

**PONTIFICIA UNIVERSIDAD
CATÓLICA DEL PERÚ**

Escuela de Posgrado



Revisión sistemática de la literatura sobre comparativa de los
estilos arquitectónicos SOA y microservicios en entornos de
desarrollo ágil

Tesis para obtener el grado académico de Maestro en Informática con
mención en Ingeniería de Software que presenta:

Davis Gereda Hernandez

Asesor:

Dennis Stephen Cohn Muroy

Lima, 2024

Informe de Similitud

Yo, **Dennis Stephen Cohn Muroy**, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada “**Revisión sistemática de la literatura sobre comparativa de los estilos arquitectónicos SOA y microservicios en entornos de desarrollo ágil**”, del autor **Davis Gereda Hernandez**, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 26%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el **30/07/2024**.
- He revisado con detalle dicho reporte y la Tesis, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: **Lima, 30 de julio del 2024.**

Apellidos y nombres del asesor: Cohn Muroy, Dennis Stephen	
DNI: 43513429	Firma: 
ORCID: 0000-0003-4820-0178	

Dedicatoria

Dedicado a mis padres, por su amor incondicional, respaldo constante y sacrificios invaluable que hicieron posible este logro.



Agradecimientos

A mis padres, Gregorio Gereda y Leonidas Hernandez por ser mi ejemplo de vida y brindarme siempre su apoyo incondicional.

A mi familia, por su comprensión, ánimo y aliento durante este camino.

A mi asesor Dennis Cohn, por su orientación y acertadas recomendaciones durante la elaboración de la presente investigación.

A mis profesores de la maestría, por su sabiduría, guía y enseñanzas que ampliaron mi horizonte académico.



Resumen

(ANTECEDENTES) La búsqueda de arquitecturas eficientes y adaptables ha sido constante en el desarrollo de software, destacando la Arquitectura Orientada a Servicios (SOA) por su capacidad para integrar aplicaciones mediante servicios independientes y reutilizables. Con el avance tecnológico y la necesidad de sistemas escalables y ágiles, han surgido los microservicios, que dividen aplicaciones en partes más pequeñas con contextos y funcionalidades propias, promoviendo modularidad y facilidad de desarrollo. Comparar SOA y microservicios es esencial para entender sus diferencias, ventajas y desventajas en entornos ágiles, y para elegir el enfoque adecuado según los objetivos del proyecto.

(OBJETIVOS) El objetivo es analizar comparativamente SOA y microservicios en entornos de desarrollo ágil, identificando sus contribuciones, similitudes, diferencias, problemas y soluciones.

(MÉTODOS) Se realizó una revisión sistemática de la literatura en bases de datos digitales relevantes.

(RESULTADOS) Se recuperaron 135 estudios, y luego del proceso de selección, se obtuvieron 30 estudios primarios los cuales permitieron dar respuesta a las preguntas planteadas en la presente investigación. Las conferencias es el medio de publicación que presenta mayor número de publicaciones con 19 estudios (63%), seguido de artículos con 8 estudios (27%) y finalmente las revisiones con 3 estudios (10%).

(CONCLUSIONES) Se puede concluir que las similitudes entre SOA y microservicios se dan principalmente en aspectos como interoperabilidad, orientación a servicios, acoplamiento, reutilización y computación distribuida. De igual manera sus diferencias se encuentran en la composición del servicio, autonomía, tamaño, granularidad, comunicación, escalabilidad y gestión de datos.

Palabras clave

Ingeniería de software, Arquitectura de software, SOA, Microservicios, Agilidad.

Abstract

(BACKGROUND) The quest for efficient and adaptable architectures has been a constant trend in software development, with Service-Oriented Architecture (SOA) standing out for its ability to integrate applications through independent and reusable services. With technological advancements and the need for scalable and agile systems, microservices have emerged, breaking applications into smaller parts, each with its own context and functionality, promoting modularity and ease of development. Comparing SOA and microservices is crucial to understanding their differences, advantages, and disadvantages in agile environments, and to choosing the most suitable approach based on project objectives.

(OBJECTIVES) The goal is to perform a comparative analysis of SOA and microservices in agile development environments, identifying their contributions, similarities, differences, issues, and solutions.

(METHODS) A systematic review of the literature was conducted using relevant digital databases.

(RESULTS) 135 studies were recovered, and after the selection process, 30 primary studies were obtained which allowed us to answer the questions posed in this research. Conferences are the publication medium that presents the highest number of publications with 19 studies (63%), followed by articles with 8 studies (27%) and finally reviews with 3 studies (10%).

(CONCLUSIONS) It can be concluded that the similarities between SOA and microservices are mainly in aspects such as interoperability, service orientation, coupling, reuse and distributed computing. Likewise, their differences are found in the composition of the service, autonomy, size, granularity, communication, scalability and data management.

Keywords

Software engineering, Software architecture, SOA, Microservices, Agility.

Índice

Resumen	iv
Índice	vi
Índice de tablas	viii
Índice de figuras	ix
1. Introducción	1
1.1. Problema de Investigación	1
1.2. Antecedentes	1
1.3. Justificación	2
1.4. Objetivos del Estudio	2
1.5. Estructura de la tesis	3
2. Marco Conceptual	4
2.1. Arquitectura Orientada a Servicios (SOA)	4
2.2. Microservicios	5
2.3. Metodologías Ágiles	6
2.3.1. Scrum	6
2.3.2. Kanban	7
2.3.3. Extreme Programming	8
2.3.4. Lean Software Development	9
3. Planeamiento de la Revisión Sistemática de la Literatura	10
3.1. Metodología	10
3.2. Protocolo de Revisión	10
3.3. Preguntas de Investigación	11
3.4. Estrategia de Búsqueda y Selección de Fuentes	14
3.4.1. Definición de Términos y Cadena de Búsqueda	14
3.4.2. Selección de Fuentes	15
3.4.3. Criterios de Inclusión y Exclusión	15
3.4.4. Criterios de Estimación de Calidad	16
3.5. Estrategia de Extracción de Datos	16
4. Ejecución de la Revisión Sistemática de Literatura	18
4.1. Estudios relacionados	18
4.2. Búsqueda de Estudios Primarios	18
4.3. Selección de Estudios Primarios	20
	vi

4.4.	Evaluación de la Calidad	22
4.5.	Extracción y Síntesis de Datos	23
6.	Resultados de la Revisión Sistemática de Literatura	25
5.1.	Preguntas Bibliométricas	25
5.1.1.	Distribución de estudios (PB-1)	25
5.1.2.	Tendencia de los estudios (PB-2)	26
5.1.3.	Contribución de países (PB-3)	26
5.1.4.	Medios de publicación de estudios (PB-4)	26
5.1.5.	Congresos y revistas de publicación (PB-5)	27
5.2.	Preguntas de Investigación	27
5.2.1.	Contribuciones y tipos de investigación (PI-1)	27
5.2.2.	Sectores de aplicación de estilos arquitectónicos (PI-2)	28
5.2.3.	Principales similitudes y diferencias entre los estilos arquitectónicos (PI-3)	29
5.2.4.	Factores de influencia en elección de estilos arquitectónicos (PI-4)	30
5.2.5.	Problemas de aplicación y soluciones propuestas (PI-5)	31
5.2.6.	Enfoques de definición de estilos arquitectónicos (PI-6)	32
5.2.7.	Uso combinado de estilos arquitectónicos (PI-7)	32
7.	Amenazas a la Validez	34
6.1.	Validez del Constructo	34
6.2.	Validez Interna	34
6.3.	Validez Externa	34
6.4.	Validez de las Conclusiones	35
8.	Conclusiones y Trabajos Futuros	36
	Referencias bibliográficas	38
	Anexos	43
	Anexo A: Cadenas de búsqueda por base de datos	43
	Anexo B: Proceso de filtrado de estudio	46
	Anexo C: Listado detallado de estudios primarios	55
	Anexo D: Listado detallado de fuentes de publicación	57

Índice de tablas

Tabla 3.1: Criterios de la técnica PICOC	11
Tabla 3.2: PICOC para las preguntas PB-1 al PB-5	11
Tabla 3.3: PICOC para la pregunta PI-1 al PI-3	12
Tabla 3.4: PICOC para la pregunta PI-4	13
Tabla 3.5: PICOC para la pregunta PI-5	13
Tabla 3.6: PICOC para la pregunta PI-6 y PI-7	13
Tabla 3.7: Términos de búsqueda derivados del PICOC	14
Tabla 3.8: Cadena de búsqueda por pregunta de investigación	15
Tabla 3.9: Bases de datos consultadas	15
Tabla 3.14: Preguntas para el aseguramiento de la calidad	16
Tabla 3.15: Formulario de extracción de datos	17
Tabla 4.1: Pasos en la búsqueda de literatura	19
Tabla 4.2: Resumen del proceso de selección de estudios	20
Tabla 4.3: Listado de estudios primarios	22
Tabla 4.4: Resultado de la evaluación de la calidad	23
Tabla 4.5: Fuentes de publicación relevantes	24
Tabla A.1: Cadena de búsqueda por base de datos – Cadena 1	43
Tabla A.2: Cadena de búsqueda por base de datos – Cadena 2	43
Tabla A.3: Cadena de búsqueda por base de datos – Cadena 3	44
Tabla A.4: Cadena de búsqueda por base de datos – Cadena 4	45
Tabla B.1: Detalle del proceso de filtrado de estudio	46
Tabla C.1: Lista detallada de estudios primarios	55
Tabla D.1: Lista de fuentes de publicación de los estudios	57

Índice de figuras

Figura 2.1: Arquitectura Básica Orientada a Servicios	5
Figura 2.2: Arquitectura de Microservicios – Aplicación de Base de Datos	5
Figura 2.3: Flujo del Proceso Scrum	7
Figura 2.4: Tablero Kanban	8
Figura 2.5: Ciclo de Programación Extrema	8
Figura 3.1: Protocolo de Revisión Sistemática propuesto por Ahmad et al.	10
Figura 4.1: Resultado de Búsqueda Primaria	19
Figura 4.2: Resultado de búsqueda y selección de estudios	21
Figura 4.3: Publicaciones por año	24
Figura 4.4: Estudios por tipo de publicación	24
Figura 5.1: Estudios por base de datos	25
Figura 5.2: Estudios por año	26
Figura 5.3: Estudios por país	26
Figura 5.4: Estudios por medio de publicación	27
Figura 5.5: Tipo de fuente de publicación	27



Capítulo 1

Introducción

El presente capítulo está compuesto por 5 secciones. La Sección 1.1 presenta la problemática sobre la cual se basa la presente investigación, la Sección 1.2 presenta estudios previos relacionados al tema de investigación, la Sección 1.3 presenta la justificación de la investigación; finalmente la Sección 1.4 presenta los objetivos de la investigación.

1.1. Problema de Investigación

En la actualidad muchas organizaciones están en busca de una arquitectura de software que permita brindar un adecuado soporte a sus modelos de negocio, presentándose la dificultad de tener que elegir un estilo arquitectónico distribuido que mejor se adapte a sus objetivos organizacionales.

De acuerdo con Power et al. [2], muchas organizaciones luchan con enfoques de toma de decisiones de arquitectura eficiente. A menudo, los enfoques de toma de decisiones no se articulan ni se entienden. Este problema es particularmente evidente en organizaciones grandes distribuidas globalmente con múltiples productos y sistemas grandes.

Además, según Yli-Huumo et al. [3], existe una deuda técnica por la inadecuada decisión de la arquitectura que se aplica. El autor realizó una investigación en una gran organización de desarrollo de software para recopilar evidencia empírica relacionada con el concepto de gestión técnica de la deuda (TDM).

Asimismo, según Gordon et al. [4], en su estudio destaca los altos gastos en adquisición de hardware y/o software que soportan las arquitecturas. Finalmente, de acuerdo con Blinowski et al. [5], nos dice que la principal preocupación es el costo de la escalabilidad.

1.2. Antecedentes

En la literatura, se han realizado algunos estudios relacionados al tema de investigación donde los autores presentan trabajos que se centran en métricas de medición automática de la capacidad de mantenimiento de sistemas basados en servicios (SBS). Asimismo, algunos autores presentan estudios que plantean diferentes desafíos relacionados a la Arquitectura Orientada a Servicios (SOA) y Arquitectura de Microservicios (MSA).

Un estudio realizado por Bogner et al. [16] del 2017 propone una selección de métricas de mantenibilidad para sistemas basados en servicios (SBS) considerando limitaciones de Microservicios como el número de pequeños servicios, heterogeneidad tecnológica y descentralización del control. Si bien el estudio de Bogner et al. [16] no presenta un enfoque comparativo entre estilos arquitectónicos, plantea que los paradigmas ágiles y la orientación a servicios llevan hacia los

Microservicios y que estos son considerados como una evolución a partir de SOA. En el mismo sentido, el autor Newman [17], propone pensar la relación entre Microservicios y SOA del mismo modo que XP o Scrum son enfoques específicos para el desarrollo ágil de software.

Asimismo, se han encontrado algunos estudios fuera del contexto ágil como los de Hamzehloui et al. [18] y Vural et al. [19] que presentan definiciones sobre Microservicios, sus tamaños, límites de la arquitectura y su relación con SOA y Diseño Dirigido por el Dominio (DDD). Además, Vural et al. [19] presenta las nuevas tendencias sobre los estilos arquitectónicos Microservicios y SOA, analiza los estándares emergentes, los tipos de investigación realizados y las motivaciones prácticas para llevar a cabo la arquitectura de microservicios.

Por otro lado, Mito de Oliveira dos Santos et al. [20] el 2020 efectuó un estudio que se enfoca en identificar métodos de selección de criticidad asociados a la Arquitectura de Microservicios (MSA), SOA y componentes, considerando que estos últimos forman parte de los antecedentes y la evolución de los microservicios.

Finalmente, estudios como los de Baresi et al. [21], Di Francesco [22] y Bogner et al. [23] quienes enfocan sus estudios en la identificación de desafíos sobre Microservicio y SOA. Baresi et al. [21] identifica en su estudio los desafíos a través de una visión evolutiva desde los primeros años de SOA hasta los microservicios para luego analizar cómo se abordan estos desafíos en la práctica, mientras que Bogner et al. [23] en su estudio realiza una exploración sobre los procesos de aseguramiento de la capacidad de evolución para microservicios en la industria, qué herramientas, métricas y patrones utilizan, y qué desafíos se perciben para la capacidad de evolución de los sistemas. Además Di Francesco [22] aborda los desafíos relacionados a la Arquitectura de Microservicios (MSA) como (i) la identificación de las propiedades clave de las arquitecturas de microservicios, (ii) la identificación e investigación de un lenguaje de descripción para diseñar y analizar arquitecturas, (iii) la identificación de los factores que impactan el proceso de migración de aplicaciones existentes hacia MSA.

Si bien estos artículos cubren los puntos previamente mencionados, se ha detectado algunos aspectos que no son cubiertos como la comparación de los estilos arquitectónicos SOA y Microservicios centrándose en sus similitudes y diferencias. Del mismo modo, ninguno de los estudios realiza un análisis de las deficiencias y problemas en la aplicación de los estilos arquitectónicos.

1.3. Justificación

En los últimos años las empresas vienen realizando migraciones de arquitecturas monolíticas a arquitecturas distribuidas, siendo entre ellas las más aplicadas la arquitectura orientada a servicios (SOA) y arquitectura de microservicios según Reupo-Musayón [1]. Además, según Yli-Huumo et al. [3] muchas organizaciones luchan con enfoques de toma de decisiones de arquitectura eficiente. Por ello surge la necesidad de un estudio que permita explorar y analizar los artículos relacionados al uso de los estilos arquitectónicos SOA y Microservicios para identificar beneficios e inconvenientes de su aplicación en entornos de desarrollo ágil.

1.4. Objetivos del Estudio

El objetivo general de este estudio es realizar un análisis comparativo entre los estilos arquitectónicos SOA y Microservicios en la aplicación en entornos de desarrollo ágil. Para poder cubrir el objetivo central del estudio, se definieron los siguientes objetivos específicos que requieren ser completados:

- OE1: Identificar las contribuciones acerca del tema de investigación y los enfoques de definición sobre estilos arquitectónicos propuestos en los estudios.
- OE2: Analizar las principales similitudes y diferencias de los estilos arquitectónicos en estudio y su aplicación en distintos sectores de la industria.
- OE3: Identificar los factores que influyen en la elección de alguno de los estilos arquitectónicos en estudio y analizar los problemas presentados en su adopción.
- OE4: Analizar los casos de aplicación combinada de los estilos arquitectónicos para la solución de problemas.

1.5. Estructura de la tesis

La organización del documento de tesis es el siguiente:

- Capítulo 2: Presenta el marco conceptual del estudio definiendo los principales conceptos empleados en el estudio.
- Capítulo 3: Presenta el planeamiento de la tesis, muestra la metodología aplicada en la revisión sistemática de literatura siguiendo el protocolo de revisión propuesto por Ahmad et al. [25], define las preguntas de investigación planteadas. Además, muestra la estrategia de búsqueda y selección de fuentes del estudio.
- Capítulo 4: Presenta la ejecución de la revisión sistemática de literatura, especifica la búsqueda y selección de estudios primarios; así como, la evaluación de la calidad, extracción y síntesis de datos.
- Capítulo 5: Presenta los resultados de la revisión sistemática de literatura.
- Capítulo 6: Presenta las amenazas a la validez del estudio.
- Capítulo 7: Presenta las conclusiones de la tesis; además, los trabajos futuros propuestos.
- Anexos: Presenta listados detallados y complementarios con información usada en la investigación.
 - Anexo A: Cadenas de búsqueda por base de datos
 - Anexo B: Proceso de filtrado de estudio
 - Anexo C: Listado detallado de estudios primarios
 - Anexo D: Listado detallado de fuentes de publicación

Capítulo 2

Marco Conceptual

El presente capítulo está compuesto por 3 secciones. La Sección 2.1 presenta definiciones sobre Arquitectura Orientada a Servicios (SOA), la sección 2.2 presenta definiciones sobre Microservicios; finalmente la sección 2.3 presenta definiciones sobre Metodologías Ágiles que son usadas en la presente investigación.

2.1. Arquitectura Orientada a Servicios (SOA)

Según Jammes y Smith [6], una arquitectura orientada a servicios (SOA) es esencialmente una colección de servicios. Estos servicios se comunican entre sí. SOA es un conjunto flexible de principios de diseño que proporcionará un conjunto de servicios poco integrados que se puede utilizar en múltiples dominios empresariales. Se crean de forma independiente entre sí y funcionan independientemente de su entorno, ya que proporcionan una funcionalidad autónoma.

Según Papazoglou [7], SOA es una forma de reorganizar una cartera de aplicaciones de software e infraestructura de soporte previamente aisladas en un conjunto interconectado de servicios, cada uno accesible a través de interfaces estándar y protocolos de mensajería. Una vez que todos los elementos de una arquitectura empresarial están en su lugar, las aplicaciones existentes y futuras pueden acceder a estos servicios según sea necesario sin necesidad de enrevesadas soluciones punto a punto basadas en protocolos propietarios inescrutables. Este enfoque arquitectónico es particularmente aplicable cuando varias aplicaciones que se ejecutan en diversas tecnologías y plataformas necesitan comunicarse entre sí. De esta forma, las empresas pueden mezclar y combinar servicios para realizar transacciones comerciales con un mínimo esfuerzo de programación.

