

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE ESMERALDAS**



CARRERA:

INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
PREVIO AL GRADO ACADÉMICO DE INGENIERÍA DE TECNOLOGÍA
DE LA INFORMACIÓN Y COMUNICACIÓN

TEMA DE INVESTIGACIÓN:

MAPEO SISTEMÁTICO DE HERRAMIENTAS Y PLATAFORMAS PARA
MICROSERVICIOS WEB.

LÍNEA DE INVESTIGACIÓN:

ESTUDIO, DISEÑO E IMPLEMENTACIÓN DE SOFTWARE

PREVIO A LA OBTENCIÓN DE TÍTULO DE:

INGENIERÍA DE TECNOLOGÍAS DE LA INFORMACIÓN

AUTOR:

YUSLEIDY LISSETH GODOY OSORIO

ASESOR:

MGT. JAIME SAYAGO HEREDIA

ESMERALDAS, 2022

TRIBUNAL DE GRADUACIÓN

Título: Mapeo sistemático de herramientas y plataformas para microservicios web.

Autor(a): Yusleidy Lisseth Godoy Osorio

Mgt. Jaime Sayago Heredia

f. _____

Asesor

Mgt. José Luis Carbajal

f. _____

Lector #1

Mgt. Xavier Quiñonez Ku

f. _____

Lector #2

Mgt. Xavier Quiñonez Ku

f. _____

Coordinador de Carrera

AUTORÍA

Yo, **Yusleidy Lisseth Godoy Osorio** con número de cédula de identidad 0850300948 manifiesto que mediante la presente investigación sobre el tema “**MAPEO SISTEMÁTICO DE HERRAMIENTAS Y PLATAFORMAS PARA MICROSERVICIOS WEB.**” los resultados obtenidos como tesis de grado, previo a la obtención del título de “**INGENIERO EN TECNOLGÍAS DE LA INFORMACIÓN**” son de total responsabilidad del autor, y que se ha respetado las fuentes de información consultadas, realizando las citas correspondientes y los resultados alcanzados son totalmente legítimos. Al mismo tiempo declaro que todo el contenido incluyendo resultados, discusión, conclusiones, recomendaciones y otros efectos legales y académicos que se desglosan, son y serán exclusiva responsabilidad legal y académica del autor y de la PUCESE.

Godoy Osorio Yusleidy Lisseth

C.I 0850300948

AGRADECIMIENTOS

Las palabras no pueden expresar mi gratitud al creador por mantenerme con vida, gozar de buena salud, mantener a mis padres con vida y proveerme de sabiduría e inteligencia a lo largo de esta etapa.

Quisiera expresar mi más profundo agradecimiento a mis padres quienes estuvieron presentes en cada momento crucial e importante de mi vida, por haber depositado su confianza en mí, por todos los valores que me han inculcado gracias, Adrián Godoy y Yemanya Osorio por sus consejos; ayuda y apoyo incondicional en los momentos más difíciles.

A mi hermana Kennya Godoy por el apoyo y amor incondicional, estar siempre presente acompañándome y sobre todo por jamás dejar de creer en mi capacidad de superación.

A mis buenos amigos y compañeros de aula Santiago, Adonis, Mariana, Denys, Pablo y Adrián por brindarme su amistad y por demostrarme que en la universidad no solo vamos a continuar con nuestra formación como profesionales, sino que también podemos encontrar grandes amistades.

A mis tíos, abuelos y demás amistades que de una u otra manera han estado conmigo y han contribuido con mi proyecto de vida y ahora son parte de este logro.

Estoy enormemente agradecida con todos los docentes, por ser excelentes guías, impartir sus conocimientos y consejos, en especial al Mgt. Jaime Sayago, Mgt. Xavier Quiñonez y Mgt. Marc Grob por su apoyo, correcciones y directrices para realizar correctamente el proyecto de investigación y así finalmente alcanzar la meta anhelada.

DEDICATORIA

Este trabajo es dedicado a mis padres por darme el mejor regalo que un padre puede brindar a sus hijos que es la educación además porque ellos me han brindado su apoyo incondicional y porque siempre serán mi mayor fuente de motivación e inspiración.

RESUMEN

El término “microservicios” ha sido tendencia en los últimos años. Los microservicios o arquitectura de microservicios remplazaron las arquitecturas monolíticas. Debido a la basta cantidad de información referente a las herramientas que suelen ser utilizadas para el despliegue de microservicios fue desarrollada la presente investigación, esta pretende plantear algunas herramientas utilizadas por desarrolladores en el mundo del despliegue de microservicios.

El principal propósito del presente trabajo es la investigación de las principales herramientas utilizadas en el mundo del despliegue de microservicios web. Para llevar a cabo dicha investigación fue necesario implementar un mapeo sistemático en el cual se consideró criterios de inclusión y exclusión a estudios publicados dentro del periodo 2018-2022. El estudio fue desarrollado a partir de una metodología de carácter formal la cual es aplicada en investigaciones relacionadas con el software, desarrollo y ciencias de la computación. El estudio de mapeo sistemático en cuestión fue realizado siguiendo los parámetros propuestos por Peterson con la diferencia de que en este estudio fueron planteadas 3 preguntas de investigación, además, fueron recuperados 69 estudios en los cuales se identificaron algunas herramientas utilizadas para el monitoreo, mensajería y lenguajes de programación utilizados frecuentemente. Entre los resultados obtenidos en cada una de estas categorías se escogieron tres para posterior realizar una recategorización y estudio de estas.

Palabras claves: Microservicios web, Mapeo sistemático, Herramientas,

Abstract

The term "microservices" has been trending in recent years. Microservices or microservices architecture replaced monolithic architectures. Due to the vast amount of information regarding the tools that are usually used for the deployment of microservices, this research was developed, it aims to raise some tools used by developers in the world of microservices deployment.

The main purpose of this work is the investigation of the main tools used in the world of web microservices deployment. To carry out such research it was necessary to implement a systematic mapping in which inclusion and exclusion criteria were considered to studies published within the period 2018-2022. The study was developed from a formal methodology which is applied in research related to software, development and computer science. The systematic mapping study in question was conducted following the parameters proposed by Peterson with the difference that in this study 3 research questions were posed, in addition, 69 studies were retrieved in which some tools used for monitoring, messaging and frequently used programming languages were identified. Among the results obtained in each of these categories, three were chosen for subsequent re-categorization and study.

Keywords: Web microservices, Systematic mapping, Tools.

ÍNDICE DE CONTENIDOS

RESUMEN	V
ÍNDICE DE ILUSTRACIONES	8
ÍNDICE DE TABLAS	8
INTRODUCCIÓN	9
Presentación de la investigación	9
Planteamiento del problema	9
Justificación.....	10
Objetivos	11
Objetivo general	11
Objetivos específicos	11
CAPÍTULO I: MARCO TEÓRICO	12
1.1 Bases teóricas científicas.....	12
1.1.1 Microservicios.....	12
1.1.2 Arquitectura de microservicios	15
1.1.3 Lenguaje de programación.....	16
1.1.4 Monitoreo de aplicaciones	16
1.1.5 Impacto operacional.....	16
1.1.6 Orquestadores de contenedores más utilizados.....	17
1.2 Antecedentes de la investigación	19
CAPÍTULO II: METODOLOGÍA	22
2.1 Delimitación	22
2.2 Tipos de investigación.....	22
2.3 Métodos.....	22
2.4 Técnicas.....	23
2.5 Descripción de instrumentos	23
2.6 Técnicas de procesamiento y análisis de datos	23
2.7 Normas éticas	24
CAPÍTULO III: RESULTADOS.....	25
3.1 Descripción del estudio de mapeo sistemático.....	25
3.2 Preguntas de investigación	27
3.2.1 ¿Cuáles son las principales razones para utilizar microservicios?	27
3.2.2 ¿Cuáles son las herramientas más utilizadas en el área de lenguaje de programación, mensajería y monitoreo en el área de despliegue de microservicios?.....	28
3.3 Definición de estrategia de búsqueda.....	31
3.4 Definición de fuentes de búsqueda.....	32

3.5	Estrategia de búsqueda	32
3.6	Criterio de inclusión y exclusión.....	32
CAPÍTULO IV: DISCUSIÓN		43
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES		44
	Conclusiones	44
	Recomendaciones.....	45
REFERENCIAS BIBLIOGRÁFICAS.....		46
ANEXOS		51

ÍNDICE DE ILUSTRACIONES

Ilustración 1	Death Star microservicios implementados en Amazon y Netflix.....	12
Ilustración 2	Termino Microservicios clasificado por países	13
Ilustración 3	Termino Cloud Computing clasificado por países	13
Ilustración 5	Proceso de refinamiento de información extraída	33
Ilustración 6	Cantidad de papers extraídos/seleccionados según la base de datos	38
Ilustración 7	Porcentaje de estudios extraídos por año	38
Ilustración 8	Número de artículos seleccionados según el país.....	39
Ilustración 9	Clasificación de los papers seleccionados según el tipo de documento	40
Ilustración 10	Porcentaje de artículos seleccionados según el continente	41
Ilustración 11	Gráfico metodología bubble	42

ÍNDICE DE TABLAS

Tabla 1	Principales características de la implementación de la arquitectura de microservicios	14
Tabla 2	Lenguajes de programación más utilizados para el desarrollo de microservicios	28
Tabla 3	Numero de empresas a nivel mundial que utilizan los lenguajes de programación (Python, C, Java).....	30
Tabla 4	Herramientas de monitoreo.....	30
Tabla 5	Herramientas extraídas para mensajería de microservicios web	31
Tabla 6	Bases de datos científicas empleadas para la recolección de información	32
Tabla 7	Estudios extraídos y orientados al desarrollo de la investigación	33
Tabla 8	Resultado de los papers seleccionados para el desarrollo de la investigación.....	34
Tabla 9	Cronograma de actividades para el desarrollo de la investigación	51
Tabla 10	Timelapse segunda parte.....	51

INTRODUCCIÓN

Presentación de la investigación

El mapeo sistemático es un método de investigación de un asunto en específico y se utiliza para recopilar, comparar o presentar evidencia de una investigación [1]. Uno de los objetivos de este método de investigación es estructurar un área de esta. Los microservicios se pueden definir como un enfoque arquitectónico que sirve para crear aplicaciones en la nube [2]. Cada aplicación está construida como un conjunto de servicios, a su vez cada servicio ejecuta sus propios procesos y se comunica a través de una API (Application Programming Interfaces).

Si se aplica este estilo de arquitectura se puede tener como resultado: agilidad de procesos, aumento en cuanto a la productividad del desarrollador, flexibilidad, resistencia, confiabilidad, capacidad de mantenimiento, división de preocupaciones, facilidad en cuanto a su implementación; pero, esos beneficios conllevan desafíos, como el descubrimiento de servicios en la red, la gestión de la seguridad y la optimización de las comunicaciones. Sin embargo, cuando se abordan adecuadamente, estos desafíos permiten que el sistema obtenga la mayor cantidad de los beneficios antes mencionados. No todo en los microservicios web es color de rosa, ya que para un óptimo funcionamiento es necesario gestionar una arquitectura distribuida, entre ellos se puede encontrar: latencia, falta de fiabilidad de la red, tolerancia a fallos e incoherencia de datos [3].

Planteamiento del problema

El tema de microservicios se popularizó a raíz de que empresas grandes como Amazon, SoundCloud, Netflix, Twitter, entre otras utilizaran este tipo de enfoque [4]. Luego de tomar la importante decisión de migrar de un sistema monolítico a uno enfocado a microservicios las empresas antes mencionadas aseguran que muchos de sus problemas fueron solventados. A pesar de las muchas ventajas que ofrecen los microservicios web, no siempre son la mejor opción para todas las organizaciones. Antes de comprometerse, es necesario asegurarse de que la migración de microservicios se alinee a los objetivos que se intenta cumplir. Uno de los primeros pasos para determinar si la implementación de microservicios es adecuada, es evaluar los desafíos a los que probablemente enfrentará durante la transformación. A partir de ahí, deberá averiguar si su equipo u organización está equipado para manejarlos [5] [6].

Los microservicios son un patrón de estilo arquitectónico para desarrollar aplicaciones grandes, complejas y simples para la empresa; en esta arquitectura, las aplicaciones grandes y complejas se descomponen en varios pequeños servicios individuales que se pueden mantener,

cada servicio tiene una sola responsabilidad y esta es comunicarse entre sí mediante los protocolos HTTP (Hypertext Transfer Protocol) / HTTPS (Hypertext Transfer Protocol) / WebSocket. Para desarrollar las aplicaciones usando este patrón, los desarrolladores necesitan herramientas y tecnologías para construir y monitorear estos microservicios [7].

Si no se cuenta con herramientas para el desarrollo o gestión de microservicios podrían presentarse varios problemas, por ejemplo, correr el riesgo de que la API del servicio web no esté funcionando correctamente, para que esto no ocurra por ejemplo se debería implementar una herramienta para realizar pruebas a la API y así tener un mejor control de esta [8].

Lo que es bueno para Netflix puede no serlo para su empresa u organización. Las grandes empresas, como Netflix, que emplean varios equipos de desarrollo, pueden beneficiarse enormemente de la naturaleza modular de los microservicios y los servicios independientes que estos ofrecen. El costo de implementar una arquitectura basada en microservicios tiende a ser más alto que construir una aplicación monolítica (debe tener en cuenta un sistema de monitoreo de registros, seguimiento de solicitudes y la capacidad de implementar todo el sistema, así como cualquier microservicio individual). Puede que la implementación de los microservicios no valga la pena para los sistemas que no esperan un gran tráfico, problemas de escalabilidad y la combinación de varias tecnologías [9] [10].

