

Samuel David Roncal Vidal

9042__1734456467ART02.docx

 My Files My Files Universidad Peruana Union

Detalles del documento

Identificador de la entrega

trn:oid:::29566:420053023

Fecha de entrega

9 ene 2025, 4:48 p.m. GMT-5

Fecha de descarga

9 ene 2025, 4:58 p.m. GMT-5

Nombre de archivo

9042__1734456467ART02.docx

Tamaño de archivo

527.1 KB

11 Páginas

5,946 Palabras

33,265 Caracteres

2% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Trabajos entregados
- Base de datos de Crossref
- Base de datos de contenido publicado de Crossref

Exclusiones

- N.º de fuente excluida
- N.º de coincidencias excluidas

Fuentes principales

- 2%  Fuentes de Internet
- 0%  Publicaciones
- 0%  Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alertas de integridad para revisión

No se han detectado manipulaciones de texto sospechosas.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Fuentes principales

- 2% Fuentes de Internet
- 0% Publicaciones
- 0% Trabajos entregados (trabajos del estudiante)

Fuentes principales

Las fuentes con el mayor número de coincidencias dentro de la entrega. Las fuentes superpuestas no se mostrarán.

1	Internet	www.coursehero.com	0%
2	Internet	lirias.kuleuven.be	0%
3	Internet	upc.aws.openrepository.com	0%
4	Internet	repositorio.ucv.edu.pe	0%
5	Internet	biostats.bepress.com	0%
6	Internet	elib.dlr.de	0%
7	Internet	repositorio.unap.edu.pe	0%
8	Internet	www.redhat.com	0%
9	Internet	ddd.uab.cat	0%
10	Internet	elib.uni-stuttgart.de	0%
11	Internet	www.rediris.es	0%

12

Internet

www.slideshare.net	0%
--------------------	----

13

Internet

www.certima.net	0%
-----------------	----

Marco de trabajo para la mejora del desempeño de servicios en aplicaciones web apoyado en protocolos de gestión de mensajes basado en colas

Samuel D. Roncal Vidal¹, Omar L. Loaiza², Fernando Asín³, Nemias Saboya

¹First affiliation, Address, City, Country (e-mail)

²Second affiliation, Address, City, Country (e-mail)

³Third affiliation, Address, City, Country (e-mail)

Abstract – El Marco de trabajo para la mejora del desempeño de aplicaciones web apoyado en protocolos de gestión de mensajes basado en colas, permite mejorar la disponibilidad de la información, la tasa de error y el tiempo de respuesta de la aplicación web, el marco de trabajo es escalable, flexible y de uso factible en organizaciones que desarrollan sus propias aplicaciones web. El protocolo de colas AMQP permite integrar soluciones indistintamente del lenguaje de programación, definir consumidores escalables, reconocer mensajes fuera de línea y manejo de errores.

Keywords – Framework, service performance improvement, AMQP Model

1. Introduction

El mundo es cada vez más competitivo en el intercambio de productos y servicios en todos los sectores de la economía, según la OMC (OMC, 2019) se prevé que el volumen del comercio mundial de mercancías aumente un 4,0% en 2022, en lo correspondiente al sector de los servicios (educación, consultorías, etc.) podría aumentar en un 50%. Por su parte, la UNCTAD (UNCTAD, 2020) prevé un aumento del PIB mundial del 2.0% para la agenda del 2030. La dinámica descrita suscita una gran interacción de actividades entre diversos agentes, con un caudal de información gigantesco, fruto de la necesidad de cumplir con normas derivados de tratados y convenios multilaterales.

Los datos necesarios para la gestión de actividades empresariales en productos y servicios entre diferentes agentes, con manejados por sistemas de información y una infraestructura de equipos de TI que sirven para almacenar gran cantidad de datos y procesarlos (Longatt, 2017). Hoy en día la gran parte de sistemas de información se basan en el uso de internet y sus protocolos, cuestión que se ha convertido en recurrente. Es así como, la interacción de un cliente con una organización se efectúa a través de aplicaciones web, escritorio o móviles. Estas

interacciones generan solicitudes enviadas por el protocolo HTTP hacia los servidores de la organización. El desempeño de los servicios de una aplicación web obtiene una merma cuando la demanda aumenta, o las reglas que gobiernan el acceso restrictivas, o los servidores llegan a su capacidad máxima. Una de las arquitecturas que genera mayor merma en el desempeño de los servicios es la arquitectura monolítica, una aplicación monolítica cuenta con un solo repositorio de código base y proporciona servicios que utilizan diferentes clientes. (Mario Villamizar, 2015)

El problema que aborda esta investigación tiene que ver con el bajo desempeño de los servicios de aplicaciones web no monolíticas; esto se refleja en la baja disponibilidad de los servicios REST, alta tasa de error y alto tiempo de respuesta. De acuerdo con X. J. Hong, H. Sik Yang y Y.H. Kim (X. J. Hong, 2018) el desempeño de servicios en aplicaciones web tiene que ver con el estudio del tiempo de respuesta y la tasa de error en aplicaciones web, destacan la conveniencia de un protocolo en proporción al número de solicitudes, aspecto a considerar en el estudio para la mejora del desempeño; En el mismo sentido J. W. Cane (Cane, 2004) define el desempeño basado en el tiempo de demora y los recursos del servidor utilizados para responder las solicitudes. Por su parte C. H. Kao (C. H. Kao, 2013) refuerza la importancia del estudio del desempeño en cuanto a tiempo de respuesta. Una cuestión en los estudios mencionados es que resaltan la existencia de soluciones parciales y limitadas en la solución de estos problemas.

La necesidad de un adecuado desempeño de servicios en aplicaciones web requiere buenas prácticas que describan cómo solucionar los problemas de disponibilidad, considerando prácticas y tecnologías actuales. R. Fielding define a REST como un estilo de abstracción arquitectónico orientado exclusivamente a describir técnicamente sus componentes (Fielding, 2000), es por ello que existe la necesidad de describir en un marco de trabajo cómo abordar problemas de desempeño; cuestión que

existen iniciativas aisladas u orientadas con un enfoque distinto (Oracle, 2021).

