguntas fueron negadas, para evitar respuestas monótonas. El cuestionario también incluía 3 preguntas abiertas para recabar la opinión de los participantes. El instrumento de medición se encuentra disponible en: *https://goo.gl/ 9mwiwA*. El valor agregado para cada variable subjetiva fue calculado como la media aritmética de las respuestas a las preguntas asociadas a cada variable dependiente.

4.4 Diseño, operación y ejecución

Al tratarse de un cuasi-experimento no realizamos una asignación de los participantes a métodos y ejercicios diferentes. Es decir, todos los sujetos que participaron en el experimento realizaron el mismo ejercicio utilizando Cloud MoS@RT para evaluar el servicio cloud de la aplicación de subastas en línea.

Hemos realizado una sesión de entrenamiento de 120 minutos antes de la sesión experimental con el objetivo de presentar los conceptos de computación en la nube y el método de monitorización a los participantes, sin embargo se proporciona también una guía del método para que se pueda prescindir de esta sesión. El entrenamiento incluyó el uso del prototipo del Configurador de la Monitorización y las tareas envueltas en el proceso de configuración. Como un ejemplo en la sesión de entrenamiento se utilizó el Caso de Referencia Abierta (ORC) el cual fue utilizado dentro del proyecto europeo SLA@SOI [16]. El ORC es una extensión de la implementación de CoCoMe, el cual provee una solución orientada a servicios de un supermercado para manejar las ventas y los procesos de inventario. El conjunto de servicios fue desplegado como SaaS sobre la plataforma Microsoft Azure. En el entrenamiento, los participantes usaron el configurador para la configuración de dos requisitos no funcionales (eficiencia y precisión del servicio) para el servicio de Inventarios. Todo el material de entrenamiento y el material experimental está disponible en: goo.gl/nmQYiL.

El experimento fue realizado en un aula de la UPV. La ejecución fue controlada para evitar interacciones entre los participantes. El experimento fue conducido en dos sesiones (una sesión con el grupo de la mañana y otra con el de la tarde). Cada sesión tuvo una duración de 120 minutos. Sin embargo, se permitió a los participantes finalizar el experimento incluso si el tiempo llegaba a su fin con el objetivo de mitigar el efecto techo. El experimentador ha resuelto las dudas lo largo de las sesiones experimentales. Después de realizar las tareas experimentales, los participantes rellenaron un cuestionario que nos ha permitido evaluar las variables del experimento.

Para el análisis de los resultados, se usaron estadística descriptiva, test estadísticos para la prueba de hipótesis y gráficos de densidad para analizar los datos recogidos. Debido a que ambos grupos de participantes tenían el mismo perfil, combinamos los datos en un solo grupo. Los datos fueron analizados de acuerdo a las hipótesis establecidas. Los resultados fueron obtenidos usando SPSS v20, con un $\alpha = 0.05$.

4.5 Resumen de los resultados

La Fig. 3 muestra los gráficos de densidad para cada variable de percepción en las cuales podemos ver que la media para cada variable es mayor que el valor neutro de la escala de Likert, que es el 3.

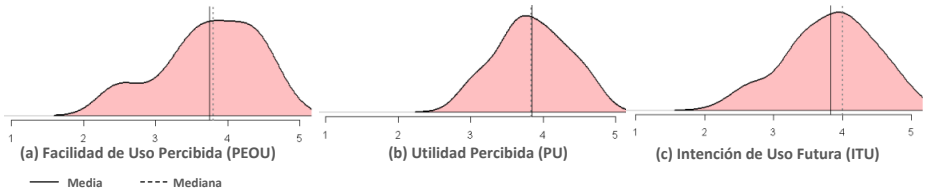

Fig. 3. Variables de Percepción. Cuasi-experimento UPV

La Tabla 3 muestra las estadísticas descriptivas y los resultados de la prueba Shapiro-Wilk para analizar la distribución de las variables estudiadas. La media y la desviación estándar se utilizaron como estadística descriptiva también para las variables PEOU, PU y ITU, siendo la escala de Likert de cinco puntos adoptada para su medición una escala de intervalo. Se han aplicado pruebas para verificar las hipótesis comparando si la media de las respuestas a las preguntas relacionadas con una variable dada, fueron significativamente más altas que el valor neutro de la escala de Likert.