Asimismo, para Papazoglou [7] SOA no es una arquitectura que se refiere únicamente a servicios, es una relación de tres participantes: el proveedor de servicios, la agencia de descubrimiento de servicios y el solicitante de servicios (cliente). Las operaciones de publicación, búsqueda y vinculación son parte de las interacciones. Además, define una interacción entre agentes de software como un intercambio de mensajes entre clientes y proveedores de servicios como se muestra en la Figura 2.1.

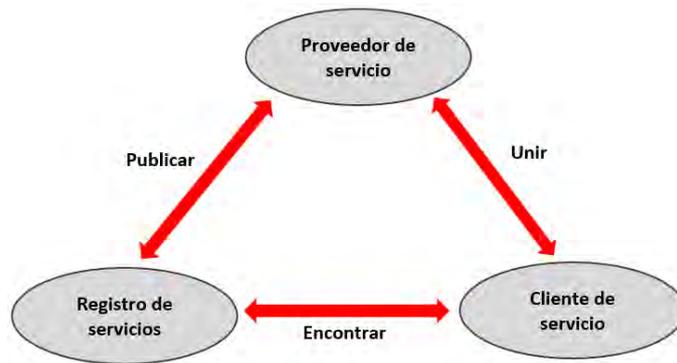


Figura 2.1: Arquitectura Básica Orientada a Servicios [7]

2.2. Microservicios

Según Fowler y Lewis [8], el estilo arquitectónico de microservicios es un enfoque para desarrollar una sola aplicación como un conjunto de pequeños servicios, cada uno ejecutándose en su propio proceso y comunicándose con mecanismos livianos, a menudo una API de recursos HTTP. Estos servicios se basan en capacidades comerciales y se implementan de forma independiente mediante maquinaria de implementación totalmente automatizada. Hay un mínimo indispensable de gestión centralizada de estos servicios, que pueden estar escritos en diferentes lenguajes de programación y utilizar diferentes tecnologías de almacenamiento de datos. Además de descentralizar las decisiones sobre modelos conceptuales, los microservicios también descentralizan las decisiones de almacenamiento de datos. Los microservicios prefieren dejar que cada servicio administre su propia base de datos, ya sea instancias diferentes de la misma tecnología de base de datos o sistemas de base de datos completamente diferentes, un enfoque llamado persistencia políglota como se muestra en la Figura 2.2.

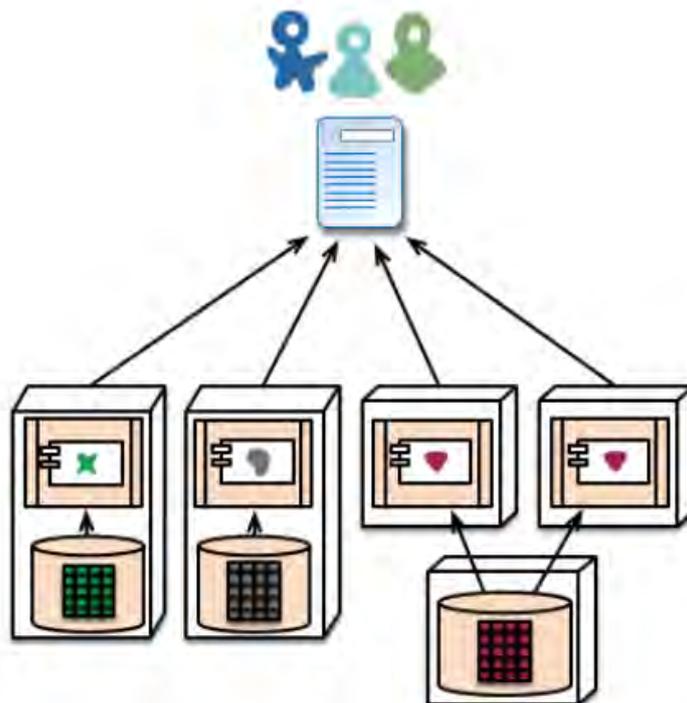


Figura 2.2: Arquitectura de Microservicios – Aplicación de Base de Datos [8]

Para Thönes [9] un microservicio es una pequeña aplicación que se puede implementar de forma independiente, escalar de forma independiente y probar de forma independiente y que tiene una sola responsabilidad. Es una sola responsabilidad en el sentido original de que tiene una sola razón para cambiar y/o una sola razón para ser reemplazado. Pero el otro eje es una sola responsabilidad en el sentido de que hace una sola cosa y solo una cosa y se puede entender fácilmente.

Según Avritzer et. al [10], el estilo arquitectónico de microservicios es un enfoque para crear aplicaciones de software como una colección de componentes de software débilmente acoplados. Estos componentes se denominan microservicios y se supone que son autónomos, implementables de forma automática e independiente y cohesivos. Esta arquitectura se presta a la implementación descentralizada y a la integración e implementación continuas por parte de los desarrolladores.

2.3. Metodologías Ágiles

De acuerdo a Muñoz et. al [11], la agilidad es la capacidad de crear y responder a cambios para obtener ganancias en entornos turbulentos. Por lo tanto, las metodologías ágiles definen cómo debe ser el desarrollo del software de acuerdo con los valores y principios y el manifiesto ágil para responder rápidamente a los problemas y resolverlos a lo largo de las etapas del desarrollo del software.

A continuación, se describen los principales marcos de trabajo ágiles referenciados en la presente investigación.

2.3.1. Scrum

Para Sutherland et. al [62], Scrum es un marco de trabajo sencillo y adaptable utilizado para la gestión de proyectos de desarrollo de software. Este ayuda en la organización de equipos y a tener el trabajo hecho de manera productiva y con mayor calidad. Según Faniran et. al [12], Jeff Sutherland y su equipo de desarrollo idearon Scrum a principios de los años 1990, luego fue formalizado para la industria del software por Ken Schwaber.

Scrum tiene como objetivo aumentar la energía, el enfoque, la claridad y la transparencia en los equipos de proyecto que desarrollan sistemas de software. Los procedimientos que se utilizan en Scrum siguen los principios del enfoque ágil. Los principios del enfoque ágil se centran en: satisfacer al cliente mediante la entrega temprana de software valioso; dar la bienvenida a los requisitos cambiantes; colaboración entre desarrolladores y empresarios; software que funcione (como medida del progreso); mantener el diseño simple y hacer que los equipos piensen en cómo ser más eficientes durante el curso del desarrollo.

Scrum es conocido por importantes procesos de software que lo distinguen de otros enfoques ágiles, desarrollo de software eficiente, procesos unificados ágiles, etc. como se muestra en la Figura 2.3. Los procesos Scrum incluyen:

- **Backlogs:** Es una lista de requisitos funcionales en una escala de preferencia, que podría cambiar durante el desarrollo.
- **Sprints:** Son unidades de trabajo necesarias para lograr un requisito en el trabajo pendiente en un plazo (normalmente 30 días).

- **Reuniones de Scrum (Daily Scrum):** El Daily Scrum es un evento de 15 minutos (máximo) para los desarrolladores del equipo de Scrum. El propósito del Daily Scrum es inspeccionar el progreso hacia el Objetivo Sprint y adaptar el Sprint Backlog según sea necesario, ajustando el próximo trabajo planeado. Las reuniones de Scrum mejoran la comunicación, identifican impedimentos, promueven una rápida para la toma de decisiones, y en consecuencia, eliminan la necesidad de otras reuniones [63].
- **Demostraciones (Sprint Review):** Son los incrementos de software entregados al cliente para su evaluación, que incluyen funcionalidades solicitadas para ser entregadas dentro de un plazo establecido.

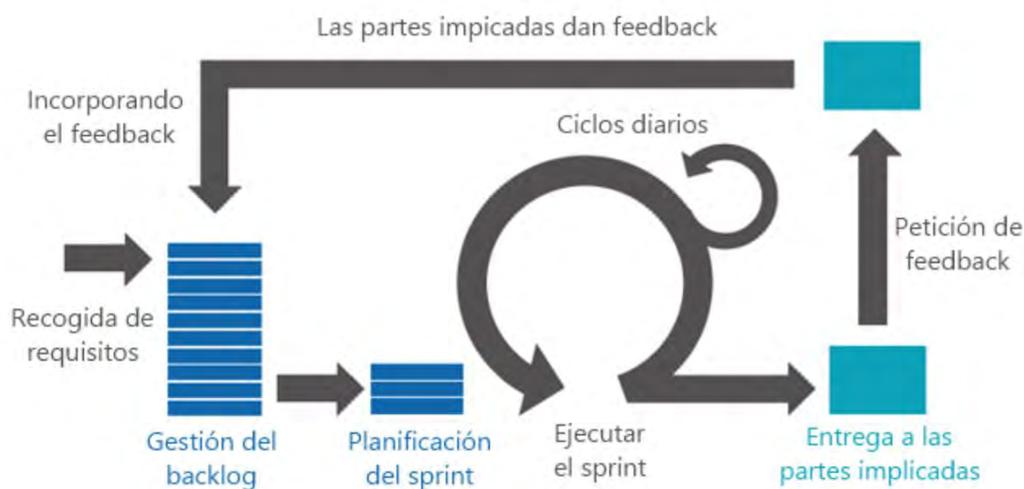


Figura 2.3: Flujo del Proceso Scrum [61]

2.3.2. Kanban

De acuerdo a Saranya y Fumio [13], Kanban proviene del término japonés que significa señal o registro visible. Kanban simplemente es una herramienta para ayudar a los sistemas de extracción y flujo de producto a trabajar juntos, no establece planes de producción. Una tarjeta Kanban se puede usar para informar a un empleado si necesita materiales o tareas de la temporada anterior permitiendo mover o trabajar piezas. Los sistemas Kanban exigen que cada pieza o lote se mueva o se trabaje sin una de estas tarjetas. Para eliminar los cuellos de botella, la demanda de cada proceso utiliza la tarjeta Kanban para llamar al producto y producir el producto por el que pasan los materiales de manera uniforme y equilibrada.

Kanban se organiza con un tablón grande dividido en columnas como se muestra en la Figura 2.4. El número de columnas pueden variar dependiendo del nivel de complejidad o fases del proceso [64]. A continuación, se explica las más usadas para el desarrollo de software:

- **Lista de tareas (To do):** Se incluyen las tareas que aún están pendientes y que se pueden completar de inmediato. Las tareas de mayor prioridad se colocan arriba y las demás se colocan en orden descendente para que puedan pasar a otras columnas.

- **En desarrollo (Doing):** Una tarea se sitúa hasta que se complete. La tarea debe volver a la columna anterior en caso de que algo falle.
- **Pruebas:** Las pruebas o comprobaciones necesarias se realizan para determinar si la tarea se ha completado con éxito. Si todo está correcto, avanzará a la siguiente fase; si no, regresará a la fase de desarrollo. Esta columna puede estar dentro de "Doing" como una columna independiente o como una subcolumna, aunque el resultado y el orden son los mismos.
- **Terminado (Done):** Aquí se incluyen en las tareas que están finalizadas por completo.

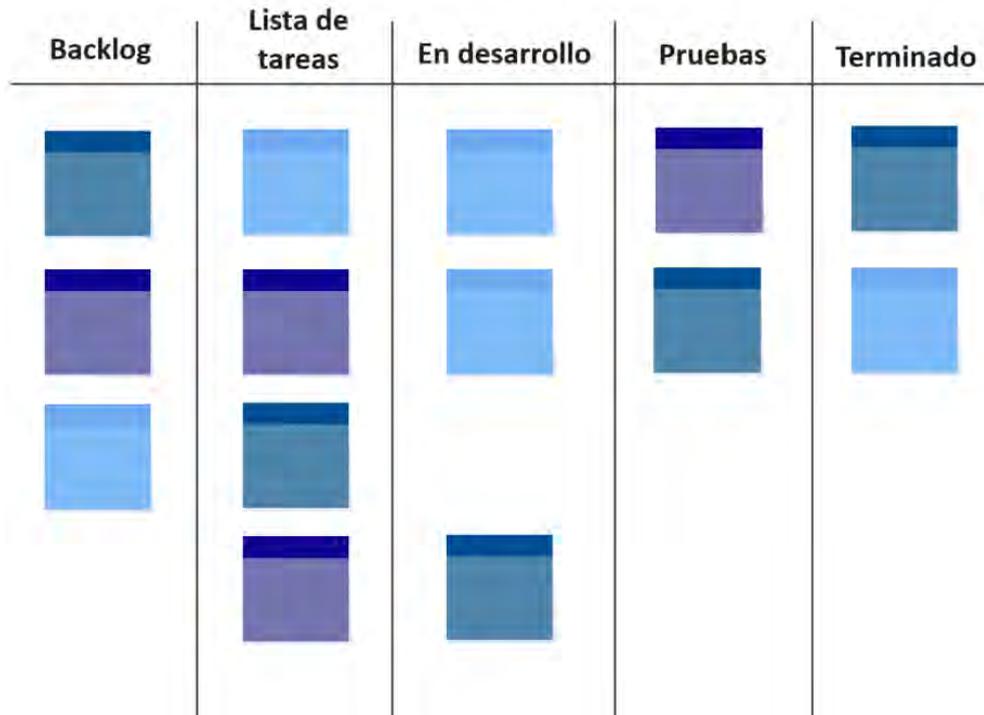


Figura 2.4: Tablero Kanban [65]

2.3.3. Extreme Programming

Según Goto et. al [14], el enfoque de desarrollo ágil de programación extrema (XP) está compuesto por un ciclo de desarrollo de software que consta de las siguientes etapas: planificación, diseño, codificación y prueba como se ilustra en la Figura 2.5. Este ciclo se utiliza para extraer una función del sistema deseado y se repite hasta que se completa el desarrollo del sistema.

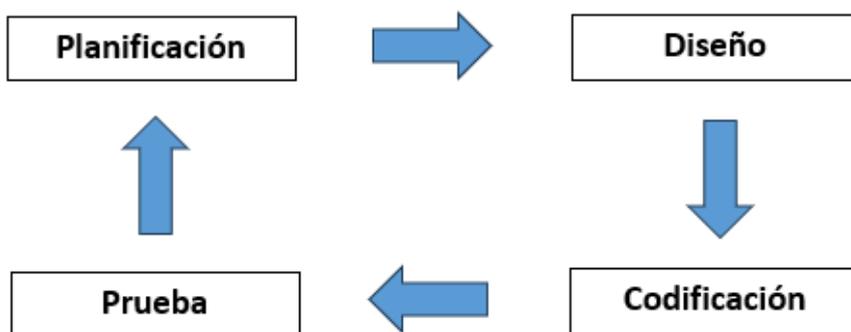


Figura 2.5: Ciclo de Programación Extrema [14]

2.3.4. Lean Software Development

De acuerdo a Anwaar et. al [15], Lean Software Development (LSD) es un proceso de reduplicación que desarrolla todo el proceso para maximizar los resultados. La idea detrás de los procesos ágiles de desarrollo de software se presenta en LSD, que también proporciona un conjunto de ideas o sugerencias para llevar a cabo procesos de software que se adaptarán mejor a sus clientes y partes interesadas, su dominio de trabajo y su capacidad de evolución. El objetivo de todo el proceso es diseñar, desarrollar y realizar todo un conjunto de actividades desde el inicio hasta el producto o servicio específico, con el objetivo de reducir el desperdicio y optimizar el sistema productivo para satisfacer al cliente final.

El desarrollo iterativo, el enfoque en el cliente, el empoderamiento del equipo, la mejora continua y la amplificación del aprendizaje son las principales características de LSD. LSD se define como el proceso de hacer más con cada vez menos esfuerzo humano, equipo, tiempo y espacio, mientras nos acercamos cada vez más a brindar a los clientes lo que quieren.

La metodología básica gira en torno a siete principios: eliminar el desperdicio, construir calidad, crear conocimiento, aplazar el compromiso, entregar rápido, respetar a las personas y optimizar el conjunto.



Capítulo 3

Planeamiento de la Revisión Sistemática de la Literatura

El presente capítulo está compuesto por 5 secciones. La Sección 3.1 presenta la metodología empleada en la investigación y la Sección 3.2. define el protocolo seguido para la revisión, la Sección 3.3 presenta las preguntas de investigación planteadas en el presente estudio, la Sección 3.4 presenta la estrategia de búsqueda aplicada, términos de búsqueda, selección de fuentes, criterios de estimación de calidad, criterios de inclusión y exclusión; finalmente la Sección 3.5 presenta la estrategia de extracción de datos empleada en la investigación.

3.1. Metodología

Para el presente estudio se ha determinado realizar una revisión sistemática de la literatura (Systematic Literature Review, SRL). Según Kitchenham et al. [24], una revisión sistemática de la literatura es un medio para evaluar e interpretar toda la investigación disponible relevante para una pregunta de investigación, área temática o fenómeno de interés en particular. Las revisiones sistemáticas tienen como objetivo presentar una evaluación justa de un tema de investigación mediante el uso de una metodología confiable, rigurosa y auditable.

3.2. Protocolo de Revisión

El presente estudio tiene como objetivo realizar un análisis comparativo entre los estilos arquitectónicos SOA y Microservicios en la aplicación en entornos de desarrollo ágil. Para el proceso de revisión se ha empleado el protocolo propuesto por Ahmad et al. [25] mostrado en la Figura 3.1.

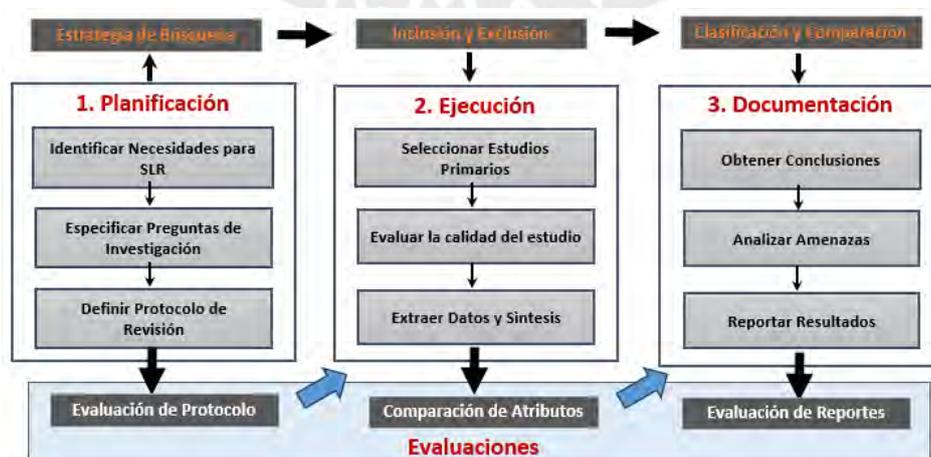


Figura 3.1: Protocolo de Revisión Sistemática propuesto por Ahmad et al. [25]

3.3. Preguntas de Investigación

Las preguntas de investigación planteadas en el presente estudio se han estructurado usando los criterios PICOC (Población, Intervención, Comparación, Resultados y Contexto); los cuales se encuentran detallados en la Tabla 3.1.

Tabla 3.1: Criterios de la técnica PICOC

Criterio	Significado
Población	¿Quién o Qué?
Intervención	¿Cómo?
Comparación	¿Cuál es la alternativa con la que se está comparando?
Resultados	¿Qué estamos tratando de obtener, mejorar, proponer?
Contexto	¿Bajo qué circunstancias?