Existen iniciativas que buscan incrementar el rendimiento del software mediante principios y características que permitan desacoplar la lógica del software y ejecutarlas en un plano aparte como el modelo AMQP. Empresas como Solace han hecho un trabajo enorme difundiendo la Arquitectura basada en eventos y sus beneficios logrando mostrar un panorama muy positivo para el futuro inmediato (Solace., 2017), la arquitectura basada en eventos es sin duda una de las propuestas que mayor valor otorgan al software y cubren aspectos complementarios para poder seguir un enfoque de microservicio. este trabajo de investigación utiliza la arquitectura basada en eventos en combinación con el modelo AMQP para poder obtener un mejor rendimiento.

A lo señalado en el párrafo anterior, los marcos de trabajo son un conjunto articulado de actividades y procedimientos, roles y documentos que ayudan a lograr resultados con una orientación específica, dependiendo de la naturaleza para los cuales son definidos. Así Mnkandla (Mnkandla, 2009) indica que un marco de trabajo generalmente proporciona una abstracción esquelética de una solución a una serie de problemas similares, es por ello que se eligió realizar un marco de trabajo en lugar de una metodología.

El marco de trabajo para la mejora del desempeño de servicios en aplicaciones web apoyado en protocolos de gestión de mensajes basado en colas es una solución que describe en forma práctica y entendible cómo abordar problemas de desempeño en aplicaciones web mediante una serie de pasos, artefactos y roles con libertad y creatividad. Esta solución utilizará el protocolo AMQP el cual permite utilizar cualquier lenguaje de programación para la comunicación, definir consumidores escalables, reconocimiento de mensajes fuera de línea y manejo de errores (VMWare, 2008).

Por otro lado, las organizaciones que utilicen el marco de trabajo en su desarrollo de software permitirán a sus clientes disponer de sus aplicativos indistintamente del volumen de datos y concurrencia, así mismo brindará una mejor experiencia de usuario gracias a tiempos de carga reducidos.

Además, la investigación es relevante porque el marco de trabajo propuesto es capaz de eliminar las brechas de comunicación y las ralentizaciones entre diferentes plataformas, sistemas críticos y componentes de aplicaciones (Oasis, 2010), a mayor volumen de datos transferidos existirá menor tasa de error al no existir un proxy limitante como sucede en REST. (X. J. Hong, 2018)

El objetivo de la investigación es determinar en qué medida el marco de trabajo apoyado en protocolos de gestión de mensajes basado en colas mejora el desempeño de servicios en aplicaciones web.

2. Literatura relacionada

Las referencias siguientes pueden ser útiles por su metodología, el campo de aplicación, su alcance u otro

factor que se señale al final de cada párrafo. Así, Haupt et al en su investigación "A Framework for the Structural Analysis of REST APIs" presentada el año 2017 en IEEE International Conference on Software Architecture (ICSA) tuvo como objetivo construir un metamodelo canónico que permita comprender y realizar un análisis estructurado de un API REST. La investigación utilizó el análisis de la documentación y estructura del API para la consecuente construcción del metamodelo canónico obteniendo como resultado el planteamiento del marco de trabajo y métricas como: número de raíces, número de enlaces, máxima profundidad. Las métricas brindan una descripción general de alto nivel permiten caracterizar un conjunto de APIs. De esta referencia interesa la metodología utilizada para formular el marco de trabajo y sus indicadores.

Así mismo, Fernandes et al en su publicación "Performance evaluation of RESTful web services and AMQP protocol" planteó como objetivo presentar un estudio comparativo de desempeño de los servicios web RESTful y el protocolo AMQP considerando el intercambio de mensajes entre cliente y servidor; En base a tres experimentos, los autores encontraron las principales ventajas con respecto al uso de un servicio web, en comparación con el uso del software de código abierto Message Broker, o viceversa, mediante el envío continuo de mensajes con una concurrencia de 24 usuarios. El resultado fue que existe una gran diferencia entre el uso de colas por medio de RabbitMQ y el servicio web, ya que RabbitMQ envía mensajes al servidor solicitante y almacena mensajes en una cola. De esta referencia es interés a la propuesta de ingeniería las pruebas realizadas y sus conclusiones como punto de apoyo para la elección de un protocolo basado en colas. (J. L. Fernandes, 2013)

Por otra parte, Sajedi et al con el fin de tener una guía para la codificación y diseño de software orientado a la usabilidad, los autores plantearon como objetivo presentar las pautas fundamentales para aumentar la usabilidad. Consideraron nuevos aspectos, incluido el acceso de los usuarios, la selección del idioma y otras opciones técnicas. Los resultados fueron una lista de sugerencias y mejoras para diversos aspectos técnicos como, por ejemplo: reducción de tasa de error y latencia. Con respecto a esta referencia interesa la definición de indicadores y la descripción de usabilidad para guiar la implementación en la capa software. (A. S. Badashian, 2008)

Finalmente, C. B. Gemirter et al en su trabajo "A Comparative Evaluation of AMQP, MQTT and HTTP Protocols Using Real-Time Public Smart City Data" en la 6th International Conference on Computer Science and Engineering (UBMK) hicieron una comparativa entre los protocolos anteriormente descritos, los resultado que obtuvieron son de vital

interés para esta investigación ya que concluyeron que los protocolos AMQP y MQTT son cuatro veces más rápidos que HTTP. Los protocolos orientados a mensajes ofrecen resultados más estables en comparación con el protocolo HTTP utilizando servicios REST. (C. B. Gemirter, 2021)

Operaciones de TI o quien haga a sus veces, un reporte de los aplicativos backend, el formato del reporte incluye las columnas: url de solicitud, fecha, tamaño, tiempo de respuesta del upstream en milisegundos. La finalidad de esta etapa es obtener un conocimiento profundo del funcionamiento del diseño del Sistema el cual es útil para plantear hipótesis plausibles sobre lo que está fallando. (Betsy Beyer, 2016)

3. Metodología

Esta investigación se divide en tres etapas: i) Identificar requerimientos que debe cumplir el marco de trabajo, ii) elaboración del marco de trabajo y iii) aplicación del marco de trabajo.

Respecto a la identificación de requerimientos, se debe comprender y describir el desempeño de servicios en aplicaciones web, se describe el flujo y ciclo de vida de una solicitud a un servicio web.