La arquitectura de microservicios obliga a pensar en muchas variables relacionadas con el despliegue que no se dan en las aplicaciones monolíticas. Desarrollar aplicaciones utilizando la arquitectura de microservicios es mucho más complejo que hacerlo con una arquitectura monolítica. El proceso de desarrollo de software siempre debe ajustarse a las necesidades reales del proyecto. No hay una arquitectura que sea intrínsecamente mejor que otra. Las estructuras monolíticas todavía tienen su lugar en el desarrollo de software moderno y no han quedado obsoletas por la cada vez más popular arquitectura de microservicios [11] [12].

Justificación

El implementar microservicios web a una organización es fundamental ya que le aportará significativas ventajas como por ejemplo flexibilidad, reusabilidad, reducción de costos y riesgos, escalabilidad, adaptabilidad, entre otros. Debido a los beneficios que esta arquitectura ofrece, existe gran cantidad de herramientas o plataformas para el desarrollo de microservicios web es por esto por lo que se realizará la presente investigación, para que se pueda conocer las diferentes características que ofrecen las distintas herramientas o plataformas. Esta investigación también pretende de cierta forma ayudar a conocer cuál sería la plataforma o

herramienta que se adapte mejor a las necesidades de su sistema. Elegir las herramientas adecuadas para el desarrollo o despliegue de microservicios web suele ser un proceso que lleva mucho tiempo debido a que hay muchas situaciones, asuntos o puntos a considerar como por ejemplo la funcionalidad, licencia, ventajas y desventajas. Siempre se recomienda entender primero los retos existentes y luego buscar una herramienta adecuada que se adapte a los requisitos del sistema.

Objetivos

Objetivo general

Investigar el estado del arte de las principales herramientas para el despliegue de microservicios web utilizando el método de mapeo sistemático para determinar su potencial industrial.

Objetivos específicos

- a) Describir el marco teórico de tecnologías de microservicios.
- b) Definir la metodología para el desarrollo del Systematic Mapping Study (SMS).
- c) Explicar los resultados del análisis de datos de los artículos.
- d) Determinar conclusiones y recomendaciones de los resultados obtenidos.

CAPÍTULO I: MARCO TEÓRICO

1.1 Bases teóricas científicas

Para poder llevar a cabo este proyecto de investigación es necesario conocer conceptos fundamentales, los cuales sirven para comprender el estudio que se realiza en esta investigación, esta sección ha sido elaborada en base a artículos científicos, revistas, conferencias y libros acerca de los microservicios web.

1.1.1 Microservicios

Grandes empresas como lo son Netflix, Amazon y BBC prefieren implementar la arquitectura de microservicios respecto a la monolítica debido a que los microservicios encajan perfectamente en el modelo y metodología de negocios que poseen estas empresas. Las empresas tienen la oportunidad de hacer evolucionar rápidamente su software y hacerse notar o diferenciar en un amplio panorama tecnológico el cual es cada vez más competitivo. Un informe reciente de Forrester [13] muestra que el 76 % de las empresas consideran que los microservicios son una agenda clave [14].

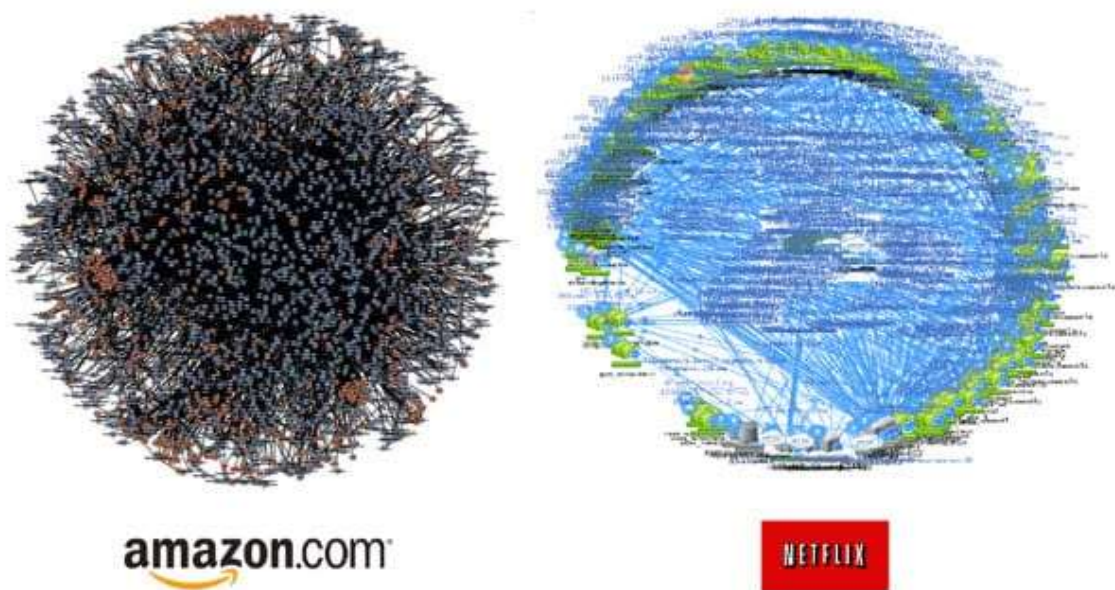


Ilustración 1 Death Star microservicios implementados en Amazon y Netflix

La arquitectura de microservicios es considerada una metodología usualmente implementada en el mundo desarrollo de aplicaciones basadas en servicios. Esta metodología, divide una aplicación grande en pequeñas unidades de servicio las cuales a su vez logran ser independientes. Resumiendo lo anterior, un microservicio web consiste en una arquitectura orientada a servicios (SOA) la cual divide una aplicación completa en una colección de servicios interconectados en donde cada uno de estos servicios atiende solo una necesidad

empresarial [15].

Términos como microservicios web y computación en la nube son actualmente tendencia en búsqueda tal y como se muestra en la Ilustración 2 e Ilustración 3, según el gigante en tendencias de búsqueda de la mundialmente conocida empresa Google muestra que dentro del top 5 de países con mayor cantidad de búsquedas realizadas con el termino Cloud Computing está; Etiopía, Zimbabue, India, Nepal y Camerún.

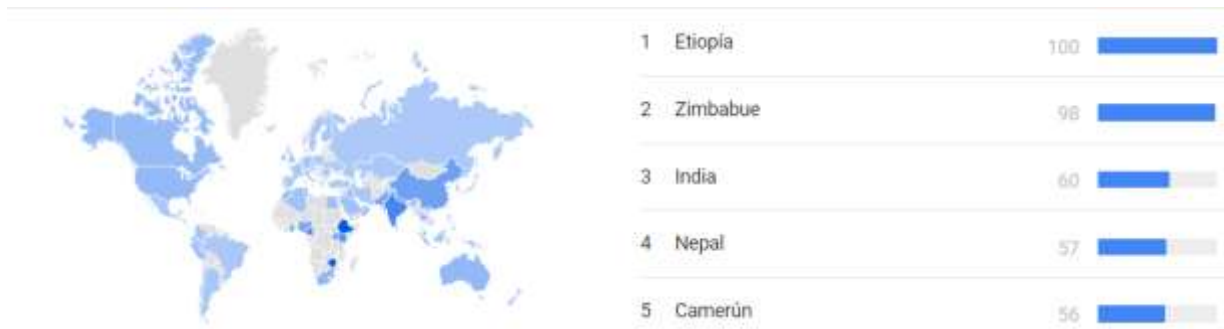


Ilustración 2 Termino Microservicios clasificado por países

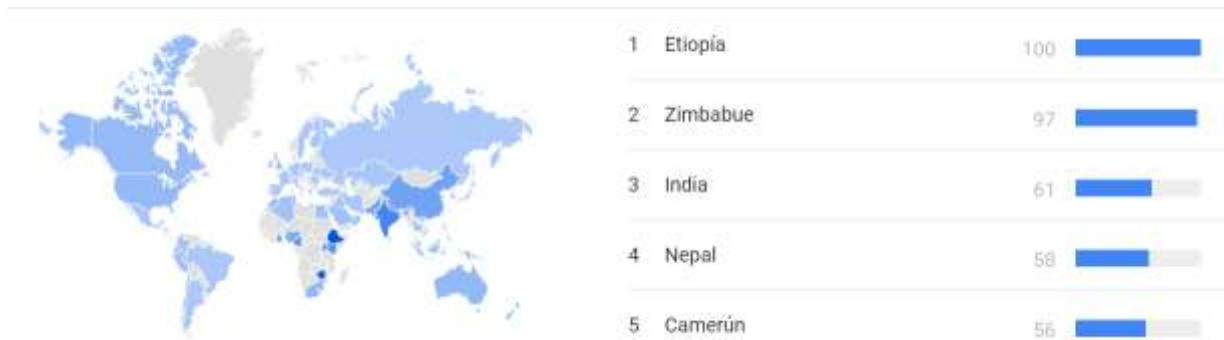


Ilustración 3 Termino Cloud Computing clasificado por países

Los microservicios fueron desarrollados con la principal idea de resolver los problemas que normalmente suelen encontrarse en la arquitectura o diseño monolítico. En resumen, los microservicios describen el proceso por el que las aplicaciones son fragmentadas a servicios más pequeños y poco acoplados. La Tabla 1 muestra las características más importantes presentadas por Mazlami et al. acerca de la implementación de la arquitectura de microservicios [16].

Tabla 1 Principales características de la implementación de la arquitectura de microservicios

Descripción	
a)	La complejidad se reduce a que cada microservicio se convierte en una miniaplicación que es mucho más simple y fácil de entender.
b)	Los ciclos de desarrollo se aceleran ya que el equipo de desarrollo se enfoca únicamente en ese servicio y a su vez pueden ser utilizados por muchas aplicaciones.
c)	Pueden ser ejecutados en la nube en una VM, en un contenedor Docker o en instancias sin servidor, lo que lo hace ideal para los servicios en la nube modernos
d)	Son más fáciles de escalar, además, es más rápido identificar y resolver cuellos de botella

Un ejemplo exitoso del uso de microservicios se puede visualizar en el gigante de streaming Netflix. Esta compañía inicialmente desarrolló sus operaciones bajo un modelo monolítico hasta el año 2008, año en el que sufrió una interrupción de TI debido a una corrupción masiva de la base de datos, luego de sufrir este gran inconveniente, los altos mandos de esta empresa decidieron dividir la aplicación, de tal manera que pasaron de utilizar una arquitectura monolítica (servicios monolíticos) a una arquitectura de servicios múltiples e independientes (microservicios), de esta manera lograron mejorar pilares fundamentales dentro de una empresa tales como la escalabilidad, confiabilidad y disponibilidad [17].

En el informe de Forrester [13], muestra que el 29% de las empresas luchan con largos ciclos de implementación y el 16% de ellas no cumple con los plazos. Los ciclos de implementación extendidos de los microservicios aceleran el crecimiento de los ingresos y la reputación, lo que permite múltiples lanzamientos por año o incluso varias veces al día [18].

Una vez desarrollados, estos servicios también se pueden implementar de forma independiente entre sí y, por lo tanto, es fácil identificar los servicios, sin necesidad de escalar la aplicación completa. Los microservicios también ofrecen un mejor aislamiento de fallas por lo que, en el caso de que aparezca un error en un servicio, la aplicación completa no necesariamente dejará de funcionar [19].

Una de las características principales de una arquitectura basada en microservicios es que le permite pensar en los servicios como aplicaciones independientes y autónomas. El equipo encargado del desarrollo de cada servicio solo se preocupa por comprender las complejidades de este [20].

El uso de microservicios reduce la cantidad de esfuerzo necesario para identificar los cuellos de botella lentos en una aplicación. También permite escalar microservicios individuales para resolver esos cuellos de botella y brindar una mejor experiencia general del

usuario. Al simplificar la forma en que se pueden extraer los datos para varios usuarios finales, los microservicios permiten a los desarrolladores adaptar fácilmente la presentación de los datos para diferentes audiencias.

Debido a su arquitectura modular, los microservicios permiten un alto nivel de reutilización de código y datos, lo que agiliza y facilita la implementación de casos de uso y soluciones adicionales basados en datos para obtener un valor comercial adicional. Los microservicios utilizan tecnologías de transmisión de eventos para permitir una fácil integración en comparación con los protocolos de comunicaciones entre procesos de gran peso de las arquitecturas SOA tradicionales [21].

Las herramientas para el desarrollo de microservicios se dividen principalmente en las siguientes categorías: lenguaje de programación, framework de desarrollo y monitoreo de aplicaciones, estas desempeñan un papel esencial en el desarrollo de microservicios y también son consideradas categorías que ayudan a aumentar la productividad de los desarrolladores.

1.1.2 Arquitectura de microservicios

Arquitectura de microservicios vs arquitectura monolítica

Arquitectura monolítica

Una arquitectura o aplicación monolítica está diseñada para manejar múltiples tareas relacionadas. Suelen ser aplicaciones complejas que abarcan varias funciones estrechamente acopladas. Por ejemplo, considere una aplicación SaaS de comercio electrónico monolítica, esta puede contener un servidor web, un equilibrador de carga, un servicio de catálogo que ofrece imágenes de productos, un sistema de pedidos, una función de pago y un componente de envío. Debido a su amplio alcance, las herramientas monolíticas tienden a tener enormes bases de código y hacer un pequeño cambio en una sola función puede requerir compilar y probar toda la plataforma, lo que va en contra del enfoque ágil que prefieren los desarrolladores actualmente.

La arquitectura monolítica se puede utilizar en proyectos que no requieren una respuesta en tiempo real y pueden soportar el tiempo de inactividad. Este tipo de proyectos están limitados a un cierto tamaño si se espera que el tamaño de la aplicación web supere ese límite, entonces se debe adoptar por una arquitectura de microservicios [22].

