


## Problemas en la implementación de pruebas en sistemas de microservicios: estudio de mapeo sistemático

### Problems in microservice systems testing: a systematic mapping study

Christian Candela Uribe  Luis Sepúlveda Rodríguez   
Universidad Tecnológica de Pereira - Universidad del Quindío, Colombia

Daniel Gómez Betancur  Julio Chavarro Porras   
Universidad Tecnológica de Pereira, Colombia

John Sanabria Ordoñez   
Universidad del Valle, Colombia

OPEN  ACCESS

Recibido: 22/09/2021

Aceptado: 22/10/2021

Publicado: 13/12/2021

Correspondencia de  
autores:

daniel33.gomez@utp.edu.co



Copyright 2020  
by Investigación e  
Innovación en Ingenierías

#### Resumen

**Objetivo:** En este artículo se construyó un SMS (Systematic Mapping Study) para la identificación y clasificación de problemas y causas en la realización de pruebas en la MSA (Micro-Service Architecture). **Metodología:** Este SMS adoptó prácticas establecidas en ingeniería de software basada en evidencias. El SMS consideró artículos publicados desde 2016 en bases de datos como ACM, IEEE Xplore, Science Direct, WoS, Springer y Scopus. Además, la estrategia de búsqueda incluyó cuatro iteraciones de bola de nieve. **Resultados:** El SMS comenzó con 842 estudios y terminó con 32 estudios seleccionados, identificando nueve problemas y ocho causas relacionadas con las pruebas de software en MSA. Finalmente, los estudios se clasificaron por problemas y causas. **Conclusiones:** El SMS permitió inferir que la Alta Complejidad en pruebas MSA es el problema más representativo. Igualmente, las causas más destacadas son el Alto Número de Servicios, la Falta de Investigación y la proliferación de cada versión de servicio (Service Versioning).

**Palabras clave:** MSA, Microservicios, Arquitectura de microservicios, Pruebas, SMS.

#### Abstract

**Objective:** This article aimed to build a SMS (Systematic Mapping Study) for the identification and classification of problems and causes in testing in the MSA (Micro-Service Architecture). **Methodology:** This SMS adopted established practices in evidence-based software engineering. The SMS considered articles published since 2016 in databases such as ACM, IEEE Xplore, Science Direct, WoS, Springer, and Scopus. In addition, the search strategy included four snowball iterations. **Results:** The SMS started with 842 studies and ended with 32 selected studies, identifying nine problems and eight causes related to software testing in MSA. Finally, the studies were classified by problems and causes. **Conclusions:** The SMS allowed deducing that High Complexity in MSA testing is the most representative problem. Likewise, the most outstanding causes are the High Number of Services, the Lack of Research, and the proliferation of each service version (Service Versioning).

**Keywords:** MSA, Microservices, Microservices architecture, Testing, SMS.

**Como citar (IEEE):** D. Gómez-Betancur., C. Candela-Uribe., L. Sepúlveda-Rodríguez., J. Chavarro-Porras., y J. Sanabria-Ordoñez. "Problemas en la Implementación de Pruebas en Sistemas de Microservicios: Estudio de Mapeo Sistemático", vol. 9, n°3, pp. 3-17, 2021. DOI: <https://doi.org/10.17081/invinno.9.3.5314>

## Introducción

MSA ha recibido atención los últimos años en la industria y academia [1], entre otras cosas al permitir desarrollar aplicaciones como conjuntos de pequeños servicios comunicados mediante mecanismos ligeros [2]. MSA, comparado con las aplicaciones monolíticas, tiene ventajas como la facilidad de desarrollo, despliegue, mantenimiento y escalabilidad [3]. Sin embargo, MSA tiene desafíos como la descomposición de servicios, el monitoreo y la realización de pruebas [4, 5].

Dada la naturaleza distribuida de MSA, el desafío en la realización de pruebas se intensifica por el uso intensivo de la comunicación asincrónica, la cual causa problemas relacionados con la coherencia de los datos [6].

Considerando lo anterior, el presente trabajo se centra en la realización de pruebas en MSA presentando un SMS con la identificación y clasificación de las principales problemas y causas. Este SMS adopta prácticas establecidas en ingeniería de software basada en evidencias [7, 8], además de otros elementos del proceso establecidos en [9] y considerar artículos publicados desde 2016 en bases de datos como ACM, IEEE Xplore, ScienceDirect, WoS, Springer y Scopus. De igual forma, se apoyó en el uso de software especializado como Software SMS Builder, Mendeley y Google Scholar. La estrategia de búsqueda incluyó cuatro iteraciones de bola de nieve. El SMS comenzó con 842 estudios y terminó con 32 estudios seleccionados, identificando nueve problemas y ocho causas relacionadas con las pruebas de software en MSA.

El resto del documento comprende las siguientes secciones: 2) Trabajos relacionados. 3) Metodología. 4) Resultados y discusión. 5) Amenazas a la validez y 7) Conclusiones.

## Trabajo relacionado

A continuación, se presentan estudios que pertenecen al ámbito de la ingeniería del software testing y los sistemas de microservicios; su fecha de publicación está entre los años 2017-2021 y tienen una temática similar a nuestra propuesta de investigación. Ver Tabla 1.

**Tabla 1. Trabajos relacionados**

Categorías	Estudios
SMS con información general de la MSA	[10, 11, 12]
SMS con la tendencia y los retos de la MSA	[5, 13, 14]
Estudios con estrategias/métodos para la realización de pruebas en MSA	[15, 16, 17, 18]

**Fuente:** Elaboración propia

Los trabajos listados en la Tabla 1 hacen una contribución significativa en el área de MSA desde el enfoque de su arquitectura, atributos de calidad, y procesos de implementación de pruebas. Este SMS extiende el trabajo de [14], adicionando la identificación las causas y actualizando los estudios.

## Metodología

Este SMS adopta prácticas establecidas para realizar revisiones sistemáticas en ingeniería de software basada en evidencias [7, 8], se compone por seis etapas: 1) Planeación. 2) Búsqueda de estudios. 3) Evaluación de calidad. 4) Extracción de datos. 5) Clasificación de estudios. 6) Análisis de datos.