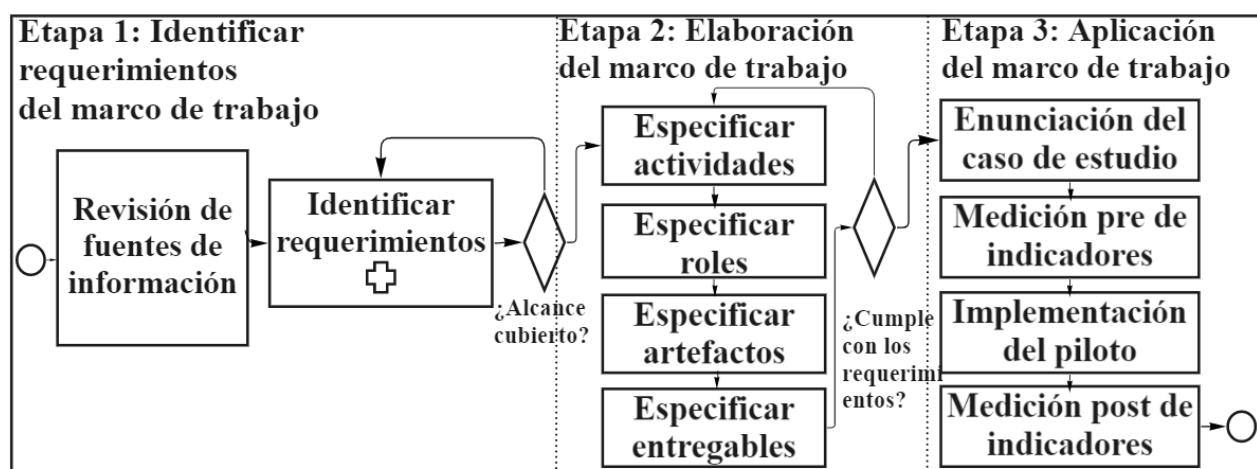


Figura 1. Metodología del Marco de trabajo para la mejora del desempeño de servicios en aplicaciones web apoyado en protocolos de gestión de mensajes basado en colas.

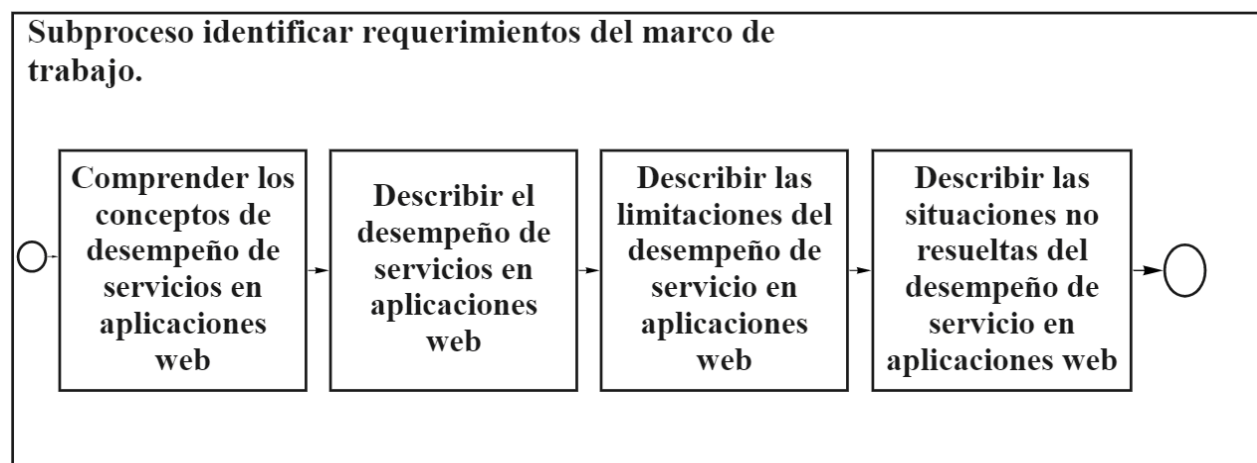


Figura 2. Subproceso identificar requerimientos del marco de trabajo.

En lo referente a la etapa uno, se revisó las fuentes de información, para ello se accede al historial de logs de incidencias, servicios inactivos en dashboards y reportes de usuarios. El Site Reliability Engineer (ver la sección 4.2.2) se encarga de solicitar, al área de

El Site Reliability Engineer elabora un marco referencial. Finalmente, se describirán las limitaciones y situaciones no resueltas.

El entregable de esta etapa es el Documento de objetivo de nivel del servicio (ver tabla 2) donde estarán definidos los objetivos de nivel de servicio, las

8

buenas prácticas de la ingeniería de la confiabilidad de sitios web indican que se determinan en función del tiempo de inactividad que se acordó como aceptable, y representa el límite máximo de interrupciones y errores permitidos. (Redhat, 2020)

Por su parte en la etapa dos, para la elaboración del marco de trabajo, se elaborará la secuencia de actividades mediante el análisis del alcance de los requerimientos identificados en la etapa 1 y determinando puntos de control. A continuación, se especificarán los roles y actividades. Luego, se especificará los artefactos y entregables. Marco de trabajo

Finalmente, en la etapa tres, en la aplicación del marco de trabajo, se llevará a cabo la enunciación del caso de estudio, se especifica el alcance en el plan de implementación del marco de trabajo. Acto seguido, el Site Reliability Engineer hace la medición pre de indicadores usando fichas de observación. Seguidamente, se ejecutará el piloto. Finalmente se hace la medición post para su respectivo análisis comparativo y conclusión.

4. Resultados

4.1. Identificar requerimientos del marco de trabajo

La primera actividad de la etapa requiere revisar las fuentes de información de la organización y finalmente se elaboran las fichas de observación. La revisión de las fuentes de información apunta a poder cubrir algunos de los aspectos que REST no abarca por definición propia, al ser REST un estilo de abstracción no detalla el cómo implementar y monitorear el rendimiento de los servicios elaborados. [8]

Para los requerimientos del marco de trabajo se ha considerado lo necesario para definir y ejecutar el alcance y finalmente medirlo en un entorno de producción. La definición de los objetivos e indicadores de nivel de servicio serán la guía para la ejecución del marco de trabajo. (Betsy Beyer, 2016)

Tabla 1. Lista de requerimientos del marco de trabajo

N°	Descripción
REQ1	El marco de trabajo debe guiar la delimitación y el diagnóstico de servicios
REQ2	El marco de trabajo debe definir los objetivos de nivel de servicio.
REQ3	El marco de trabajo debe definir los indicadores de nivel del servicio.
REQ4	El marco de trabajo debe guiar en la elicitación de requerimientos del software que se gestione con el marco de trabajo.