Con relación a lo antes expuesto, para el presente estudio se han formulado cinco preguntas bibliométricas y siete preguntas de investigación las cuales se detalla a continuación.

PB-1: ¿Cuál es la distribución de estudios relacionados con el tema de investigación en cada una de las Bases de Datos seleccionadas?

En la estructuración de esta pregunta se ha usado los elementos Población, Intervención, Comparación y Contexto (PICoC) de la técnica PICOC como se muestra en la Tabla 3.2. La motivación de esta pregunta es identificar los artículos relacionados con el tema de investigación publicados por base de datos.

Tabla 3.2: PICOC para las preguntas PB-1 al PB-5

Criterio	Ámbito de Investigación
Población	Proceso de Construcción de Software, Arquitectura de software, Arquitectura de aplicaciones, Arquitectura ágil
Intervención	Estilo arquitectónico SOA (Arquitectura Orientado a Servicios)
Comparación	Estilo arquitectónico Microservicios
Contexto	Aplicación de los estilos arquitectónicos en entornos de desarrollo ágil

PB-2: ¿Cuál es la tendencia de interés sobre las arquitecturas de software SOA y Microservicios?

Con la formulación de esta pregunta se pretende identificar los años en que se publican los estudios relacionados al tema de investigación. La estructuración de esta pregunta es la usada en todas las preguntas bibliométricas de la presente investigación.

PB-3: ¿Cuáles son los países con mayor contribución de este tipo de investigación?

Esta pregunta busca identificar los países que concentran la mayor cantidad de estudios sobre el tema de investigación. La estructuración de esta pregunta es la usada en todas las preguntas bibliométricas de la presente investigación.

PB-4: ¿Cuáles son los medios de publicación de los artículos con relación al tema de investigación?

El objetivo de esta pregunta es identificar los medios de publicaciones que concentran la mayor cantidad de estudios sobre el tema de investigación. La estructuración de esta pregunta es la usada en todas las preguntas bibliométricas de la presente investigación.

PB-5: ¿En qué Journals y Congresos se publican los artículos con relación al tema de investigación?

La motivación de la formulación de esta pregunta es identificar los journals y congresos que publican mayor cantidad de investigaciones sobre el tema de investigación. La estructuración de esta pregunta es la usada en todas las preguntas bibliométricas de la presente investigación.

PI-1: ¿Qué tipo de investigaciones y contribuciones se han propuesto sobre el tema de investigación?

Esta pregunta busca clasificar los artículos encontrados en el estudio considerando la clasificación propuesta por Wieringa et al. [26] en la cual determina 6 tipos (propuesta de solución, investigación de validación, artículo de opinión, investigación de evaluación, artículo filosófico y artículo de experiencia). El objetivo de esta pregunta es identificar las contribuciones y tipo de investigaciones realizadas sobre el tema del presente estudio. La estructura usada en esta pregunta no aplica el elemento "Resultados" de la técnica PICOC como se muestra en la Tabla 3.3.

PI-2: ¿Cómo se aplican los estilos arquitectónicos del tema de investigación en los distintos sectores?

Esta pregunta busca identificar los sectores como educación, gobierno, construcción, salud u otros que usan los estilos arquitectónicos del tema de investigación. Además, de analizar cómo estos estilos arquitectónicos son aplicados en los sectores identificados. La Tabla 3.3 muestra la estructuración de la pregunta.

PI-3: ¿Cuáles son las principales similitudes y diferencias entre los estilos arquitectónicos del tema de investigación?

El objetivo de esta pregunta es identificar y analizar las similitudes y diferencias entre los estilos arquitectónicos del tema de investigación. La estructura de esta pregunta se presenta en la Tabla 3.3.

PI-4: ¿Cómo influyen los factores encontrados en los estudios para la elección de los estilos arquitectónicos del tema de investigación?

La motivación de la formulación de esta pregunta es identificar y analizar los factores que permiten elegir alguna de las arquitecturas distribuidas que forman parte de la investigación. La estructura de esta pregunta se presenta en la Tabla 3.4.

Tabla 3.3: PICOC para la pregunta PI-1 al PI-3

Criterio	Ámbito de Investigación
Población	Proceso de Construcción de Software, Arquitectura de software, Arquitectura de aplicaciones, Arquitectura ágil
Intervención	Estilo arquitectónico SOA (Arquitectura Orientado a Servicios)
Comparación	Estilo arquitectónico Microservicios
Contexto	Aplicación de los estilos arquitectónicos en entornos de desarrollo ágil

Tabla 3.4: PICOC para la pregunta PI-4

Criterio	Ámbito de Investigación
Población	Proceso de Construcción de Software, Arquitectura de software, Arquitectura de aplicaciones, Arquitectura ágil
Intervención	Estilo arquitectónico SOA (Arquitectura Orientado a Servicios)
Comparación	Estilo arquitectónico Microservicios
Resultados	Ventajas y desventajas de los estilos arquitectónicos. Además, las características de cada uno de los estilos
Contexto	Aplicación de los estilos arquitectónicos en entornos de desarrollo ágil

PI-5: ¿Qué problemas se presentan en la aplicación de los estilos arquitectónicos del tema de investigación en el entorno de desarrollo ágil y cómo son abordados por los autores?

La motivación de la formulación de esta pregunta es identificar y analizar los problemas presentados en la aplicación de los estilos arquitectónicos y cómo estos son abordados por los autores. La estructura de esta pregunta se presenta en la Tabla 3.5.

Tabla 3.5: PICOC para la pregunta PI-5

Criterio	Ámbito de Investigación
Población	Proceso de Construcción de Software, Arquitectura de software, Arquitectura de aplicaciones, Arquitectura ágil
Intervención	Estilo arquitectónico SOA (Arquitectura Orientado a Servicios)
Comparación	Estilo arquitectónico Microservicios
Resultados	Adopción, aplicación o uso de los estilos arquitectónicos. Además de factores que influyen en la elección de uso de los estilos arquitectónicos
Contexto	Aplicación de los estilos arquitectónicos en entornos de desarrollo ágil

PI-6: ¿De qué manera los autores enfocan la definición de estilo arquitectónico relacionado al tema de investigación?

La motivación de la formulación de esta pregunta es analizar el enfoque aplicado por los autores para referirse a las arquitecturas SOA y Microservicios e identificar si el enfoque es de estilos arquitectónicos u otros. La estructura de esta pregunta se presenta en la Tabla 3.6.

PI-7: ¿Es posible el uso combinado de los estilos arquitectónicos del tema de investigación para la solución de problemas y qué mencionan los autores al respecto?

La motivación de la formulación de esta pregunta es analizar el uso combinado de los estilos arquitectónicos y cómo estas se complementan para solucionar problemas. La estructura de esta pregunta se presenta en la Tabla 3.6.

Tabla 3.6: PICOC para la pregunta PI-6 y PI-7

Criterio	Ámbito de Investigación
Población	Proceso de Construcción de Software, Arquitectura de software, Arquitectura de aplicaciones, Arquitectura ágil
Intervención	Estilo arquitectónico SOA (Arquitectura Orientado a Servicios)
Comparación	Estilo arquitectónico Microservicios
Resultados	Problemas y dificultades que se presentan en la aplicación o uso de los estilos arquitectónicos
Contexto	Aplicación de los estilos arquitectónicos en entornos de desarrollo ágil

3.4. Estrategia de Búsqueda y Selección de Fuentes

3.4.1. Definición de Términos y Cadena de Búsqueda

Los términos de búsqueda definidos en la presente investigación se desarrollaron utilizando las sugerencias de Brereton et al. [10], considerando principalmente las siguientes:

- Descomposición de la pregunta de investigación en elementos individuales aplicando la técnica PICOC para obtener los principales términos de búsqueda.
- Evaluación de palabras claves obtenidas de estudios primarios conocidos para la obtención de otros términos de búsqueda principales.
- Identificación de sinónimos para los términos de búsqueda principales.
- Uso del operador booleano “AND” para unir los términos de búsqueda principales y “OR” para incluir sinónimos.

Los términos de búsqueda fueron identificados a partir de los elementos PICOC definidos anteriormente en las Tablas 3.2, 3.3, 3.4, 3.5 y 3.6, teniendo como resultado los términos por cada pregunta de investigación mostrados en la Tabla 3.7.

Tabla 3.7: Términos de búsqueda derivados del PICOC

Criterio	Términos de Búsqueda				
	PB-1 al PB-5	PI-1 al PI-3	PI-4	PI-5	PI-6 y PI-7
Población	(software OR application OR agile) AND (process* OR engineering OR develop* OR design OR architecture)				
Intervención	("service oriented architecture" OR SOA OR "service-oriented")				
Comparación	(microservice* OR "micro-service*")				
Resultados			(advantage* OR disadvantage* OR feature* OR characteristic* OR factor*)	(adoption* OR app* OR factor*)	(difficult* OR problem* OR app* OR use)
Contexto	(agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD)				

A partir de lo expuesto en la Tabla 3.7 y considerando el protocolo propuesto por Ahmad et al. [25] se definió la estrategia de búsqueda para cada una de las preguntas de investigación planteadas, de tal manera que se definieron las siguientes cadenas de búsqueda.

- **Cadena 1:** ((software OR application OR agile) AND (process* OR engineering OR develop* OR design OR architecture)) AND (("service oriented architecture" OR SOA OR "service-oriented")) AND ((microservice* OR "micro-service*")) AND ((agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD))
- **Cadena 2:** Cadena 1 AND ((advantage* OR disadvantage* OR feature* OR characteristic* OR factor*))
- **Cadena 3:** Cadena 1 AND ((adoption* OR app* OR factor*))
- **Cadena 4:** Cadena 1 AND ((difficult* OR problem* OR app* OR use))

La relación entre las cadenas de búsqueda y las preguntas de investigación planteadas se detalla en la Tabla 3.8.

Tabla 3.8: Cadena de búsqueda por pregunta de investigación

Cadena de Búsqueda	Pregunta de Investigación
Cadena 1	PB-1, PB-2, PB-3, PB-4, PB-5, PI-1, PI-6 y PI-7
Cadena 2	PI-4
Cadena 3	PI-5
Cadena 4	PI-6 y PI-7

3.4.2. Selección de Fuentes

Las bases de datos consultadas en la investigación se detallan en la Tabla 3.9. Para la selección de base de datos se tomó como criterio lo siguiente: (i) sugerencia del recurso en línea de la Pontificia Universidad Católica del Perú para las áreas de Ciencias e Ingeniería, (ii) facilidad de aplicación de las cadenas de búsqueda, (iii) la cantidad de estudios resultantes de una búsqueda preliminar de la cadena 1 y (iv) la accesibilidad a los artículos completos.

Tabla 3.9: Bases de datos consultadas

Base de Datos	URL
ACM Digital Library	http://dl.acm.org/
IEEE Xplore	http://ieeexplore.ieee.org/
Scopus	https://www.scopus.com/
Web of Science	http://isiknowledge.com/

A partir de la definición de cadenas de búsqueda por cada pregunta de investigación planteada según lo expuesto en la Tabla 3.8. se determinaron las cadenas de búsqueda usadas en las bases de datos seleccionadas las cuales se detallan en el Anexo A del presente documento de investigación.

3.4.3. Criterios de Inclusión y Exclusión

Para la selección de los estudios se consideró la aplicación de los siguientes criterios de selección.

- **Criterios de Inclusión**

CI1: El artículo está relacionado al campo de arquitectura de software.

CI2: El artículo está relacionado a los estilos arquitectónicos SOA y/o Microservicios.

CI3: El artículo presenta una comparación relacionada al tema de investigación.

CI4: El documento está publicado como artículo, revista o conferencia.

CI5: El artículo se encuentra disponible en texto completo.

- **Criterios de Exclusión**

CE1: El artículo está relacionado a temas ajenos al desarrollo ágil.

CE2: El artículo está escrito en un idioma diferente al inglés, portugués o español.

CE3: El artículo es secundario o terciario.

CE4: El artículo es publicado antes del año 2014.

CE5: El artículo es duplicado.

Respecto al criterio de exclusión CE4, se consideró el año 2014 por ser la fecha en que Martin Fowler introdujo la definición de Microservicios. Acerca del criterio de exclusión CE5, si hubiera artículos duplicados se seleccionará el artículo que contenga una información más completa.

3.4.4. Criterios de Estimación de Calidad

El criterio de estimación de calidad está basado en la propuesta de Zarour et al. [28]. La propuesta presenta los siguientes niveles de calificación para evaluar la calidad de los estudios: Sí cumple = 1 punto, Cumple parcialmente = 0.5 puntos y No cumple = 0 puntos.

El rango de puntajes obtenido es de 0 a 5 puntos como máximo por cada estudio evaluado, en la cual son considerados como aceptados los que obtienen un puntaje mayor o igual a 2.5 puntos. A continuación, se presentan en la Tabla 3.14 las preguntas aplicadas para el aseguramiento de calidad.

Tabla 3.14: Preguntas para el aseguramiento de la calidad

ID	Pregunta
QA1	¿El objetivo de la investigación ha sido suficientemente explicado?
QA2	¿La idea o el enfoque presentado ha sido claramente explicado?
QA3	¿Se han considerado amenazas a la validez?
QA4	¿Hay una adecuada descripción del contexto en donde se ha llevado a cabo la investigación?
QA5	¿Se mencionan de forma clara los hallazgos del estudio?

3.5. Estrategia de Extracción de Datos

Como parte de la estrategia de extracción de datos de los estudios primarios seleccionados para responder a las preguntas de investigación, se definió el formulario de extracción compuesto por campos de información general y campos específicos relacionados a las preguntas de investigación planteadas. A continuación, se detalla el formulario de extracción de datos en la Tabla 3.15.

Tabla 3.15: Formulario de extracción de datos

Campo	Descripción	RQ
ID	Identificador del estudio.	General
Base de datos	Nombre de la base de datos fuente. Responde a la primera pregunta bibliométrica.	PB-1
Año	Año de publicación del estudio. Responde a la segunda pregunta bibliométrica.	PB-2
País	País de publicación del estudio. Responde a la tercera pregunta bibliométrica.	PB-3
Idioma	Idioma del estudio.	General
DOI	Código único del estudio.	General
Medio de publicación	Medio de publicación de estudio. Responde a la cuarta y quinta pregunta bibliométrica.	PB-4 PB-5
Título	Título de estudio.	General
Resumen	Resumen del estudio.	General
Autores	Nombre de los autores del estudio.	General
Palabras claves	Palabras claves de los autores del estudio.	General
Afiliaciones	Instituciones afiliadas de los investigadores del estudio.	General
Enlace web	Enlace de la base de datos donde se encuentra publicado el estudio.	General
Tipo investigación	Indica el tipo de investigación según clasificación propuesto por Wieringa et al. [26]. Asimismo, presenta las contribuciones del estudio. Responde a la primera pregunta de investigación.	PI-1
Sectores	Identifica los sectores y cómo aplican los estilos arquitectónicos del tema de investigación. Responde a la segunda pregunta de investigación.	PI-2
Similitudes y diferencias	Identifica las similitudes y diferencias de los estilos arquitectónicos del tema de investigación. Responde a la tercera pregunta de investigación.	PI-3
Factores	Determina cómo los factores influyen en la elección de estilos arquitectónicos. Responde a la cuarta pregunta de investigación.	PI-4
Problemas	Identifica los problemas que se presentan en la aplicación de los estilos arquitectónicos. Responde a la quinta pregunta de investigación.	PI-5
Enfoque de definición	Identifica los enfoques utilizados por los autores para definir los estilos arquitectónicos. Responde a la sexta pregunta de investigación.	PI-6
Uso combinado	Identifica la posibilidad de uso combinado de estilos arquitectónicos. Responde a la séptima pregunta de investigación.	PI-7

Capítulo 4

Ejecución de la Revisión Sistemática de Literatura

El presente capítulo está compuesto por 5 secciones. La Sección 4.1 presenta los estudios relacionados, la Sección 4.2 presenta la búsqueda de estudios primarios, la Sección 4.3 presenta la selección de estudios primarios; la Sección 4.4 presenta la evaluación de calidad del estudio; finalmente la Sección 4.5 presenta la extracción y síntesis de datos realizada en la investigación.

4.1. Estudios relacionados

En el mes de mayo de 2023 se realizó una búsqueda de estudios relacionados al tema de investigación en la base de datos Scopus con el propósito de verificar que no existan estudios previos que aborden los mismos objetivos planteados en la investigación y así validar la ejecución del presente estudio. De la búsqueda mencionada se obtuvieron como resultado 30 artículos; sin embargo, ninguno de ellos aborda directamente el tema y objetivos planteados en la investigación. La cadena de búsqueda ejecutada fue la siguiente:

```
(TITLE-ABS-KEY ("review of studies" OR "structured review" OR "systematic review" OR "literature review" OR "literature analysis" OR "in-depth survey" OR "literature survey" OR "meta-analysis" OR "past studies" OR "subject matter expert" OR "analysis of research" OR "empirical body of knowledge" OR "evidence-based software engineering" OR "overview of existing research" OR "body of published research" OR "study aggregation" OR "study synthesis" OR "critical review" OR "mapping study" OR "literature mapping" OR "systematic mapping") AND TITLE-ABS-KEY ((software OR application OR agile) AND (process* OR engineering OR develop* OR design OR architecture) AND ("service oriented architecture" OR SOA OR "service-oriented") AND (microservice* OR "micro-service*")))
```

4.2. Búsqueda de Estudios Primarios

Para la búsqueda de estudios primarios se realizó una búsqueda en 2 etapas que se detallan a continuación.

Búsqueda Primaria

En el mes de junio de 2023 se ejecutaron las cadenas de búsqueda según lo especificado en la Tabla 3.8 para las preguntas planteadas en la investigación sin ningún tipo de restricción en las fechas de publicación. La ejecución de las cadenas se realizó en cada una de las fuentes indicadas en la Tabla

3.9. Para el desarrollo de la búsqueda primaria mencionada se siguió el proceso de cinco pasos propuesto por Ahmad et. al [25] detallado en la Tabla 4.1.

Tabla 4.1: Pasos en la búsqueda de literatura

Pasos	Descripción
Elaborar cadenas de búsqueda	Se elaboraron en base a los términos derivados del PICOC detallado en la Tabla 4.7.
Considerar sinónimos	Se consideraron sinónimos de los términos definidos para la búsqueda.
Combinar términos de búsqueda	Se usó el operador AND para vincular los términos principales y el operador OR para incorporar sinónimos.
Dividir cadena de búsqueda	Se dividió las cadenas para su aplicación en las diferentes bases de datos.
Gestionar referencias	Se usó citas con notas finales.

Se obtuvo como resultado de la búsqueda primaria artículos publicados entre los años 2015 y 2022 para cada una de las cadenas. Las cantidades de artículos obtenidos se especifican a continuación:

Para la cadena 1 se obtuvo un total de 135 artículos, para la cadena 2 se obtuvo un total de 45 artículos, para la cadena 3 se obtuvo un total de 112 artículos y finalmente para la cadena 4 se obtuvo un total de 115 artículos como se ilustra en la Figura 4.1. La fuente con mayor número de artículos fue Scopus con 75 artículos, seguido por IEEE Xplore con 49 artículos.