### Etapa 1: Planeación

Se establece aspectos como: 1) Propósito general, 2) Objetivos, 3) Preguntas de investigación, 4) Métricas, 5) Tópicos de clasificación, 6) Criterios de inclusión/exclusión y 7) Atributos de calidad. El modelo GQM (Goal-Question-Metric) [19, 20] se aplicó a los aspectos 2,3, 4 y 5; el modelo PICOC [21, 22] fue aplicado en los aspectos 4 y 5.

**Objetivo:** Identificar estudios relacionados con problemas y causas para la ejecución de pruebas de software en MSA.

**Preguntas de investigación (RQs):** Las preguntas de investigación se relacionan en la Tabla 2.

Tabla 2. Preguntas de investigación.

Código	Descripción
RQ1	¿Cuáles son los problemas más recurrentes en la realización de pruebas en sistemas basados en arquitectura de microservicios?
RQ2	¿Cuáles son las causas más relevantes de los problemas asociados a la realización de pruebas en sistemas basados en arquitectura microservicios?

Fuente: Elaboración propia

**Métrica:** Cantidad de estudios relacionados con cada uno de los problemas asociados a la realización de pruebas en sistemas basados en microservicios.

**Tópicos:** Se definieron a partir de las preguntas de investigación y el modelo PICOC. Los detalles se muestran en la Tabla 3.

Tabla 3. Tópicos de clasificación

Categoría	Tópico
Tipo de prueba	Pruebas funcionales, Pruebas no funcionales
Pruebas funcionales	Pruebas de integración, Pruebas de sistema, Pruebas de regresión
Pruebas no funcionales	Pruebas de rendimiento, Pruebas de resiliencia, Pruebas de carga
Problema	Dificultad de replicar todo el sistema, Las pruebas tienden a ser engañosas, Alta complejidad, Baja facilidad para pruebas, Difícil de monitorear, Alto consumo de recursos, Dificultad para plantear una estrategia, Dificultad para determinar el componente que falla
Causa	Alto número de servicios, Falta de herramientas, Versiones de servicios, Uso de diversos lenguajes de programación, Tiempo de respuesta lento, Falta de investigación, Tiempos de espera

Fuente: Elaboración propia

**Criterios de inclusión:** Artículo de revista escritos en inglés o español y publicados en el periodo 2016-2021.

**Atributos de calidad:** Se adoptaron tres atributos de calidad relacionados con la percepción de validez, la citación y la relación con las RQs. El primer atributo usó el índice CVI [23, 24]. El segundo atributo usó el índice SCI (Study Citation Index) Ver Ecuación 1. El tercer atributo usó el índice IRRQ (Index of Relationship to Research Questions). Ver ecuación 2.

$$SCI = \frac{C}{Y} \quad (1)$$

$C$  es el número de Citaciones en el periodo de tiempo (2016-2021),  $Y$  es el número de años desde la publicación del estudio.

$$IRRQ = \frac{N}{T} \quad (2)$$

$N$  corresponde al número de RQs con el cual se relaciona un estudio.  $T$  es el total de las RQs de este estudio.

## Etapa 2: Búsqueda de estudios

### Definición de la estrategia de búsqueda

En la realización de una SMS es posible utilizar una estrategia de búsqueda híbrida [25, 26]. Se combinaron las estrategias Búsqueda en bases de datos y Búsqueda por bola de nieve. Tomamos como referencia el trabajo realizado por [27].

### Ejecución de búsqueda en bases de datos

Para la identificación de estudios en las bases de datos ACM, IEEE Xplore, ScienceDirect, Web of Science, Springer y Scopus se definieron los términos de búsqueda. Ver Tabla 4.

Tabla 4. Términos de búsqueda

Palabra clave	Sinónimos
Testing	tests, test, assess, assessing, assessment
Issue	trouble, problem, disadvantage, drawback
microservice system	microservice based system, microservice architecture, microservice based architecture

Fuente: Elaboración propia

Se construyeron las cadenas de búsqueda para cada una de las bases de datos según los términos definidos. En el caso de ScienceDirect, se tuvo que dividir la cadena de búsqueda en dos debido a que dicha base de datos no soporta la cantidad de operadores booleanos "OR" y "AND" que se necesitaban. Ver Tabla 5.

**Tabla 5. Cadenas de búsqueda**

Librería	Cadena de búsqueda
ACM	[[All: testing] OR [All: tests] OR [All: test] OR [All: assess] OR [All: assessing] OR [All: assessment]] AND [[All: issue] OR [All: trouble] OR [All: problem] OR [All: disadvantage] OR [All: drawback] OR [All: issues] OR [All: troubles] OR [All: problems] OR [All: disadvantages] OR [All: drawbacks]] AND [[All: "microservice system"] OR [All: "microservice based system"] OR [All: "microservice architecture"] OR [All: "microservice based architecture"] OR [All: "microservices system"] OR [All: "microservices based system"] OR [All: "microservices architecture"] OR [All: "microservices based architecture"]]]
IEEE Xplore	(testing OR tests OR test OR assess OR assessing OR assessment) AND (issue OR trouble OR problem OR disadvantage OR drawback OR issues OR troubles OR problems OR disadvantages OR drawbacks ) AND ("microservice system" OR "microservice based system" OR "microservice architecture" OR "microservice based architecture" OR "microservices system" OR "microservices based system" OR "microservices architecture" OR "microservices based architecture")
Science Direct	(testing OR test) AND (issue OR trouble OR problem) AND ("microservice system" OR "microservice based system") (assess OR assessing OR assessment) AND (disadvantage OR drawback) AND ("microservice architecture" OR "microservice based architecture")
Web of Science	TEMA: ((testing OR tests OR test OR assess OR assessing OR assessment) AND (issue OR trouble OR problem OR disadvantage OR drawback OR issues OR troubles OR problems OR disadvantages OR drawbacks) AND ("microservice system" OR "microservice based system" OR "microservice architecture" OR "microservice based architecture" OR "microservices system" OR "microservices based system" OR "microservices architecture" OR "microservices based architecture"))
Springer	(testing OR test OR assess OR assessing OR assessment) AND (issue OR trouble OR problem OR disadvantage OR drawback) AND ("microservice system" OR "microservice based system" OR "microservice architecture" OR "microservice based architecture")
Scopus	TITLE-ABS-KEY ( ( testing OR tests OR test OR assess OR assessing OR assessment ) AND ( issue OR trouble OR problem OR disadvantage OR drawback OR issues OR troubles OR problems OR disadvantages OR drawbacks ) AND ( "microservice system" OR "microservice based system" OR "microservice architecture" OR "microservice based architecture" OR "microservices system" OR "microservices based system" OR "microservices architecture" OR "microservices based architecture" ) )