REQ5	El marco de trabajo debe orientar la mejora de la arquitectura del software en cuestión a gestionar por el marco de trabajo
REQ6	El marco de trabajo debe orientar la implementación de mejoras
REQ7	El marco de trabajo debe medir las mejoras mediante indicadores de desempeño
REQ8	El marco de trabajo debe orientar el despliegue a producción

4.2. Elaboración del marco de trabajo

4.2.1. Estructura del marco de trabajo

El marco de trabajo propuesto se divide en 4 etapas: diagnóstico, preparación, implementación y medición (ver Figura 3). Este apartado hace una descripción de alto nivel de la estructura del marco de trabajo, cuestión que se aborda con mayor detalle en la sección 4.3

Respecto a la etapa de diagnóstico, las actividades son: revisión de logs, delimitar los servicios de la aplicación a mejorar y finalmente valoración de indicadores de desempeño; sus artefactos son: Logs de errores como fuente de información, lista de servicios y el dashboard para cumplir los requerimientos 1 al 3. La entrada de la etapa son los logs de la aplicación, se analizan los servicios y en base a ello la salida es el informe diagnóstico de la aplicación web.

En consideración a la etapa de preparación, las actividades son: elaborar requerimientos de mejora, elaborar diagrama de arquitectura de aplicaciones web y desplegar la infraestructura de TI; sus artefactos son: Requerimientos, diagrama de clases, diagrama de componentes y diagrama de despliegue. La entrada de la etapa son los requerimientos del informe diagnóstico, se especifican los requerimientos y se despliega la infraestructura de TI.

Referente a la etapa de implementación, las actividades son: implementar requerimientos de mejora, hacer pruebas; sus artefactos son: Ficheros físicos de la aplicación web, y ficheros de casos de prueba. La entrada de la etapa son los ficheros físicos, creados en base a la especificación de requerimientos, y la salida de la etapa es la aplicación web mejorada.

En lo que concierne a la etapa de medición, las actividades son: despliegue a producción y valoración de indicadores de desempeño; sus artefactos son: Aplicación web y el dashboard. La entrada de la etapa es la aplicación web mejorada, esto permite realizar la valoración final de los indicadores y generar el informe de mejora, que es la salida de la etapa.

12

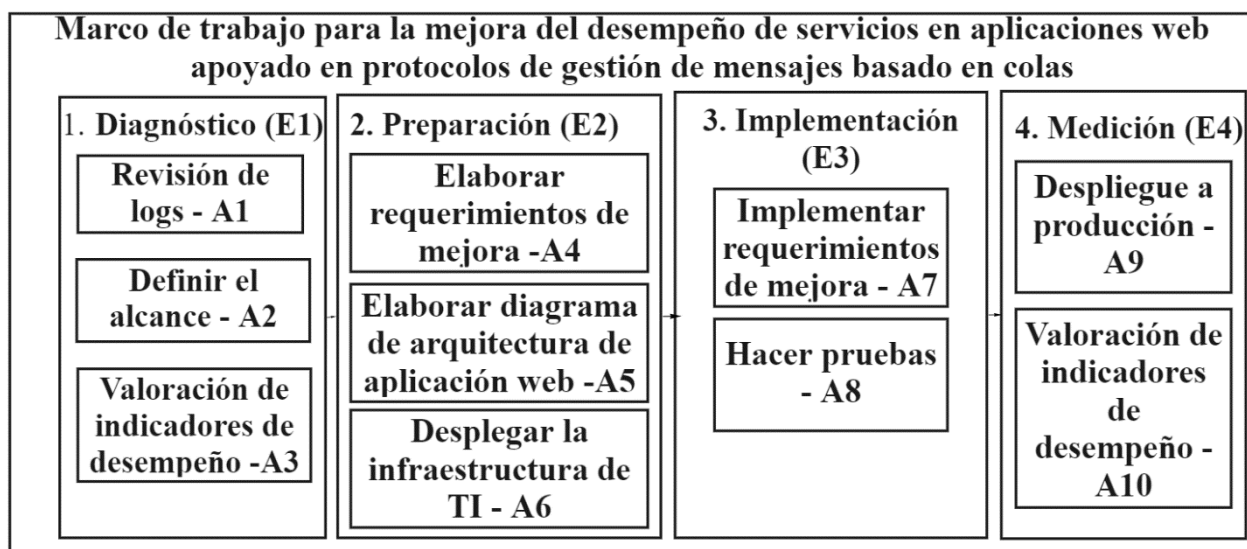


Figura 3. Marco de trabajo para la mejora del desempeño de servicios en aplicaciones web apoyado en protocolos de gestión de mensajes basado en colas.

4.2.2. Artefactos, roles y entregables

Los artefactos se utilizan para modelar entidades conceptuales clave relevantes para el negocio que progresan a través de las actividades, entre los artefactos podemos encontrar a los logs, la lista de servicio y el dashboard para la elaboración del documento de objetivo de nivel del servicio. Así mismo, los requerimientos y diagramas sirven para que la etapa de preparación concluya en la infraestructura de TI desplegada. Finalmente, los ficheros físicos del código fuente y casos de prueba reciben las mejoras a través de los desarrolladores y testers, las mediciones finales demostrarán el progreso a través del dashboard.