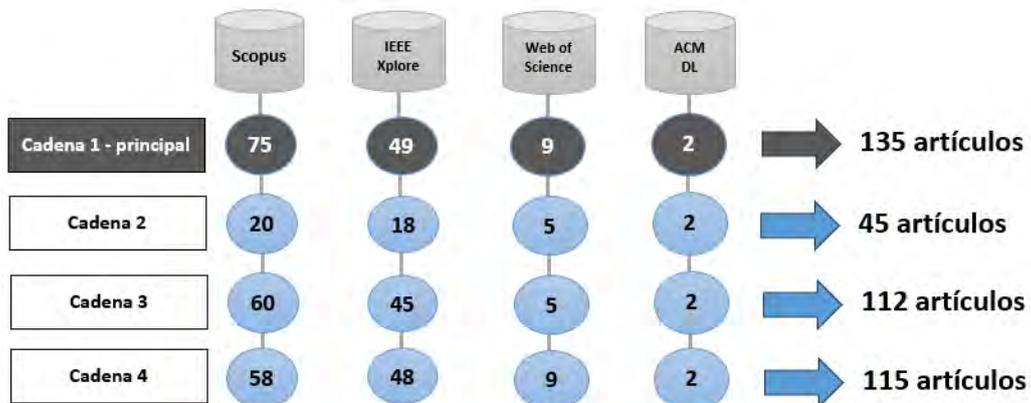


Figura 4.1: Resultado de Búsqueda Primaria

En esta primera etapa de búsqueda primaria no se eliminó artículos por duplicidad; sin embargo, se identificaron 12 artículos duplicados que se descartaron durante el proceso de selección que se detalla en la siguiente sección del presente documento. Asimismo, en la sección 4.3 se detallan los 17 estudios únicos obtenidos después de la aplicación de los criterios de selección.

Búsqueda Secundaria

Después de obtener los 17 estudios resultantes de la búsqueda primaria, se usó la técnica de Bola de Nieve (Snowballing) a un segundo nivel revisando cada uno de los estudios citados por los artículos seleccionados. El total de artículos citados al aplicar la técnica en primer nivel fue 469 y al aplicar el segundo nivel de la técnica se tuvo un total de 305 artículos. Estos artículos fueron sometidos al

cumplimiento de los criterios de selección que se detallan en la siguiente sección de la presente investigación.

4.3. Selección de Estudios Primarios

Considerando el total de 135 artículos encontrados en la búsqueda primaria se realizó la selección de estudios en 2 fases que se detallan a continuación.

Selección Inicial

En la primera fase de selección se revisó los títulos y resúmenes de los estudios descartando rápidamente aquellos que no cumplían con los criterios de selección mencionados previamente en el numeral 3.4.3 de la presente investigación. Como resultado de la selección inicial se excluyeron 93 estudios, quedando 42 estudios que pasaron a la siguiente fase de selección final.

Selección Final

En la fase de selección final se recuperó el texto completo de los 42 estudios y se aplicó los criterios de inclusión y exclusión en cada uno de los estudios detectándose 12 estudios duplicados los cuales fueron descartados. Como resultado de la aplicación de los criterios de selección se excluyeron en total 25 estudios quedando 17 estudios únicos seleccionados a los cuales se le aplicó la técnica de Bola de Nieve (Snowballing) a dos niveles.

Al aplicar el primer nivel del Snowballing se obtuvo un total de 469 publicaciones citadas por los 17 estudios previamente seleccionados, de los cuales después de aplicar los criterios de inclusión y exclusión se obtuvo 8 estudios que fueron incluidos a la lista de estudios primarios seleccionados para la investigación. De igual manera, al aplicar el segundo nivel del Snowballing, se obtuvo un total de 305 publicaciones citadas por los 8 nuevos estudios previamente seleccionados. Estos estudios también fueron sometidos a los criterios de inclusión y exclusión, obteniendo 5 nuevos estudios adicionales que fueron incluidos al listado de estudios primarios, haciendo un total de 30 estudios considerados para la investigación.

En la Figura 4.2 se muestra el resumen del proceso de búsqueda y selección de estudios identificados en cada etapa. En total, la búsqueda realizada obtuvo como resultado 135 estudios, de los cuales, al aplicar los criterios de selección, se excluyeron 118 estudios (incluyendo duplicados) quedando 17 estudios primarios en la búsqueda primaria. Luego se sumaron 13 estudios resultantes de la aplicación de la técnica Snowballing como búsqueda secundaria. El listado de estudios a los que se aplicó el proceso de selección se presenta en el Anexo B del presente documento de investigación.

El resultado final del proceso de búsqueda y selección de los estudios primarios es resumido en la Tabla 4.2.

Tabla 4.2: Resumen del proceso de selección de estudios

Búsqueda	Técnica	Resultados	Fase Inicial - Criterios en Título y Resumen	Fase Final – Revisión Texto Completo
Primaria	PICOC	135	42	17
Secundaria	Snowballing – nivel 1	469	29	8
	Snowballing – nivel 2	305	13	5

Finalmente, se obtuvieron 30 estudios primarios que se muestran en la Tabla 4.3 indicando su año de publicación y la cadena de búsqueda con la cual fue encontrada. El listado detallado de los estudios primarios seleccionados se muestra en la Tabla C.1 del anexo C del presente documento de investigación.

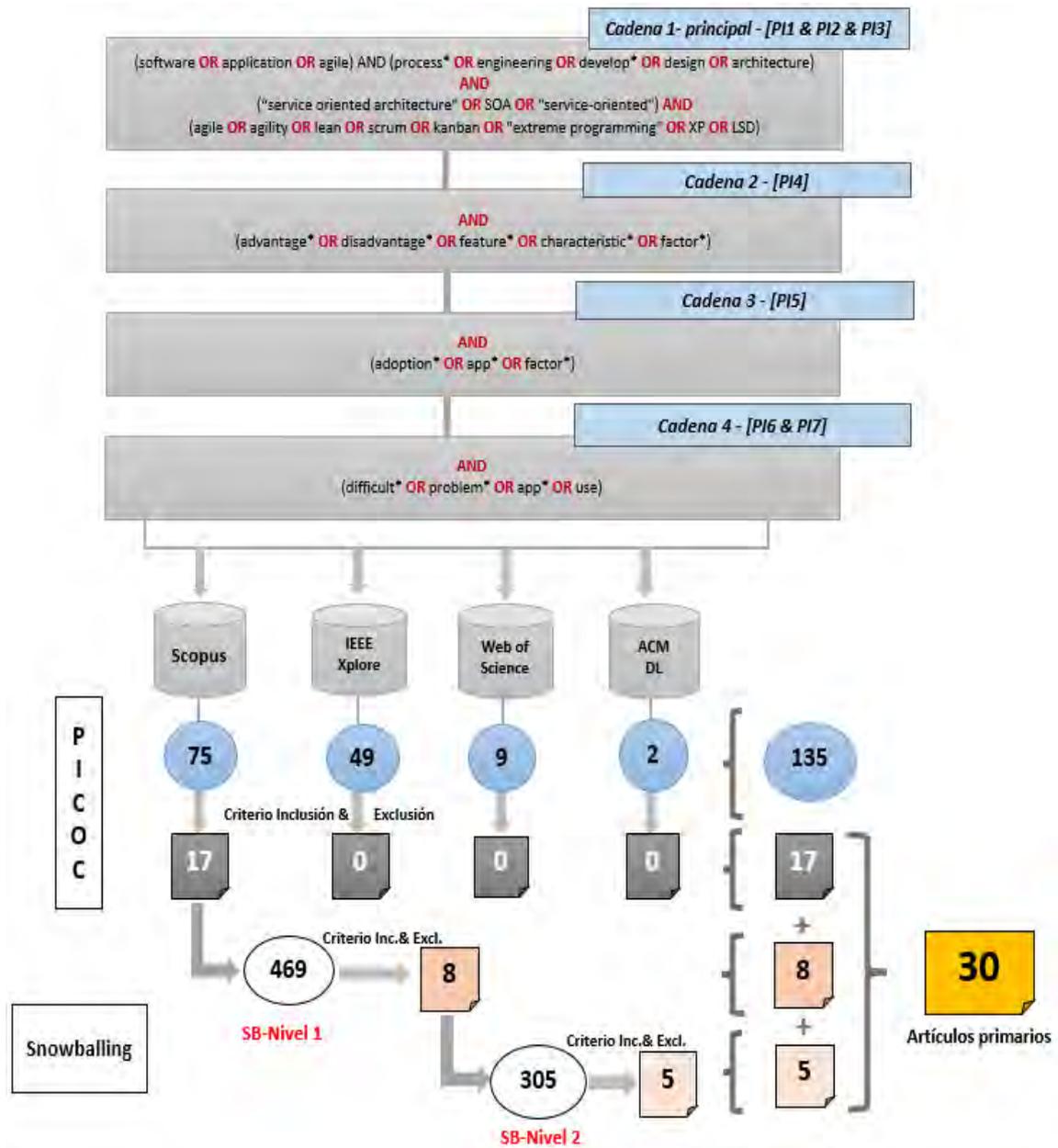


Figura 4.2: Resultado de búsqueda y selección de estudios

Tabla 4.3: Listado de estudios primarios

ID	Ref.	Año	País	Cadenas
S01	[31]	2022	Egipto	1, 2, 3, 4
S02	[32]	2021	Canada	1, 2, 3, 4
S03	[33]	2021	EE.UU	1, 3, 4
S04	[34]	2020	India	1, 3, 4
S05	[35]	2020	Suiza	1
S06	[36]	2020	China	1, 3, 4
S07	[37]	2019	Austria	1, 2, 3, 4
S08	[38]	2019	Holanda	1, 3, 4
S09	[39]	2019	Italia	1, 4
S10	[40]	2019	Francia	1, 3, 4
S11	[41]	2018	EE.UU	1, 3, 4
S12	[42]	2018	Brasil	1, 3, 4
S13	[43]	2018	China	1, 2, 3, 4
S14	[44]	2017	Peru	1, 3, 4
S15	[45]	2017	Suiza	1, 3, 4
S16	[46]	2017	Colombia	1, 3, 4
S17	[47]	2017	Suiza	1, 2, 3, 4
S18	[48]	2021	Turquía	1, 2, 3, 4
S19	[49]	2017	Dinamarca	1, 2, 3, 4
S20	[50]	2018	Alemania	1, 2, 3, 4
S21	[51]	2017	Suiza	1, 2, 3, 4
S22	[52]	2018	Grecia	1, 2, 3, 4
S23	[53]	2017	Reino Unido	1, 2, 3, 4
S24	[54]	2016	Alemania	1, 2, 3, 4
S25	[55]	2017	EE.UU	1, 2, 3, 4
S26	[56]	2017	Alemania	1, 2, 3, 4
S27	[57]	2018	Alemania	1, 2, 3, 4
S28	[58]	2017	China	1, 2, 3, 4
S29	[59]	2017	Suiza	1, 2, 3, 4
S30	[60]	2018	Dinamarca	1, 2, 3, 4

4.4. Evaluación de la Calidad

Los 30 estudios primarios seleccionados para la investigación fueron sometidos al proceso de evaluación de la calidad aplicando las preguntas presentadas en la Tabla 3.14. de la sección 3.4.4 del capítulo 4 basados en la propuesta de Zarour et al. [28]. En la Tabla 4.4 se presentan los resultados obtenidos en la evaluación de la calidad en la cual todos los estudios seleccionados cumplieron con la escala mínima requerida para ser considerados en la investigación.

Tabla 4.4: Resultado de la evaluación de la calidad

ID	QA1	QA2	QA3	QA4	QA5	Puntaje
S01	1.0	1.0	0.0	1.0	1.0	4.0
S02	1.0	1.0	1.0	1.0	1.0	5.0
S03	1.0	1.0	0.0	0.5	0.5	3.0
S04	1.0	1.0	0.0	1.0	0.5	3.5
S05	1.0	1.0	1.0	0.0	1.0	4.0
S06	1.0	1.0	0.0	1.0	1.0	4.0
S07	1.0	1.0	0.0	1.0	1.0	4.0
S08	1.0	1.0	0.0	1.0	1.0	4.0
S09	1.0	1.0	0.0	1.0	0.5	3.5
S10	1.0	1.0	0.0	1.0	1.0	4.0
S11	1.0	1.0	0.0	1.0	1.0	4.0
S12	1.0	1.0	0.0	1.0	0.5	3.5
S13	1.0	1.0	0.0	1.0	0.5	3.5
S14	1.0	1.0	0.0	1.0	1.0	4.0
S15	1.0	1.0	0.0	1.0	1.0	4.0
S16	1.0	1.0	0.0	1.0	0.5	3.5
S17	1.0	1.0	0.0	1.0	1.0	4.0
S18	1.0	1.0	1.0	1.0	1.0	5.0
S19	1.0	1.0	0.0	0.5	0.5	3.0
S20	1.0	1.0	0.0	1.0	0.5	3.5
S21	1.0	1.0	0.0	1.0	0.5	3.5
S22	1.0	1.0	0.0	1.0	1.0	4.0
S23	1.0	1.0	0.0	1.0	1.0	4.0
S24	1.0	1.0	0.0	0.5	1.0	3.5
S25	1.0	1.0	0.0	0.5	1.0	3.5
S26	1.0	1.0	0.0	1.0	1.0	4.0
S27	1.0	1.0	0.0	1.0	0.5	3.5
S28	1.0	1.0	0.0	1.0	1.0	4.0
S29	1.0	1.0	0.0	1.0	1.0	4.0
S30	1.0	1.0	0.0	1.0	0.5	3.5

4.5. Extracción y Síntesis de Datos

Se realizó la lectura completa de los 30 estudios seleccionados y se extrajo los datos relevantes según el formulario establecido en la Tabla 3.15. de la sección 3.5 del capítulo 3. Asimismo, se analizó los datos extraídos de cada uno de los estudios con el propósito de elaborar la síntesis que se describe a continuación.

Publicaciones por año

En la Figura 4.3 se presenta el número acumulado de publicaciones seleccionadas. Se aprecia que las publicaciones se realizaron a partir del año 2016 hasta el 2022, siendo el año 2017 cuando se produjo el mayor incremento de publicaciones con 11 artículos, seguido por el año 2018 con 7 artículos.

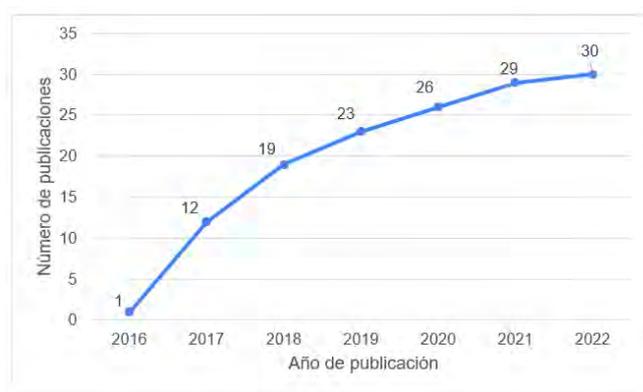


Figura 4.3: Publicaciones por año

Fuentes de publicación

En la Tabla 4.5 se muestra las fuentes de publicación de los 30 estudios seleccionados. Asimismo, se indica el tipo de publicación y el número de estudios por cada fuente identificada. Según los datos mostrados la fuente IEEE Software es la que presenta mayor número de publicaciones relacionadas al tema de investigación. El listado detallado de las fuentes de publicación de estudios primarios seleccionados se muestra en la Tabla D.1 del anexo D del presente documento de investigación.

Tabla 4.5: Fuentes de publicación relevantes

Fuente	Tipo	Cantidad
IEEE Software	Revista	3
IEEE Access	Revista	2
Computer Science - Research and Development	Revista	2
LNCS - Lecture Notes in Computer Science	Conferencia	2
ACM International Conference Proceeding Series	Conferencia	1
CEUR Workshop Proceedings	Conferencia	1
Empirical Software Engineering	Revista	1
EPIC Series in Computing	Conferencia	1
ES - International Conference on Enterprise Systems	Conferencia	1
Present and Ulterior Software Engineering	Libro	1
Otras fuentes	Conferencia/Revista	15

Asimismo, en la Figura 4.4 se muestra que el 64% de las publicaciones fueron presentadas en Conferencias, el 33% en Revistas y finalmente, el 3% fueron publicados en libros.

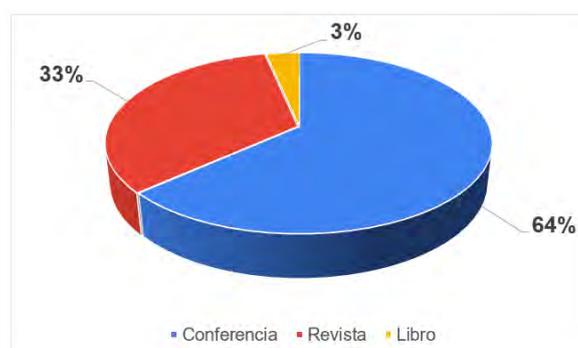


Figura 4.4: Estudios por tipo de publicación

Capítulo 5

Resultados de la Revisión Sistemática de Literatura

El presente capítulo está compuesto por 2 secciones. La Sección 5.1 presenta los resultados de las preguntas bibliométricas y la Sección 5.2 presenta los resultados de las preguntas de investigación considerando los datos obtenidos del formulario de extracción y de la síntesis realizada en la investigación.

5.1. Preguntas Bibliométricas

A continuación, se presentan los resultados de las 5 preguntas bibliométricas planteadas en la investigación.

5.1.1. Distribución de estudios (PB-1)

¿Cuál es la distribución de estudios relacionados con el tema de investigación en cada una de las Bases de Datos seleccionadas?

Resultados

El número total de publicaciones encontradas fue 135 antes de la aplicación de los criterios de selección. El mayor número de publicaciones encontradas fue la fuente Scopus con 75 estudios (56%), luego la fuente IEEE Xplore con 49 estudios (49%), seguido por la fuente Web of Science con 9 estudios y finalmente la fuente ACM Digital Library (7%) con 2 estudios (1%).

En la Figura 5.1 se muestra la distribución de estudios por base de datos.

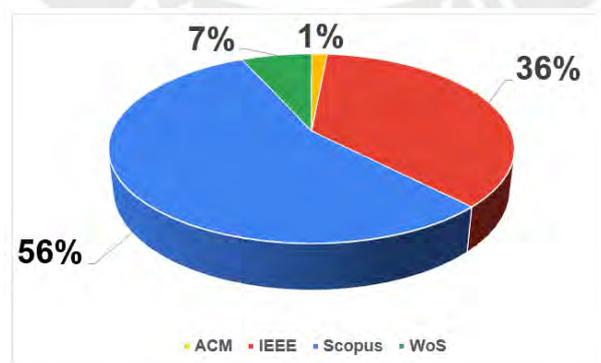


Figura 5.1: Estudios por base de datos

5.1.2. Tendencia de los estudios (PB-2)

¿Cuál es la tendencia de interés sobre las arquitecturas de software SOA y Microservicios?

Resultados

De los 30 estudios primarios seleccionados se tiene que en el año 2017 se publicaron la mayor cantidad de estudios relacionados al tema de investigación con 11 estudios, en el año 2018 se publicaron 7 estudios, en el año 2019 se publicaron 4 estudios seguidos de los años 2020 y 2021 con 3 estudios, finalmente los años 2016 y 2022 tuvieron solo 1 publicación como se muestra en la Figura 5.2.