Fuente: Elaboración propia

Tras la ejecución de las consultas se obtuvo 817 estudios de los cuales finalmente se seleccionaron 7. Ver Tabla 6.

**Tabla 6. Resultado de ejecución: búsqueda en bases de datos**

	Bases de datos						
	ACM	IEEE	SD	WoS	Springer	Scopus	Total
Sin criterios de exclusión	275	19	61	8	410	44	817
Con criterios de exclusión	14	1	49	5	33	11	113
Tamizaje (Screening)	1	0	4	1	1	0	7
Participación final	14%	0%	58%	14%	14%	0	100%

Fuente: Elaboración propia

### Estrategia de búsqueda 2: Bola de nieve

Los 7 estudios identificados en bases de datos conformaron la línea base para la búsqueda por bola de nieve, realizando 4 iteraciones. Se identificaron 25 estudios. Ver Tabla 7.

Tabla 7. Resultado de ejecución: búsqueda por bola de nieve

Bola de nieve	Iteración 1	Iteración 2	Iteración 3	Iteración 4	Total
Hacia atrás	6	1	2	0	9
Hacia adelante	7	9	0	0	16

Fuente: Elaboración propia

### Resultados de búsqueda

Esta sección presenta un resumen de los resultados de la Etapa 2. Búsqueda de estudios. Esta etapa incluyó dos estrategias de búsqueda, la primera de ella (Bases de datos) permitió identificar inicialmente 817 estudios. A partir de estos datos se realizaron procesos de refinamiento (aplicación de criterios de exclusión y tamizaje), lo que condujo a reducir a 7 los estudios identificados (Ver Tabla 6). A partir de estos 7 estudios se dio inicio a la segunda estrategia de búsqueda (Bola de nieve), que a través de sus iteraciones adelante y hacia atrás permitió identificar 25 nuevos estudios, para un total de 32 estudios identificados.

### Etapa 3: Evaluación de calidad

Aunque en un SMS no es necesario realizar una evaluación de calidad, hacerla aporta mucho al resultado y acerca el SMS a una Revisión Sistemática de la Literatura [27, 28]. Con el fin de disminuir el sesgo en el SMS, se utilizaron los índices CVI, SCI y IRRQ definidos durante la etapa de planeación; con esto, se pudieron identificar los estudios de mayor valor según su relevancia temática (CVI), su citado (SCI) y su relación con el estudio (IRRQ). Ver Figuras 1, 2 y 3.

### Etapa 4: Extracción de datos

Los 32 estudios seleccionados fueron nombrados con el prefijo SPS (Selected Primary Study) seguido de un número. Ver Tabla 8.

Tabla 8. Estudios primarios

SPS	Ref	SPS	Ref	SPS	Ref	SPS	Ref
SPS01	[13]	SPS09	[29]	SPS17	[30]	SPS25	[31]
SPS02	[32]	SPS10	[33]	SPS18	[34]	SPS26	[35]
SPS03	[36]	SPS11	[11]	SPS19	[14]	SPS27	[37]
SPS04	[38]	SPS12	[39]	SPS20	[40]	SPS28	[41]
SPS05	[42]	SPS13	[12]	SPS21	[43]	SPS29	[44]
SPS06	[45]	SPS14	[5]	SPS22	[46]	SPS30	[47]
SPS07	[48]	SPS15	[49]	SPS23	[50]	SPS31	[51]
SPS08	[52]	SPS16	[53]	SPS24	[54]	SPS32	[10]

Fuente: Elaboración propia

### Etapa 5: Clasificación de estudios

La clasificación de los estudios del SMS fue realizada por tres evaluadores usando los tópicos definidos en la etapa de planeación (ver Tabla 3). El resultado de la clasificación se muestra en las tablas 9, 10, 11 y 12. La Tabla 9 muestra información de análisis cruzado entre los SPSs y tipos específicos de pruebas. La Tabla 10 muestra los SPSs relacionados con pruebas funcionales y no funcionales. La Tabla 11 muestra los SPSs relacionados con cada uno de los problemas en la ejecución de pruebas de software en MSA. La Tabla 12 muestra los SPSs relacionados con cada una de las causas relacionadas a problemas en la ejecución de pruebas de software en MSA.

Tabla 9. Sps por tipo específicos de prueba.

Tipo específico de prueba	SPS
Pruebas de integración	SPS001 SPS003 SPS007 SPS013 SPS020 SPS022 SPS023 SPS025 SPS027 SPS028
Pruebas de sistema	SPS023 SPS026 SPS027
Pruebas de regresión	SPS017 SPS022
Pruebas de rendimiento	SPS002 SPS007 SPS009 SPS027 SPS029 SPS030
Pruebas de resiliencia	SPS030
Pruebas de carga	SPS029

Fuente: Elaboración propia

Tabla 10. Sps por pruebas funcionales y pruebas no funcionales.