Arquitectura de microservicios

Por otro lado, una arquitectura de microservicios implica aplicaciones más pequeñas implementadas de forma independiente como servicios poco acoplados, unidas entre sí a través de la integración de aplicaciones. Con las aplicaciones de microservicio, la lógica empresarial puede abarcar varias plataformas, incluido el software como servicio, bases de datos locales y aplicaciones desarrolladas internamente que satisfacen necesidades que ninguna aplicación SaaS maneja [23] [12].

Los microservicios pueden ser más sencillos de desarrollar, tienen un alcance más pequeño y, por lo tanto, un tamaño más pequeño, lo que facilita a los desarrolladores mejorarlos a través de la integración y entrega continuas (CI / CD). Además, los microservicios se pueden escribir o desarrollar en cualquier lenguaje de programación y pueden comunicarse con otros microservicios a través de una API [24] [25].

1.1.3 Lenguaje de programación

Los microservicios web se pueden desarrollar utilizando diferentes lenguajes de programación y tecnología, la ventaja del desarrollo de microservicios es que el lenguaje de programación que se elija depende completamente del requisito de la función comercial. Diferentes servicios en la misma aplicación pueden usar diferentes lenguajes de programación y tecnología, lo que da libertad a los desarrolladores. El lenguaje de programación basado en JDK (Java) es el más popular en el desarrollo de microservicios en Spring Boot y Elixir [26].

1.1.4 Monitoreo de aplicaciones

Una vez desarrollada y desplegada la aplicación, es muy importante supervisar los eventos de funcionamiento de las aplicaciones en el entorno de producción. El servicio/aplicación desplegada puede ser manejado y monitoreado proactivamente y evitar que el sistema se caiga, lo que puede impactar en el negocio [27][28].

Cuando se empieza a desarrollar aplicaciones usando microservicios, es necesario asegurarse de que todos los servicios individuales puedan comunicarse entre sí usando una API, para esto se debe calibrar todas las API mientras se desarrolla la aplicación para pruebas de API, las herramientas más famosas son Cartero, Apache y JMeter (para probar el equilibrio de carga en el servicio / aplicación).

1.1.5 Impacto operacional

Al considerar cualquier infraestructura, una de las preguntas que se podría plantear en primera instancia es: "¿Cuáles serán los impactos operativos de la nueva tecnología?" En caso de que

haya decidido adoptarlos, sin duda hay algunos impactos importantes que se debe considerar:

Costo

El costo es un factor importante que viene a la mente de los desarrolladores al tomar la decisión final de adoptar cualquier arquitectura de software.

Fiabilidad

Cuando se trata de confiabilidad, los microservicios tienen ventaja en este punto. Los sistemas monolitos no son más que una gran cantidad de binarios de aplicaciones. Por casualidad, si la implementación falla, toda la aplicación se cae. En comparación con los sistemas monolitos, los microservicios son muy confiables. Incluso en los casos en que falla un servicio, la aplicación no se desactivará en su totalidad [3]. Existen varias herramientas de detección de servicios como Hashicorp Consul, que verifica el estado de cada servicio [29].

Escalabilidad

Los monolitos se pueden escalar de varias formas. Uno de ellos es utilizar numerosas máquinas virtuales y luego enrutar la solicitud mediante un equilibrador de carga. La arquitectura de microservicio es más detallada y, por lo tanto, la escalabilidad de cada microservicio es más precisa y flexible. La escalabilidad es un factor de contraste de cualquier aplicación empresarial. Por lo tanto, existen múltiples técnicas disponibles en el mercado que garantizan que los microservicios sean escalables. Algunas de las herramientas populares disponibles en el mercado son Amazon EKS, Amazon ECS, Docker y Kubernetes [29].

Tiempo de comercialización

Debido al gran tamaño y número de dependencias, los mecanismos que se encuentran en la construcción e implementación de monolitos son catalogados más complicados y consumen más tiempo en comparación a los microservicios, estos son sensibles a los recursos y se construyen para escalar. Como los módulos están desacoplados entre sí, es más fácil construir y desplegar. Esto aumenta la agilidad de la aplicación que se ejecuta en los microservicios y reduce significativamente el tiempo para que las aplicaciones de microservicios lleguen al mercado [29].

1.1.6 Orquestadores de contenedores más utilizados

Los contenedores han revolucionado la forma en que son distribuidas las aplicaciones al permitir entornos de prueba replicados, portabilidad, eficiencia de recursos, escalabilidad y capacidades de aislamiento inigualables. Si bien los contenedores ayudan a empaquetar

aplicaciones para facilitar su implementación y actualización, se debe implementar un conjunto de herramientas especializadas para administrarlas [23].

Para ayudar con esto, las herramientas de orquestación proporcionan un marco el cual permite la optimización de las distintas cargas de trabajo en contenedores. Estas herramientas contribuyen al apoyo en equipos de DevOps a de cierta forma administrar el ciclo de vida de los contenedores y, por lo tanto, implementar sus redes, equilibrio de carga, aprovisionamiento, escalado y más. Como resultado, las herramientas de orquestación ayudan a los equipos a aprovechar todos los beneficios de la contenedorización al ofrecer resiliencia de aplicaciones, seguridad mejorada y operaciones simplificadas.

Docker

Docker es una plataforma open source en donde se puede desarrollar, enviar y ejecutar varias aplicaciones de forma más rápida. Docker permite que las aplicaciones se ejecuten por separado de la infraestructura del host y traten la infraestructura como una aplicación administrada. Docker también ayuda a enviar código más rápido, probar más rápido, implementar más rápido y acortar el ciclo entre escribir código y ejecutar código. Docker lo hace combinando una plataforma de virtualización de contenedores liviana con flujos de trabajo y herramientas que ayudan a administrar e implementar aplicaciones Docker [30].

Azure

Azure es una plataforma de nube pública que proporciona servicios como software como servicio o SaaS; Plataforma como Servicio, o PaaS; e Infraestructura como Servicio, o IaaS. Estos servicios se pueden utilizar en lugar de o además de los servidores locales. Azure también ofrece soluciones para redes, almacenamiento y más [31].

Kubernetes

Kubernetes es un sistema de orquestación de contenedores la cual sirve para administrar aplicaciones distribuidas desde contenedores. La plataforma Kubernetes le permite implementar y administrar contenedores a gran escala. De esta manera, no tiene que preocuparse por qué máquinas virtuales específicas (en un centro de datos) realizan ciertas subtarefas de la aplicación. Kubernetes también se conoce como K8s, que es un numerónimo de Kubernetes [32] [33].

1.2 Antecedentes de la investigación

A continuación, se presentará el análisis y las descripciones de las investigaciones hechas en relación con el tema de estudio “Mapeo sistemático de herramientas y plataformas para microservicios web” que sirven de base de apoyo a nivel de entendimiento para realizar el análisis de las variables a estudiar, las cuales son: mapeo sistemático y microservicios web.

La información de la presente investigación fue recopilada de varias fuentes bibliográficas digitales entre ellas, IEEE Xplore, Google Académico, Scopus y Web Of Science. Para la ejecución de la búsqueda se utilizó una metodología basada en un protocolo de búsqueda científica. De esta manera se utilizó la cadena de búsqueda “(Web Microservices) AND (Systematic Mapping Study) OR (tools) OR (plataforms)”, esta metodología fue de gran ayuda, ya que se pudo recolectar información relacionada con el tema de estudio. Es así como se han seleccionado 6 estudios que abarcan relación directa a los microservicios y el estudio de mapeo sistemático. Los estudios seleccionados para el desarrollo de la investigación fueron escogidos en un rango de cinco años (2016-2021) y serán detallados a continuación:

La primera investigación seleccionada titula “*Architecting with microservices: A systematic mapping study*”, el propósito de esta investigación fue examinar el estado de la arquitectura con microservicios. Esto fue posible identificando, clasificando y evaluando el estado del arte. Los conocimientos clave se extrajeron a través de un método de mapeo sistemático que involucró 103 estudios primarios; Durante este proceso también se definió un marco de clasificación. Este estudio contribuyó con un marco de clasificación para futuras investigaciones de arquitectura de microservicios, una evaluación del potencial para la adopción industrial de los resultados de la investigación y una discusión de los hallazgos emergentes. En consecuencia, este trabajo proporcionó una imagen clara, rigurosa y replicable del estado del arte con la arquitectura de microservicios [19]. Esta investigación proporciona información sobre la arquitectura de los microservicios. También proporciona detalles sobre la metodología específica utilizada para realizar el estudio. Tanto los profesionales como los investigadores se benefician de los resultados de esta investigación debido a su aplicabilidad en el campo.

Dentro de la segunda investigación seleccionada titulada “A Systematic Mapping Study of Software Development With GitHub”, su objetivo principal fue identificar el número, temas y métodos empíricos del trabajo de investigación destinado a analizar cómo las prácticas de desarrollo de software se ven influenciadas por las plataformas de codificación social distribuida como lo es GitHub. La metodología que los investigadores implementaron para el

desarrollo de este estudio fue la conocida mapeo sistemático la cual consistió en realizar cuatro preguntas de investigación en donde fue necesario evaluar 80 publicaciones entre 2009 y 2016. Entre los resultados presentados: Este trabajo demuestra un esfuerzo de investigación muy activo en el campo de la colaboración de código abierto, especialmente en el dominio del software, revelando una serie de fallas y proponiendo medidas para mitigarlas. Este documento también puede ser de suma ayuda sentar las bases en futuras investigaciones sobre otras actividades colaborativas (como la escritura de libros) que también se transfieren a GitHub [34]. Este estudio es relevante para el tema de investigación, ya que la información presentada será el punto de partida y referencia para seleccionar diferentes estrategias y herramientas para poder comprender con mayor facilidad el mapeo sistemático.

Otra de las investigaciones estudiadas en relación con el tema fue la propuesta por Muhammad Waseem et al. la cual titula “*A Systematic Mapping Study in Microservices Architecture in DevOps*”, tiene como objetivo identificar, analizar y clasificar sistemáticamente la literatura sobre MSA en DevOps. El método de estudio empleado en esta investigación comprende un estudio de mapeo sistemático (SMS) sobre una literatura publicada entre enero de 2009 y julio de 2018. Entre los resultados de la investigación: se informó la demografía, clasificación y mapeo de los temas de investigación, también fueron analizados los métodos de descripción de MSA, patrones y los controles de calidad. Los resultados y los hallazgos beneficiarán a investigadores y profesionales para realizar más investigaciones y brindar soluciones más específicas para los problemas de MSA en DevOps [6]. Esta investigación se relaciona con el tema de estudio debido a que proporciona información acerca de cómo debería realizarse el estudio de mapeo sistemático relacionado a la arquitectura de microservicios desde el punto de análisis de los DevOps.

Como cuarta investigación seleccionada titula “*Understanding and Addressing Quality Attributes of Microservices Architecture: A Systematic Literature Review*”, la cual tuvo como objetivo principal investigar el estado del arte basado en evidencia de QA de sistemas basados en microservicios. La metodología Revisión de Literatura Sistemática (SLR) es utilizada para identificar y sintetizar los estudios relevantes que reportan evidencia relacionada con QAs de MSA. Entre los resultados aportados identificaron 19 tácticas que abordan arquitectónicamente los controles de calidad críticos en MSA, incluidas dos tácticas para la escalabilidad, cuatro para el rendimiento, cuatro para la disponibilidad, cuatro para la monitorización, tres para la seguridad y dos para la prueba. Luego de analizar esta investigación se concluye que para los sistemas basados en MSA: 1) Aunque la escalabilidad es el beneficio comúnmente reconocido

de MSA, sigue siendo una preocupación indispensable entre los QA identificados, especialmente cuando se intercambian con otros QA, por ejemplo, el rendimiento. Aparte de los seis QA identificados en este estudio, otros QA para MSA como la mantenibilidad necesitan más atención para una mejora y evaluación efectivas en el futuro. 2) Los profesionales deben tomar cuidadosamente la decisión de migrar a MSA en función del retorno de la inversión, ya que este estilo arquitectónico también causa algunos problemas en la práctica [35]. Esta investigación se relaciona con el tema de estudio ya que proporciona información sobre los atributos de calidad de la arquitectura de microservicios utilizando para ello una revisión sistemática.

La quinta investigación seleccionada titula “*Securing microservices and microservice architectures: a systematic mapping study*” este estudio tuvo como objetivo principal proporcionar una guía útil a los desarrolladores sobre las amenazas que normalmente se presentan en los microservicios y cómo se pueden detectar, mitigar o prevenir. La metodología aplicada fue un mapeo sistemático el cual ayudó a categorizar las amenazas en MSA con sus propuestas de seguridad. Entre los resultados aportados: clasificación de mecanismos de seguridad de MSA, lista de factores que deben considerarse implementar arquitecturas de microservicios [36]. Esta investigación tiene una gran relación con el tema de estudio debido a que la información proporcionada servirá potencialmente de ayuda para el desarrollo de esta, además este estudio engloba 3 de las variables de estudio.

Otra de las investigaciones estudiadas en relación con el tema fue la propuesta por Cleber Santana, Breno Alencar y Cássio Prazeres. la cual titula “*Microservices: A Mapping Study for Internet of Things Solutions*”. Esta investigación se relaciona con el tema de estudio debido a que proporciona un enfoque sistemático a dos de las variables de estudio. En cuanto al principal objetivo de esta investigación, este fue presentar una visión general del estado del arte en relación con los Microservicios en el desarrollo de aplicaciones IoT mediante la metodología de estudio de mapeo. Entre los resultados: una lista de factores que deben considerarse cuidadosamente al implementar aplicaciones que utilizan la arquitectura de microservicios en campos como IoT y Mobile [37].