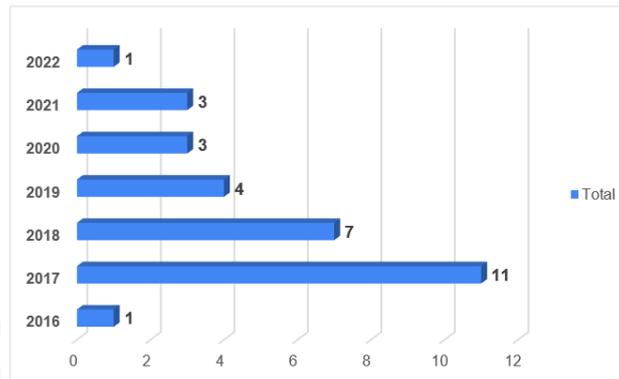


Figura 5.2: Estudios por año

5.1.3. Contribución de países (PB-3)

¿Cuáles son los países con mayor contribución de este tipo de investigación?

Resultados

En el país de Suiza se realizaron 5 publicaciones (20%) relacionadas al tema de investigación, seguido por Alemania con 4 publicaciones (13%); además, los países de China y Estados Unidos tienen 3 publicaciones (10%), seguidos de Egipto y Canadá con 1 publicación (3%). Asimismo, existen 11 países que presentan una publicación como se puede visualizar en la Figura 5.3.

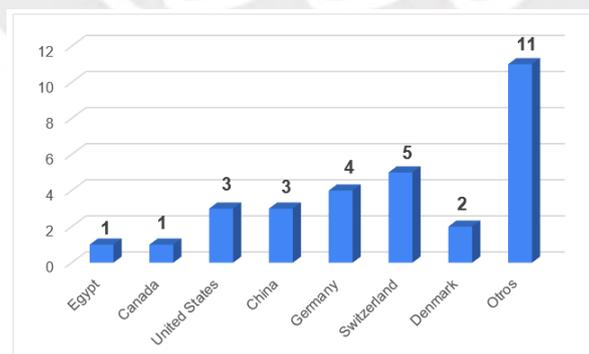


Figura 5.3: Estudios por país

5.1.4. Medios de publicación de estudios (PB-4)

¿Cuáles son los medios de publicación de los artículos con relación al tema de investigación?

Resultados

Las conferencias es el medio de publicación que presenta mayor número de publicaciones con 19 estudios (63%), seguido de artículos con 8 estudios (27%) y finalmente las revisiones con 3 estudios (10%). En la Figura 5.4 se muestra la distribución de estudios por medio de publicación.

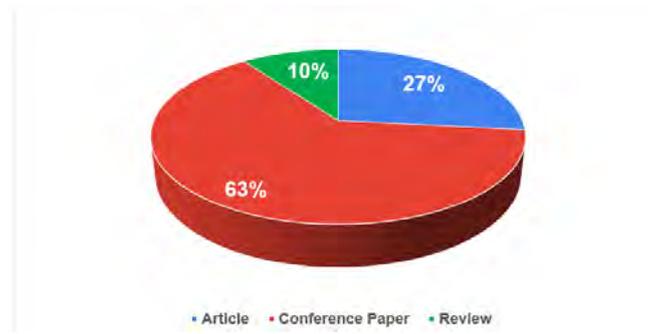


Figura 5.4: Estudios por medio de publicación

5.1.5. Congresos y revistas de publicación (PB-5)

¿En qué Journals y Congresos se publican los artículos con relación al tema de investigación?

Resultados

De los datos extraídos se tiene 19 estudios (64%) que corresponden a conferencias o congresos, 10 estudios corresponden a revistas o Journals (33%) y 1 estudio (3%) corresponde a libros. El listado detallado de las fuentes de publicación se presenta en la Tabla D.1 del anexo D del presente documento de investigación.

Asimismo, en la Figura 5.5 se visualiza la distribución por tipo de fuente de publicación.

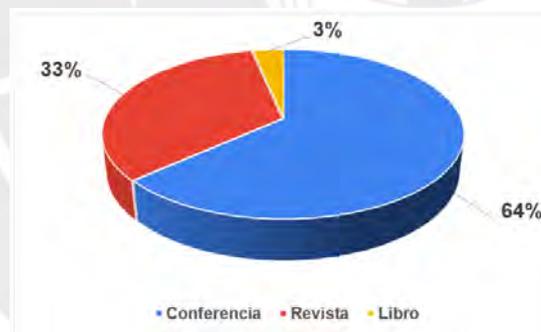


Figura 5.5: Tipo de fuente de publicación

5.2. Preguntas de Investigación

Las respuestas a las 7 preguntas de investigación planteadas se detallan a continuación.

5.2.1. Contribuciones y tipos de investigación (PI-1)

¿Qué tipo de investigaciones y contribuciones se han propuesto sobre el tema de investigación?

Resultados

Considerando la clasificación de investigaciones propuesto por Wieringa et al. [26], se identificaron propuestas de solución como en S01 [31], en la cual Zaki et al. proponen un marco basado en microservicios para desarrollar aplicaciones de atención médica. De igual manera Villamizar

et al. en S16 [46], presenta una comparación de costos de infraestructura y el rendimiento de las arquitecturas monolíticas, microservicios y una arquitectura híbrida.

En el mismo sentido, se identificaron estudios de tipo artículos filosóficos, siendo la mayoría de los estudios de este tipo, que brindan una descripción y comparación de los estilos arquitectónicos, entre ellos se tiene a Pautasso et al. en el estudio S07 [46] realiza una comparación de la arquitectura de microservicios con la arquitectura orientada a servicios, destacando las ventajas y desventajas. Asimismo, Shadija et al. en el estudio S23 [53], proporciona información sobre los beneficios y limitaciones de cada arquitectura como la necesidad de comprender el dominio y las mayores demandas de recursos para monitorear cada servicio.

Finalmente, se tiene estudios de tipo artículo de opinión (S17 [47] y S21 [51]), investigación de validación (S08 [38]) y artículos de experiencia (S04 [34], S06 [36], S12 [42] y S18 [48]) que no presentan explícitamente contribuciones relevantes sobre el tema de investigación a excepción de estudio S18 [48] en la que Vural et al. realiza una exploración de diferentes técnicas para identificar los límites de los microservicios.

5.2.2. Sectores de aplicación de estilos arquitectónicos (PI-2)

¿Cómo se aplican los estilos arquitectónicos del tema de investigación en los distintos sectores?

Resultados

Se tiene estudios (S04 [34], S05 [35], S07 [37], S08 [38], S12 [42], S18 [48], S19 [49], S20 [50], S21 [51], S22 [52], S23 [53], S25 [55], S26 [56], S27 [57], S29 [59], S30 [60]) que no presentan de manera explícita información sobre la aplicación de los estilos arquitectónicos abordados en la investigación; sin embargo, de los estudios restantes seleccionados se tiene que los estilos arquitectónicos son aplicados en sectores como Salud, Industria, Informática y Tecnología de Información principalmente en desarrollo y arquitectura de Software. En S01 [31], de acuerdo con Zaki et al. en el sector Salud se hace uso de enfoques tradicionales como arquitectura monolítica y SOA, estos estilos arquitectónicos se aplican estructurando una aplicación en base a los principios y características del estilo elegido. Zaki et al. también menciona que los estilos arquitectónicos se pueden combinar o personalizar para cumplir con los requisitos específicos de un sistema. Por otro lado, según Xiao et al. en S28 [58], en el sector Industria se utiliza la arquitectura de microservicios en las empresas modernas y a su vez sugiere una aplicación combinada de los estilos considerando la agilidad de los microservicios con la estabilidad de los sistemas heredados.

Respecto al sector de la informática y tecnología de la información, según Butzin et al. en S24 [54], la aplicación de la Arquitectura Orientada a Servicios (SOA) en el Internet de las Cosas (IoT) implica el uso de marcos, como el Perfil de Dispositivos para Servicios Web (DPWS) y el Protocolo de Aplicación Restringida (CoAP), que están diseñados para adaptarse a las limitaciones de los dispositivos y redes de IoT. Estos marcos proporcionan mecanismos de comunicación livianos y permiten el desarrollo de servicios que representan dispositivos físicos como sensores y actores.

En la Arquitectura de Software, para Rossi en S09 [39] su aplicación se basa en la exploración de diferentes alternativas de diseño para manejar cadenas de llamadas entre microservicios, incluidos enfoques basados en coreografía y orquestación, así como soluciones basadas en mensajería. Por otro lado, Pautasso et al. en S17 [47] menciona que los estilos arquitectónicos se aplican considerando las propiedades deseadas y las limitaciones del sistema. Además, SOA se puede restringir de manera similar a los microservicios. Pautasso et al. proponen un modelo de capacidades que vincula las

propiedades deseadas del sistema con las restricciones que deben aplicarse en la implementación. Este enfoque ayuda a garantizar el éxito de las implementaciones SOA.

Finalmente, en el Desarrollo de Software para Wang et al. en S02 [32], la arquitectura de microservicios se aplica de varias maneras según el contexto específico y los requisitos del sistema que se está desarrollando y para Villamizar et al. en S16 [46], la arquitectura SOA se aplica organizando el sistema de software como una colección de servicios que se comunican entre sí. Estos servicios están débilmente acoplados y pueden desarrollarse, implementarse y escalarse de forma independiente. La arquitectura a menudo implica el uso de un Enterprise Service Bus (ESB) para enrutar y enviar mensajes entre aplicaciones.

5.2.3. Principales similitudes y diferencias entre los estilos arquitectónicos (PI-3)

¿Cuáles son las principales similitudes y diferencias entre los estilos arquitectónicos del tema de investigación?

Resultados

Según los resultados obtenidos de la extracción y síntesis de datos, en S11 [41] Gallipeau y Kudrle resaltan la similitud de que ambos estilos arquitectónicos dependen de servicios que se comunican a través de un protocolo común y ambos apuntan a dividir las aplicaciones en elementos funcionales más pequeños. De igual manera otros autores mencionan aspectos de similitudes que se presentan a continuación.

- **Interoperabilidad**, de acuerdo con Zaki et al. en S01 [31], ambos enfoques permiten compartir datos entre diferentes servicios. Wang et al. en S02 [32] complementa indicando que los estilos arquitectónicos en estudio promueven el uso de API para la comunicación entre servicios. Del mismo modo Rasheedh et al. y Posadas en S04 [34], S14 [44] respectivamente, indican que los estilos tienen como objetivo permitir la comunicación y la interacción entre diferentes servicios, a menudo utilizando protocolos e interfaces estandarizadas.
- **Orientada a servicios**, se basan en el concepto de dividir las aplicaciones en servicios modulares más pequeños y poco acoplados que se pueden desarrollar, implementar y escalar de forma independiente, según los estudios S03 [33], S05 [35], S06 [36], S10 [40] y S14 [44].
- **Acoplamiento**, tienen como objetivo lograr un acoplamiento flexible entre servicios, permitiéndoles evolucionar y cambiar de forma independiente sin afectar a otros servicios, de acuerdo con los estudios S03 [33], S08 [38], S09 [39].
- **Reutilización**, de acuerdo con He y Deng en S06 [36], priorizan la reutilización de servicios para mejorar la eficiencia del desarrollo y reducir la redundancia.
- **Computación distribuida**, ambas arquitecturas implican la distribución de servicios entre diferentes sistemas o nodos, lo que permite flexibilidad y escalabilidad según Posadas en S14 [44]. En ese sentido, Rossi en S09 [39] añade que los servicios se pueden implementar en diferentes máquinas o incluso en diferentes ubicaciones geográficas.

También se presentan diferencias entre los estilos arquitectónicos detalladas a continuación.

- **Composición de servicio**, según Hoday et al. en S07 [37], SOA permite una composición horizontal del servicio, donde un servicio de grano grueso maneja todo el proceso de negocio,

mientras que Microservicios promueve servicios independientes con funcionalidades relacionadas que satisfacen necesidades comerciales específicas.

- **Autonomía**, de acuerdo a la arquitectura de microservicios enfatiza la autonomía de los servicios, lo que les permite desarrollarse e implementarse de forma independiente. En SOA, los servicios suelen estar más estrechamente acoplados y pueden requerir coordinación mediante orquestación.
- **Tamaño y granularidad**, según S03 [33], S05 [35], S07 [37], S09 [39] los microservicios suelen ser más pequeños y detallados en comparación con los servicios en SOA. Además, los microservicios se centran en capacidades empresariales específicas y están diseñados para ser implementables y escalables de forma independiente, siguiendo el principio de responsabilidad única, mientras que los servicios SOA tienden a ser más grandes y abarcar múltiples capacidades.
- **Gobernanza**, de acuerdo con S02 [32], S03 [33], S06 [36], S09 [39], S10 [40], las arquitecturas SOA suelen tener mecanismos de gobernanza centralizados para gestionar y controlar los servicios, con un enfoque en la estandarización y la reutilización, mientras que las arquitecturas de microservicios promueven la gobernanza descentralizada, permitiendo a los equipos tener más autonomía y flexibilidad para diseñar y gestionar sus propios servicios.
- **Estilo de comunicación**, según lo expuesto por Hoday et al. en S07 [37], SOA se basa principalmente en la orquestación de servicios, donde los servicios se coordinan para lograr funcionalidades complejas, mientras que Microservicios prefiere la coreografía de servicios, utilizando eventos y mecanismos de publicación/suscripción para la colaboración entre servicios.
- **Protocolo de comunicación**, Wang et al. en S02 [32] indica que los microservicios suelen estar asociados con protocolos ligeros, como HTTP REST, para la comunicación, mientras que la arquitectura SOA puede implicar varios protocolos de comunicación, incluido SOAP. De igual manera S03 [33], S04 [34], S12 [42], S14 [44] añaden que las arquitecturas SOA a menudo dependen de protocolos estandarizados como SOAP (Protocolo simple de acceso a objetos) y WS-* para la comunicación, mientras que las arquitecturas de microservicios comúnmente usan protocolos livianos como REST (Transferencia de estado representacional) y HTTP.
- **Implementación y escalabilidad**, las arquitecturas SOA pueden tener requisitos de implementación y escalamiento más complejos debido al mayor tamaño y las interdependencias de los servicios de acuerdo con los estudios S03 [33], S04 [34], S05 [35], S06 [36].
- **Gestión de datos**, según He y Deng, y Cojocarú et al. en S06 [36] y S08 [38] respectivamente concuerdan que SOA a menudo se basa en un modelo de datos centralizado y bases de datos compartidas, mientras que los microservicios abogan por una gestión de datos descentralizada y cada servicio tiene su propia base de datos.

5.2.4. Factores de influencia en elección de estilos arquitectónicos (PI-4)

¿Cómo influyen los factores encontrados en los estudios para la elección de los estilos arquitectónicos del tema de investigación?

Resultados

Para Zaki et al. en S01 [31] los factores para elegir arquitecturas se determinan en función de varias consideraciones, incluida la gestión de requisitos, la interoperabilidad, la resiliencia, las pruebas, la agilidad, la complejidad operativa, el mantenimiento y la utilización de recursos en la nube. Estos factores juegan un papel crucial en la evaluación y selección del enfoque arquitectónico más adecuado

para un proyecto de desarrollo de software determinado. Además, Pautasso et al. en S21 [51] incluyen consideraciones como la distribución de inteligencia en la infraestructura, el mal uso de estilos arquitectónicos, la elección del protocolo de transporte, los patrones de intercambio de mensajes, los protocolos de aplicación de dominio, el versionado semántico y la evolución y versionado de los microservicios.

Por otro lado, Gallipeau et al. en S11 [41] presenta como contexto la industria de los medios y la radiodifusión en la cual enfatiza factores como la escalabilidad y la flexibilidad para la elección de la arquitectura de microservicios debido a que se centra en la necesidad de agilidad, flexibilidad y la capacidad de mantenerse al día con los avances en virtualización y tecnologías de nube para construir flujos de trabajo más flexibles y rentables. Asimismo, Pautasso et al. en S17 [47] menciona que los factores para elegir arquitecturas, particularmente microservicios, incluyen un enfoque en el valor comercial, la capacidad de lanzar funciones y correcciones rápidamente, resolución de problemas, clarificación de dominios, desacoplamiento de interfaces compartidas, cambio de tamaño de los componentes del sistema.

Finalmente, Zimmermann et al. en S29 [59] enfatiza la importancia de alinear la arquitectura con los requisitos funcionales y no funcionales, considerando la pila de tecnología, aprovechando la experiencia del equipo, abordando los requisitos de escalabilidad e integración y equilibrando las limitaciones de costo y tiempo.

5.2.5. Problemas de aplicación y soluciones propuestas (PI-5)

¿Qué problemas se presentan en la aplicación de los estilos arquitectónicos del tema de investigación en el entorno de desarrollo ágil y cómo son abordados por los autores?

Resultados

De los resultados obtenidos de la revisión sistemática, se identifican problemas en la aplicación de los estilos arquitectónicos que se presentan a continuación.

- **Complejidad**, los estudios S02 [32] y S04 [34] presentan el problema de complejidad en las arquitecturas SOA debido a la gran cantidad de servicios y sus interacciones, lo que dificulta la gestión y el mantenimiento del sistema. Ante ello, Wang et al. en S02 [32] recomienda seguir los principios de control de versiones semántico y utilizar técnicas como el control de versiones de API.
- **Gastos generales de rendimiento**, según Wang et al. en S02 [32] el uso de protocolos pesados, como SOAP, en arquitecturas SOA puede introducir gastos generales de rendimiento debido al procesamiento adicional y la transformación de datos necesarios. Asimismo, Rasheedh et al. en S04 [34] precisa que los gastos generales de rendimiento se dan especialmente en los casos en que los servicios necesitan comunicarse con frecuencia. Al respecto, el mismo autor menciona que en lugar de utilizar protocolos pesados como SOAP, considere usar protocolos livianos como REST para mejorar el rendimiento y reducir los gastos generales.
- **Gestión de datos**, los estudios S02 [32] y S04 [34] presentan el problema de la complejidad de garantizar la coherencia de los datos y mantener la integridad de los datos en múltiples microservicios, especialmente en escenarios donde los datos deben compartirse o sincronizarse entre servicios. Al respecto, Rasheedh en S04 [34] recomienda implementar la gestión de datos distribuidos a través del uso de técnicas como bases de datos distribuidas o abastecimiento de eventos para gestionar la coherencia y la sincronización de los datos entre microservicios.

- **Gobernanza centralizada**, según Wang et al. en S02 [32] las arquitecturas SOA a menudo implican una gobernanza centralizada, lo que puede provocar cuellos de botella y retrasos en la toma de decisiones y obstaculizar la autonomía de los equipos de desarrollo. Además, Rasheedh et al. en S04 [34] menciona que garantizar una gobernanza y gestión adecuadas de los servicios en un entorno distribuido puede ser un desafío en SOA, lo que genera problemas como el control de versiones, la seguridad y la supervisión. Al respecto, el mismo autor recomienda implementar una gobernanza y gestión adecuadas estableciendo procesos y herramientas de gobernanza para gestionar el control de versiones, la seguridad y el seguimiento de los servicios.

5.2.6. Enfoques de definición de estilos arquitectónicos (PI-6)

¿De qué manera los autores enfocan la definición de estilo arquitectónico relacionado al tema de investigación?