Tipo de prueba	SPS
Pruebas funcionales	SPS001 SPS003 SPS004 SPS005 SPS006 SPS007 SPS008 SPS009 SPS010 SPS011 SPS012 SPS013 SPS014 SPS015 SPS016 SPS017 SPS018 SPS019 SPS020 SPS021 SPS022 SPS023 SPS024 SPS025 SPS026 SPS027 SPS028 SPS030 SPS031 SPS032
Pruebas no funcionales	SPS002 SPS004 SPS005 SPS006 SPS007 SPS008 SPS009 SPS010 SPS011 SPS012 SPS013 SPS014 SPS015 SPS016 SPS018 SPS019 SPS021 SPS024 SPS027 SPS029 SPS030 SPS031 SPS032

Fuente: Elaboración propia

Tabla 11. Sps por problema

Problema	SPS
Difícil de monitorear	SPS002 SPS028 SPS030
Alto consumo de recursos	SPS004 SPS017
Tedioso de realizar	SPS005 SPS025
Dificultad para plantear una estrategia	SPS015 SPS017 SPS030
Dificultad para determinar el componente que falla	SPS012 SPS023
Dificultad de replicar todo el sistema	SPS003 SPS006 SPS019 SPS027 SPS031
Las pruebas tienden a ser engañosas	SPS006 SPS010
Alta complejidad	SPS002 SPS003 SPS004 SPS005 SPS007 SPS008 SPS009 SPS010 SPS011 SPS013 SPS016 SPS018 SPS020 SPS021 SPS022 SPS023 SPS024 SPS025 SPS027 SPS028 SPS029 SPS030 SPS032
Baja facilidad para pruebas	SPS001 SPS004 SPS030

Fuente: Elaboración propia

Tabla 12. Sps por causa

Causa	SPS
Tiempos de espera	SPS002
Versiones de servicios	SPS002 SPS006 SPS015 SPS030 SPS032
Uso de diversos lenguajes de programación	SPS026 SPS027 SPS030
Tiempo de respuesta lento	SPS010 SPS017 SPS023
Falta de investigación	SPS009 SPS010 SPS013 SPS017 SPS018 SPS030
La necesidad de usar Dobles de Prueba	SPS008
Alto número de servicios	SPS001 SPS002 SPS009 SPS014 SPS019 SPS025 SPS028 SPS031
Falta de herramientas	SPS001 SPS009 SPS018 SPS030

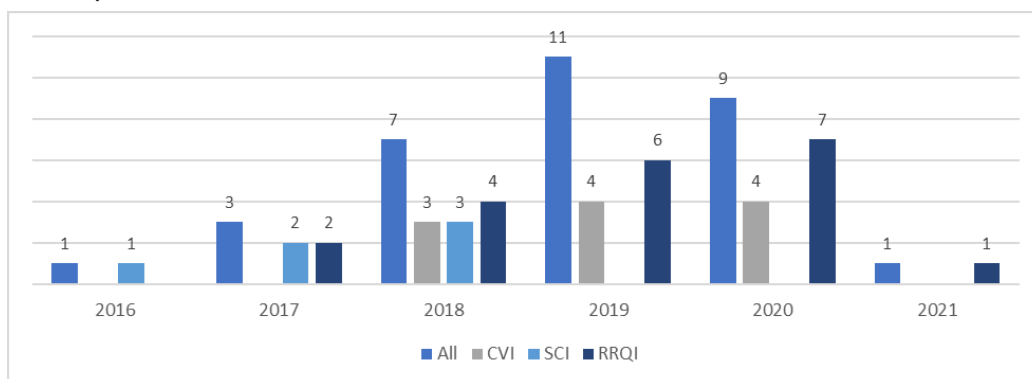
Fuente: Elaboración propia

### Etapa 6: Análisis de datos

#### Descripción general de los SPSs

La Figura 1 evidencia para el 2019 un aumento en la publicación de estudios que documentan problemas para la realización de pruebas en MSA, que corresponde al 34% de los SPSs. Por otra parte, los años 2016 y 2021 son los años con menos relevancia aportando cada uno sólo el 3% de los estudios seleccionados para este SMS.

Figura 1. SPSs por año

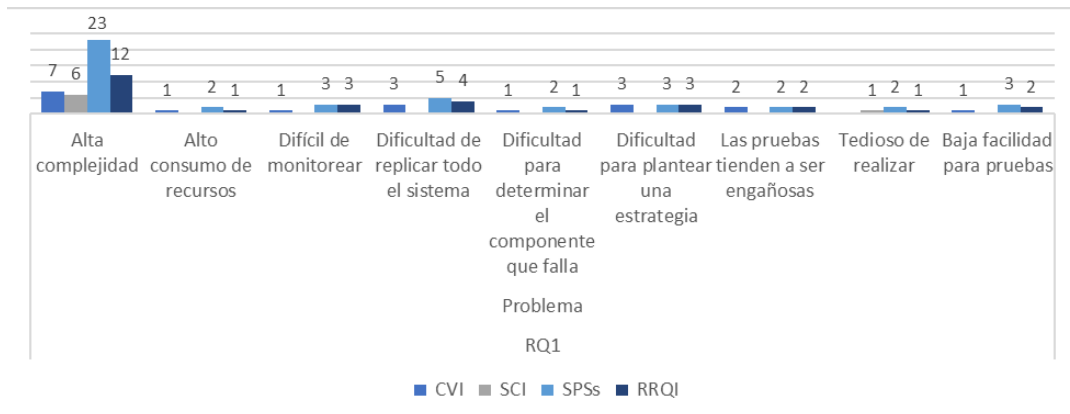


Fuente: Elaboración propia

La Figura 2 muestra que el problema “Alta complejidad” se destaca en aspectos como los siguientes: a) Es el más frecuente con 23 SPSs (72%), b) Contiene 6 SPSs (18.7%) altamente citados, c) Cuenta con 7 SPSs (21.8%) que han sido identificados como muy relevantes y d) 12 SPSs (33.33%) tiene relación con ambas preguntas de investigación. Adicionalmente, el problema “Dificultad de replicar todo el sistema” se destaca en los siguientes aspectos: a) Se relaciona en 5 SPSs (15,6%), b) Cuenta con 3 SPSs (9.3%) que han sido identificados como muy relevantes y c) 4 SPSs (12,5%) tiene relación con ambas preguntas de investigación.