Los roles ejecutan las actividades del marco de trabajo, en esta investigación se utilizaron roles conocidos dentro del ámbito del desarrollo de software como son: Analista, Arquitecto de software, Desarrollador y Tester. Así mismo, se utilizaron roles definidos en DevOps como el Arquitecto de infraestructura y el Coordinador de despliegue. Finalmente, el rol de Ingeniero de confiabilidad de sitio forma parte de los principios y practicas de la Ingeniería de Condiabilidad de Sitio definida por Google. (Betsy Beyer, 2016)

Los entregables son salidas de las actividades durante las etapas, cada uno de los roles interactúa a través de las actividades procesando los artefactos para generar resultados. Los informes de diagnóstico y de mejora sirven para evidenciar la definición del alcance y la mejora lograda, otros entregables como el Documento de objetivo de nivel del servicio definen los indicadores y la cuantificación de los mismos, mientras que el Diagrama de Arquitectura y la infraestructura de TI desplegada son resultados de la etapa de preparación, donde los arquitectos de

software y arquitecto de infraestructura trabajan para tener todo un entorno de trabajo de lenguaje agnóstico para implementar los consumidores y productores AMQP. Finalmente la aplicación web es mejorada y estos resultados son evidenciados en el informe de mejora.

Tabla 2. Resumen del marco de trabajo

Actividad	Artefacto	Entregable	Rol
E1:A1	Logs	Informe diagnóstico	Ingeniero de confiabilidad del sitio
E1:A2	Alcance	Informe diagnóstico	Ingeniero de confiabilidad del sitio
E1:A3	Dashboa rd	Documento de objetivo de nivel de servicio	Ingeniero de confiabilidad del sitio
E2:A4	Requeri mientos	Especificaci ón de requerimient os	Analista
E2:A5	Diag. Clases Diag. Compon entes	Diagrama de arquitectura	Arquitecto de software
E2:A6	Diag. de desplieg ue	Infraestructu ra de TI desplegada	Arquitecto de infraestructura TI

E3:A7	Ficheros físicos	Aplicación web mejorada	Desarrollador
E3:A8	Archivos de casos de prueba		Tester
E4:A9	Aplicación web	Informe de mejora	Coordinador de despliegue
E4:A10	Dashboards	Informe de mejora	Ingeniero de confiabilidad del sitio

4.3. Definición de la aplicación del marco de trabajo

4.3.1. Diagnóstico

El diagnóstico tiene como objetivo poder realizar la valoración de indicadores de desempeño de servicios en la aplicación web, mediante el levantamiento de información y la delimitación de los servicios a mejorar.

El Site reliability engineer elabora el Documento de Objetivo de Nivel de Servicio, este documento describe los objetivos a cumplir por cada indicador de nivel de servicio, los indicadores son: Disponibilidad, Tasa de error, y Tiempo de respuesta.

El ingeniero de confiabilidad del sitio solicita que analicen los logs de error del proxy reverso del API según la especificación de cada objetivo de nivel de servicio. El análisis de los logs se puede realizar por medio de un lenguaje para realizar consultas como PromQL y un web scrapper, otra opción es extraer los logs en formato texto para su análisis mediante comandos en el terminal. Es recomendable elaborar un dashboard para monitorear las métricas según la periodicidad definida en el documento de Objetivo de Nivel de Servicio.

Acto seguido, el ingeniero de confiabilidad del sitio agrupa por aplicación las peticiones erróneas y delimita cuáles aplicaciones y servicios serán considerados en la iteración. Toda la documentación previamente analizada se condensa en un informe diagnóstico. Finalmente, el Ingeniero de confiabilidad del sitio, realiza la valoración de los indicadores de desempeño antes de la aplicación del marco de trabajo, este documento servirá de insumo para la etapa de preparación.

4.3.2. Preparación

El Analista de software recibe como insumo de entrada el informe diagnóstico de la primera etapa y procede a elaborar la especificación de requerimientos de mejora, se considera la funcionalidad del servicio

existente, la concurrencia del servicio e información técnica como el lenguaje de programación, el framework utilizado y los formatos de respuesta.

El Arquitecto de software analiza la especificación de requerimientos de mejora y define el número de productores, consumidores y workers que serán necesarios para dar soporte a los servicios de la aplicación, se definen las rutas de comunicación, los métodos de envío y los retornos aceptados. Todo lo anterior se describe en el diagrama de arquitectura.

El Arquitecto de infraestructura instancia los entornos necesarios para la aplicación, así mismo habilita un monitor para los consumidores y workers, también habilita los puertos de comunicación entre las aplicaciones involucradas. Finalmente, se configuran las tuberías de integración continua en los entornos especificados por el Arquitecto de software.

4.3.3. Implementación

Los Desarrolladores de software reciben como entrada cada requerimiento de mejora con su documentación respectiva, y comienzan el proceso de implementación de la solución para cada servicio, los requerimientos de mejora son implementados e interconectados con el gestor de colas.

El equipo desarrollador debe mapear el envío y retorno de información respetando las rutas según los patrones definidos por el Arquitecto de software, el número de consumidores se configuran en base a la fórmula $(2 \times \text{número de cores}) + 1$, y monitoreando el rendimiento en base a los criterios TTIN (Incremento del número de procesos en uno) y TTOU (Decremento del número de procesos en uno. [20] De igual forma, el arquitecto de software define el número de colas, y exchanges en el gestor de colas. Finalmente, el arquitecto de software debe coordinar con Infraestructura de TI la habilitación de los puertos y el servicio STOMP para la comunicación de los sockets.

Los Desarrolladores definen los casos de prueba y codificarlos en los archivos de prueba, los Tester deben realizar las pruebas funcionales y de rendimiento para asegurar que los servicios cumplen con la función determinada y aportan al objetivo del indicador. En caso el requerimiento no haya sido cumplido, los Tester realizan un informe declarando la necesidad de una nueva iteración, el punto de partida de la iteración estará definido por la recomendación del Tester.

4.3.4. Medición

El Coordinador de despliegue recibe la confirmación de las pruebas satisfactorias y ejecuta las acciones del pase a producción, realiza las fusiones de las ramas respectivas y dispara los eventos para las tuberías de despliegue. El software de integración

continúa retorna el informe del pase, el Ingeniero de confiabilidad del sitio realiza una nueva medición posterior a la aplicación del marco de trabajo.

5. Resultados

Los resultados que se presentan están acordes a la ejecución de la etapa 3 de la metodología de este artículo.

5.1. Descripción del caso de aplicación

El marco de trabajo se implementó en la Universidad Peruana Unión, la cual cuenta con 3 campus en Lima, Juliaca y Tarapoto. La Universidad brinda programas de Pregrado y Posgrado a la comunidad universitaria, así mismo es parte de la red educativa mundial adventista de 118 universidades.