Tabla 3. Prueba Shapiro-Wilk para PEOU, PU e ITU. Cuasi-experimento base UPV1

Var	Min	Max	Media	Desv. Std.	E. Std.	prueba t unilateral/p-value	Shapiro-Wilk prueba p-value
PEOU	2.00	4.80	3.710	0.6978	0.09496	<0.001**	0.012*
PU	2.83	4.83	3.833	0.5005	0.06811	<0.001	0.302
ITU	2.33	5.00	3.809	0.6364	0.08661	<0.001	0.102

*La variable no corresponde a una distribución normal. **Resultados de la prueba de Wilcoxon

Para las variables PU e ITU, las cuales los datos tienen una distribución normal ($p>0.05$), se ha probado la hipótesis aplicando el t-test one-tailed mientras que para la variable PEOU, donde los datos no siguen una distribución normal ($p<0.05$), se ha aplicado el test Wilcoxon one-tailed one-sample con un valor de prueba igual a 3, ya que este valor corresponde al valor neutral de la escala de Likert del cuestionario. Los resultados de estos test nos han permitido rechazar las hipótesis nulas $H1_0$, $H2_0$ y $H3_0$, lo que significa que los participantes han percibido que el método Cloud MoS@RT es fácil de usar, útil, y también han expresado su intención de utilizar este método en el futuro si tuvieran que monitorizar la calidad de servicios cloud.

5 Replicación Interna

El objetivo de la replicación es comprobar los resultados del experimento base en un contexto distinto y ampliar la validez de los resultados. Una motivación adicional es que el tamaño muestral del experimento base puede afectar a la magnitud del efecto,

por lo que es conveniente ejecutar replicaciones para aumentar el tamaño de la muestra y así mejorar la efectividad de la confirmación de las hipótesis del experimento.

5.1 Cambios en el diseño inicial

La replicación se llevó a cabo en otro contexto pero bajo las mismas condiciones experimentales (replicación exacta). Al igual que en el cuasi-experimento original, después de las sesiones de entrenamiento, su ejecución se realizó según lo previsto. La ejecución fue controlada para evitar interacciones entre los participantes. El experimento fue realizado en una sesión de 120 minutos de duración. Sin embargo, se permitió a los participantes finalizar el experimento incluso si el tiempo llegaba a su fin con el objetivo de mitigar el efecto techo. La persona encargada de conducir el experimento ha esclarecido cualquier pregunta a lo largo de la sesión experimental. Al igual que el cuasi-experimento base los estudiantes han realizado tareas de configuración en base a unos requisitos (incluida una tarea de modificación de los parámetros de monitorización) y han rellenado un cuestionario de evaluación.

5.2 Muestra y participantes

Los participantes son un grupo de 14 profesionales, con una experiencia media de 3 años en el desarrollo de software, que participaban en un Máster Profesional en Ingeniería del Software en la Universidad Nacional de Asunción (UNA) en Paraguay. Los participantes asistieron un curso sobre Calidad y Métricas del Software y el experimento fue organizado como parte del curso.

5.3 Análisis de las Percepciones de Usuario

Para el análisis de los resultados, se usaron estadística descriptiva, test de prueba de hipótesis y gráficos de densidad para analizar los datos obtenidos. Los datos fueron analizados de acuerdo a las hipótesis establecidas en el cuasi-experimento base. Los resultados obtenidos usando SPSS v20 tienen un $\alpha = 0.05$.

La Fig. 4 muestra los gráficos de densidad para cada variable (PEOU, PU e ITU) en las cuales podemos ver que la media para cada variable es mayor que el valor neutral de la escala de Likert (valor = 3).

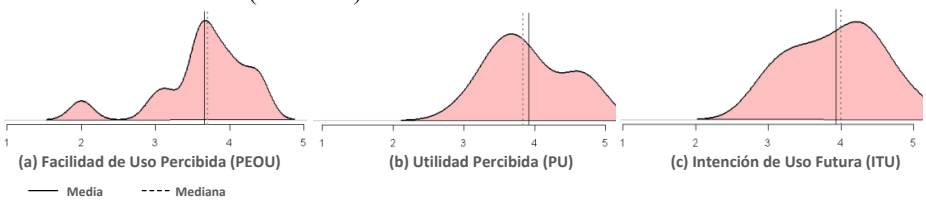

Fig. 4. Variables de Percepción. Cuasi-experimento UNA

A continuación, se ha aplicado la prueba de Shapiro-Wilk para comprobar si los datos estaban normalmente distribuidos para seleccionar el test que podría usarse para