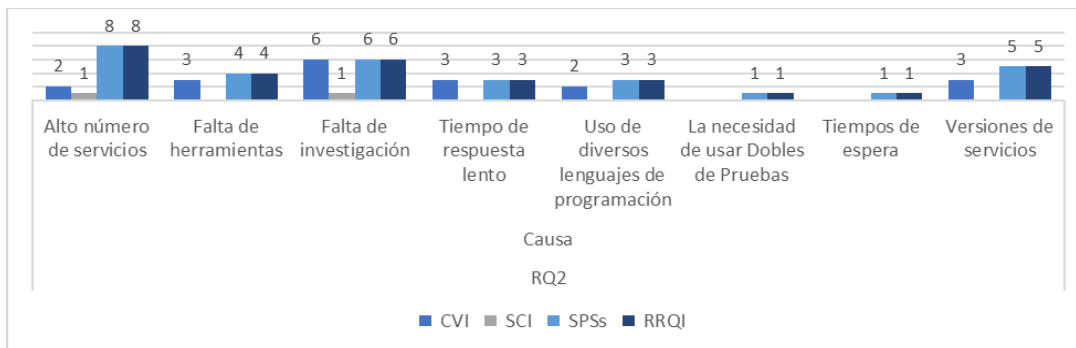
Figura 2. Calidad por categoría Problema



Fuente: Elaboración propia

La Figura 3 muestra que la causa “Alta número de servicio” es la más frecuente con 8 SPSs (25%), que han sido identificados como muy relevantes. La causa “Falta de investigación” tiene 6 SPSs (18,75%) que han sido identificados como muy relevantes y tiene relación con ambas preguntas de investigación. De igual forma también sobresale la causa “Versiones de servicios” con 5 SPSs (15,62%) que tienen una relación con ambas preguntas de investigación.

Figura 3. Calidad por categoría Causa



Fuente: Elaboración propia

## Resultados y Discusión

Con el mapeo realizado fue posible identificar y agrupar en nueve tópicos los problemas reportados sobre la realización de pruebas en MSA, descritos en la Tabla 13, siendo destacados **Alta complejidad y Dificultad al replicar todo el sistema**. Ver Tabla 11. En este sentido, también se identificaron y agruparon en ocho tópicos las causas, descritas en la Tabla 14, destacándose **Alto número de servicios, Falta de investigación y Versiones de servicios** como las causas más documentadas. Ver Tabla 12.

Tabla 13. Definición de problemas

Problema	Definición
Alta complejidad	En los sistemas de microservicios hay una combinación de factores como el alto número de componentes, patrones y elementos interrelacionados que dificultan la realización de pruebas.
Dificultad al replicar todo el sistema	Considerando la naturaleza distribuida de los microservicios, la replicación y configuración de ambiente que emule las condiciones del ambiente de producción no es una actividad trivial.
Dificultad al monitorear	Este elemento hace referencia a la incapacidad de rastrear el flujo de una solicitud realizada por un usuario final a través de pasarelas y múltiples microservicios [32].
Baja facilidad para pruebas	Los sistemas de microservicios no se prestan para la realización de pruebas, no es un sistema típico donde la realización de pruebas sea intuitiva.
Dificultad para plantear una estrategia	En los sistemas de microservicios, es complicado determinar los modelos y mecanismos de prueba para evaluar los atributos de calidad fundamentales de cada microservicio y de la aplicación general desde una perspectiva de autoadaptación [49].
Las pruebas tienden a ser engañosas	Al ejecutar la misma prueba dos veces o más veces se podrían obtener resultados diferentes, las pruebas también podrían comportarse de manera diferente [42].
Tedioso de realizar	Las pruebas en sistemas de microservicios pueden volverse muy complicadas, especialmente cuando el sistema que se está probando es muy grande y hay demasiadas conexiones entre los componentes [31].
Dificultad para determinar el componente que falla	Dado que en los MSA suelen haber muchos componentes generalmente conectados mediante comunicación asíncrona, cuando una ocurre una falla durante la prueba, es difícil ubicar su procedencia.
Alto consumo de recursos	La realización de pruebas en microservicios puede tomar mucho tiempo e incrementar el consumo de memoria en el equipo [38]

Fuente: Elaboración propia

Tabla 14. Definición de causas

Causa	Definición
Alto número de servicios	Los sistemas MSA suelen estar conformados por una gran cantidad de servicios
Versiones de servicios	Se refiere a los diferentes tipos de cambios que pueden ocurrir en un servicio y las políticas de gobierno para administrar estos cambios en sus servicios [55].
Falta de investigación	No hay suficientes estudios publicados relacionados con las pruebas de sistemas de microservicios.
Falta de herramientas	No hay suficientes herramientas o marcos que se puedan usar en la implementación de pruebas en sistemas de microservicios.
Tiempo de respuesta lento	Probar exhaustivamente una gran aplicación de microservicios puede ser muy lento [33], lo cual se ve agravado debido al uso de comunicación asíncrona donde no se tiene certeza del tiempo que tomará una respuesta [55]
Uso de diferentes lenguajes de programación	Los servicios en un sistema MSA son independientes, se pueden desarrollar en un lenguaje de programación diferente, cada uno con sus propias características y limitaciones
Tiempos de espera	Los requerimientos que se hacen desde un microservicio a otro suelen tener un tiempo de espera máximo, en muchas ocasiones la respuesta tarda más del tiempo establecido y el sistema debe manejar esta inconsistencia en la red.
Necesidad de usar Dobles de Prueba	Doble de Prueba es un término genérico para cualquier caso en el que reemplace un objeto de producción con otro que simule su funcionamiento, esto se realiza para evitar la configuración y levantamiento de un servicio externo en los entornos de prueba. Esta práctica ocasiona que no se obtenga un resultado real sobre cómo se están interrelacionando los diferentes servicios.

Fuente: Elaboración propia

### ***Amenazas de validez***

Algunas amenazas a la validez de este estudio son: a) Sesgo en la selección de estudios, b) Errores en la clasificación de estudios, c) Falta de exactitud en la extracción de datos y e) Errores en la aplicación del protocolo de búsqueda.