El marco de trabajo se ejecutó en los sistemas de admisión y encuestas de la Universidad Peruana Unión, el sistema de admisión tiene desarrollado su frontend en AngularJs, y por su parte el backend en Python, así mismo el sistema de encuestas tiene desarrollado su frontend en Angular 10 y el backend está desarrollado en PHP. El uso de cuestionarios para los distintos procesos académicos, admisión, evaluación de personal son bastante concurrentes y requieren de un alto rendimiento ininterrumpido. Por otro lado, el sistema de Admisión permite a los interesados postular hacia un programa universitario de los distintos niveles de enseñanza disponibles en los campus. El reporte de postulantes, la generación de ingresos y códigos universitarios se realizan por este sistema.

5.2. Implementación del marco de trabajo

La revisión de logs se realizó por medio de Prometheus, un software scraper que consolida métricas periódicamente, en el caso de estudio se habilitó el módulo stub_status al proxy reverso NGINX. Seguidamente, el Ingeniero de Confiabilidad del Sitio elabora el Documento de Objetivo de Nivel de Servicio, los objetivos definidos a nivel de servicio se pueden visualizar en la tabla 3.

Tabla 3. Objetivos de nivel de servicio

Categoría	SLI	SLO
Disponibilidad (del tipo agregación)	La proporción de solicitudes exitosas. Cualquier estado HTTP que no sea 500-599 se considera correcto.	97% de éxito

	Recuento de solicitudes http "api" que no tienen un código de estado 5XX dividido por recuento de todas las solicitudes http "api".	
Tiempo de respuesta	La proporción de solicitudes suficientemente rápidas. “Suficientemente rápido” se define por < 400 ms.	90% de solicitudes < 400 ms
	Recuento de solicitudes http http_requests "api" con una duración menor o igual a "0,4" segundos dividido por recuento de todas las solicitudes http "api" http_requests.	
Tasa de error	La proporción de solicitudes no exitosas permitidas. Cualquier estado HTTP que no sea 500-599 se considera correcto.	3% como límite de error
	Recuento de solicitudes http "api" que tienen un código de estado 5XX dividido por recuento de todas las solicitudes http "api"	

Seguidamente, luego de un lapso definido por el equipo de trabajo, el ingeniero de confiabilidad del sitio solicitó a Infraestructura de TI el consolidado de logs de acceso y error que tengan como estatus valores mayores a 500. El siguiente procedimiento puede variar según las herramientas de monitoreo de la organización, en el caso de estudio, se elaboraron los consolidados por medio de comandos en el terminal del servidor. Se utilizó una expresión regular que filtra el contenido del archivo físico, realiza también una agrupación y finalmente ejecuta un ordenamiento del

resultado. Finalmente se procedió a delimitar el alcance para los requerimientos de mejora.

Seguidamente, el ingeniero de confiabilidad del sitio utiliza las ecuaciones planteadas en el documento de objetivo de nivel de servicio en el software Grafana, mediante expresiones PromQL. Se utilizó los recursos de Prometheus como fuente de datos, así mismo el resultado de los querys se representaron como gráficos Gauge.

Al finalizar la etapa 1, se realizó la medición al aplicar las configuraciones del dashboard, los resultados fueron Disponibilidad del 92.931%, Tasa de error de 7.070% y Tiempo de respuesta de 36.500%.

La siguiente actividad de la etapa 2 fue la elaboración del diagrama de arquitectura de la aplicación web, la naturaleza del framework está orientada al uso del protocolo de apoyo basado en gestión de colas, para ello se elaboró un diagrama de arquitectura basada en eventos. La arquitectura basada en eventos utiliza eventos para desencadenar y comunicar entre servicios desacoplados y es común en las aplicaciones modernas construidas con microservicios. (Amazon, 2022) Para el caso de estudio, se propuso contar con productores tanto frontend como backend según el requerimiento de mejora; Como bróker se utilizó al software RabbitMQ, habilitando el plugin Web STOMP para tener disponible el web socket para enviar y recibir los mensajes y resultados por medio de la suscripción a un Exchange. Finalmente se configuran consumidores para poder procesar los mensajes gestionado por el bróker y ser llevados hacia el aplicativo solicitante.

agiliza en gran manera el tráfico de la petición web y se obtiene retroalimentación del estado de la cola.

Para el despliegue de la arquitectura, el área de infraestructura de TI utilizó el software de despliegue continuo Jenkins y GIT como software integración continua propuesta, se realizó la configuración de la tuberías y ramas de los repositorio, de esa forma los cambio realizado por el equipo de desarrollo se desplegaban en los entornos correspondientes.

En cuanto a la etapa 3, la implementación de los requerimientos de mejora se realiza progresivamente, un ejemplo es el servicio para calcular ingresantes de admisión de la aplicación frontend, antes de la implementación del marco de trabajo la aplicación angular enviaba una petición a la aplicación Django Rest para ejecutar un procedimiento almacenado, sin embargo, se cuenta con un límite de tiempo de un minuto incrementando la tasa de error. Por lo tanto, se propone anular la petición a la aplicación backend, conectarse al websocket STOMP desde la aplicación angular, suscribirse a un Exchange y anidar a una cola la solicitud de generación de ingreso, luego se ejecutaría el procedimiento almacenado desde un consumidor que lea la ruta configurada para comunicar a la aplicación angular el resultado.

A continuación, se configuraron los consumidores de los mensajes gestionados por el bróker. El consumidor está compuesto por la conexión, recepción de parámetros, ejecución de método según la ruta, devolución del mensaje hacia el Exchange para retornar a la aplicación frontend.