verificar las hipótesis H1, H2 y H3. La Tabla 4 muestra los resultados de la prueba Shapiro-Wilk para las variables estudiadas. Se han aplicado pruebas para verificar las hipótesis comparando si la media de las respuestas a las preguntas relacionadas con una variable dada, fueron significativamente más altas que el valor neutral de la escala de Likert. Las variables PEOU, PU e ITU han presentado una distribución normal ($p>0.05$) por lo que se ha probado la hipótesis aplicando el t-test. Estos resultados nos han permitido rechazar las hipótesis nulas $H1_0$, $H2_0$ y $H3_0$, lo que significa que los participantes perciben que Cloud MoS@RT es fácil de usar, útil y tendrían la intención de usarlo en el futuro en caso de necesitar monitorizar servicios cloud.

Tabla 4. Prueba Shapiro-Wilk para PEOU, PU e ITU. Replicación UNA

Var	Min	Max	Media	Desv. Std.	E. Std.	prueba t unilateral/p-value	Shapiro-Wilk pruebap-value
PEOU	2.40	4.20	3.3143	0.5067	0.13541	0.037	0.725
PU	2.67	4.67	3.9286	0.5536	0.14796	0.000	0.069
ITU	2.67	5.00	4.1429	0.7814	0.20882	0.000	0.107

5.4 Discusión y Comparación

Los resultados obtenidos confirman los resultados del cuasi-experimento base. Se han aplicado el test de Shapiro-Wilk para verificar si los datos tenían una distribución normal en ambos experimentos. Para determinar si los resultados fueron o no significativos se observó el p-value, en donde, mientras en el cuasi-experimento UPV1 se rechazaron todas las hipótesis nulas con una significancia muy alta[2], en el cuasi-experimento UNA se rechazó la hipótesis nula $H1_0$ con una significancia media y las hipótesis nulas $H2_0$ y $H3_0$ con una significancia muy alta.

Esta diferencia probablemente puede ser debido al tamaño de la muestra, más reducida en la replicación. Por tanto, teniendo en cuenta los contextos en los que se ha aplicado el método Cloud MoS@RT, podemos concluir que independientemente del grado de experiencia de los participantes, el método se percibe como fácil de usar, útil y los participantes han manifestado su intención de uso en el futuro. Esto probablemente se debe a que el método proporciona guías claras que ayudan a los usuarios a realizar la configuración de la monitorización de una manera sistemática y de alto nivel sin que tengan que ser expertos en conocer los distintos mecanismos de extracción de información de los servicios cloud y/o los parámetros y contadores de las plataformas que permitan medir los atributos de calidad de interés. Esto se ha reflejado en las respuestas de los participantes a las preguntas abiertas del cuestionario: *"Creo que es un método bastante sencillo o el ejemplo estaba muy bien guiado", "El método en si no es muy complicado, me costó un poco entenderlo por los atributos y características de calidad, algunas de ellas no tan obvias y fáciles de categorizar, por lo que tuve que recurrir a los anexos"*. Sin embargo, como trabajo futuro, pretende-

[2] Para el análisis de datos, hemos utilizado los siguientes niveles de significancia sugeridos por Moody [13]: no significativo=$p>0.1$; baja significancia=$p<0.1$; media significancia=$p<0.05$; alta significancia=$p<0.01$; muy alta significancia=$p<0.001$)

mos estudiar empíricamente el efecto que la experiencia y la habilidad de los participantes tienen sobre los resultados del experimento.

6 Amenazas a la validez

Las *amenazas a la validez interna* son relevantes en los estudios que intentan establecer relaciones causales. Las principales amenazas a la validez interna fueron: la experiencia de los participantes, los sesgos del autor y los sesgos relacionados al método de monitorización y herramienta que lo soporta.

Para reducir la amenaza relacionada con la experiencia de los participantes, nosotros preparamos un ejemplo de entrenamiento representativo, el cual muestra cada paso del proceso y provee a los usuarios un alto entendimiento tanto de la monitorización de servicios cloud y cómo monitorizar la calidad de los servicios utilizando nuestro método. Tanto los sesgos del autor como las producidas por la entendibilidad del material fueron reducidas durante la ejecución de un experimento piloto, en el cual un grupo de investigadores expertos en el área evaluaron el material experimental para así reducir posibles errores o malos entendidos relacionados con el experimento. Los sesgos con respecto a la herramienta de monitorización fueron reducidos a través de su validación en el experimento piloto y pruebas sucesivas para mejorar la usabilidad de la herramienta.