CAPÍTULO II: METODOLOGÍA

2.1 Delimitación

En esta sección se presentan los aspectos relacionados con la delimitación de la investigación propuesta, debido a la naturaleza del tema de investigación este no está restringido a un espacio físico en particular. En cuanto al tipo de estudio de mapeo sistemático se consideró relevantes revistas, conferencias, estudios de impacto de bases de datos científicas tales como: IEEE Xplore, Web of Science, ACM, Scopus, Google Scholar, Elsevier, Scielo y Springer. En cuanto a la delimitación temporal, se planteó que la investigación se encuentre en un periodo de máximo cinco años (2018 – 2022) debido a que se pretende abarcar la mayor cantidad de información sobre el tema de estudio, de esta manera también se podrá evitar caer en picos de moda o información desactualizada.

2.2 Tipos de investigación

Esta investigación es considerada explicativa ya que se llevó a cabo para ayudar a encontrar la solución a un problema que no se ha estudiado antes en profundidad. La investigación explicativa no se utiliza para dar pruebas concretas, sino para ayudar a entender el problema de forma más eficiente.

El tipo de estudio también se consideró como investigación documental debido a que este incluye en su desarrollo información de investigación que anteriormente fue publicada en informes de investigación y documentos con información similar. Estos documentos están disponibles en sitios web, bases de datos científicas y bibliotecas públicas.

2.3 Métodos

Para lograr los objetivos planteados de la investigación, fue necesario la implementación de varios métodos, entre ellos el método inductivo también conocido como el método de causa y efecto ya que luego de investigar un tema en específico se obtuvieron conclusiones. El método descriptivo también fue útil para el desarrollo del tema de investigación, debido a que es un tipo de método que describe el fenómeno o situación que está siendo estudiado, además este se centra en responder las preguntas de cómo, qué, cuándo y porqué. También se aplicó el método deductivo, como su nombre lo indica, ayudará a deducir una conclusión fundamentada y lógica de la investigación.

2.4 Técnicas

Referente a la técnica para la recolección de información fue necesario utilizar la conocida Mapping Study propuesta por los autores Petersen et. al [38], debido a que para el desarrollo de la investigación se recopiló gran cantidad de información de distintas fuentes bibliográficas.

2.5 Descripción de instrumentos

En el presente apartado se detallarán los instrumentos utilizados para el desarrollo de la investigación. Se desarrolló el análisis documental en donde se consideró artículos publicados en distintas bases de datos científicas tales como ACM, Springer, Elsevier, Scielo, Scopus, IEEE Xplore y Google Scholar, con una antigüedad máxima de cinco años. En cuanto a las cadenas de búsqueda que se utilizaron para la recopilación de información en las distintas bases de datos científicas:

- a) (Microservices) AND (tools) OR (plataforms)
- b) (TITTLE(*Systematic Mapping Study*) AND TITTLE (*Microservices*))
- c) (TITTLE(*Web Microservices*) AND TITTLE (*Microservices Architecture*) AND NOT TITTLE (*Monolithic System*))
- d) (Microservices) AND (tools) OR (Software)

2.6 Técnicas de procesamiento y análisis de datos

Referente a las técnicas que se emplearon para el procesamiento y análisis de datos fue necesario emplear el protocolo definido por la metodología que se utilizó en el SMS debido a que esta técnica utiliza metadatos de distintas publicaciones de carácter científico para la extracción de información entre estudios además, los papers de otros autores analizados con anterioridad, utilizan esta técnica debido a que son utilizadas frecuentemente en mapeos sistemáticos en el campo de la ingeniería de software.

Debido a que se recopiló gran cantidad de información otra técnica que fue de suma ayuda para el desarrollo de la investigación fue la implementación de criterios de inclusión y exclusión ya que estos ayudaron a establecer los límites del SMS y a su vez ayudaron a que la información recopilada sea de mayor fiabilidad y calidad.

La extracción de datos fue otra técnica aplicada al desarrollo de la investigación ya que permitió obtener información necesaria acerca de las variables de esta, lo que permitió abordar el tema con mayor facilidad, esta técnica es utilizada frecuentemente por investigadores que realizan investigaciones de tipo Systematic Mapping Study (SMS).

En cuanto a las herramientas utilizadas para esta técnica, fue de suma utilidad el emplear una búsqueda de nivel científico en bases de datos tales como: Scopus, IEEE Xplore, Google Scholar, ACM y Mendelay. Para el desarrollo de la búsqueda también fue necesario emplear cadenas de búsqueda para obtener mejores resultados.

2.7 Normas éticas

La presente investigación se mantendrá bajo las normas éticas establecidas en el Reglamento de Grados de la Pontificia Universidad Católica del Ecuador Sede Esmeraldas (PUCESE), esto con la finalidad de cumplir con las normas y leyes que establece la institución. También se respetó todos los derechos de autor en cuanto a los trabajos relacionados a este proyecto de investigación tomados de bases documentales como Scopus, Web of Science, IEEE Xplore y ACM, por lo cual fueron citados correctamente para evitar conflictos con los autores propios de dichos documentos, de esta manera será garantizado el respeto a los derechos de autor de las diferentes investigaciones y artículos abordados a lo largo de la investigación.

CAPÍTULO III: RESULTADOS

3.1 Descripción del estudio de mapeo sistemático

Se puede definir como revisión o mapeo sistemático al método de investigación el cual es utilizado para recopilar, identificar, comparar, evaluar y presentar evidencias o estudios más relevantes de un tema en específico. Para llevar a cabo este estudio fue necesario implementar las pautas propuestas por Petersen et al. [39] para realizar el SMS, y las estrategias presentadas por Kitchenham et al. [40] el cual fue implementado para el desarrollo de revisiones sistemáticas de literatura (SLR).

El motivo por el cual fue utilizado este tipo de método es porque luego de realizar la revisión sistemática de papers relacionados con la temática de investigación se obtuvo como resultado que el 34% (Petersen) y 36% (Kitchenham) de estos aplican dichas estrategias para el desarrollo de este tipo de investigación, tal y como se aprecia en la

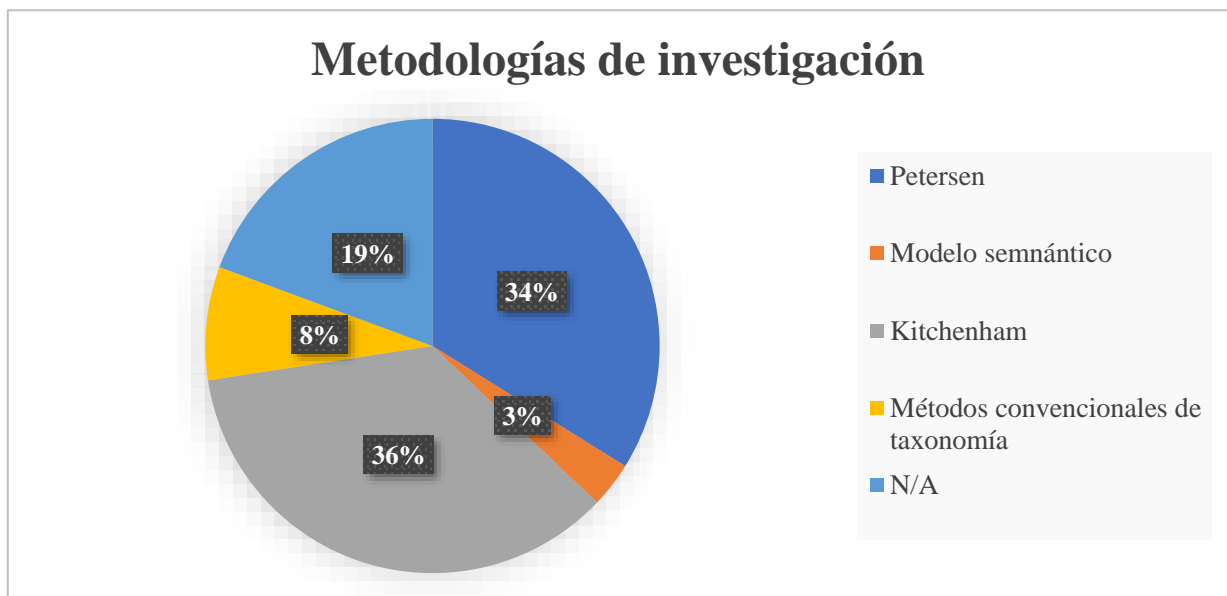


Ilustración 4 además, esta metodología es muy sencilla de aplicar y consta de cuatro etapas, la primera que fue donde se definió las preguntas de investigación luego, fue necesario realizar la búsqueda de estudios para sustentar la investigación después, se planteó palabras clave para en base a ello poder extraer categorías las cuales sirvieron para un mejor manejo de la información finalmente, se extrajeron los datos e información recopilada según la clasificación definida en un principio.

Metodologías de investigación

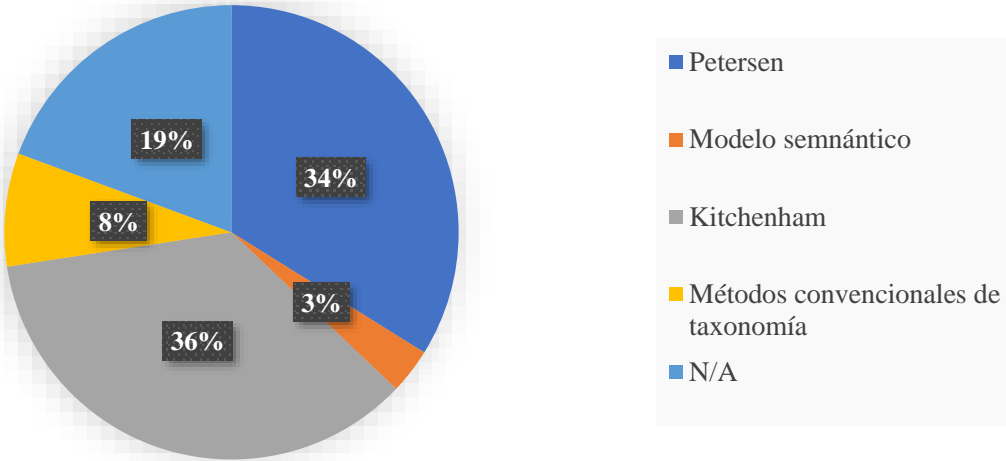


Ilustración 4 Metodologías empleadas en los papers extraídos

3.2 Preguntas de investigación

3.2.1 ¿Cuáles son las principales razones para utilizar microservicios?

A lo largo del desarrollo del documento, se han especificado las grandes ventajas que tiene la implementación de microservicios de forma más específica en el presente apartado de la investigación. Los microservicios funcionan independientemente unos de otros, pero permanecen poco conectados. A diferencia del estilo monolítico, el enfoque de microservicios para el desarrollo de software permite una mejor escalabilidad. Puede ampliar las aplicaciones en función de la demanda de los usuarios sin afectar a otros microservicios [41].

Las pequeñas empresas y las empresas de comercio electrónico se benefician específicamente de la arquitectura de microservicios. Esta arquitectura permite manejar diferentes tareas sin interferir entre sí. Por ejemplo, la facturación puede ser manejada por una sección de la arquitectura mientras que otra maneja otras funciones. La arquitectura de microservicios se basa en la nube y los desarrolladores pueden realizar un seguimiento de los cambios en la infraestructura de la aplicación en cualquier lugar del mundo [42].

Los microservicios utilizan un modelo de entrega constante para administrar todo el ciclo de vida de una aplicación. Cuando los desarrolladores y los equipos de prueba trabajan juntos en un solo servicio, el proceso de prueba se vuelve fácil e instantáneo. Este método permite a las organizaciones desarrollar y probar continuamente el código, así como utilizarlo de las bibliotecas existentes [43].

En relación con su funcionalidad concentrada, los microservicios permiten a las organizaciones optimizar la eficiencia del equipo. Los miembros del equipo pueden concentrarse completamente en servicios específicos según sus capacidades.

Esa es una de las principales razones detrás de la decisión de Netflix de pasarse a los microservicios. En su caso, 25 ingenieros enfrentaron problemas y dificultades para trabajar juntos en un mismo proyecto. Sin embargo, al migrar los microservicios, Netflix dividió un equipo en muchos pequeños equipos de ingeniería que ahora trabajan de manera independiente y eficiente [44].








3.2.2 ¿Cuáles son las herramientas más utilizadas en el área de lenguaje de programación, mensajería y monitoreo en el área de despliegue de microservicios?

Luego de realizar una exhaustiva recolección de papers referente a las herramientas más implementadas en las áreas mencionadas anteriormente, se obtuvo como resultado lo siguiente:

Lenguaje de programación

De acuerdo con investigación realizada por TIOBE [45] (empresa especializada en la valoración y monitoreo de la calidad del software), a julio de 2022, los lenguajes de programación con un mayor porcentaje de uso para el desarrollo de microservicios son:

Tabla 2 Lenguajes de programación más utilizados para el desarrollo de microservicios

Posición jul-22	Posición jul-21	Cambio	Logo	Lenguaje de Programación	Ratings	Cambio
1	2	↑		Python	15,42%	+3.56%
2	1	↓		C	14.59%	+2.03%
3	3			Java	12.40%	+1.96%
4	4			C++	10.17%	+2.81%
5	5			C#	5.59%	+0.45%
6	6	↑		JavaScript	2.33%	-0.61%
7	7	↑		SQL	1.70%	+0.23%
8	8	↓		PHP	1.39%	-0.80%
9	9	↑		Swift	1.27%	+0.30%
10	10	↑		Obtective-C	1.22%	+0.31%

Encabezando la lista de lenguajes más utilizados se encuentra Python, el cual cuenta con una votación del 15,42% además, este también cuenta con una mejora respecto a su votación ya que especifica que en Julio año pasado (2021) se encontraba en segundo puesto. En la Tabla 2 también se puede apreciar que ha presentado una mejora de su puntuación, para ser exactos 3.56%. Cabe mencionar que en el mismo estudio realizado por la compañía alega que este lenguaje ha tenido el premio como lenguaje del año en los años 2007, 2010, 2018, 2020, 2021.