Resultados

En S01 [31] Zaki et al. definen a los microservicios basados en el concepto de "Principio de Responsabilidad Única" asumiendo el hecho de que cada microservicio es responsable de realizar una y sólo una función. De igual forma los estudios S02 [32], S04 [34], S06 [36], S09 [39], S10 [40], S14 [44], S16 [46], S18 [48], S20 [50], S25 [55], S27 [57] definen microservicios como una colección de servicios pequeños, poco acoplados y que se pueden implementar de forma independiente. Además, según Thones en S03 [33] manifiesta que los microservicios pueden verse como un estilo de implementación específico de la Arquitectura Orientada a Servicios (SOA), donde los servicios son pequeños, detallados y desplegados de forma independiente.

Por otro lado, en S11 [41] y S16 [46] se define la arquitectura orientada a servicios (SOA) como "esencialmente una colección de servicios que se comunican entre sí". De igual manera, Pautasso et al. en S21 [51] definen a SOA como un paradigma arquitectónico que implica hacer que los servicios estén disponibles a través de una red para permitir que diferentes sistemas se comuniquen entre sí. Además, en los estudios S13 [43], S14 [44], S15 [45] y S18 [48] definen a SOA como un estilo arquitectónico de software que estructura una aplicación como una colección de servicios reutilizables y débilmente acoplados.

5.2.7. Uso combinado de estilos arquitectónicos (PI-7)

¿Es posible el uso combinado de los estilos arquitectónicos del tema de investigación para la solución de problemas y qué mencionan los autores al respecto?

Resultados

Según Zaki et al. en S01 [31], sí es posible el uso combinado de los estilos arquitectónicos, el enfoque híbrido permite a las organizaciones aprovechar la modularidad y escalabilidad de los microservicios y, al mismo tiempo, beneficiarse del acoplamiento flexible y la interoperabilidad que proporciona SOA.

Por otro lado, Wang et al. en S02 [32] precisa que los microservicios pueden verse como una evolución del estilo arquitectónico SOA y enfatiza que las arquitecturas de microservicios pueden beneficiarse de las lecciones aprendidas en el campo de SOA, como la importancia del diseño de servicios, el control de versiones de los servicios y la gobernanza de los servicios. Además, Rossi et al. en S09 [39], menciona que los microservicios pueden verse como una forma más específica y restringida de SOA, donde los microservicios imponen requisitos más estrictos sobre el acoplamiento

flexible y el desarrollo independiente. En este sentido, los microservicios pueden considerarse como una implementación especializada del estilo arquitectónico SOA más amplio.

Finalmente, Rasheedh et al. en S04 [34] menciona que al combinar SOA y microservicios, las organizaciones pueden beneficiarse de la flexibilidad y modularidad de los microservicios mientras aprovechan las capacidades de gobernanza e integración de SOA. Este enfoque híbrido permite la creación de servicios implementables de forma independiente y poco acoplados que se pueden orquestar y gestionar utilizando principios SOA.



Capítulo 6

Amenazas a la Validez

El presente capítulo presenta las amenazas a la validez propuesto por Wohlin et al. [29] las cuales se detallan en las siguientes 4 secciones. La Sección 6.1 presenta la validez del constructo utilizados en el presente estudio, la Sección 6.2 presenta la validez interna del estudio, la Sección 6.3 presenta la validez externa del estudio; finalmente la Sección 6.4 presenta la validez de las conclusiones del presente estudio.

6.1. Validez del Constructo

De acuerdo a Jedlitschka et al. [30] la validez del constructo se refiere al grado en que la operacionalización de las medidas en un estudio refleja los constructos en la vida real. De igual manera para Wohlin et al. [29] la validez refleja el grado en que las medidas operativas que se examinan corresponden con los objetivos previstos y las preguntas de investigación planteadas por el investigador.

En el presente estudio para mitigar la amenaza se ha utilizado cadenas de búsqueda para cada una de las preguntas de investigación. Las cadenas de búsqueda fueron elaboradas usando los términos al aplicar la técnica PICOC. Además, se ha considerado los sinónimos correspondientes a los términos usados para cada pregunta de investigación. Las cadenas de búsqueda fueron ejecutadas en cada una de las bases de datos seleccionadas. Asimismo, los términos utilizados en las cadenas de búsqueda fueron validados por otros investigadores durante el desarrollo del presente estudio.

6.2. Validez Interna

Para Jedlitschka et al. [30] la validez interna se refiere al grado en que el tratamiento o la variable independiente es realmente responsable del efecto observado en la variable dependiente. Por otro lado, según Wohlin et al. [29] la validez es un problema en la investigación causal. Cuando los investigadores prueban si un factor afecta al factor en estudio, existe el riesgo de que el factor en estudio también esté influenciado por un tercer factor.

En el presente estudio es posible la existencia de un sesgo por parte del investigador en las etapas de extracción y análisis de datos del estudio. Para mitigar esta amenaza se aplicó criterios de inclusión, exclusión y criterios de calidad de los estudios.

6.3. Validez Externa

De acuerdo a Jedlitschka et al. [30] la validez externa se define como la medida en que los resultados del estudio se pueden aplicar a otras poblaciones o entornos participantes. Para experimentos controlados en entornos artificiales, donde las mismas condiciones pueden no ser

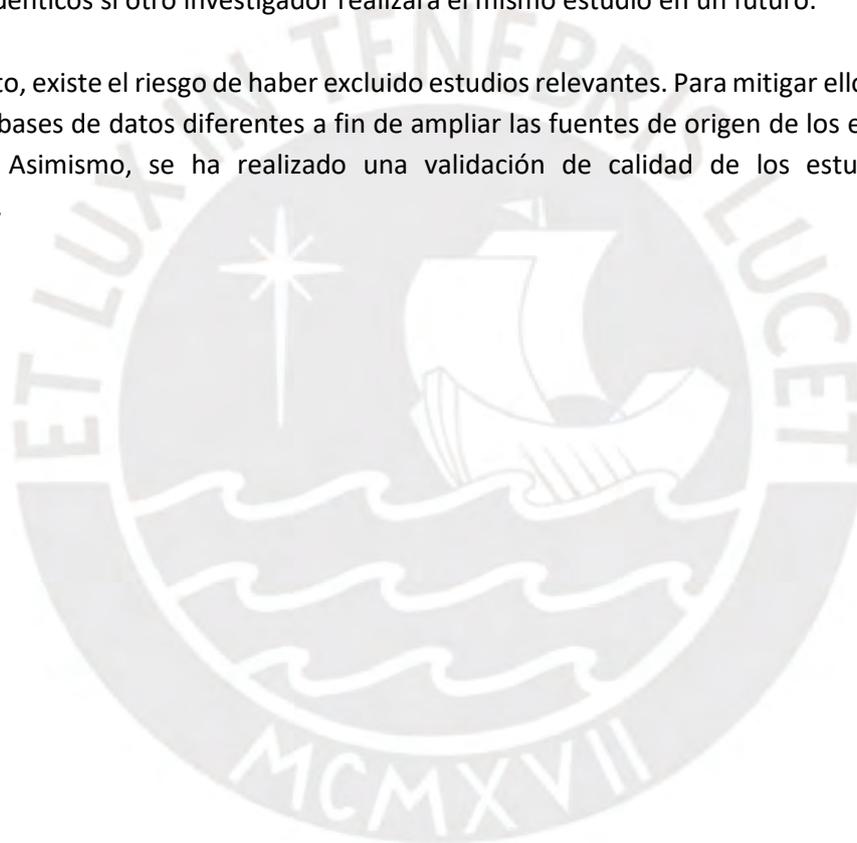
válidas en el mundo real, la validez externa con frecuencia puede ser un problema. Del mismo modo para Wohlin et al. [29] la validez se refiere a la medida en que los hallazgos pueden generalizarse y si son relevantes para otras personas que no están involucradas en el caso investigado.

En el presente estudio para mitigar esta amenaza se siguió el protocolo propuesto por Ahmad et al. [25], repitiendo la etapa de ejecución específicamente la actividad relacionada a la búsqueda y selección de estudios primarios.

6.4. Validez de las Conclusiones

Para Jedlitschka et al. [30] la validez de la conclusión se refiere a la veracidad de las conclusiones de un estudio. Por otro lado, según Wohlin et al. [29] la validez de la conclusión se refiere a qué tan dependientes están los datos y el análisis de los investigadores específicos. En teoría, los resultados deberían ser idénticos si otro investigador realizara el mismo estudio en un futuro.

Al respecto, existe el riesgo de haber excluido estudios relevantes. Para mitigar ello se ha extraído estudios de 4 bases de datos diferentes a fin de ampliar las fuentes de origen de los estudios para la investigación. Asimismo, se ha realizado una validación de calidad de los estudios primarios seleccionados.



Capítulo 7

Conclusiones y Trabajos Futuros

El objetivo de la presente revisión sistemática es realizar un análisis comparativo entre los estilos arquitectónicos SOA y Microservicios en la aplicación en entornos de desarrollo ágil. Para tal fin, se plantearon 5 preguntas bibliométricas y 7 preguntas de investigación las cuales fueron respondidas en base a los datos extraídos después de seguir el proceso de búsqueda y selección establecido en el estudio.

En la investigación realizada se obtuvieron un total de 30 artículos primarios como resultado de las búsquedas primaria y secundaria y además del proceso de selección inicial y final.

De los resultados se tiene que las similitudes entre los estilos arquitectónicos en estudio se dan principalmente en aspectos como interoperabilidad, orientación a servicios, acoplamiento, reutilización, computación distribuida. De igual manera sus diferencias identificadas destacan en composición de servicio, autonomía, tamaño, granularidad, comunicación, escalabilidad y gestión de datos.

Se presentaron resultados acerca de las contribuciones dadas por otros autores acerca del tema de investigación, principales similitudes y diferencias de los estilos arquitectónicos en estudio y su aplicación en distintos sectores como Salud, Industria, Informática y Tecnología de Información principalmente en el campo de desarrollo y arquitectura de Software. Además, se presentaron los resultados relacionados a la identificación de factores que influyen en la elección de alguno de los estilos arquitectónicos en estudio, así como también un análisis de los problemas presentados en su adopción. Asimismo, los casos de aplicación combinada de arquitecturas para la solución de problemas mencionados por los autores permiten concluir que es posible el uso combinado de los estilos arquitectónicos en las organizaciones de manera que puedan aprovechar los principios y conceptos de SOA mientras adoptan una arquitectura de microservicios para lograr una mayor modularidad, escalabilidad y flexibilidad.

Como trabajo futuro, se propone realizar un experimento empírico para la comparación de rendimiento y escalabilidad de las arquitecturas SOA y Microservicios con el propósito de obtener una visión práctica sobre cómo estos estilos arquitectónicos pueden comportarse en condiciones reales. A su vez, determinar la arquitectura que ofrece mejor rendimiento, escalabilidad y eficiencia en el contexto de la aplicación desarrollada.

De igual manera, otra propuesta de trabajo futuro es realizar una evaluación de tolerancia a fallos entre las arquitecturas SOA y Microservicios en condiciones controladas con el objetivo de evaluar la resiliencia y la capacidad de recuperación ante fallos de una aplicación implementada.

Las propuestas antes mencionadas ayudarán a tomar decisiones fundamentadas en la elección de una arquitectura. También considerando que los microservicios abogan por una gestión de datos descentralizada se sugiere que se lleve a cabo un estudio sobre cómo administrar de manera efectiva la persistencia de datos en entornos de microservicios, teniendo en cuenta las opciones de

almacenamiento distribuido, la consistencia de datos y los modelos de bases de datos específicos para este tipo de arquitectura.

Finalmente, considerando las tendencias relacionadas al tema de investigación, se puede realizar una revisión sistemática de literatura sobre la adopción, desafíos y mejores prácticas de los microservicios en entornos de computación en la nube. Esta propuesta de estudio debería proporcionar una visión general amplia y detallada de la literatura existente sobre microservicios en la nube, abarcando desde su adopción hasta los desafíos, las mejores prácticas y las tendencias futuras.



Referencias bibliográficas

- [1] Reupo-Musayón, N. (2020). Mapeo sistemático sobre las arquitecturas de software en el desarrollo ágil. [Tesis de licenciatura, Pontificia Universidad Católica del Perú]. Repositorio Institucional PUCP. <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/20825>
- [2] Power, K., & Wirfs-Brock, R. (2018). Understanding architecture decisions in context. In C. Cuesta, D. Garlan, & J. Pérez (Eds.), *Software architecture. ECSA 2018. Lecture notes in computer science* (Vol. 11048). Springer, Cham. https://doi.org/10.1007/978-3-030-00761-4_19
- [3] Yli-Huumo, J., Maglyas, A., & Smolander, K. (2016). How do software development teams manage technical debt? – An empirical study. *Journal of Systems and Software*, 120, 195-218. <https://doi.org/10.1016/j.jss.2016.05.018>
- [4] Gorton, I., & Teja Rayavarapu, V. (2022). Foundations of scalable software architectures. In 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C) (pp. 233-236). IEEE. <https://doi.org/10.1109/ICSA-C54293.2022.00052>
- [5] Blinowski, G., Ojdowska, A., & Przybyłek, A. (2022). Monolithic vs. microservice architecture: A performance and scalability evaluation. *IEEE Access*, 10, 20357-20374. <https://doi.org/10.1109/ACCESS.2022.3152803>
- [6] Jammes, F., & Smit, H. (2005). Service-oriented paradigms in industrial automation. *IEEE Transactions on Industrial Informatics*, 1(1), 62-70. <https://doi.org/10.1109/TII.2005.844419>
- [7] Papazoglou, M. P. (2003). Service-oriented computing: Concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE 2003)* (pp. 3-12). IEEE. <https://doi.org/10.1109/WISE.2003.1254461>
- [8] Fowler, M., & Lewis, J. (2014). *Microservices*. Martin Fowler. <https://martinfowler.com/articles/microservices.html>
- [9] Thönes, J. (2015). *Microservices*. *IEEE Software*, 32(1), 116. <https://doi.org/10.1109/MS.2015.11>
- [10] Avritzer, A., Ferme, V., Janes, A., Russo, B., Schulz, H., & van Hoorn, A. (2018). A quantitative approach for the assessment of microservice architecture deployment alternatives by automated performance testing. In C. Cuesta, D. Garlan, & J. Pérez (Eds.), *Software architecture. ECSA 2018. Lecture notes in computer science* (Vol. 11048, pp. 128-141). Springer, Cham. https://doi.org/10.1007/978-3-030-00761-4_11
- [11] Muñoz, M., Mejia, J., & Corona, B. (2016). Toward assessing the implementation and use of agile methodologies in SMEs: An analysis of the actual agile methodologies assessment tools. In 2016 11th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE. <https://doi.org/10.1109/CISTI.2016.7521520>
- [12] Faniran, V. T., Badru, A., & Ajayi, N. (2017). Adopting Scrum as an agile approach in distributed software development: A review of literature. In 2017 1st International Conference on Next Generation Computing Applications (NextComp) (pp. 36-40). IEEE. <https://doi.org/10.1109/NEXTCOMP.2017.8016173>

- [13] Saranya, P., & Fumio, A. (2014). The design of Kanban system for improving bottleneck problem of multiproduct. In 2014 IEEE 6th International Conference on Awareness Science and Technology (ICAST) (pp. 1-5). IEEE. <https://doi.org/10.1109/ICAwST.2014.6981834>
- [14] Goto, T., Tsuchida, K., & Nishino, T. (2014). EPISODE: An extreme programming method for innovative software based on systems design. In 2014 IIAI 3rd International Conference on Advanced Applied Informatics (pp. 780-784). IEEE. <https://doi.org/10.1109/IIAI-AAI.2014.157>
- [15] Anwaar, Z., Khan, F. U., Naz, M., & Farooq, M. U. (2021). Effect of lean software development on small organizations using LSD approach. In 2021 International Conference on Innovative Computing (ICIC) (pp. 1-8). IEEE. <https://doi.org/10.1109/ICIC53490.2021.9692992>
- [16] Bogner, J., Wagner, S., & Zimmermann, A. (2017). Automatically measuring the maintainability of service- and microservice-based systems: A literature review. In ACM International Conference Proceeding Series (pp. 107-115). <https://doi.org/10.1145/3143434.3143443>
- [17] Newman, S. (2015). Building microservices: Designing fine-grained systems (1st ed.). O'Reilly Media.
- [18] Hamzehloui, M. S., Sahibuddin, S., & Ashabi, A. (2019). A study on the most prominent areas of research in microservices. *International Journal of Machine Learning and Computing*, 9(2), 242-247. <https://doi.org/10.18178/ijmlc.2019.9.2.793>
- [19] Vural, H., Koyuncu, M., & Guney, S. (2017). A systematic literature review on microservices. In *Proceedings of the International Conference on Software Engineering* (pp. 199-209). Springer. https://doi.org/10.1007/978-3-319-62407-5_14
- [20] Mito de Oliveira dos Santos, E. F., Werner, C. M. L., & de França Costa, M. (2020). Selection methods for criticality in microservices architectures: A systematic mapping. In *Proceedings of the 23rd Iberoamerican Conference on Software Engineering (CibSE 2020)*
- [21] Baresi, L., & Garriga, M. (2019). Microservices: The evolution and extinction of web services? In *Microservices: Science and engineering* (pp. 3-28). Springer. https://doi.org/10.1007/978-3-030-31646-4_1
- [22] Di Francesco, P. (2017). Architecting microservices. In *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops (ICSAW 2017): Side Track Proceedings* (pp. 224-229). IEEE. <https://doi.org/10.1109/ICSAW.2017.65>
- [23] Bogner, J., Fritsch, J., Wagner, S., & Zimmermann, A. (2021). Industry practices and challenges for the evolvability assurance of microservices: An interview study and systematic grey literature review. *Empirical Software Engineering*, 26(5). <https://doi.org/10.1007/s10664-021-09999-9>
- [24] Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering (Vol. 2, p. 1051). *Engineering*
- [25] Ahmad, A., Jamshidi, P., & Pahl, C. (2012). Protocol for systematic literature review
- [26] Wieringa, R., Maiden, N., Mead, N., & Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requirements Engineering*, 11(2), 102-107. <https://doi.org/10.1007/s00766-006-0022-4>