### ***Sesgo en la selección de estudios***

Esta amenaza se minimizó mediante la aplicación de las siguientes actividades: 1) Se aplicó el protocolo de construcción de SMS sugerido en los estudios [7, 8], incluyendo los modelos GQM y PICOC. 2) La búsqueda en bases de datos se realizó en 6 librerías reconocidas y con alto impacto en el contexto científico. 3) Se agregaron sinónimos a las cadenas de búsqueda para ampliar el rango de posibles resultados. 4) Las cadenas de búsqueda se construyeron de forma iterativa realizando ajustes para identificar la mayor cantidad de estudios y de mejor calidad. 5) Se utilizó una estrategia híbrida de búsqueda combinando la búsqueda en bases de datos y por bola de nieve; además, se realizaron varias iteraciones de búsqueda por bola de nieve para encontrar la mayor cantidad de estudios relevantes. 6) Se crearon alertas para identificar estudios primarios utilizando Mendeley y Google Scholar. 7) Se realizó la validación de estudios mediante los índices CVI, SCI y RRQI.

### ***Errores en la clasificación de estudios***

Los estudios fueron agrupados de acuerdo con los esquemas de clasificación y tópicos definidos en la fase de planeación relacionándolos con las preguntas de investigación.

### ***Falta de exactitud en la extracción de datos***

Se utilizó el software *SMS-Builder* para la extracción de datos. Este software facilita la extracción de datos de los SPSs mediante la clasificación acordada en la fase de planeación. Además, para reducir posibles sesgos o errores en la extracción de información se realizó una revisión de pares como lo recomienda [7].

### ***Errores en la aplicación del protocolo de búsqueda***

El protocolo de búsqueda también se realizó a través de una revisión de pares, para evitar posibles errores. Un evaluador siguió el protocolo de búsqueda mientras que otro revisaba su trabajo. El software *SMS-Builder* se utilizó con el objetivo de evitar el procesamiento manual de datos evitando así errores durante la aplicación del protocolo de búsqueda.

## **Conclusiones**

Este trabajo presenta los resultados de una investigación tipo SMS enfocada en identificar problemas y causas con relación a pruebas en MSA. Se consideraron estudios desde el 2016 hasta principios de 2021. El SMS muestra que en la realización de pruebas en MSA la “Alta Complejidad” es el problema más representativo en el conjunto de estudios. Del mismo modo, las causas más destacadas son el “Alto Número de Servicios”, la “Falta de Investigación” y las “Versiones de servicios (Service Versioning)”.

Dado los resultados obtenidos, este trabajo realiza un aporte al establecer una base para la proposición de nuevos métodos y estrategias de pruebas en MSA desde una perspectiva de confrontación a las causas y los problemas más relevantes; por otra parte, la disposición de los estudios mapeados a través de este trabajo busca favorecer la localización artículos según las estructuras de clasificación planteadas en la sección de planificación. Como trabajo futuro, se propondrá una estrategia para la realización de pruebas en sistemas de microservicios que pueda confrontar los problemas actuales.

## Referencias bibliográficas

1. P. di Francesco, I. Malavolta, and P. Lago, "Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption," in *Proceedings - 2017 IEEE International Conference on Software Architecture, ICSA 2017*, 2017, pp. 21–30. doi: <https://doi.org/10.1109/ICSA.2017.24>.
2. J. Lewis, "Microservices - A definition of this new architectural term," *Martinfowler.Com*, vol. 25, pp. 1–16, 2014, [Online]. Available: <http://martinfowler.com/articles/microservices.html>
3. N. Dragoni, I. Lanese, S. T. Larsen, M. Mazzara, R. Mustafin, and L. Safina, "Microservices: How to make your application scale," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 10742 LNCS, pp. 95–104. doi: [https://doi.org/10.1007/978-3-319-74313-4\\_8](https://doi.org/10.1007/978-3-319-74313-4_8).
4. N. Alshuqayran, N. Ali, and R. Evans, "A systematic mapping study in microservice architecture," in *Proceedings - 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications, SOCA 2016*, 2016, pp. 44–51. doi: <https://doi.org/10.1109/SOCA.2016.15>.
5. F. H. Vera-Rivera, C. M. Gaona Cuevas, and H. Astudillo, "Desarrollo de aplicaciones basadas en microservicios: tendencias y desafíos de investigación," *Revista Ibérica de Sistemas e Tecnologías de Informação*, vol. E23, no. E23, pp. 107–120, 2019.
6. T. Engel, M. Langermeier, B. Bauer, and A. Hofmann, "Evaluation of microservice architectures: A metric and tool-based approach," in *Lecture Notes in Business Information Processing*, 2018, vol. 317, pp. 74–89. doi: [https://doi.org/10.1007/978-3-319-92901-9\\_8](https://doi.org/10.1007/978-3-319-92901-9_8).
7. B. Kitchenham *et al.*, "Systematic literature reviews in software engineering—a tertiary study," *Information and software technology*, vol. 52, no. 8, pp. 792–805, 2010, doi: <https://doi.org/10.1016/j.infsof.2010.03.006>.
8. D. Budgen, M. Turner, P. Brereton, and B. A. Kitchenham, "Using Mapping Studies in Software Engineering.," in *Ppig*, 2008, vol. 8, pp. 195–204. [Online]. Available: <http://www.ppig.org/papers/20th-budgen.pdf>
9. L. E. Sepúlveda-Rodríguez *et al.*, "Study-based Systematic Mapping Analysis of Cloud Technologies for Leveraging IT Resource and Service Management: The Case Study of the Science Gateway Approach," *Journal of Grid Computing*, vol. 19, no. 4, p. 41, 2021, doi: <https://doi.org/10.1007/s10723-021-09587-7>.
10. S. Li *et al.*, "Understanding and addressing quality attributes of microservices architecture: A Systematic literature review," *Information and Software Technology*, vol. 131, p. 106449, 2021, doi: <https://doi.org/10.1016/j.infsof.2020.106449>.
11. D. Taibi, V. Lenarduzzi, and C. Pahl, "Architectural patterns for microservices: A systematic mapping study," in *CLOSER 2018 - Proceedings of the 8th International Conference on Cloud Computing and Services Science*, 2018, vol. 2018-Janua, pp. 221–232. doi: <https://doi.org/10.5220/0006798302210232>.