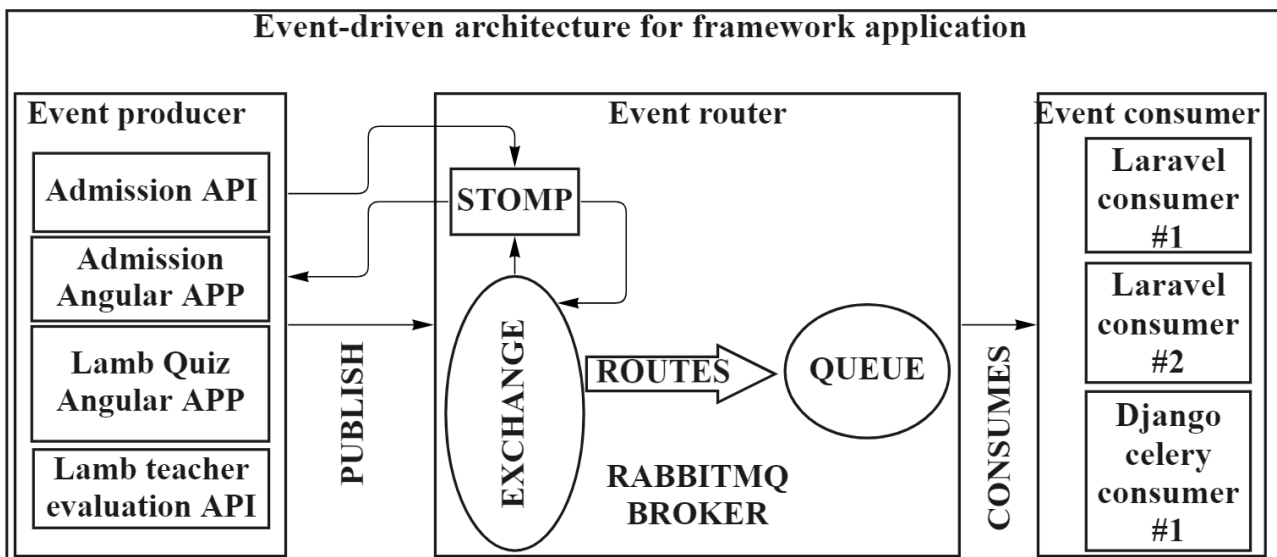


Figura 4. Arquitectura basada en eventos correspondiente al marco de trabajo.

RabbitMQ permite comunicar al cliente frontend con el Exchange anidado a la cola, de esta forma se

El siguiente paso es configurar el suscriptor de los mensajes, para ello se procede a utilizar el método

watch del servicio STOMP para suscribirse a los mensajes emitido por el Exchange, el cual hace referencia a una cola en RabbitMQ.

La etapa cuatro es la medición, para ello el coordinador de despliegue aprueba las solicitudes de merge para poder visualizar los cambios en producción.

Finalmente se realiza la medición de indicadores posterior a la aplicación del marco de trabajo, en el caso de estudio de la aplicación del marco de trabajo, se consideró el tiempo de una semana para el recalcule de los indicadores tal como lo indica el Documento de Objetivos de Nivel de Servicio. La medición de los indicadores posterior a la aplicación del marco de trabajo es: disponibilidad de 99.379%, tasa de error de 0.621% y tiempo de respuesta de peticiones ubicadas en el cuartil 90 es de 91.303%.

5.3. Comparación de resultados

Según el Documento de Objetivo de Nivel de Servicio, se realiza la medición de los indicadores previa implementación del marco de trabajo, posteriormente se aplica el piloto del marco de trabajo y finalmente se realiza la medición posterior a la implementación del marco de trabajo.

En cuanto al indicador disponibilidad de información, la medición previa retornó un resultado de 92.391%, mientras que la medición posterior retornó un resultado de 99.379%, se observa un cambio de +6.988% lo cual permite sobrepasar el objetivo en 2.379%. La principal razón de la mejora en el indicador se da por la estabilidad que proporciona el protocolo AMQP, el cual es 4 veces más rápido que el protocolo HTTP.[17]

Así mismo, el indicador tasa de error, la medición previa retornó un resultado de 7.070%, mientras que la medición posterior retornó un resultado de 0.621%, se observa un cambio de -6.449% lo cual permite sobrepasar el objetivo en 2.379%. La tasa de error se reduce porque el marco de trabajo apunta directamente a los servicios con mayor índice de error, especialmente los derivados de tiempos de respuesta.

Finalmente, el indicador tiempo de respuesta de la aplicación, la medición previa retornó un resultado de 36.500 %, mientras que la medición posterior retornó un resultado de 91.303%, se observa un cambio de +54.803% lo cual permite sobrepasar el objetivo en 1.303%. Los tiempos de respuesta mejoran debido a que la latencia es mucho menor utilizando el protocolo propuesto por el marco de trabajo. (C. B. Gemirter, 2021)

Tabla 4. Comparación de resultado pre y post

Indicador	Medición pre	Medición post	Objetivo	Conclusión
-----------	--------------	---------------	----------	------------

Disponibilidad de información [18]	92.391 %	99.379 %	Mín. 97%	Objetivo cumplido por 2.379%
Tasa de error [18]	7.070 %	0.621 %	Hasta 3%	Objetivo cumplido Por 2.379%
Tiempo de respuesta de la aplicación [18]	36.500 %	91.303 %	Mín. 90%	Objetivo cumplido por 2.379%

En cuanto a disponibilidad de información, el objetivo de 97% fue superado por un 2.379% logrando un 99.379%, esto sucedió porque los servicios de la aplicación son más estables al estar alineados a una arquitectura basada en eventos. Por otro lado, el objetivo de la tasa de error era de reducirse a un 3%, sin embargo, el objetivo fue superado en un 2.449% debido a que los errores por tiempo de respuesta fueron mitigados. Finalmente, el tiempo de respuesta de la aplicación tenía como objetivo mejorar a un 90%, al realizar la medición se sobrepasó el objetivo por 1.303% logrando un 91.303%, este último beneficio se logró gracias a que los tiempos de respuesta son reducidos al tener tareas ejecutadas en un segundo plano.

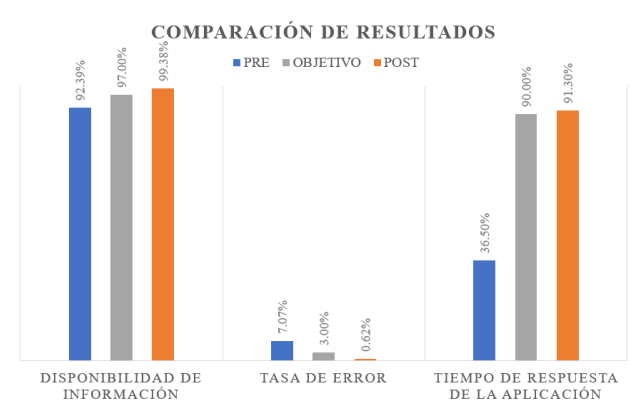


Figura 5. Comparación de mediciones pre y post

DOI: 10.18421/TEMxx-xx

<https://doi.org/10.18421/TEMxx-xx>

Corresponding author: - ,
affiliation
Email: -

Received: ----.