Según el Índice TIOBE [45], Python es el lenguaje más utilizado en julio de 2022. Y hay una razón para ello. Python es un lenguaje versátil: es fácil de aprender, bueno para experimentar, permite a los desarrolladores escribir código sangrado limpio y, al ser un lenguaje interpretado, puede poner en marcha cambios rápidamente. Es por eso por lo que Python es presentado como el primero de los cinco lenguajes más populares para aprender.

Los desarrolladores usan Python para crear todo tipo de aplicaciones, desde scripts de sistema simples que ejecutan tareas en un servidor hasta aplicaciones orientadas a objetos grandes y complejas que brindan servicios a millones de usuarios.

En segundo lugar, se encuentra el lenguaje C el cual cuenta con una votación del 14,59% además, se detalla de igual manera que entre el presente y pasado año hubo menor elección respecto a este lenguaje ya que el anterior año se encontraba en primer puesto y ahora está en segundo puesto también, se puede apreciar que a pesar de contar con el segundo lugar ha tenido una votación positiva de 2.03%.

En cuanto al tercer lugar se encuentra Java con el 12.40%, este lenguaje en comparación a los dos anteriores se ha mantenido en su posición desde el año pasado además este presenta un cambio de 1.96%.

Según el Índice TIOBE [45], Java está dentro del top tres lenguajes más populares en julio de 2022, este ofrece muchos recursos y bibliotecas de programación. Además, es fácil encontrar desarrolladores de Java, y hay muchos proveedores de la nube que pueden escalar microservicios basados en Java.

Java es ideal para escribir microservicios. Entre otras razones, su sintaxis de anotaciones es fácil de leer. Las anotaciones de Java hacen que escribir microservicios sea mucho más fácil, sobre todo cuando se utiliza un marco de trabajo como Spring Boot. Hay mucho valor en la legibilidad, especialmente cuando se trata de trabajar en sistemas complejos [20].

Tabla 3 Numero de empresas a nivel mundial que utilizan los lenguajes de programación (Python, C, Java)

Número de empresas que lo utilizan	
Python	11,757
C	8,584
Java	26,269

Tal y como se muestre en la Tabla 3 existe una gran cantidad de empresas a nivel mundial que utilizan los lenguajes de programación C, Python y Java, siendo este último el más utilizado, con una cantidad de 26.269 de empresas en todo el mundo.

Monitoreo

Tabla 4 Herramientas de monitoreo

Nombre	Licencia	Versión más reciente	Fecha de publicación	Código abierto	Gran cantidad de documentación
Prometheus	Apache 2.0	9.0.7		Si	Si
Logstash	Apache 2.0	1.4.2	24 / 06 / 2021	Si	Si
Grafana	GNU	5.0.4	03 / 12 / 2018	Si	Si

Luego de realizar la respectiva revisión sistemática se obtuvieron tres herramientas empleadas en el monitoreo de microservicios web entre ellas:

Prometheus, referente a esta herramienta se puede acotar que la licencia que esta maneja es una Apache 2.0 lo que la hace una licencia de tipo permisiva creada por Apache Software Foundation, su versión más reciente lanzada es la 9.0.7 y cuenta con gran cantidad de documentación, además, es de código abierto.

Otra de las herramientas extraídas es Logstash, esta herramienta cuenta con una licencia de tipo Apache y su versión reciente más estable es la 1.4.2 lanzada el 24 de junio de 2014, es de código abierto y cuenta con gran cantidad de documentación.

Como tercera herramienta se encuentra Grafana, esta es una herramienta con una licencia tipo GNU y una versión de software 5.0.4. esta versión fue lanzada el 3 de diciembre de 2018 además cuenta con código abierto y gran cantidad de documentación.

Mensajería

Luego de realizar el análisis de cada uno de los papers extraídos, se obtuvo como resultado 3 herramientas principales para el monitoreo de microservicios a continuación, más detalles sobre estas.

Tabla 5 Herramientas extraídas para mensajería de microservicios web

Nombre	Licencia	Versión más reciente	Fecha de publicación	Código abierto	Gran cantidad de documentación
Apache Kafka	Licencia Apache 2.0	3.2.1	29-jul-22	Si	Si
RabbitMQ	Mozilla Public License	3.10.2	06-06-2022	Si	Si
Amazon MQ	Software Propietario	5.16.4	04-may-22	Si	Si

La primera herramienta extraída fue Apache Kafka, tal y como se menciona en la Tabla 5 esta conocida herramienta cuenta con una licencia de tipo Apache 2.0, la versión más reciente de esta herramienta es la 3.2.1 y fue lanzada el 29 de julio del presente año. Las herramientas de código abierto actualmente juegan un papel muy importante dentro del mundo del desarrollo y esta no es la excepción. Apache Kafka es una herramienta de código abierto además, esta posee una gran cantidad de documentación para la implementación de esta.

Como segunda herramienta RabbitMQ, esta posee una licencia de tipo Mozilla Public License. Referente a la versión más reciente de esta herramienta se puede acotar que fue lanzada el 6 de junio del 2022, la versión en cuestión es la 3.10.2. Esta herramienta al igual que la primera también es de código abierto y no está de más mencionar que en la web se puede encontrar mucha documentación referente a esta.

Finalmente, Amazon MQ, esta herramienta con una licencia de tipo Software propietario además de contar con una versión lanzada el 4 de mayo del 2022, la versión en cuestión es la 5.16.4. Esta herramienta es de código abierto y también cuenta con gran cantidad de información documental.

3.3 Definición de estrategia de búsqueda

La búsqueda realizada en esta revisión consideró trabajos disponibles en revistas, conferencias y estudios de impacto de bases de datos científicas sobre estudios que han abordado el tema de mapeo sistemático a herramientas de microservicios web. La estrategia de búsqueda y selección se divide en dos fases: Fase I: (Definición de fuentes de búsqueda, definición de

términos de búsqueda, selección de estudios y extracción de datos) Fase II: (Esta fase está relacionada con el proceso de Bola de Nieve o también conocida como Snowballing Process). Cabe recalcar que se identificó más estudios relevantes aplicando el procedimiento de bola de nieve.

3.4 Definición de fuentes de búsqueda

El primer paso de la estrategia de búsqueda fue identificar y definir las fuentes de búsqueda se emplearon para realizar la revisión sistemática. Para el desarrollo de la investigación se consideró relevante y necesario la extracción de información proveniente de revistas, conferencias, estudios de impacto de bases de datos científicas tales como: IEEE Xplore, Google Springer, Elsevier, Scielo, Scholar, ACM y Scopus, también fue necesario aplicar una delimitación temporal a la investigación para que de esta manera la información que se extraiga sea de un periodo de máximo cinco años (2017 – 2022) debido a que se pretende abarcar el mayor porcentaje información respecto al tema de estudio, de esta manera también se podrá evitar picos de moda o información desactualizada. Los detalles para el acceso a dichas fuentes de información se observan en la Tabla 6.

Tabla 6 Bases de datos científicas empleadas para la recolección de información

Nombre de fuente de información	Dirección URL
SCOPUS	https://www.scopus.com/
IEEE XPLORE	https://www.ieeexplore.ieee.org/
GOOGLE SCHOLAR	https://scholar.google.com/
ACM	https://dl.acm.org/
SPRINGER	https://www.springer.com/la
SCIELO	https://scielo.org/es/

3.5 Estrategia de búsqueda

Se formularon varias cadenas de búsqueda las cuales se emplearon para poder obtener información y responder a las preguntas de investigación anteriormente planteadas, además, este proceso fue de suma ayuda para a responder las preguntas formuladas para el desarrollo de la investigación.

3.6 Criterio de inclusión y exclusión

Referente al proceso de recolección de información se aplicó sobre las revistas, conferencias, estudios de impacto de bases de datos mencionadas anteriormente. Cabe mencionar que luego de aplicar el proceso de recolección se clasificó la información según el título, tipo de investigación, año, tipo de publicación, metodología implementada, resultados obtenidos y

enfoque. En la Tabla 7 muestra cada uno de los resultados obtenidos en las distintas bases de datos.

Tabla 7 Estudios extraídos y orientados al desarrollo de la investigación

	Artículos encontrados en las diferentes Bases de Datos Científicas	Artículos seleccionados
ELSEVIER	180	13
IEEE XPLORE	250	33
GOOGLE SCHOLAR	175	9
ACM	94	7
SPRINGER	142	6
SCIELO	75	6
UNIVERSIDAD DISTRITAL DE CALDAS	109	1

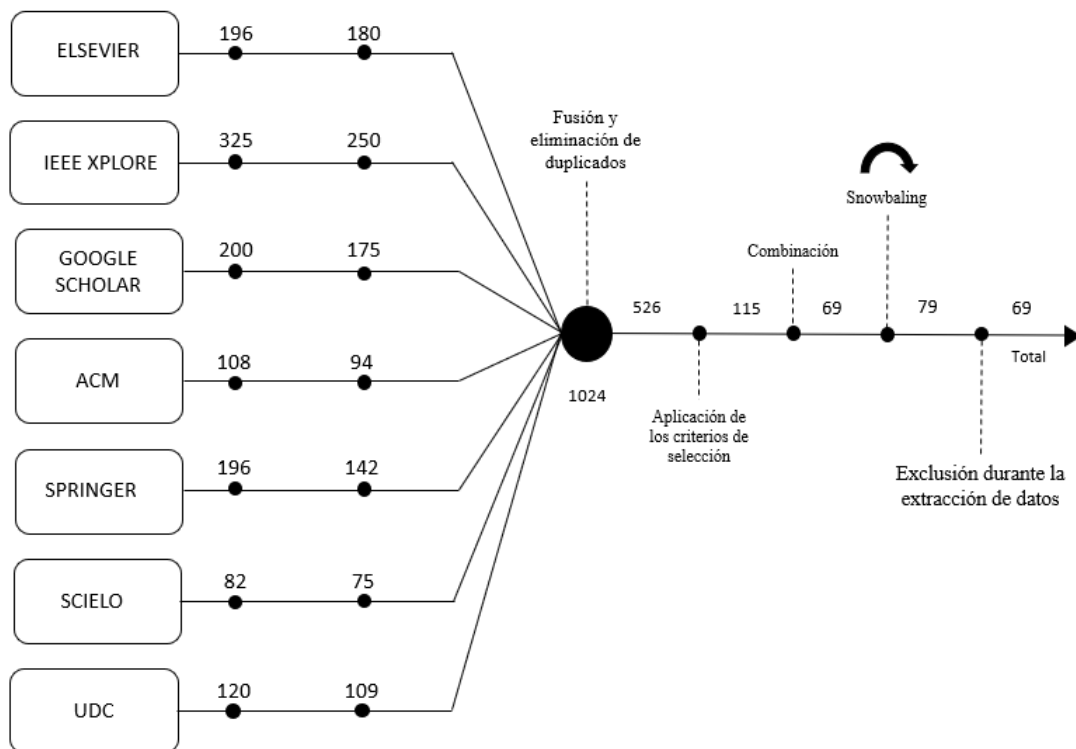


Ilustración 5 Proceso de refinamiento de información extraída

Luego de realizar el respectivo tratamiento a la información recopilada, se procedió a clasificar nuevamente la información debido a que la cantidad de información abarcada fue bastante extensa, teniendo como resultado 1025 papers entre todas las diferentes bases de datos, tal y como se muestra en la Ilustración 5. A continuación, se describen los papers utilizados para el desarrollo de la investigación. Como se muestra en Tabla 8, los papers fueron clasificados según el id. tema, tipo, país, base de datos y año.