12. T. Cerny, M. J. Donahoo, and M. Trnka, "Contextual understanding of microservice architecture," *ACM SIGAPP Applied Computing Review*, vol. 17, no. 4, pp. 29–45, 2018, doi: <https://doi.org/10.1145/3183628.3183631>.
13. P. di Francesco, P. Lago, and I. Malavolta, "Architecting with microservices: A systematic mapping study," *Journal of Systems and Software*, vol. 150, pp. 77–97, 2019, doi: <https://doi.org/10.1016/j.jss.2019.01.001>.
14. I. Ghani, W. M. N. Wan-Kadir, A. Mustafa, and M. I. Babir, "Microservice testing approaches: A systematic literature review," *International Journal of Integrated Engineering*, vol. 11, no. 8, pp. 65–80, 2019, doi: <https://doi.org/10.30880/ijie.2019.11.08.008>.
15. L. M. Hillah et al., "Automation and intelligent scheduling of distributed system functional testing: Model-based functional testing in practice," *International Journal on Software Tools for Technology Transfer*, vol. 19, no. 3, pp. 281–308, 2017, doi: <https://doi.org/10.1007/s10009-016-0440-3>.
16. Y. Wang, L. Cheng, and X. Sun, "Design and Research of Microservice Application Automation Testing Framework," in *Proceedings - 2019 International Conference on Information Technology and Computer Application, ITCA 2019*, 2019, pp. 257–260. doi: <https://doi.org/10.1109/ITCA49981.2019.00063>.
17. M. J. Kargar and A. Hanifzade, "Automation of regression test in microservice architecture," in *2018 4th International Conference on Web Research, ICWR 2018*, 2018, pp. 133–137. doi: <https://doi.org/10.1109/ICWR.2018.8387249>.
18. S. Sivanandan, "Test Automation Framework as a Service (TAFaaS) - scale test automation & devops practices with cloud, containers, and microservice.," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 7C2, pp. 108–111, 2019, [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85070741957&partnerID=40&md5=879423b8e93c0fc922a389dddafa709e>
19. V. R. Basili and G. Caldiera, "The Goal Question Metric Paradigm," *Encyclopedia of Software Engineering - 2 Volume Set*, pp. 528–532, 2000, [Online]. Available: <https://www.cs.umd.edu/~basili/publications/technical/T89.pdf>
20. V. R. Basili, "Software Modeling and Measurement," 1992.
21. I. G. Needleman, "A guide to systematic reviews," *Journal of Clinical Periodontology*, vol. 29, no. SUPPL. 3, pp. 6–9, 2002, doi: <https://doi.org/10.1034/j.1600-051x.29.s3.15.x>.
22. M. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences: A Practical Guide*. John Wiley & Sons, 2008. doi: <https://doi.org/10.1002/9780470754887>.
23. E. Almanasreh, R. Moles, and T. F. Chen, "Evaluation of methods used for estimating content validity," *Research in Social and Administrative Pharmacy*, vol. 15, no. 2, pp. 214–221, 2019, doi: <https://doi.org/10.1016/j.sapharm.2018.03.066>.
24. F. YAGHMAEI, "Content validity and its estimation," *Journal of Medical Education*, vol. 3, no. 1, pp. 25–27, 2003, doi: <https://doi.org/10.22037/jme.v3i1.870>.
25. E. Mourao, M. Kalinowski, L. Murta, E. Mendes, and C. Wohlin, "Investigating the Use of a Hybrid Search Strategy for Systematic Reviews," in *International Symposium on Empirical Software Engineering and Measurement*, 2017, vol. 2017-Novem, pp. 193–198. doi: <https://doi.org/10.1109/ESEM.2017.30>.
26. P. H. Nguyen, M. Kramer, J. Klein, and Y. Le Traon, "An extensive systematic review on the Model-Driven Development of secure systems," *Information and Software Technology*, vol. 68, pp. 62–81, 2015, doi: <https://doi.org/10.1016/j.infsof.2015.08.006>.