Accepted: ----.

Published: ----.

© 2022. ----; published by UIKTEN. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License.

The article is published with Open Access at <https://www.temjournal.com/>

6. Conclusiones

El marco de trabajo para la mejora del desempeño de servicios en aplicaciones web apoyado en protocolos de gestión de mensajes basado en colas, permite mejorar el indicador disponibilidad de información, al superar el objetivo del indicador en 2.379%. Así mismo, el marco de trabajo permite reducir la tasa de error, al superar el objetivo del indicador en un 2.379%. Finalmente, el marco de trabajo permite mejorar el indicador tiempo de respuesta de la aplicación, al superar el objetivo del indicador en 1.303%.

Se recomienda implementar un latido para los consumidores, especialmente en lenguajes sin implementación nativa como php.

El marco de trabajo define actividades, roles, artefactos y entregables organizados de una forma entendible y replicable en distintos tipos de software, como aplicaciones móviles, aplicaciones de escritorio que utilicen la comunicación entre cliente y servidor.

Referencias –

- [1]. A. S. Badashian, M. M., A. Pourshirmohammadi and M. M. nejad. (2008). Fundamental Usability Guidelines for User Interface Design. Paper presented at the 2008 International Conference on Computational Sciences and Its Applications, Perugia, Italy. Retrieved from: <https://ieeexplore.ieee.org/document/4561210/>
- [2]. Amazon. (2022). What is an Event-Driven Architecture? Retrieved from: <https://aws.amazon.com/es/event-driven-architecture/>
- [3]. Betsy Beyer, C. J., Jennifer Petoff and Niall Richard Murphy. (2016). Site Reliability Engineering (O'Reilly Ed. Vol. 1).
- [4]. C. B. Gemirter, Ç. Ş. a. Ş. B. (2021). A Comparative Evaluation of AMQP, MQTT and HTTP Protocols Using Real-Time Public Smart City Data. Paper presented at the 6th International Conference on Computer Science and Engineering (UBMK), Ankara, Turkey. Retrieved from: <https://ieeexplore.ieee.org/document/9559032>
- [5]. C. H. Kao, C. C. L. a. J. C. (2013). Performance Testing Framework for REST-Based Web Applications. Paper presented at the 13th International Conference on Quality Software, Naging, China.
- [6]. Cane, J. W. (2004). Measuring Performance of Web Applications: Empirical Techniques and Results. Paper presented at the IEEE SoutheastCon, 2004. Proceedings, Greensboro, NC, USA.
- [7]. F. Haupt, F. L., A. Scherer and K. Vukojevic-Haupt. (2017). A Framework for the Structural Analysis of REST APIs. Paper presented at the 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg, Sweden. Retrieved from: <https://ieeexplore.ieee.org/document/7930199>
- [8]. Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. (Doctoral). University of California, Irvine. Retrieved from: https://www.ics.uci.edu/~fielding/pubs/dissertation/fie lding_dissertation.pdf
- [9]. GUnicorn. (2009). GUnicorn Design Docs. Retrieved Retrieved from: <https://docs.gunicorn.org/en/latest/design.html>
- [10]. J. L. Fernandes, I. C. L., J. J. P. C. Rodrigues and S. Ullah. (2013). Performance evaluation of RESTful web services and AMQP protocol. Paper presented at the 2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN), Da Nang, Vietnam. Retrieved from: <https://ieeexplore.ieee.org/document/6614932/>
- [11]. Longatt, F. G. (2017). Introducción a los Sistemas de Información. Fundamentos. Introducción a los Sistemas de Información. Fundamentos. Morelos, México.
- [12]. Mario Villamizar, O. G., Harold Castro, Mauricio Verano, Lorena Salamanca, Rubby Casallas, Santiago Gil. (2015). Evaluating the Monolithic and the Microservice Architecture Pattern to
- [13]. Deploy Web Applications in the Cloud. Paper presented at the 10th Colombian Computing Conference, (10CCC 2015).
- [14]. Mnkandla, E. (2009). About software engineering frameworks and methodologies. Paper presented at the IEEE AFRICON 2009, Nairobi, Kenya. Retrieved from: <https://ieeexplore.ieee.org/abstract/document/5308117>
- [15]. Oasis. (2010). AMQP is the Internet Protocol for Business Messaging. Retrieved from: <https://www.amqp.org/about/what>
- [16]. OMC. (2019). El comercio mundial listo para una recuperación sólida, aunque desigual, después de la pandemia de COVID-19 [Press release]. Retrieved from: https://www.wto.org/spanish/news_s/pres21_s/pr876_s.htm
- [17]. Oracle. (2021). REST API for Oracle Responsys Marketing Cloud Service. F22015-13. Retrieved from: <https://docs.oracle.com/en/cloud/saas/marketing/respo nsys-rest-api/Throttling.html>

- [18]. Redhat. (2020). ¿Qué es la ingeniería de confiabilidad del sitio (SRE)?
Retrieved from:
<https://www.redhat.com/es/topics/devops/what-is-sre>
- [19]. Solace. (2017). AMQP 1.0 Protocol Conformance.
Retrieved from:
<https://docs.solace.com/Open-APIs-Protocols/AMQP/AMQP-Protocol-Conformance.htm>
- [20]. UNCTAD. (2020). Informe sobre el comercio y el desarrollo 2020.
Retrieved from New York:
https://unctad.org/system/files/official-document/tdr2020_es.pdf
- [21]. VMWare. (2008). AMQP 0-9-1 Model Explained.
Retrieved from:
<https://www.rabbitmq.com/tutorials/amqp-concepts.html>
- [22]. X. J. Hong, e. a. (2018). Performance Analysis of RESTful API and RabbitMQ for Microservice Web Application. Paper presented at the 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South).
Retrieved from:
<https://ieeexplore.ieee.org/document/8539409>