Tabla 8 Resultado de los papers seleccionados para el desarrollo de la investigación

NUM	TEMA	TIPO	AÑO	PAÍS	BASE DE DATOS
1	Detecting Malicious Behavior in Microservice Based Web Applications	Conferencia	2019	Turquía	IEEE
2	Visualization Tool for Designing Microservices with the Monolith Approach	Artículo	2018	España	IEEE
3	Efficient Resources Utilization by Different Microservices Deployment Models	Artículo	2018	USA	IEEE
4	A Declarative Approach For Updating Distributed Microservices	Poster	2018	Suecia	ACM
5	Guidelines for performing Systematic Literature Reviews in Software Engineering	Artículo	2007	Reino Unido	Google Scholar
6	Review Of Methods For Migrating Software systems to microservices architecture	Revista	2021	Serbia	ACM
	A systematic Literature Review on Microservices	Conferencia	2018	Turquía	Springer
8	A systematic mapping Study of infrastructure as code research	Artículo	2019	USA	Elsevier
9	A Systematic Mapping Study of Software Development with Github	Artículo	2018	España	IEEE
10	Kuber: Cost-Efficient Microservice Deployment Planner	Conferencia	2020	USA	IEEE
11	A Systematic Mapping Study on Mining Software	Conferencia	2018	China	ACM
12	A progressive Web Applications Based on Microservices Combining Geospatial Data and the Internet of Things	Artículo	2019	España	IEEE
13	A systematic mapping study on microservices Architecture in DevOps	Artículo	2020	USA	Elsevier
14	A comparative Study of Meta-Data-Based Microservice Extraction Tools	Artículo	2022	USA	Elsevier
15	Approaches to Cross-Domain Sentiment Analysis: A Systematic Literature Review	Artículo	2018	Malasia	IEEE
16	Architecting with microservices: A systematic mapping Study	Revista	2019	Londres	Elsevier
17	Arquitectura basada en micro-servicios para aplicaciones web	Artículo	2019	Colombia	Universidad Distrital Francisco José de Caldas
18	Arquitectura de microservicios como estrategia para el desarrollo de aplicaciones web en la UCLV	Libro	2019	Cuba	Google Scholar
19	Analysis and Design of Microservices: Results from Turkey	Conferencia	2020	Turquía	IEEE
20	Challenges of microservices Architecture: A survey on the State of the Practice	Artículo	2018	Alemania	IEEE

21	Comparison of Runtime Testing Tools for Microservices	Conferencia	2019	USA	IEEE
22	The process of Data Mapping for Data Integration Projects Data Mapping-A Key Work Product for Data Warehouse, Data Integration, and Data Migration Projects	Artículo	2019	USA	IEEE
23	Estudio del Framework Spring, Spring Boot y Microservicios	Artículo	2020	España	Google Scholar
24	Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report	Conferencia	2018	USA	IEEE
25	Performance evaluation in the migration process from a monolithic application to microservices	Conferencia	2018	España	IEEE
26	Using Dependency graph and graph theory concepts to identify anti-patterns in a microservices system: A tool-based approach	Artículo	2021	Colombo	IEEE
27	Lihombga - A Microservice-Based Virtual Learning Environment	Conferencia	2018	India	IEEE
28	Mapeo sistemático y evaluación de arquitecturas de software para contextos de big data	Artículo	2018	Argentina	Scielo
29	Extraction of Microservices from Monolithic Software Architectures	Conferencia	2018	USA	IEEE
30	Microservices: A systematic Mapping Study	Conferencia	2020	Italia	ACM
31	Microservice Testing Approaches: A systematic Literature Review	Artículo	2019	Malasia	Google Scholar
32	Microservices: A Mapping Study for Internet of Things Solutions	Artículo	2018	USA	IEEE
33	Microservices in Practice: A Survey Study	Conferencia	2018	Brazil	IEEE
34	Migración de un sistema monolítico a una arquitectura de microservicios	Tesis de maestría	2019	Chile	Google Scholar
35	Microservices: A perfect SOA based solution for Enterprise Applications compared to Web Services	Conferencia	2018	India	IEEE
36	The Pains and Gains of Microservicios: A Systematic Grey Literature Review	Artículo	2018	Italia	Elsevier
37	Understanding and Addressing Quality Attributes of Microservices Architecture: A Systematic Literature Review	Artículo	2021	China	Elsevier
38	Design and Research of Microservice Application Automation Testing Framework	Conferencia	2019	China	IEEE
39	An Microservices-Based Openstack Monitoring Tool	Artículo	2019	China	IEEE
40	ViewBovine: A Microservices-Powered Web Application to Support Interactive Investigation of Bovine Tuberculosis Infection Pathways	Conferencia	2020	China	IEEE

41	Modelo de composición de Microservicios para la implementación de una aplicación web de comercio electrónico utilizando Kubernetes	Tesis de maestría	2018	Peru	Google Scholar
42	A platform-independent communication framework for the simplified development of shop-floor applications as microservice components	Artículo	2019	Taiwan	IEEE
43	Detecting Anti-Patterns in a MSA using Distributed Traicing	Tesis de maestría	2019	Paises Bajos	Springer
44	Security in Microservices Architectures	Artículo	2021	Portugal	Elsevier
45	Comparison of Runtime Testing Tools for Microservices	Artículo	2019	USA	IEEE
46	A Semantic Model for Interchangeable Microservices in Cloud Continuum Computing	Artículo	2021	Grecia	Elsevier
47	Microservices Monitoring with Event Logs and Black Box Execution Tracing	Conferencia	2022	USA	IEEE
48	Information Security Practice and Experience	Conferencia	2019	Malasia	Springer
49	Securing microservices and microservice architectures: A systematic mapping study	Artículo	2021	Argelia	Elsevier
50	Microservice transition and its granularity problem: A systematic mapping study	Artículo	2020	Inglaterra	ACM
51	Kubernetes Cluster for Automating Software Production Environment	Artículo	2021	Polonia	Elsevier
52	Elastic Scheduling for Microservice Applications in Clouds	Artículo	2021	China	IEEE
53	A Microservices-Based Framework for Smart Design and Optimization of PV Installations	Conferencia	2021	Italia	IEEE
54	Design of Modern Distributed Systems based on Microservices Architecture	Artículo	2021	Kosovo	ACM
55	Microservices Backlog—A Genetic Programming Technique for Identification and Evaluation of Microservices From User Stories	Artículo	2021	USA	IEEE
56	A microservice architecture for real-time IoT data processing: A reusable web of things approach for smart ports	Artículo	2022	España	Elsevier
57	A Model-Driven Architecture for Automated Deployment of Microservices	Artículo	2021	España	Elsevier
58	Intelligent Autoscaling of Microservices in the Cloud for Real-Time Applications	Artículo	2021	India	IEEE
59	Automated identification of security discussions in microservices systems: Industrial surveys and experiments	Artículo	2021	China	Elsevier
60	Microservices: Migration of a Mission Critical System	Artículo	2021	USA	IEEE

61	A microservices persistence technique for cloud-based online social data analysis	Artículo	2021	Reino Unido	Springer
62	Human microservices: A framework for turning humans into service providers	Artículo	2021	España	Google Scholar
63	Microservice-Based Approach to Simulating Environmentally Friendly Equipment of Infrastructure Objects Taking into Account Meteorological Data	Artículo	2021	N/E	ACM
64	Deployment and verification of machine learning tool-chain based on kubernetes distributed clusters	Artículo	2021	China	Springer
65	Research on online teaching platform system based on microservice architecture	Artículo	2022	China	Springer
66	Systematic Approach for Generation of Feasible Deployment Alternatives for Microservices	Artículo	2021	Turquía	IEEE
67	Microservices Backlog—A Genetic Programming Technique for Identification and Evaluation of Microservices From User Stories	Artículo	2021	USA	IEEE
68	State of Microservices	A. Revista	2020	N/E	N/A
69	Automatic performance monitoring and regression testing during the transition from monolith to microservices	Artículo	2019	Alemania	IEEE

A continuación, se describen los resultados de los papers seleccionados. Como se muestra en la Ilustración 6. IEEE encabeza la lista de bases de datos en donde se extrajo información (33 papers), en un segundo lugar Elsevier (13), tercer lugar Google Scholar (9), cuarto lugar ACM (7), quinto lugar Springer (7), para finalizar Scielo y Universidad Distrital de Caldas con un solo documento extraído respectivamente (1).

Ilustración 6 Cantidad de papers extraídos/seleccionados según la base de datos

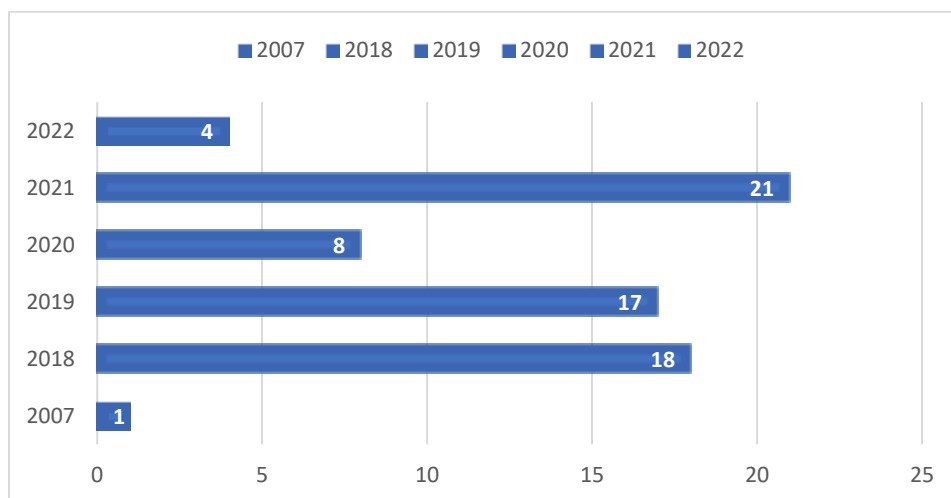
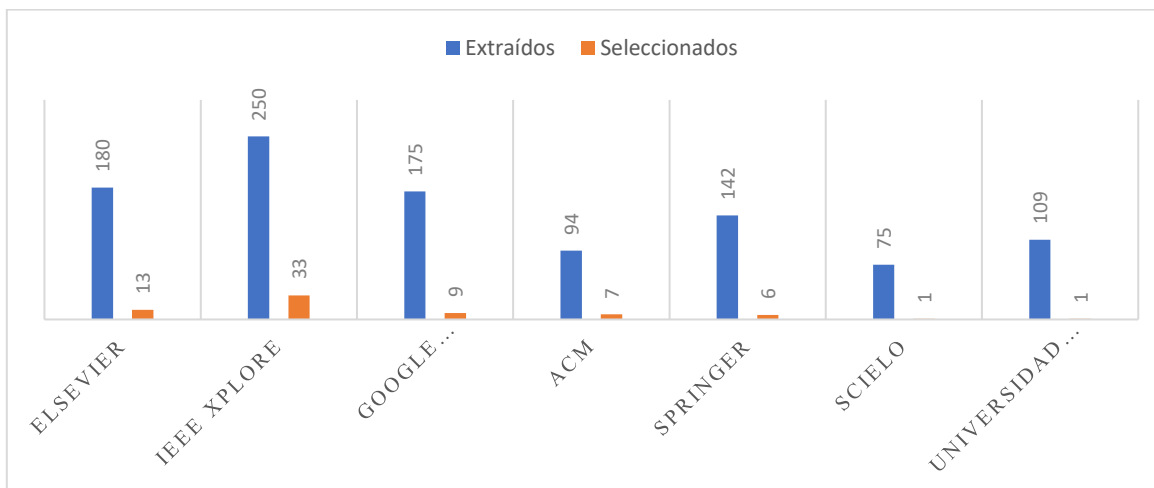


Ilustración 7 Porcentaje de estudios extraídos por año

Como se puede apreciar en la Ilustración 7 la mayor cantidad de los artículos fueron extraídos del año 20221, de este año se extrajeron 21 documentos lo que significa un 30% del porcentaje total. En cuanto al segundo año con mayor cantidad de artículos extraídos está el año 2018, en el cual se recopiló la cantidad de 18 artículos que corresponde al 26% del porcentaje total. Siguiendo con el año de 2019 en donde se extrajo 17 artículos los cuales

corresponden al 25% del porcentaje. Del año 2020 únicamente se extrajeron ocho artículos los cuales corresponden al 12% y correspondiente al año 2022 se extrajeron cuatro artículos que significan el 6%, finalmente en 2007 se obtuvo un único documento el cual fue utilizado como guía para la metodología de la investigación.

La Ilustración 8 muestra la distribución de los estudios seleccionados según el año en que fueron publicados. En esta ilustración se puede apreciar que el país de donde se extrajo mayor cantidad de documentos es Estados Unidos con 15 documentos (21.74%) luego, en segundo lugar, se encuentra China con nueve documentos extraídos (13.04%), en cuarto lugar, está España país del que se extrajo ocho documentos (11.59%) y también Malasia, de donde se pudo extraer únicamente tres documentos (4.35%). A pesar de estos países ocupan los primeros puestos, la documentación acerca de microservicios y temáticas relacionadas con este término es extensa y se puede encontrar en muchos países del mundo.

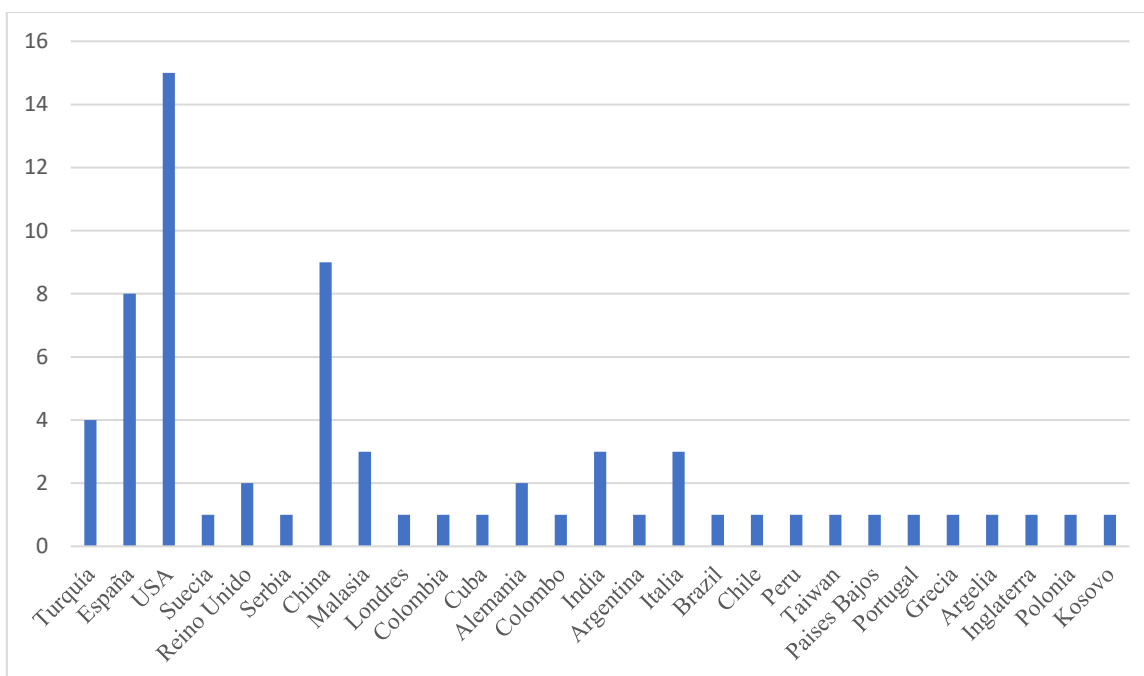


Ilustración 8 Número de artículos seleccionados según el país

En cuanto a los tipos de documentos seleccionados para el desarrollo de la investigación, se clasificó los documentos según su tipo, teniendo como resultado siete categorías: conferencias, artículos, poster, revistas, libros, tesis de maestría y artículos de revista. Como se puede apreciar en la Ilustración 9 el mayor tipo de documento seleccionado fue artículo con 43 documentos seleccionados (62%), en segundo lugar, están los documentos de tipo

conferencia con 18 documentos (26%) y tercer lugar los de tipo tesis de maestría con tres documentos (3%).