Las *amenazas a la validez externa* se refieren a la habilidad para generalizar los resultados en diferentes contextos. La principal amenaza a la validez externa es la representatividad de los resultados que puede verse afectada por el diseño de la evaluación, el contexto de participantes seleccionados y el tamaño y complejidad de las tareas experimentales. El diseño de la evaluación puede tener un impacto en la generalización de los resultados debido a la complejidad de la plataforma cloud, sus características particulares, las herramientas usadas para recuperar los datos y los RNF a ser considerados. Nosotros intentamos reducir este problema seleccionando una plataforma popular y comúnmente utilizada, la cual comparta conceptos con otras plataformas, y considerando un escenario muy común desde el cual recoger datos crudos. Además, nosotros hemos seleccionado los RNF, los cuales son representativos para servicios cloud. Con respecto a la experiencia de los participantes, el cuasi-experimento fue conducido con alumnos de Ciencias de la Computación, Ingeniería de Sistemas y alumnos de máster en Ciencias de la Computación, quienes además en los cuatro casos asistieron al curso de Calidad de Software y tienen un buen conocimiento de modelos de calidad y métricas. Además, ellos fueron entrenados en el uso de nuestro enfoque y herramienta durante una cantidad razonable de tiempo. Será, sin embargo, necesario ejecutar futuros experimentos con participantes de la industria. El tamaño y complejidad de las tareas podría además afectar la validez externa. Para ello, hemos propuesto un conjunto de tareas experimentales con un nivel suficiente de complejidad, dado el tiempo que se tenía para las sesiones.

La principal *amenaza a la validez de constructo* es la confiabilidad del cuestionario. Para analizar la confiabilidad del cuestionario, un análisis de la prueba de confiabilidad del alfa de Cronbach fue realizado para cada conjunto de preguntas relaciona-

das a cada variable subjetiva. Esas fueron mayores al umbral mínimo aceptado α= 0.70, donde el α Cronbach de la facilidad de uso percibida es 0.834, utilidad percibida es 0.776 e intención de uso es 0.820.

Finalmente, las *amenazas a la validez de conclusión,* se refieren a las conclusiones estadísticas. Ejemplos de estas son la elección de los métodos estadísticos, y la elección del tamaño de la muestra, entre otros. Uno de los principales problemas de validez son los tamaños de la muestra en la réplica del cuasi-experimento. Para controlar el riesgo de la variación debido a diferencias individuales que se pueden hacer mayores debido al tratamiento, hemos seleccionado un grupo homogéneo de participantes. Y en cuanto a la recogida de datos, se aplicó el mismo procedimiento en cada experimento individual con el fin de extraer los datos, y se aseguró que cada variable dependiente se calculara mediante la misma función de medición.

7 Conclusiones y Trabajo Futuro

Este artículo ha presentado los resultados de una replicación interna para verificar los resultados obtenidos en el cuasi-experimento base pero con participantes que tienen experiencia profesional y en un contexto diferente. Los resultados obtenidos están alineados con los resultados del cuasi-experimento base, confirmando que Cloud MoS@RT puede ser utilizado de manera efectiva tanto por estudiantes como por profesionales noveles sin la necesidad de un entrenamiento intensivo ni con conocimientos muy expertos en las plataformas cloud. En cualquier caso, estos resultados tienen que ser tomados con cautela, y más réplicas internas y externas son necesarias.

Los trabajos futuros pretenden abordar algunas de las limitaciones identificadas para este estudio, por ejemplo, en cuanto al grado de experiencia de los participantes, que radica principalmente en que, si bien todos ellos son profesionales de la industria con experiencia en el desarrollo de software, no trabajan cotidianamente con servicios desplegados en la nube, por lo cual sería importante replicar el experimento con expertos en tecnologías cloud y con experiencia en la monitorización de servicios.