- [27] Brereton, P., Kitchenham, B., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4), 571-583. <https://doi.org/10.1016/j.jss.2006.07.009>
- [28] Zarour, M., Abran, A., Desharnais, J.-M., & Alarifi, A. (2015). An investigation into the best practices for the successful design and implementation of lightweight software process assessment methods: A systematic literature review. *Journal of Systems and Software*, 101, 180-192. <https://doi.org/10.1016/j.jss.2014.12.019>
- [29] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering*. Springer. <https://doi.org/10.1007/978-3-642-29044-2>
- [30] Jedlitschka, A., Ciolkowski, M., & Pfahl, D. (2008). Reporting experiments in software engineering. In F. Shull, J. Singer, & D. I. K. Sjøberg (Eds.), *Guide to advanced empirical software engineering* (pp. 201-228). Springer. https://doi.org/10.1007/978-1-84800-044-5_8
- [31] Zaki, J., Islam, S. M. R., Alghamdi, N. S., Abdullah-Al-Wadud, M., & Kwak, K.-S. (2022). Introducing cloud-assisted micro-service-based software development framework for healthcare systems. *IEEE Access*, 10, 33332-33348. <https://doi.org/10.1109/ACCESS.2022.3161455>
- [32] Wang, Y., Kadiyala, H., & Rubin, J. (2021). Promises and challenges of microservices: An exploratory study. *Empirical Software Engineering*, 26, 63. <https://doi.org/10.1007/s10664-020-09910-y>
- [33] Fella, A., & Bandi, A. (2021). Microservice-based architectures: An evolutionary software development model. *EPIc Series in Computing*, 78, 206-216
- [34] Rasheedh, J. A., & Saradha, S. (2020). Review of micro-services architectures and runtime dynamic binding. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)* (pp. 1130-1137). IEEE. <https://doi.org/10.1109/I-SMAC49090.2020.9243335>
- [35] Neri, D., Soldani, J., Zimmermann, O., et al. (2020). Design principles, architectural smells and refactorings for microservices: A multivocal review. *SICS Software-Intensive Cyber-Physical Systems*, 35(1), 3-15. <https://doi.org/10.1007/s00450-019-00407-8>
- [36] He, X., & Deng, F. (2020). Research on architecture of Internet of Things platform based on service mesh. In *2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*. IEEE
- [37] Homa, A., Zoitl, A., Sousa, M. D., & Wollschlaeger, M. (2019). A survey: Microservices architecture in advanced manufacturing systems. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)* (pp. 1165-1168). IEEE. <https://doi.org/10.1109/INDIN41052.2019.8972079>
- [38] Cojocaru, M.-D., Uta, A., & Oprescu, A.-M. (2019). Attributes assessing the quality of microservices automatically decomposed from monolithic applications. In *2019 18th International Symposium on Parallel and Distributed Computing (ISPDC)* (pp. 84-93). IEEE. <https://doi.org/10.1109/ISPDC.2019.00021>
- [39] Rossi, D. (2019). Consistency and availability in microservice architectures. In M. Escalona, F. Domínguez Mayo, T. Majchrzak, & V. Monfort (Eds.), *Web information systems and technologies: WEBIST 2018* (Vol. 372, pp. 39-58). Springer. https://doi.org/10.1007/978-3-030-35330-8_3

- [40] Abidi, S., Essafi, M., Guegan, C. G., Fakhri, M., Wittl, H., & Ghezala, H. H. B. (2019). A web service security governance approach based on dedicated micro-services. *Procedia Computer Science*, 159, 372-386. <https://doi.org/10.1016/j.procs.2019.09.192>
- [41] Gallipeau, D., & Kudrle, S. (2018). Microservices: Building blocks to new workflows and virtualization. *SMPTE Motion Imaging Journal*, 127(4), 21-31. <https://doi.org/10.5594/JMI.2018.2811599>
- [42] Jamshidi, P., Pahl, C., Mendonça, N. C., Lewis, J., & Tilkov, S. (2018). Microservices: The journey so far and challenges ahead. *IEEE Software*, 35(3), 24-35. <https://doi.org/10.1109/MS.2018.2141039>
- [43] Kang, R., Zhou, Z., Liu, J., Zhou, Z., & Xu, S. (2018). Distributed monitoring system for microservices-based IoT middleware system. In X. Sun, Z. Pan, & E. Bertino (Eds.), *Cloud computing and security: ICCCS 2018* (Vol. 11063, pp. 458-469). Springer. https://doi.org/10.1007/978-3-030-00006-6_43
- [44] Posadas, J. V. (2017). Application of mixed distributed software architectures for social-productive projects management in Peru. In 2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON) (pp. 1-4). IEEE. <https://doi.org/10.1109/INTERCON.2017.8079698>
- [45] Zimmermann, O. (2017). Microservices tenets. *Computing Research and Development*, 32(4), 301-310. <https://doi.org/10.1007/s00450-016-0337-0>
- [46] Villamizar, M., Garcés, O., Ochoa, L., et al. (2017). Cost comparison of running web applications in the cloud using monolithic, microservice, and AWS Lambda architectures. *Journal of Systems and Software*, 11, 233-247. <https://doi.org/10.1007/s11761-017-0208-y>
- [47] Pautasso, C., Zimmermann, O., Amundsen, M., Lewis, J., & Josuttis, N. (2017). Microservices in practice, part 1: Reality check and service design. *IEEE Software*, 34(1), 91-98. <https://doi.org/10.1109/MS.2017.24>
- [48] Vural, H., & Koyuncu, M. (2021). Does domain-driven design lead to finding the optimal modularity of a microservice? *IEEE Access*, 9, 32721-32733. <https://doi.org/10.1109/ACCESS.2021.3060895>
- [49] Dragoni, N., et al. (2017). Microservices: Yesterday, today, and tomorrow. In M. Mazzara & B. Meyer (Eds.), *Present and ulterior software engineering* (pp. 195-216). Springer. https://doi.org/10.1007/978-3-319-67425-4_12
- [50] Ghofrani, J., & Lübke, D. (2018). Challenges of microservices architecture: A survey on the state of the practice. *CEUR Workshop Proceedings*, 2072, 1-8. <http://ceur-ws.org/Vol-2072/paper1.pdf>
- [51] Pautasso, C., Zimmermann, O., Amundsen, M., Lewis, J., & Josuttis, N. (2017). Microservices in practice, part 2: Service integration and sustainability. *IEEE Software*, 34(2), 97-104. <https://doi.org/10.1109/MS.2017.56>
- [52] Thramboulidis, K., Vachtsevanou, D. C., & Solanos, A. (2018). Cyber-physical microservices: An IoT-based framework for manufacturing systems. In 2018 IEEE Industrial Cyber-Physical Systems (ICPS) (pp. 232-239). IEEE. <https://doi.org/10.1109/ICPHYS.2018.8387665>

- [53] Shadija, D., Rezai, M., & Hill, R. (2017). Towards an understanding of microservices. In 2017 23rd International Conference on Automation and Computing (ICAC) (pp. 1-6). IEEE. <https://doi.org/10.23919/IConAC.2017.8082018>
- [54] Butzin, B., Golatowski, F., & Timmermann, D. (2016). Microservices approach for the internet of things. In 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA) (pp. 1-6). IEEE. <https://doi.org/10.1109/ETFA.2016.7733707>
- [55] Cerny, T., Donahoo, M. J., & Pechanec, J. (2017). Disambiguation and comparison of SOA, microservices, and self-contained systems. In Proceedings of the International Conference on Research in Adaptive and Convergent Systems (RACS '17) (pp. 228–235). Association for Computing Machinery. <https://doi.org/10.1145/3129676.3129682>
- [56] Rademacher, F., Sachweh, S., & Zündorf, A. (2017). Differences between model-driven development of service-oriented and microservice architecture. In 2017 IEEE International Conference on Software Architecture Workshops (ICSAW) (pp. 38-45). IEEE. <https://doi.org/10.1109/ICSAW.2017.32>
- [57] Rademacher, F., Sorgalla, J., Wizenty, P. N., Sachweh, S., & Zündorf, A. (2018). Microservice architecture and model-driven development: Yet singles, soon married (?). In Proceedings of the 19th International Conference on Agile Software Development: Companion (XP '18) (pp. 1–5). Association for Computing Machinery. <https://doi.org/10.1145/3234152.3234193>
- [58] Xiao, Z., Wijegunaratne, I., & Qiang, X. (2016). Reflections on SOA and microservices. In 2016 4th International Conference on Enterprise Systems (ES) (pp. 60-67). IEEE. <https://doi.org/10.1109/ES.2016.14>
- [59] Zimmermann, O. (2017). Microservices tenets. Computing Research Repository, 32, 301–310. <https://doi.org/10.1007/s00450-016-0337-0>
- [60] Dragoni, N., Lanese, I., Larsen, S. T., Mazzara, M., Mustafin, R., & Safina, L. (2018). Microservices: How to make your application scale. In Perspectives of System Informatics. PSI 2017. Lecture Notes in Computer Science (Vol. 10742, pp. 133-147). Springer. https://doi.org/10.1007/978-3-319-74313-4_8
- [61] Romero, F. (2022). Como trabaja hoy una empresa de ingeniería naval: Conceptos básicos (II). Defensa. <https://www.defensa.com/industria/como-trabaja-hoy-empresa-ingenieria-naval-conceptos-basicos-ii>
- [62] Sutherland, J., & Schwaber, K. (2007). The Scrum papers: Nuts, bolts, and origins of an agile process. Patient Keeper Inc.
- [63] Sutherland, J., & Schwaber, K. (2020). The Scrum guide. Scrum Guides. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- [64] Viewnext S.A. (2019). Kanban aplicado al desarrollo de software. <https://www.viewnext.com/kanban-desarrollo-software/>
- [65] Donetonic. (2022). El tablero Kanban. <https://donetonic.com/es/que-es-un-tablero-kanban>

Anexos

Anexo A: Cadenas de búsqueda por base de datos

Para las preguntas de investigación que aplican la **cadena 1** se muestran en la Tabla A.1.

Tabla A.1: Cadena de búsqueda por base de datos – Cadena 1

Base de Datos	Cadena de Búsqueda
ACM Digital Library	[[Abstract: software] OR [Abstract: application] OR [Abstract: agile]] AND [[Abstract: process*] OR [Abstract: engineering] OR [Abstract: develop*] OR [Abstract: design] OR [Abstract: architecture]] AND [[Abstract: "service oriented architecture"] OR [Abstract: SOA] OR [Abstract: "service-oriented"]] AND [[Abstract: microservice*] OR [Abstract: "micro-service*"]] AND [[Abstract: agile] OR [Abstract: agility] OR [Abstract: lean] OR [Abstract: scrum] OR [Abstract: kanban] OR [Abstract: "extreme programming"]] OR [Abstract: XP] OR [Abstract: LSD]]
IEEE Xplore	("Abstract": software OR application OR agile) AND ("Abstract": process* OR engineering OR develop* OR design OR architecture) AND ("Abstract": "service oriented architecture" OR SOA OR "service-oriented") AND ("Abstract": microservice* OR "micro-service*") AND ("Abstract": agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD)
Scopus	TITLE-ABS-KEY ((software OR application OR agile) AND (process* OR engineering OR develop* OR design OR architecture) AND ("service oriented architecture" OR SOA OR "service-oriented") AND (microservice* OR "micro-service*") AND (agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD))
Web of Science	TS = ((software OR application OR agile) AND (process* OR engineering OR develop* OR design OR architecture) AND ("service oriented architecture" OR SOA OR "service-oriented") AND (microservice* OR "micro-service*") AND (agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD))

Para las preguntas de investigación que aplican la **cadena 2** se muestran en la Tabla A.2.

Tabla A.2: Cadena de búsqueda por base de datos – Cadena 2

Base de Datos	Cadena de Búsqueda
ACM Digital Library	[[Abstract: software] OR [Abstract: application] OR [Abstract: agile]] AND [[Abstract: process*] OR [Abstract: engineering] OR [Abstract: develop*] OR [Abstract: design] OR [Abstract: architecture]] AND [[Abstract: "service oriented architecture"] OR [Abstract: SOA] OR [Abstract: "service-oriented"]] AND [[Abstract: microservice*] OR [Abstract: "micro-service*"]] AND [[Abstract: agile] OR [Abstract: agility] OR [Abstract: lean] OR [Abstract: scrum] OR [Abstract: kanban] OR [Abstract: "extreme programming"]] OR [Abstract: XP] OR [Abstract: LSD]] AND [[Abstract: advantage*] OR [Abstract: disadvantage*] OR [Abstract: feature*] OR [Abstract: characteristic*] OR [Abstract: factor*]]
IEEE Xplore	("Abstract": software OR application OR agile) AND ("Abstract": process* OR engineering OR develop* OR design OR architecture) AND ("Abstract": "service oriented architecture" OR SOA OR "service-oriented") AND ("Abstract": microservice* OR "micro-service*") AND ("Abstract": agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD) AND ("Abstract": advantage* OR disadvantage* OR feature* OR characteristic*) ("Abstract": software OR application OR agile) AND ("Abstract": process* OR engineering OR develop* OR design OR architecture) AND ("Abstract": "service oriented architecture" OR SOA OR "service-oriented") AND ("Abstract": microservice* OR "micro-service*") AND ("Abstract": agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD) AND ("Abstract": factor*)

Base de Datos	Cadena de Búsqueda
Scopus	TITLE-ABS-KEY ((software OR application OR agile) AND (process* OR engineering OR develop* OR design OR architecture) AND ("service oriented architecture" OR SOA OR "service-oriented") AND (microservice* OR "micro-service*")) AND (agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD) AND (advantage* OR disadvantage* OR feature* OR characteristic* OR factor*)
Web of Science	TS = ((software OR application OR agile) AND (process* OR engineering OR develop* OR design OR architecture) AND ("service oriented architecture" OR SOA OR "service-oriented") AND (microservice* OR "micro-service*")) AND (agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD) AND (advantage* OR disadvantage* OR feature* OR characteristic* OR factor*)

Para las preguntas de investigación que aplican la **cadena 3** se muestran en la Tabla A.3.

Tabla A.3: Cadena de búsqueda por base de datos – Cadena 3

Base de Datos	Cadena de Búsqueda
ACM Digital Library	[[Abstract: software] OR [Abstract: application] OR [Abstract: agile]] AND [[Abstract: process*] OR [Abstract: engineering] OR [Abstract: develop*] OR [Abstract: design] OR [Abstract: architecture]] AND [[Abstract: "service oriented architecture"] OR [Abstract: SOA] OR [Abstract: "service-oriented"]] AND [[Abstract: microservice*] OR [Abstract: "micro-service*"]] AND [[Abstract: agile] OR [Abstract: agility] OR [Abstract: lean] OR [Abstract: scrum] OR [Abstract: kanban] OR [Abstract: "extreme programming"] OR [Abstract: XP] OR [Abstract: LSD]] AND [[Abstract: adoption*] OR [Abstract: app*] OR [Abstract: factor*]]
IEEE Xplore	("Abstract": software OR application OR agile) AND ("Abstract": process* OR engineering OR develop* OR design OR architecture) AND ("Abstract": "service oriented architecture" OR SOA OR "service-oriented") AND ("Abstract": microservice* OR "micro-service*") AND ("Abstract": agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD) AND ("Abstract": adoption* OR app* OR factor*)
Scopus	TITLE-ABS-KEY ((software OR application OR agile) AND (process* OR engineering OR develop* OR design OR architecture) AND ("service oriented architecture" OR SOA OR "service-oriented") AND (microservice* OR "micro-service*")) AND (agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD) AND (adoption* OR app* OR factor*)
Web of Science	TS = ((software OR application OR agile) AND (process* OR engineering OR develop* OR design OR architecture) AND ("service oriented architecture" OR SOA OR "service-oriented") AND (microservice* OR "micro-service*")) AND (agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD) AND (adoption* OR app* OR factor*)

Para las preguntas de investigación que aplican la **cadena 4** se muestran en la Tabla A.4.

Tabla A.4: Cadena de búsqueda por base de datos – Cadena 4

Base de Datos	Cadena de Búsqueda
ACM Digital Library	[[Abstract: software] OR [Abstract: application] OR [Abstract: agile]] AND [[Abstract: process*] OR [Abstract: engineering] OR [Abstract: develop*] OR [Abstract: design] OR [Abstract: architecture]] AND [[Abstract: "service oriented architecture"] OR [Abstract: SOA] OR [Abstract: "service-oriented"]] AND [[Abstract: microservice*] OR [Abstract: "micro-service*"]] AND [[Abstract: agile] OR [Abstract: agility] OR [Abstract: lean] OR [Abstract: scrum] OR [Abstract: kanban] OR [Abstract: "extreme programming"]] OR [Abstract: XP] OR [Abstract: LSD]] AND [[Abstract: difficult*] OR [Abstract: problem*] OR [Abstract: app*] OR [Abstract: use]]
IEEE Xplore	("Abstract": software OR application OR agile) AND ("Abstract": process* OR engineering OR develop* OR design OR architecture) AND ("Abstract": "service oriented architecture" OR SOA OR "service-oriented") AND ("Abstract": microservice* OR "micro-service*") AND ("Abstract": agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD) AND ("Abstract": difficult* OR problem* OR app* OR use)
Scopus	TITLE-ABS-KEY ((software OR application OR agile) AND (process* OR engineering OR develop* OR design OR architecture) AND ("service oriented architecture" OR SOA OR "service-oriented")) AND (microservice* OR "micro-service*") AND (agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD) AND (difficult* OR problem* OR app* OR use)
Web of Science	TS = ((software OR application OR agile) AND (process* OR engineering OR develop* OR design OR architecture) AND ("service oriented architecture" OR SOA OR "service-oriented")) AND (microservice* OR "micro-service*") AND (agile OR agility OR lean OR scrum OR kanban OR "extreme programming" OR XP OR LSD) AND (difficult* OR problem* OR app* OR use)

Anexo B: Proceso de filtrado de estudio

Los estudios obtenidos de la búsqueda primaria realizada en la investigación se presentan a continuación:

Tabla B.1: Detalle del proceso de filtrado de estudio

ID	Año	Título	Paso1:Selección Inicial	Paso 2: Selección Final	CI1	CI2	CI3	CI4	CI5	CE 1	CE 2	CE 3	CE 4	CE 5	Resultado selección
1	2022	Cloud-native computing: How to design, develop, and secure microservices and event-driven applications	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
2	2022	MICROSERVICE ARCHITECTURE DESIGN: PROPOSED for E-OFFICE APPLICATION	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
3	2022	2022 International Conference on Information Technology Systems and Innovation, ICITSI 2022 - Proceedings	DUDA	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	NO	NO CUMPLE
4	2022	Design of Lightweight Satellite Status Monitoring System Based on Microservices	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
5	2022	Introducing Cloud-Assisted Micro-Service-Based Software Development Framework for Healthcare Systems	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
6	2022	A cybersecurity framework to guarantee reliability and trust for digital service chains-GUARD	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
7	2021	Promises and challenges of microservices: an exploratory study	DUDA	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
8	2021	Designing Online Healthcare Using DDD in Microservices Architecture	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
9	2021	Microservice-based architectures: An evolutionary software development model	DUDA	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
10	2021	Smart node networks orchestration: A new e2e approach for analysis and design for agile 4.0 implementation	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
11	2021	Research on Quality Model and Measurement for Microservices	DUDA	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	NO	NO CUMPLE
12	2021	Reference Architecture and Agile Development Method for a Process-Driven Web Platform based on the BPMN-Standard and Process Engines	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
13	2021	From Code Refactoring to API Refactoring: Agile Service Design and Evolution	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE

ID	Año	Título	Paso1:Selección Inicial	Paso 2: Selección Final	CI1	CI2	CI3	CI4	CI5	CE 1	CE 2	CE 3	CE 4	CE 5	Resultado selección
14	2021	15th Symposium and Summer School on Service Oriented Computing, SummerSOC 2021	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
15	2021	Algorithms for managing service-oriented research processes as services in hybrid computing environments of digital platforms	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
16	2021	Agile architecture for digital enterprises	SI	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	NO	NO CUMPLE
17	2021	Microservices: Migration of a Mission Critical System	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
18	2020	Scientific Services Consolidation Methods	DUDA	NO	SI	SI	SI	SI	NO	NO	NO	NO	NO	NO	NO CUMPLE
19	2020	Review of micro-services architectures and runtime dynamic binding	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
20	2020	Design principles, architectural smells and refactorings for microservices: a multivocal review	DUDA	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
21	2020	An Intelligent Anomaly Detection Scheme for Micro-Services Architectures with Temporal and Spatial Data Analysis	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
22	2020	The Road beyond 5G: A Vision and Insight of the Key Technologies	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
23	2020	Highly-Distributed Systems Based on Micro-Services and their Construction Paradigms	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
24	2020	Research on architecture of internet of things platform based on service mesh	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
25	2020	Domain-driven service design context modeling, model refactoring and contract generation	DUDA	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	NO	NO CUMPLE
26	2020	9th International Conference on Cloud Computing and Services Science, CLOSER 2019	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
27	2020	Domain-specific language and tools for strategic domain-driven design, context mapping and bounded context modeling	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
28	2019	Towards Novel Security Architectures for Network Functions Virtualization	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
29	2019	Management of User Equipment Context as a Radio Access Network Service	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE

ID	Año	Título	Paso1:Selección Inicial	Paso 2: Selección Final	CI1	CI2	CI3	CI4	CI5	CE 1	CE 2	CE 3	CE 4	CE 5	Resultado selección
30	2019	A survey: Microservices architecture in advanced manufacturing systems	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
31	2019	Attributes assessing the quality of microservices automatically decomposed from monolithic applications	DUDA	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
32	2019	Software Versioning with Microservices through the API Gateway Design Pattern	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
33	2019	Microservice Architectures for Advanced Driver Assistance Systems: A Case-Study	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
34	2019	Service Mesh: Challenges, state of the art, and future research opportunities	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
35	2019	Viewpoint-specific model-driven microservice development with interlinked modeling languages	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
36	2019	Exploring Microservices as the Architecture of Choice for Network Function Virtualization Platforms	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
37	2019	Design of marketplaces for smart manufacturing services	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
38	2019	Consistency and Availability in Microservice Architectures	DUDA	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
39	2019	A web service security governance approach based on dedicated micro-services	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
40	2018	Towards a continuous feedback loop for service-oriented environments	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
41	2018	Microns: Commands for building bubble microservices	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
42	2018	Laboratory information system and necessary improvements in function and programming	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
43	2018	In pursuit of architectural agility: Experimenting with microservices	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
44	2018	Microservices: Architecting for Continuous Delivery and DevOps	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
45	2018	Microservices: Building blocks to new workflows and virtualization	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
46	2018	Microservices: The journey so far and challenges ahead	DUDA	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
47	2018	Microservices	DUDA	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	NO	NO CUMPLE

ID	Año	Título	Paso1:Selección Inicial	Paso 2: Selección Final	CI1	CI2	CI3	CI4	CI5	CE 1	CE 2	CE 3	CE 4	CE 5	Resultado selección
48	2018	Scalable micro-service based approach to FHIR server with golang and No-SQL	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
49	2018	Microservices: Building blocks to new workflows and virtualization	SI	NO	SI	SI	SI	SI	SI	NO	NO	NO	NO	SI	NO CUMPLE
50	2018	12th European Conference on Software Architecture, ECSA 2018	DUDA	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	NO	NO CUMPLE
51	2018	An industrial investigation into microservice architecture and its factors [Mikroservis mimarisi ve mimari faktörleri üzerine endüstriyel bir inceleme]	DUDA	NO	SI	SI	SI	SI	SI	NO	SI	NO	NO	NO	NO CUMPLE
52	2018	TANKER: Distributed architecture for named entity recognition and disambiguation	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
53	2018	Agile services	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
54	2018	Decision-controlled digitization architecture for internet of things and microservices	DUDA	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	NO	NO CUMPLE
55	2018	Distributed monitoring system for microservices-based IoT middleware system	DUDA	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
56	2017	Decision management for micro-granular digital architecture	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
57	2017	Automatically measuring the maintainability of service- and microservice-based systems - a literature review	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
58	2017	Application of mixed distributed software architectures for social-productive projects management in peru	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
59	2017	An agile farm management information system framework for Precision Agriculture	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
60	2017	Processes, Motivations, and Issues for Migrating to Microservices Architectures: An Empirical Investigation	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
61	2017	Microservices tenets: Agile approach to service development and deployment	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
62	2017	First international workshop on architecting with microservices	DUDA	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	NO	NO CUMPLE
63	2017	Cost comparison of running web applications in the cloud using monolithic, microservice, and AWS Lambda architectures	DUDA	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE

ID	Año	Título	Paso1:Selección Inicial	Paso 2: Selección Final	CI1	CI2	CI3	CI4	CI5	CE 1	CE 2	CE 3	CE 4	CE 5	Resultado selección
64	2017	Microservices in agile software development: A workshop-based study into issues, advantages, and disadvantages	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
65	2017	Redundant and Reliable Architecture Based on Open Source Tools for Light-Rail-Transit On-Board-Systems	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
66	2017	Amorphous transdisciplinary service systems	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
67	2017	Performance evaluation of massively distributed microservices based applications	DUDA	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	NO	NO CUMPLE
68	2017	Microservices in Practice, Part 1: Reality Check and Service Design	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	CUMPLE
69	2016	Towards Integrating Microservices with Adaptable Enterprise Architecture	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
70	2016	Infrastructure Cost Comparison of Running Web Applications in the Cloud Using AWS Lambda and Monolithic and Microservice Architectures	DUDA	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	NO	NO CUMPLE
71	2016	Proceedings - 2016 IEEE Symposium on Service-Oriented System Engineering, SOSE 2016	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
72	2016	Practical Use of Microservices in Moving Workloads to the Cloud	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
73	2015	Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
74	2015	A service framework for parallel test execution on a developer's local development workstation	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
75	2015	A micro services architecture for use by microcontroller driven devices to provide interoperability with the open lean services architecture	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
76	2021	Fast and Efficient Performance Tuning of Microservices	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
77	2018	Microns: Commands for Building Bubble Microservices	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
78	2016	Workload characterization for microservices	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
79	2017	From Monolith to Microservices: A Dataflow-Driven Approach	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE

ID	Año	Título	Paso1:Selección Inicial	Paso 2: Selección Final	CI1	CI2	CI3	CI4	CI5	CE 1	CE 2	CE 3	CE 4	CE 5	Resultado selección
80	2022	Microvision: Static analysis-based approach to visualizing microservices in augmented reality	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
81	2019	Exploring Microservices as the Architecture of Choice for Network Function Virtualization Platforms	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
82	2021	Involving Stakeholders in the Implementation of Microservice-Based Systems: A Case Study in an Ambient-Assisted Living System	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
83	2020	Survey on Requirement-Driven Microservice System Evolution	DUDA	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	NO	NO CUMPLE
84	2021	Adopting Microservices for Industrial Control Systems: A Five Step Migration Path	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
85	2018	Towards Executable Specifications for Microservices	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
86	2022	An Empirical Analysis of IDS Approaches in Container Security	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
87	2022	MicroMILTS: Fault Location for Microservices Based Mutual Information and LSTM Autoencoder	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
88	2019	Design and Prototyping of Container-Enabled Cluster for High Performance Data Analytics	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
89	2020	Sizing and Coordinated Scaling of Microservices	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
90	2018	Towards Multi-Container Deployment on IoT Gateways	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
91	2018	Microservices: Building Blocks to New Workflows and Virtualization	SI	NO	SI	SI	SI	SI	SI	NO	NO	NO	NO	SI	NO CUMPLE
92	2020	Container Mapping and its Impact on Performance in Containerized Cloud Environments	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
93	2020	The Road beyond 5G: A Vision and Insight of the Key Technologies	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
94	2018	Microservices: The Journey So Far and Challenges Ahead	DUDA	NO	SI	SI	SI	SI	SI	NO	NO	NO	NO	SI	NO CUMPLE
95	2022	Introducing Cloud-Assisted Micro-Service-Based Software Development Framework for Healthcare Systems	DUDA	NO	SI	SI	SI	SI	SI	NO	NO	NO	NO	SI	NO CUMPLE
96	2020	Review of micro-services architectures and runtime dynamic binding	SI	NO	SI	SI	SI	SI	SI	NO	NO	NO	NO	SI	NO CUMPLE

ID	Año	Título	Paso1:Selección Inicial	Paso 2: Selección Final	CI1	CI2	CI3	CI4	CI5	CE 1	CE 2	CE 3	CE 4	CE 5	Resultado selección
97	2020	An Intelligent Anomaly Detection Scheme for Micro-Services Architectures With Temporal and Spatial Data Analysis	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
98	2020	Highly-Distributed Systems Based on Micro-Services and their Construction Paradigms	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
99	2019	A Survey: Microservices Architecture in Advanced Manufacturing Systems	DUDA	NO	SI	SI	SI	SI	SI	NO	NO	NO	NO	SI	NO CUMPLE
100	2019	Viewpoint-Specific Model-Driven Microservice Development with Interlinked Modeling Languages	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
101	2015	A Reusable Automated Acceptance Testing Architecture for Microservices in Behavior-Driven Development	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
102	2022	Enterprise Architecture Approach to Build API Economy	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
103	2018	Microservices	DUDA	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	SI	NO CUMPLE
104	2016	Towards Integrating Microservices with Adaptable Enterprise Architecture	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
105	2017	Decision Management for Micro-Granular Digital Architecture	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
106	2016	Practical Use of Microservices in Moving Workloads to the Cloud	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
107	2018	In Pursuit of Architectural Agility: Experimenting with Microservices	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
108	2019	Research and Application of Battery Production Data Management System Based on Microservice	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
109	2020	Research on Architecture of Internet of Things Platform Based on Service Mesh	DUDA	NO	SI	SI	SI	SI	SI	NO	NO	NO	NO	SI	NO CUMPLE
110	2019	Software Versioning with Microservices through the API Gateway Design Pattern	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
111	2015	A Service Framework for Parallel Test Execution on a Developer's Local Development Workstation	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
112	2019	Attributes Assessing the Quality of Microservices Automatically Decomposed from Monolithic Applications	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
113	2021	Microservices: Migration of a Mission Critical System	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE

ID	Año	Título	Paso1:Selección Inicial	Paso 2: Selección Final	CI1	CI2	CI3	CI4	CI5	CE 1	CE 2	CE 3	CE 4	CE 5	Resultado selección
114	2017	Microservices: Building blocks to new workflows and virtualization	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
115	2019	Towards Novel Security Architectures for Network Functions Virtualization	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
116	2019	Management of User Equipment Context as a Radio Access Network Service	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
117	2019	Service Mesh: Challenges, State of the Art, and Future Research Opportunities	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
118	2022	Design, implementation, and testing of a microservices-based Digital Twins framework for network management and control	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
119	2018	Visualization Tool for Designing Microservices with the Monolith-First Approach	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
120	2016	Infrastructure Cost Comparison of Running Web Applications in the Cloud Using AWS Lambda and Monolithic and Microservice Architectures	DUDA	NO	SI	SI	NO	SI	SI	NO	NO	NO	NO	NO	NO CUMPLE
121	2015	Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
122	2022	Design and Testing on Migration of Remiss-Supply in Banking System to Microservice Architecture	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
123	2022	A cybersecurity framework to guarantee reliability and trust for digital service chains – GUARD	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
124	2018	Migrating Towards Microservice Architectures: An Industrial Survey	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
125	2020	An Intelligent Anomaly Detection Scheme for Micro-Services Architectures With Temporal and Spatial Data Analysis	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
126	2022	Introducing Cloud-Assisted Micro-Service-Based Software Development Framework for Healthcare Systems	DUDA	NO	SI	SI	SI	SI	SI	NO	NO	NO	NO	SI	NO CUMPLE
127	2020	The Road beyond 5G: A Vision and Insight of the Key Technologies	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
128	2021	Microservices: Migration of a Mission Critical System	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
129	2018	Laboratory information system and necessary improvements in function and programming	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE

ID	Año	Título	Paso1:Selección Inicial	Paso 2: Selección Final	CI1	CI2	CI3	CI4	CI5	CE 1	CE 2	CE 3	CE 4	CE 5	Resultado selección
130	2019	Exploring Microservices as the Architecture of Choice for Network Function Virtualization Platforms	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
131	2020	Design principles, architectural smells and refactorings for microservices: a multivocal review	DUDA	NO	SI	SI	SI	SI	SI	NO	NO	NO	NO	SI	NO CUMPLE
132	2021	Smart Node Networks Orchestration: A New E2E Approach for Analysis and Design for Agile 4.0 Implementation	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
133	2021	Promises and challenges of microservices: an exploratory study	DUDA	NO	SI	SI	SI	SI	SI	NO	NO	NO	NO	SI	NO CUMPLE
134	2016	Designed and delivered today, eroded tomorrow?: towards an open and lean architecting framework balancing agility and sustainability	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE
135	2017	An Agile Farm Management Information System Framework for Precision Agriculture	NO	-	-	-	-	-	-	-	-	-	-	-	NO CUMPLE



Anexo C: Listado detallado de estudios primarios

La lista detallada de los estudios primarios seleccionados en la investigación se muestra a continuación:

Tabla C.1: Lista detallada de estudios primarios

ID	Año	País	Título	Autores
S01	2022	Egipto	Introducing Cloud-Assisted Micro-Service-Based Software Development Framework for Healthcare Systems	Zaki J., Islam S.M.R., Alghamdi N.S., Abdullah-Al-Wadud M., Kwak K.-S.
S02	2021	Canada	Promises and challenges of microservices: an exploratory study	Wang Y., Kadiyala H., Rubin J.
S03	2021	EE.UU	Microservice-based architectures: An evolutionary software development model	Fellah A., Bandi A.
S04	2020	India	Review of micro-services architectures and runtime dynamic binding	Rasheedh J.A., Saradha S.
S05	2020	Suiza	Design principles, architectural smells and refactorings for microservices: a multivocal review	Neri D., Soldani J., Zimmermann O., Brogi A.
S06	2020	China	Research on architecture of internet of things platform based on service mesh	He X., Deng F.
S07	2019	Austria	A survey: Microservices architecture in advanced manufacturing systems	Homay A., Zoitl A., De Sousa M., Wollschlaeger M.
S08	2019	Holanda	Attributes assessing the quality of microservices automatically decomposed from monolithic applications	Cojocaru M.-D., Oprescu A., Uta A.
S09	2019	Italia	Consistency and Availability in Microservice Architectures	Rossi D.
S10	2019	Francia	A web service security governance approach based on dedicated micro-services	Abidi S., Essafi M., Guegan C.G., Fakhri M., Wittl H., Ghezala H.H.B.
S11	2018	EE.UU	Microservices: Building blocks to new workflows and virtualization	Gallipeau D., Kudrle S.
S12	2018	Brasil	Microservices: The journey so far and challenges ahead	Jamshidi P., Pahl C., Mendonca N.C., Lewis J., Tilkov S.
S13	2018	China	Distributed monitoring system for microservices-based IoT middleware system	Kang R.; Zhou Z.; Liu J.; Zhou Z.; Xu S.
S14	2017	Peru	Application of mixed distributed software architectures for social-productive projects management in peru	Posadas J.V.
S15	2017	Suiza	Microservices tenets: Agile approach to service development and deployment	Zimmermann O.
S16	2017	Colombia	Cost comparison of running web applications in the cloud using monolithic, microservice, and AWS Lambda architectures	Villamizar M., Garcés O., Ochoa L., Castro H., Salamanca L., Verano M., Casallas R., Gil S., Valencia C., Zambrano A., Lang M.
S17	2017	Suiza	Microservices in Practice, Part 1: Reality Check and Service Design	Pautasso C., Zimmermann O., Amundsen M., Lewis J., Josuttis N.
S18	2021	Turquía	Does Domain-Driven Design Lead to Finding the Optimal Modularity of a Microservice?	Vural H., Koyuncu M.
S19	2017	Dinamarca	Microservices: Yesterday, today, and tomorrow	Dragoni N., Giallorenzo S., Lafuente A.L., Mazzara M.,

ID	Año	País	Título	Autores
				Montesi F., Mustafin R., Safina L.
S20	2018	Alemania	Challenges of microservices architecture: A survey on the state of the practice	Ghofrani J., Lübke D.
S21	2017	Suiza	Microservices in Practice, Part 2: Service Integration and Sustainability	Pautasso C., Zimmermann O., Amundsen M., Lewis J., Josuttis N.
S22	2018	Grecia	Cyber-physical microservices: An IoT-based framework for manufacturing systems	Thramboulidis K., Vachtsevanou D.C., Solanos A.
S23	2017	Reino Unido	Towards an understanding of microservices	Shadija D., Rezai M., Hill R.
S24	2016	Alemania	Microservices approach for the internet of things	Butzin B., Golatowski F., Timmermann D.
S25	2017	EE.UU	Disambiguation and comparison of SOA, microservices and self-contained systems	Cerny T., Donahoo M.J., Pechanec J.
S26	2017	Alemania	Differences between model-driven development of service-oriented and microservice architecture	Rademacher F., Sachweh S., Zundorf A.
S27	2018	Alemania	Microservice architecture and model-driven development: Yet singles, Soon Married (?)	Rademacher F., Sorgalla J., Wizenty P.N., Sachweh S., Zündorf A.
S28	2017	China	Reflections on SOA and Microservices	Xiao Z., Wijegunaratne I., Qiang X.
S29	2017	Suiza	Microservices tenets: Agile approach to service development and deployment	Zimmermann O.
S30	2018	Dinamarca	Microservices: How to make your application scale	Dragoni N., Lanese I., Larsen S.T., Mazzara M., Mustafin R., Safina L.

Anexo D: Listado detallado de fuentes de publicación

La distribución de estudios seleccionados por fuentes de publicación se presenta a continuación:

Tabla D.1: Lista de fuentes de publicación de los estudios

Fuente	Tipo	Cantidad
ACM International Conference Proceeding Series	Conferencia	1
CEUR Workshop Proceedings	Conferencia	1
Computer Science - Research and Development	Revista	2
Empirical Software Engineering	Revista	1
EPIc Series in Computing	Conferencia	1
ES - International Conference on Enterprise Systems	Conferencia	1
ETFa - International Conference on Emerging Technologies and Factory Automation	Conferencia	1
ICAC - International Conference on Automation and Computing	Conferencia	1
ICMTMA - International Conference on Measuring Technology and Mechatronics Automation	Conferencia	1
ICPS - Industrial Cyber-Physical Systems	Conferencia	1
ICSAW - International Conference on Software Architecture Workshops	Conferencia	1
IEEE Access	Revista	2
IEEE Software	Revista	3
INDIN - International Conference on Industrial Informatics	Conferencia	1
INTERCON - International Congress on Electronics, Electrical Engineering and Computing	Conferencia	1
ISMAC - International Conference on IoT in Social, Mobile, Analytics and Cloud	Conferencia	1
ISPDC - International Symposium on Parallel and Distributed Computing	Conferencia	1
LNBP - Lecture Notes in Business Information Processing	Conferencia	1
LNCS - Lecture Notes in Computer Science	Conferencia	2
Present and Ulterior Software Engineering	Libro	1
Procedia Computer Science	Conferencia	1
RACS - Research in Adaptive and Convergent Systems	Conferencia	1
Service Oriented Computing and Applications	Revista	1
SMPTE Motion Imaging Journal	Revista	1
Software-Intensive Cyber-Physical Systems	Conferencia	1