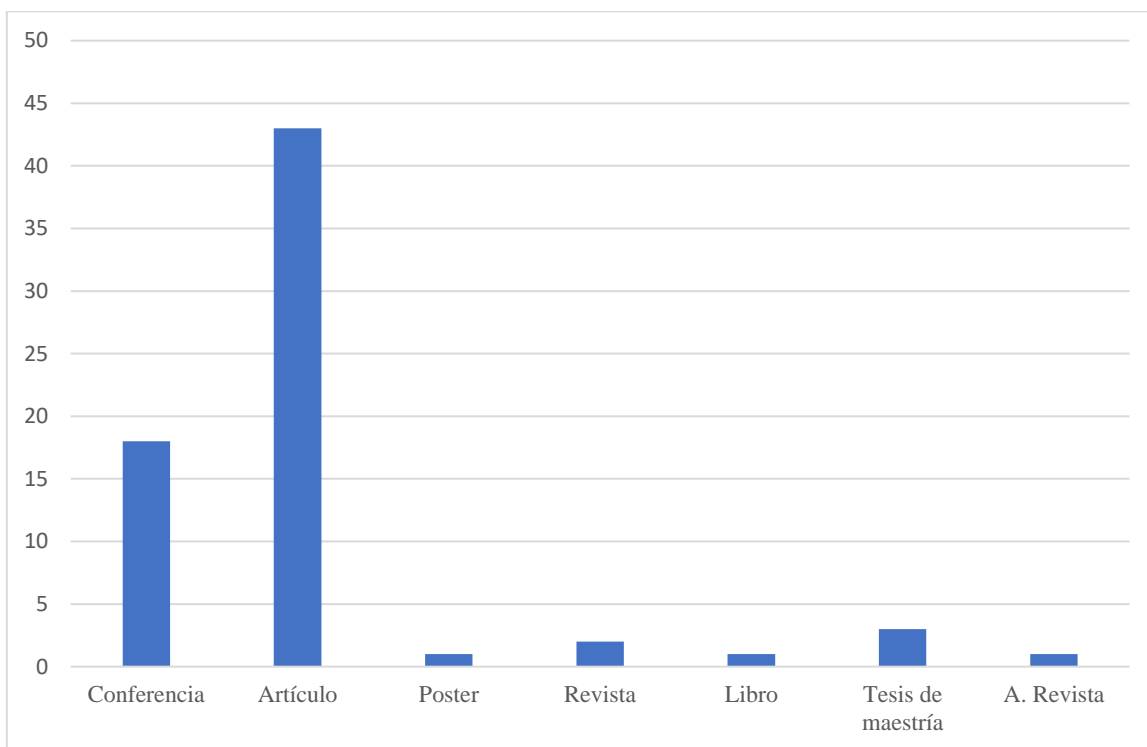


Ilustración 9 Clasificación de los papers seleccionados según el tipo de documento

De los 69 papers seleccionados se obtuvo como resultado que 24 de ellos han sido publicados en el continente europeo lo que en porcentaje corresponde al 34% de los artículos seleccionados, siguiendo con esta explicación en segundo lugar se encuentran los continentes asiático y americano con 21 artículos seleccionados por cada uno de ellos arrojando como resultado un porcentaje equivalente al 30% respectivamente. En tercer lugar, se encuentra el continente africano de donde se seleccionó un único documento y el cual representa un 2%, para finalizar se encuentra la sección de N/E la cual arrojó dos papers que representan el 3% de los papers seleccionados.

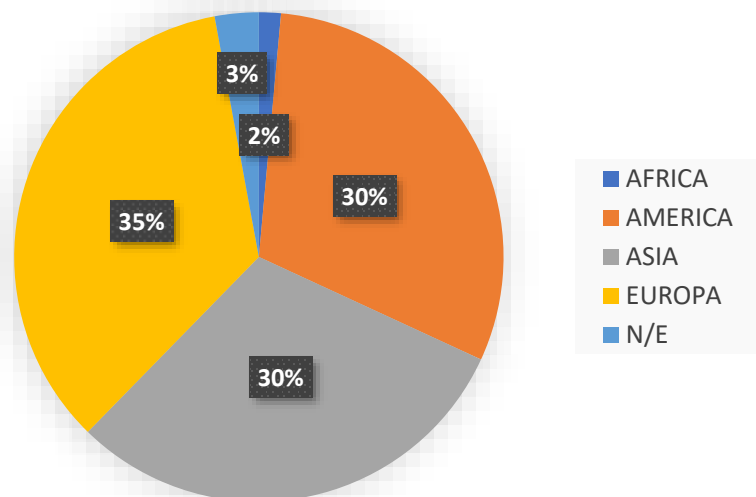


Ilustración 10 Porcentaje de artículos seleccionados según el continente

En la Ilustración 11 se puede apreciar mediante un gráfico de metodología bubble un breve resumen de unas de las incógnitas que surgieron al momento de plantear los resultados de la investigación: ¿Cuál fue la categoría de papers que más fue seleccionada?, ¿Cuál fue la categoría de herramientas que más fue utilizada?, ¿Cuál fue la categoría de papers más seleccionados para el desarrollo de la investigación? Entre las respuestas a estas incógnitas: categoría y clasificación de papers seleccionados además de categoría de herramientas encontradas.

Se puede apreciar que en la Ilustración 11 un mayor porcentaje de los artículos seleccionados con la temática de Mapping Analysis es la que se seleccionó, de la misma manera ocurrió con los papers relacionados con la categoría de Cloud y Messaging.

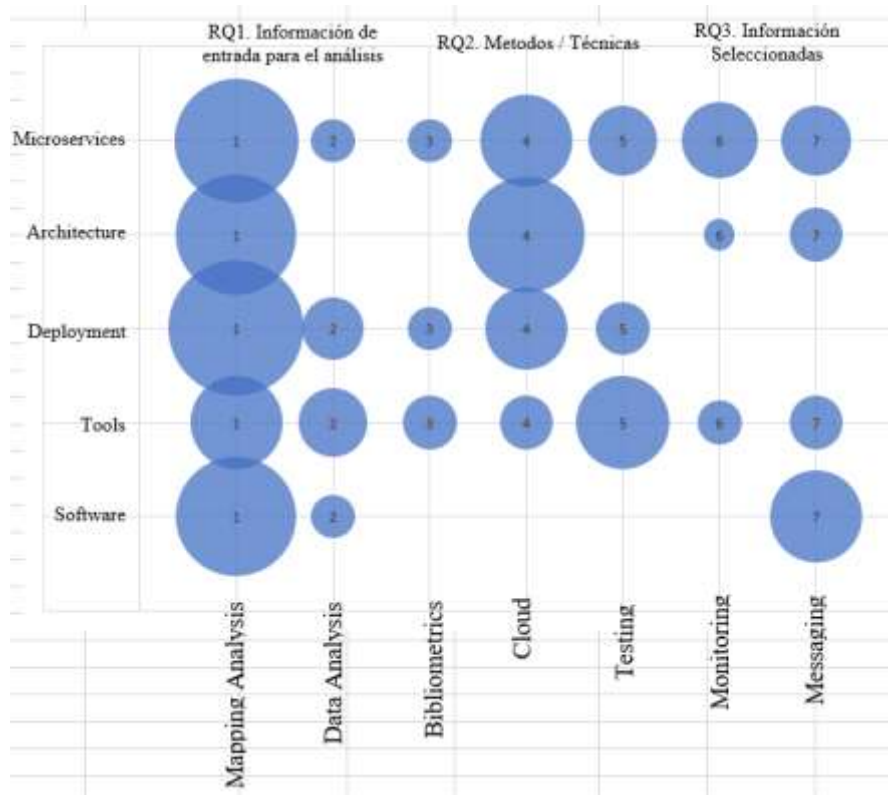


Ilustración 11 Gráfico metodología bubble

CAPÍTULO IV: DISCUSIÓN

P. Di Francesco, P. Lago, and I. Malavolta en su investigación proporcionaron un enfoque sólido, riguroso y replicable del estado del arte en arquitectura con microservicios. Sus resultados contribuyeron en la extracción de información además de presentar el enfoque de Petersen el cual sirvió como base del documento [19]. Esta investigación se relacionó con el tema a estudiar debido a que proporciona información de la arquitectura de los microservicios y también proporciona información muy precisa acerca del tipo de metodología que se utilizó para realizar la investigación.

De modo similar [34] proporcionó un punto de partida y de referencia para seleccionar las distintas estrategias y herramientas para obtener mayor entendimiento del mapeo sistemático también, este documento fue útil para la extracción de varias herramientas, de esta manera no solo contribuyó para la selección del estrategias sino también a herramientas que se describen en el documento desde el punto de vista del software.

Muhammad Waseem et al. presentaron una investigación que contribuyó en puntos clave de la investigación como, por ejemplo, identificar, analizar y clasificar sistemáticamente la literatura sobre MSA en el mundo de DevOps [6]. Esta investigación se relaciona con el tema de estudio debido a que proporcionó información acerca de cómo debe realizarse el estudio de mapeo sistemático relacionado a la arquitectura de microservicios desde el punto de análisis de los DevOps.

La metodología Revisión de Literatura Sistemática (SLR) presentada en [35] fue utilizada para identificar y sintetizar los estudios relevantes que reportan evidencia relacionada con QAs de MSA. Esta afirmación se ha podido verificar en base a los resultados debido a que en la investigación se realizaron 19 estudios de técnicas aplicadas en los distintos casos de SLR y gracias a esto se pudo continuar con el desarrollo de la investigación ya que de los 19 estudios fueron seleccionados dos para su posterior implementación dentro de la investigación.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Seleccionar la herramienta adecuada lleva tiempo porque se deben considerar muchos factores, como la funcionalidad, licencia, beneficios y los desafíos. Siempre se recomienda comprender primero los desafíos de la pila tecnológica existente y luego buscar una herramienta adecuada que se adapte a los requisitos del equipo al sopesar los beneficios/desafíos. Al final, todo se reduce a las necesidades y objetivos del equipo.

En base al primer objetivo del proyecto de investigación se concluye que, a partir de la extracción de papers relacionados con el tema de esta investigación el método a utilizar para el desarrollo del Systematic Mapping Study fue el inductivo o también llamado método de causa y efecto, además, también fue necesaria la implementación del método descriptivo, como su propio nombre lo indica este método ayuda a la descripción de fenómenos o situaciones que están siendo parte de un tema de estudio. El método descriptivo fue de suma ayuda ya que contribuyó a responder preguntas tales como: qué, cuando, cómo, porqué. Finalmente, se implementó el método deductivo, fue de ayuda para presentar una conclusión fundamentada y lógica.

Respecto al segundo objetivo se puede concluir que, las herramientas que se obtuvieron a partir de la investigación corresponden a los puntos más cruciales dentro del despliegue de microservicios debido a que estas herramientas fueron clasificadas según el lenguaje de programación, mensajería y monitoreo a su vez se clasificó si disponían de gran cantidad de documentación, licenciamiento, versión más reciente y año de la publicación de esta versión. Al ser una investigación de carácter científico y a pesar de haber recopilado gran cantidad de información no se pudo obtener una cantidad de herramientas.

Para finalizar con el tercer objetivo se puede concluir que existe gran cantidad de información relevante para el desarrollo del tema de investigación pero, exactamente ese es uno de los inconvenientes que se presentaron para el análisis de la misma, debido a que se extrajo gran cantidad de información, resultó un poco tedioso el análisis de esta pero aplicando cadenas de búsqueda, criterios de exclusión/inclusión y otras técnicas detalladas en el documento de investigación se pudo tener mejor control de la información que

finalmente fue utilizada para el desarrollo de la investigación. Los resultados de este SMS beneficiarán a los investigadores que estén interesados en comprender el estado de la investigación de MSA en DevOps

Recomendaciones

Aplicar el patrón de arquitectura de microservicios puede ofrecer un valor significativo si se hace un uso óptimo de las API RESTful. Las API RESTful ofrecen numerosas ventajas, por ejemplo, no se necesita instalar software en el lado del cliente. No necesita SDK ni marcos, ya que las solicitudes http/https son suficientes para consumir el servicio puerta de enlace de las API.

La mayoría de las aplicaciones basadas en microservicios se implementan cuando las organizaciones modernizan los sistemas monolíticos. Por lo tanto, la fase de diseño es una oportunidad ideal para mejorar la seguridad de las aplicaciones heredadas. Los equipos de desarrollo y seguridad deben hacer de la seguridad de las aplicaciones una prioridad al diseñar una arquitectura de microservicio para que la aplicación tenga la base adecuada para permanecer segura a medida que crece y evoluciona.