Otra limitación de la evaluación realizada es que ha sido centrada solo en la fase de *configuración de la monitorización* del método Cloud MoS@RT, que es donde se definen/modifican los requisitos de monitorización. También será necesario evaluar las demás fases del método (ejecución de la monitorización en múltiples plataformas y análisis de resultados) con el fin de validar la aproximación de monitorización completa y abordar aspectos de rendimiento, cambios en los requisitos de monitorización en tiempo real, sobrecarga, etc. También es necesario replicar los experimentos realizados considerando objetos experimentales con más atributos y métricas de calidad tanto de alto como de bajo nivel, que permitan determinar posibles limitaciones de la solución, empleando SLA complejos y más representativos. Finalmente, es necesario establecer evaluaciones para servicios más complejos, en entornos multi-cloud, que necesiten interacción con otros servicios y que constituyan casos reales en la industria, así como también realizar experimentación de esta solución frente a otras soluciones alternativas.

Referencias

1. G. Armellin, A. Chiasera, G. Frankova, L. Pasquale, F. Torelli, and G. Zacco, The eGovernment Use Case Scenario SLA Management Automation of Public Services. In *Service Leveal Agreenments for Cloud Computing*, Springer Science Business Media, pp. 343 – 357, 2011.
2. V. R., Basili, H. D. Rombach, The Tame Project - Towards Improvement-Oriented Software Environments. TSE 14(6), 758-773 (1988).
3. L. Bodenstaff, A. Wombacher, and M. Reichert, Empirical Validation of MoDe4SLA; Approach for Managing Service Compositions. In *14th International Conference on Business Inofrmation Systems* Poznan, Poland, 98–110, 2011.
4. J. C. Carver. Towards Reporting Guidelines for Experimental Replications: A Proposal. Proceedings of the 1st International Workshop on Replication in Software Engineering (RESER), 2010.
5. P. Cedillo, E. Insfran, J. Gonzalez-Huerta, and S. Abrahão, Design and Evaluation of a Monitoring Infrastructure for Cloud Services, *Journal of Systems and Software* (*under review*), 2017.
6. P. Cedillo., Monitorización de calidad de servicios cloud mediante modelos en tiempo de ejecución. Tesis Doctoral, Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, 2016.
7. P. Cedillo, J. Gonzalez-Huerta, S. Abrahao, and E. Insfrán, A Monitoring Infrastructure for the Quality of Cloud Services. In *24th International Conference on Information Systems Development (ISD 2015),* Harbin, China, 2015.
8. P. Cedillo, J. Jimenez-Gomez, S. Abrahao, and E. Insfran, Towards a Monitoring Middleware for Cloud Services. In *12th IEEE International Conference on Services Computing (SCC 2015*), New York, USA, IEEE Computer Society, pp. 451-458, 2015.
9. P. Cedillo, J. Gonzalez-Huerta, E. Insfrán, and S. Abrahao, Towards Monitoring Cloud Services Using Models@run time. In S. Götz, N. Bencomo, R. B. France (Eds.): Proceedings of the *9th International Workshop on Models@run.time (MRT 2014)*, collocated with MODELS 2014, Valencia, Spain, pp. 31–40, 2014.
10. F. D. Davis, R. P. Bagozzi, and P. R. Warshaw., Technology Acceptance Model for Empirically Testing New End-user Information Systems: Theory and Results. Massachusetts Institute of Technology, Sloan School of Management, 1989.
11. V. C. Emeakaroha, T. C. Ferreto, M. A. S. Netto, I. Brandic, and C. A. F. De Rose, CASViD: Application Level Monitoring for SLA Violation Detection in Clouds. In *36th Computer Software and Applications Conference (COMPSAC)*, Turkey, 499–508, 2012.
12. S. Meng and L. Liu, Enhanced monitoring-as-a-service for effective cloud management. *IEEE Transactions on Computers* 62, 9: 1705–1720, 2013.
13. D. L. Moody, *A Practical Method for Representing Large Entity Relationship Models, P.h.D. Thesis*. University of Melbourne, Australia (2001)
14. J. Montes, A. Sánchez, B. Memishi, M. S. Pérez, and G. Antoniu. GMonE: A complete approach to cloud monitoring, *Future Generation Computer Systems* 29(8): 2026–2040, 2013.
15. J. Povedano-Molina, J. M. Lopez-Vega, J. M. Lopez-Soler, A. Corradi, and L. Foschini, DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant Clouds. *Future Generation Computer Systems* 29, 8: 2041–2056, 2013.
16. W. Theilmann. SLA@SOI: Empowering the Service Economy with SLA-aware Infrastructures, Seventh Framework Program. Information Society and Media, URL: www.sla-at-soi.eu, pp. 64–65 (2011).
17. C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Springer Heidelberg New York Dordrecht London, 2012.