27. A. Q. Ali, A. B. M. Sultan, A. A. A. Ghani, and H. Zulzalil, "A Systematic Mapping Study on the Customization Solutions of Software as a Service Applications," *IEEE Access*, vol. 7, pp. 88196–88217, 2019, doi: <https://doi.org/10.1109/ACCESS.2019.2925499>.
28. K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *12th International Conference on Evaluation and Assessment in Software Engineering, EASE 2008*, 2008, pp. 1–10. doi: <https://doi.org/10.14236/ewic/ease2008.8>.
29. M. Waseem, P. Liang, and M. Shahin, "A Systematic Mapping Study on Microservices Architecture in DevOps," *Journal of Systems and Software*, vol. 170, p. 110798, 2020, doi: <https://doi.org/10.1016/j.jss.2020.110798>.
30. S. P. Ma, C. Y. Fan, Y. Chuang, I. H. Liu, and C. W. Lan, "Graph-based and scenario-driven microservice analysis, retrieval, and testing," *Future Generation Computer Systems*, vol. 100, pp. 724–735, 2019, doi: <https://doi.org/10.1016/j.future.2019.05.048>.
31. N. Dragoni *et al.*, "Microservices: Yesterday, today, and tomorrow," in *Present and Ulterior Software Engineering*, M. Mazzara and B. Meyer, Eds. Cham: Springer International Publishing, 2017, pp. 195–216. doi: [https://doi.org/10.1007/978-3-319-67425-4\\_12](https://doi.org/10.1007/978-3-319-67425-4_12).
32. M. Villamizar *et al.*, "Cost comparison of running web applications in the cloud using monolithic, microservice, and AWS Lambda architectures," *Service Oriented Computing and Applications*, vol. 11, no. 2, pp. 233–247, Jun. 2017, doi: <https://doi.org/10.1007/s11761-017-0208-y>.
33. S. Baškarada, V. Nguyen, and A. Koronios, "Architecting Microservices: Practical Opportunities and Challenges," *Journal of Computer Information Systems*, vol. 60, no. 5, pp. 428–436, 2020, doi: <https://doi.org/10.1080/08874417.2018.1520056>.
34. D. Savchenko, G. Radchenko, T. Hynninen, and O. Taipale, "Microservice Test Process: Design and Implementation," *International Journal on Information Technologies & Security*, vol. 10, no. 3, pp. 13–24, 2018.
35. T. Cerny *et al.*, "On Code Analysis Opportunities and Challenges for Enterprise Systems and Microservices," *IEEE Access*, vol. 8, pp. 159449–159470, 2020, doi: <https://doi.org/10.1109/ACCESS.2020.3019985>.
36. V. Lenarduzzi, F. Lomio, N. Saarimäki, and D. Taibi, "Does migrating a monolithic system to microservices decrease the technical debt?," *Journal of Systems and Software*, vol. 169, p. 110710, 2020, doi: <https://doi.org/10.1016/j.jss.2020.110710>.
37. Y. Wang, H. Kadiyala, and J. Rubin, "Promises and challenges of microservices: an exploratory study," *Empirical Software Engineering*, vol. 26, no. 4, pp. 1–45, 2021. doi: <https://doi.org/10.1007/s10664-020-09910-y>.
38. A. Benayache, A. Bilami, S. Barkat, P. Lorenz, and H. Taleb, "MsM: A microservice middleware for smart WSN-based IoT application," *Journal of Network and Computer Applications*, vol. 144, pp. 138–154, 2019, doi: <https://doi.org/10.1016/j.jnca.2019.06.015>.
39. T. Cerny, "Aspect-oriented challenges in system integration with microservices, SOA and IoT," *Enterprise Information Systems*, vol. 13, no. 4, pp. 467–489, 2019, doi: <https://doi.org/10.1080/17517575.2018.1462406>.
40. X. Larrucea, I. Santamaria, R. Colomo-Palacios, and C. Ebert, "Microservices," *IEEE Software*, vol. 35, no. 3, pp. 96–100, 2018.
41. A. Nebel and L. González, "Quality-based Methodology for Assessing the Applicability of Microservices Architecture," *23rd Iberoamerican Conference on Software Engineering, CibSE 2020*, 2020.

42. A. Arcuri, "RESTful API automated test case generation," in *Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security, QRS 2017*, 2017, pp. 9–20. doi: <https://doi.org/10.1109/QRS.2017.11>.
43. A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture," *IEEE Software*, vol. 33, no. 3, pp. 42–52, 2016, doi: <https://doi.org/10.1109/MS.2016.64>.
44. A. Avritzer *et al.*, "Scalability Assessment of Microservice Architecture Deployment Configurations: A Domain-based Approach Leveraging Operational Profiles and Load Tests," *Journal of Systems and Software*, vol. 165, p. 110564, 2020, doi: <https://doi.org/10.1016/j.jss.2020.110564>.
45. R. Pietrantuono, S. Russo, and A. Guerriero, "Testing microservice architectures for operational reliability," *Software Testing Verification and Reliability*, vol. 30, no. 2, Mar. 2020, doi: <https://doi.org/10.1002/stvr.1725>.
46. R. Chhonker, Ananya and de Lemos, "Microservices Architectures and Technical Debt: A Self-adaptation View," pp. 1–2, 2019.
47. C. T. Joseph and K. Chandrasekaran, "Straddling the crevasse: A review of microservice software architecture foundations and recent advancements," *Software - Practice and Experience*, vol. 49, no. 10, pp. 1448–1484, 2019, doi: <https://doi.org/10.1002/spe.2729>.
48. J. Soldani, D. A. Tamburri, and W. J. van den Heuvel, "The pains and gains of microservices: A Systematic grey literature review," *Journal of Systems and Software*, vol. 146, pp. 215–232, 2018, doi: <https://doi.org/10.1016/j.jss.2018.09.082>.
49. N. C. Mendonça, P. Jamshidi, D. Garlan, and C. Pahl, "Developing self-adaptive microservice systems: Challenges and directions," *IEEE Software*, 2019.
50. M. Vigiato, R. Terra, H. Rocha, M. T. Valente, and E. Figueiredo, "Microservices in Practice: A Survey Study," *arXiv*, 2018, [Online]. Available: <http://arxiv.org/abs/1808.04836>
51. N. Raychev, "Test automation in microservice architecture," vol. 8, no. 7. IEEE-SEM, 2020.
52. R. Boncea, A. Zamfiroiu, and I. Bacivarov, "A scalable architecture for automated monitoring of microservices," *Academy of Economic Studies. Economy Informatics*, vol. 18, no. 1, pp. 13–22, 2018, [Online]. Available: <https://search.proquest.com/docview/2172010739?accountid=14542%0Ahttp://dn3nh3eq7d.search.serialssolutions.e+for+automated+monitoring+of+microservices&title=Academy+of+Economic+Studies.+Economy+Inf>
53. F. Auer, V. Lenarduzzi, M. Felderer, and D. Taibi, "From monolithic systems to Microservices: An assessment framework," *Information and Software Technology*, vol. 137. 2021. doi: <https://doi.org/10.1016/j.infsof.2021.106600>.
54. O. Zimmermann, "Microservices tenets: Agile approach to service development and deployment," *Computer Science - Research and Development*, vol. 32, no. 3–4, pp. 301–310, Jul. 2017, doi: <https://doi.org/10.1007/s00450-016-0337-0>.
55. J. Davies, D. Schorow, S. Ray, and D. Rieber, Eds., "Versioning Services BT - The Definitive Guide to SOA: Oracle® Service Bus," Berkeley, CA: Apress, 2008, pp. 341–358. doi: [https://doi.org/10.1007/978-1-4302-1058-0\\_14](https://doi.org/10.1007/978-1-4302-1058-0_14).
56. Chris. Richardson, *Microservices patterns : with examples in Java*. Manning Publications, 2019. [Online]. Available: <https://learning.oreilly.com/library/view/microservices-patterns/9781617294549/>