Uno de los beneficios clave de los microservicios es que los equipos de desarrollo pueden implementar con mayor facilidad componentes de software individuales sin volver a implementar toda la aplicación. Si bien esto permite que los equipos de DevOps creen aplicaciones más rápidamente, también podría presentar riesgos de seguridad. Es por eso por lo que es crucial un cambio de mentalidad hacia las pruebas de seguridad durante el proceso de desarrollo.

Es por eso por lo que un enfoque de DevSecOps, donde las medidas de seguridad se integran directamente en el proceso de desarrollo, en el entorno de integración continua y de construcción, es crucial para reducir el riesgo. DevSecOps requiere automatización y crea una responsabilidad compartida para entregar software seguro sin comprometer la velocidad de desarrollo. De hecho, el enfoque DevSecOps puede incluso reducir el costo y el tiempo necesarios para la entrega segura de software porque los desarrolladores ya no necesitarán actualizar los controles de seguridad después de la implementación.

REFERENCIAS BIBLIOGRÁFICAS

- [1] J. P. Russo, “Mapeo sistemático y evaluación de arquitecturas de software para contextos de big data,” 2018, [Online]. Available: <https://dspace.ort.edu.uy/handle/20.500.11968/3876>.
- [2] G. Campeanu, “A mapping study on microservice architectures of Internet of Things and cloud computing solutions,” *2018 7th Mediterr. Conf. Embed. Comput. MECO 2018 - Incl. ECYPS 2018, Proc.*, no. June, pp. 1–4, 2018, doi: 10.1109/MECO.2018.8406008.
- [3] J. Thönes, “Microservices,” *IEEE Softw.*, vol. 32, no. 1, 2015, doi: 10.1109/MS.2015.11.
- [4] S. Bhardwaj, L. Jain, and S. Jain, “Cloud Computing : A Study of Infrastructure As a Service (IaaS),” *Int. J. Eng.*, vol. 2, no. 1, pp. 60–63, 2020, [Online]. Available: http://ijeit.org/index_files/vol2no1/CLOUD COMPUTING A STUDY OF.pdf.
- [5] Á. I. Rodríguez, J. I. Padilla, H. A. Parra, and P. Rodríguez, “Arquitectura basada en micro-servicios para aplicaciones web Microservices,” 2019.
- [6] M. Waseem, P. Liang, and M. Shahin, “A Systematic Mapping Study on Microservices Architecture in DevOps,” *J. Syst. Softw.*, vol. 170, no. August, 2020, doi: 10.1016/j.jss.2020.110798.
- [7] D. Boukhelef, J. Boukhobza, K. Boukhalfa, H. Ouarnoughi, and L. Lemarchand, “Optimizing the cost of DBaaS object placement in hybrid storage systems,” *Futur. Gener. Comput. Syst.*, vol. 93, pp. 176–187, 2019, doi: 10.1016/j.future.2018.10.030.
- [8] S. Laso, J. Berrocal, J. García-Alonso, C. Canal, and J. Manuel Murillo, “Human microservices: A framework for turning humans into service providers,” *Softw. - Pract. Exp.*, vol. 51, no. 9, pp. 1910–1935, 2021, doi: 10.1002/spe.2976.
- [9] L. L. Pérez, *Arquitectura de microservicios como estrategia para el desarrollo de aplicaciones web en la UCLV*. 2019.
- [10] I. K. Aksakalli, T. Celik, A. B. Can, and B. Tekinerdogan, “A model-driven architecture for automated deployment of microservices,” *Appl. Sci.*, vol. 11, no. 20, 2021, doi: 10.3390/app11209617.
- [11] D. A. Ruelas, “Modelo de composición de microservicios para la implementación de una aplicación web de comercio electrónico utilizando kubernetes,” *Univ. Nac. del*

Altiplano, p. 122, 2018, [Online]. Available: <http://repositorio.unap.edu.pe/handle/UNAP/6781>.

- [12] A. Stojkov and Z. Stojanov, "Review of methods for migrating software systems to microservices architecture," *J. Eng. Manag. Compet.*, vol. 11, no. 2, pp. 152–162, 2021, doi: 10.5937/jemc2102152s.
- [13] "Old Workhorse for New Tech - Mainframe In The Age Of Cloud, AI, And Blockchain: A commissioned study conducted by Forrester Consulting | Ensono." <https://www.ensonocom/resources/white-papers/old-workhorse-new-tech-mainframe-age-cloud-ai-and-blockchain-commissioned-study-conducted/> (accessed Aug. 14, 2022).
- [14] S. Hassan, R. Bahsoon, and R. Kazman, "Microservice transition and its granularity problem: A systematic mapping study," *Softw. - Pract. Exp.*, vol. 50, no. 9, pp. 1651–1681, 2020, doi: 10.1002/spe.2869.
- [15] N. A. and R. E. Nuha Alshuqayran, "A systematic mapping study in microservice architecture," *Proc. ACM Symp. Appl. Comput.*, pp. 44–51, 2018, doi: 10.1109/SOCA.2016.15.
- [16] G. Mazlami, J. Cito, and P. Leitner, "Extraction of Microservices from Monolithic Software Architectures," *Proc. ACM Symp. Appl. Comput.*, pp. 524–531, 2018, doi: 10.1109/ICWS.2017.61.
- [17] J. P. Sotomayor, S. C. Allala, P. Alt, J. Phillips, T. M. King, and P. J. Clarke, "Comparison of runtime testing tools for microservices," *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 2, pp. 356–361, 2019, doi: 10.1109/COMPSAC.2019.10232.
- [18] A. Poniszewska-Marańda and E. Czechowska, "Kubernetes cluster for automating software production environment," *Sensors*, vol. 21, no. 5, pp. 1–24, 2021, doi: 10.3390/s21051910.
- [19] P. Di Francesco, P. Lago, and I. Malavolta, "Architecting with microservices: A systematic mapping study," *J. Syst. Softw.*, vol. 150, pp. 77–97, 2019, doi: 10.1016/j.jss.2019.01.001.
- [20] D. Guamán, Lady Yaguachi, S. Cueva C, D. Jaramillo H, and F. Soto, "Performance evaluation in the migration process from a monolithic application to microservices," *Cist. 2018 - 13th Iber. Conf. Inf. Syst. Technol.*, 2018.

- [21] D. P. Bernal, “Impacto de arquitecturas de microservicios en el desarrollo Web,” 2019.
- [22] B. Bilgin, “Analysis and Design of Microservices : Results from Turkey,” 2020.
- [23] H. Cai, C. Wang, and X. Zhou, “Deployment and verification of machine learning tool-chain based on kubernetes distributed clusters,” *CCF Trans. High Perform. Comput.*, vol. 3, no. 2, pp. 157–170, 2021, doi: 10.1007/s42514-021-00065-w.
- [24] F. Tapia, M. ángel Mora, W. Fuertes, H. Aules, E. Flores, and T. Toulkeridis, “From monolithic systems to microservices: A comparative study of performance,” *Appl. Sci.*, vol. 10, no. 17, 2020, doi: 10.3390/app10175797.
- [25] M. Yang and M. Huang, “An microservices-based openstack monitoring tool,” *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS*, vol. 2019-Octob, pp. 706–709, 2019, doi: 10.1109/ICSESS47205.2019.9040740.
- [26] I. Ghani, W. M. N. Wan-Kadir, A. Mustafa, and M. I. Babir, “Microservice testing approaches: A systematic literature review,” *Int. J. Integr. Eng.*, vol. 11, no. 8, pp. 65–80, 2019, doi: 10.30880/ijie.2019.11.08.008.
- [27] M. Cinque, R. Della Corte, and A. Pecchia, “Microservices Monitoring with Event Logs and Black Box Execution Tracing,” *IEEE Trans. Serv. Comput.*, vol. 15, no. 1, pp. 294–307, 2022, doi: 10.1109/TSC.2019.2940009.
- [28] C. Y. Fan and S. P. Ma, “Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report,” *Proc. - 2017 IEEE 6th Int. Conf. AI Mob. Serv. AIMS 2017*, pp. 109–112, 2018, doi: 10.1109/AIMS.2017.23.
- [29] W. Hasselbring and G. Steinacker, “Microservice architectures for scalability, agility and reliability in e-commerce,” *Proc. - 2017 IEEE Int. Conf. Softw. Archit. Work. ICSAW 2017 Side Track Proc.*, pp. 243–246, 2017, doi: 10.1109/ICSAW.2017.11.
- [30] “Docker - ArchWiki.” <https://wiki.archlinux.org/index.php/Docker> (accessed Nov. 18, 2020).
- [31] “The Kubernetes Bundle | Microsoft Azure.” <https://azure.microsoft.com/en-us/resources/kubernetes-ebook-collection/> (accessed Feb. 25, 2021).
- [32] A. Pereira Ferreira and R. Sinnott, “A performance evaluation of containers running on managed kubernetes services,” *Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom*, vol. 2019-Decem, pp. 199–208, 2019, doi: 10.1109/CloudCom.2019.00038.

- [33] “Kubernetes.” <https://kubernetes.io/es/> (accessed Feb. 25, 2021).
- [34] V. Cosentino, J. L. C. Izquierdo, and J. Cabot, “A Systematic Mapping Study of Software Development with GitHub,” *IEEE Access*, vol. 5, pp. 7173–7192, 2018, doi: 10.1109/ACCESS.2017.2682323.
- [35] S. Li *et al.*, “Understanding and addressing quality attributes of microservices architecture: A Systematic literature review,” *Inf. Softw. Technol.*, vol. 131, p. 106449, 2021, doi: 10.1016/j.infsof.2020.106449.
- [36] A. Hannousse and S. Yahiouche, “Securing Microservices and Microservice Architectures: A Systematic Mapping Study,” no. June, pp. 0–21, 2020, doi: 10.1016/j.cosrev.2021.100415.
- [37] C. Santana, B. Alencar, and C. Prazeres, “Microservices: A mapping study for internet of things solutions,” *NCA 2018 - 2018 IEEE 17th Int. Symp. Netw. Comput. Appl.*, 2018, doi: 10.1109/NCA.2018.8548331.
- [38] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic mapping studies in software engineering,” *12th Int. Conf. Eval. Assess. Softw. Eng. EASE 2008*, no. February 2015, 2008, doi: 10.14236/ewic/ease2008.8.
- [39] F. Nasrallah, W. Zidi, M. Feki, S. Kacem, N. Tebib, and N. Kaabachi, “Biochemical and clinical profiles of 52 Tunisian patients affected by Zellweger syndrome,” *Pediatr. Neonatol.*, vol. 58, no. 6, pp. 484–489, 2017, doi: 10.1016/j.pedneo.2016.08.011.
- [40] B. Kitchenham and S. M. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering Guidelines for performing Systematic Literature Reviews in Software Engineering EBSE Technical Report EBSE-2007-01 Software Engineering Group School of Computer Science and Ma,” no. October 2021, 2007.
- [41] I. Shabani, E. Mëziu, B. Berisha, and T. Biba, “Design of Modern Distributed Systems based on Microservices Architecture,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 2, pp. 153–159, 2021, doi: 10.14569/IJACSA.2021.0120220.
- [42] J. Soldani, D. A. Tamburri, and W. J. Van Den Heuvel, “The pains and gains of microservices: A Systematic grey literature review,” *J. Syst. Softw.*, vol. 146, pp. 215–232, 2018, doi: 10.1016/j.jss.2018.09.082.
- [43] Y. Wang, L. Cheng, and X. Sun, “Design and Research of Microservice Application Automation Testing Framework,” *Proc. - 2019 Int. Conf. Inf. Technol. Comput. Appl.*

ITCA 2019, pp. 257–260, 2019, doi: 10.1109/ITCA49981.2019.00063.

- [44] J. P. Sotomayor, S. C. Allala, P. Alt, J. Phillips, T. M. King, and P. J. Clarke, “Comparison of runtime testing tools for microservices,” *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 2, no. July, pp. 356–361, 2019, doi: 10.1109/COMPSAC.2019.10232.
- [45] “TIOBE Index - TIOBE.” <https://www.tiobe.com/tiobe-index/> (accessed Aug. 14, 2022).

ANEXOS

Imágenes detalladas del cronograma de actividades que se desarrollaron para el proceso de desarrollo de la investigación.

Tabla 9 Cronograma de actividades para el desarrollo de la investigación

Objetivo 1: Identificar el método de SMS	12 days	5/18/22 8:00 a.m.
Búsqueda de artículos científicos sobre metodologías para el desarrollo de un SMS	7 days	5/18/22 8:00 a.m.
Seleccionar metodología para la construcción del SMS	7 days	5/25/22 8:00 a.m.
Objetivo 2: Realizar la clasificación de las herramientas para el despliegue de microservicio	32 days	6/8/22 8:00 a.m.
Aplicar y seleccionar la metodología	7 days	6/8/22 8:00 a.m.
Generar preguntas de investigación	7 days	6/15/22 8:00 a.m.
Generar criterios de inclusión y exclusión	7 days	6/22/22 8:00 a.m.
Búsqueda y recolección de artículos científicos	7 days	6/29/22 8:00 a.m.
Creación de base de datos con papers seleccionados para el desarrollo de la investigación	7 days	7/6/22 8:00 a.m.
Análisis de los resultados obtenidos	7 days	7/13/22 8:00 a.m.
Objetivo 3: Analizar los resultados del estudio junto con una propuesta de uso práctico.	27 days	7/27/22 8:00 a.m.
Presentación de los análisis obtenidos	7 days	7/27/22 8:00 a.m.
Presentar de propuesta de uso práctico	7 days	8/3/22 8:00 a.m.
Presentar Resultados obtenidos de la investigación realizada al asesor	7 days	8/10/22 8:00 a.m.
Elaboración de documento Final	7 days	8/17/22 8:00 a.m.
Elaboración de documento Final	7 days	8/24/22 8:00 a.m.

Tabla 10 Timelapse segunda parte

